# PROCEEDINGS

X WORKSHOP DE VISÃO COMPUTACIONAL

2014

UBERLÂNDIA MG

October 6-8

# Proceedings of
# X Workshop de Visão Computacional

Federal University of Uberlândia

ISSN 2175-6120

**Publishers:**
Ana Cláudia Martinez
Bruno Augusto Nassif Travençolo
Thiago Pirola Ribeiro

Uberlândia, October 6-8, 2014

# WVC 2014

## Conference Chairs

| | |
|---|---|
| André Ricardo Backes | FACOM-UFU |
| Bruno Augusto Nassif Travençolo | FACOM-UFU |
| Marcelo Zanchetta do Nascimento | FACOM-UFU |
| Mauricio Cunha Escarpinati | FACOM-UFU |

## Steering Committee Coordinator

| | |
|---|---|
| Adilson Gonzaga | EESC/USP |

## Steering Committee

| | |
|---|---|
| Adilson Gonzaga | EESC/USP |
| Evandro Luis Linhari Rodrigues | EESC/USP |
| Aparecido Nilceu Marana | UNESP |
| Inês Aparecida Gasparoto Boaventura | UNESP |
| Luiz Antônio Pereira Neves | UFPR |
| Maurício Marengoni | Mackenzie |
| Maurílio Boaventura | UNESP |
| Marco Antônio Piteri | UNESP |

## Local Organizing Committee

| | |
|---|---|
| Ana Cláudia Martinez | FACOM-UFU |
| André Ricardo Backes | FACOM-UFU |
| Bruno Augusto Nassif Travençolo | FACOM-UFU |
| Daniel Duarte Abdala | FACOM-UFU |
| Denise Guliato | FACOM-UFU |
| Marcelo Zanchetta do Nascimento | FACOM-UFU |
| Mauricio Cunha Escarpinati | FACOM-UFU |
| Thiago Pirola Ribeiro | FACOM-UFU |

## Program Committee

Adilson Gonzaga (USP-SC)
Aledir Silveira Pereira (UNESP)
Alexandre Xavier Falcão (UNICAMP)
Alexandre Luís M. Levada (UFSCAR)
Ana Cláudia Martinez (UFU)
Anderson da Silva Soares (UFG)
André Ricardo Backes (UFU)
Anselmo Antunes Montenegro (UFF)
Anselmo Cardoso de Paiva (UFMA)
Antônio Lopes Apolinário Júnior (UFBA)
Antônio da Luz Junior (IFTO)
Antônio Maria Tomaseli (UNESP)
Aparecido Nilceu Marana (UNESP)
Aura Conci (UFF)
Aylton Pagamisse (UNESP)
Bruno Augusto Nassif Travençolo (UFU)
Carlos E. Thomaz (FEI)
Celia Aparecida Zorzo Barcelos (UFU)
Celso Olivete Júnior (UNESP)
César A. Beltrán Castañón (PUC-Perú)
Chidambaram Chidambaram (UDESC)
Claudio Eduardo Goes (UEFS)
Clodoaldo Lima (UNICAMP)
Clodoaldo Aparecido de M. Lima (USP)
Cristina Vasconcelos (UFF)
Daniel Duarte Abdala (UFU)
Danilo Medeiros Eler (UNESP)
Emerson Carlos Pedrino (UFSCAR)
Evandro Luis Linhari Rodrigues (USP-SC)
Fabiana Rodrigues Leta (UFF)
Fabrízzio A. de Melo Nunes Soares(UFG)
Fátima N. Sombra de Medeiros (UFC)
Flávio Bortolozzi (UNICESUMAR)
Gilson Antônio Giraldi (LNCC)
Gustavo Benvenutti Borba (UTFPR)
Júlio Cesar Nievola (PUC)
Leandro Luís Galdino de Oliveira (UFG)
Maria Stela Veludo de Paiva (USP-SC)
Marilton Sanchotene de Aguiar (UFPEL)
Mauricio Cunha Escarpinati (UFU)
Mauricio Galo (UNESP)
Mauricio Marengoni (MACKENZIE)
Maurílio Boaventura (UNESP)
Maximiliam Luppe (USP-SC)
Messias Meneguette Junior (UNESP)
Michele Fúlvia Angelo (UFES)
Moacir Pereira Ponti Junior (USP-SC)
Murillo Rodrigo P. Homem (UFSCAR)
Nelson D. d'Ávila Mascarenhas (UFSCAR)
Nina Sumiko Tomita Hirata (USP)
Odemir Martinez Bruno (USP-SC)
Hélio Pedrini (UNICAMP)
Hemerson Pistori (UCDB)
Homero Schiabel (USP-SC)
Hugo Vieira Neto (UTFPR)
Iális Cavalcante de Paula Júnior (UFC)
Inês Boaventura (UNESP)
Ivan Nunes da Silva (USP-SC)
Jacob Scharcanski (UFRGS)
Jacques Facon (PUCPR)
Jander Moreira (UFSCAR)
Jarbas Joaci de Mesquita Sá Junior (UFC)
João do Espírito Santo B. Neto (USP-SC)
Joao Fernando Marar (UNESP)
João Manuel R. da Silva Tavares (FEUP)
Joao Paulo Papa (UNICAMP)
Joaquim de Mira Junior (UTFPR)
Joaquim Cezar Felipe (USP)
José Alfredo Ferreira Costa (UFRN)
José Eduardo Cogo Castanho (UNESP)
José Roberto Nogueira (UNESP)
José Hiroki Saito (UFSCAR)
Oge Marques (FLORIDA)
Paulo Mazzoncini de A. Marques (USP)

Leonardo Vidal Batista (UFPB)
Leonardo Nogueira Matos (UFS)
Liliane Ventura Schiabel (USP)
Lucas Ferrari de Oliveira (UFPR)
Luciano Cássio Lulio (USP-SC)
Lúcio André de Castro Jorge (EMBRAPA)
Luiz Marcos Garcia Gonçalves (UFRN)
Luiz Antônio Pereira Neves (UFPR)
Marcelo Nepomoceno Kapp (UNILA)
Marcelo Zanchetta do Nascimento (UFU)
Marcelo A. da Costa Vieira (USP-SC)
Marco Antônio Piteri (UNESP)
Paulo Eduardo Ambrósio (UESC)
Paulo Srgio Rodrigues (FEI)
Rafael Duarte Coelho dos Santos (INPE)
Ricardo José Ferrari (UFSCAR)
Ronaldo Martins da Costa (UFG)
Silvia Cristina Martini Rodrigues (UMC)
Sílvia Cristina Dias Pinto (USP)
Siome Klein Goldenstein (UNICAMP)
Thiago Pirola Ribeiro (UFU)
William Robson Schwartz (UFMG)
Yan Anderson Siriano Duarte (UFABC)

# Contents

1 Measure Performance of Information Visualization
Eduardo C. Oliveira (UFU), Luciene C. Oliveira (UFU), Alexandre Cardoso (UFU), Edgard A. Lamounier Jr. (UFU)
1

2 SDRF +: A Face Recognition System
Adriano da Silva Marinho (UFPB), Ed Porto Bezerra (UFPB), Leonardo Vidal Batista (UFPB)
9

3 An Effective Sequence of Operations for License Plates Recognition
Rafael Seike Ikeizumi (UNESP), Almir Olivette Artero (UNESP), Marco Antônio Piteri (UNESP), Francisco Assis da Silva (Unoeste/FIPP), Danillo Roberto Pereira (Fipp - Unoeste)
14

4 Evolutionary Optimization Applied for Fine-Tuning Parameter Estimation in Optical Flow-based Methods
Danillo R. Pereira (Fipp - Unoeste), João P. Papa (UNESP), José Delpiano (Universidad de los Andes ), Francisco Silva (Unoeste/FIPP), Marco A. Piteri (UNESP), Almir O. Artero (UNESP)
20

5 A Real-time approach for hand-gesture recognition based on computer vision and artificial neural networks
Luan Caius Ramos (CPAI), Flavio de Barros Vidal (UNB), Bruno Macchiavello (UNB), Alexandre Zaghetto (UNB)
26

6 Feature description based on Mean Local Mapped Pattern
Carolina Toledo Ferraz (USP), Osmando Pereira Junior (IFTM), Adilson Gonzaga (USP-SC)
32

7 Face recognition using complex wavelet transform, clustering and fuzzy integral
André Luiz Nogueira (IFS), Junior Leal do Prado (IFS)
38

8 Cachaça type identification using color information and computer vision
Bruno Urbano Rodrigues (UFG), Ronaldo martins da Costa (UFG), Rogerio Salvini (UFG), Anderson da Silva Soares (UFG), Flavio Alves da Silva (UFG), Márcio Caliari (UFG), Karla Cristina R. Cardoso (UFG), Tânia Isabel M. Ribeiro (Instituto Politécnico de Bragança)
45

9 Neural Filter Applied To Fissure Detection
Edson Cavalcanti Neto (UFC), Paulo César Cortez (UFC), Valberto E. R. da Silva Filho (UFC), Tarique S. Cavalcante (UFC), Pedro P. Rebouças Filho (IFCE)
50

10 True-Motion Estimation and Compensation with Multi Temporal Block Matching Search
André L. Martins (USP), Evandro L. L. Rodrigues (USP-SC)
55

11 Color Texture Classification under Varying Illumination
Tamiris Trevisan Negri (USP-SC), Adilson Gonzaga (USP-SC)
61

12 Image Enhancement Algorithms for Infrared Cameras
Lucas Rotava (USP), Evandro Luis Linhari Rodrigues (USP-SC)
67

13 Evaluation of Machine Learning and Bag of Visual Words Techniques for Pollen Grains Classification
Carolini Nascimento Martins Rodrigues (UCDB), Ariadne Barbosa Gonçalves (UCDB), Gercina Gonçalves da Silva (UCDB), Hemerson Pistori (UCDB)
73

14 Video-based Iris Recognition by Quasi-Dynamic Texture Analysis
Raissa Tavares Vieira (USP), Virgílio de Melo Langoni (UFTM), Adilson Gonzaga (USP-SC)
79

15 Detection, Extraction and Text Translation in Digital Images using Android Platform
Luiz P. Rabachini (UNESP), Marco A. Piteri (UNESP), Almir O. Artero (UNESP), Francisco A. da Silva (Unoeste/FIPP), Danillo R. Pereira (Unoeste/FIPP)
85

16 Effect of Nonrigid Alignment Using Free-Form Deformation in Frontal Face Images
Igor R. R. Xavier (FEI), Carlos E. Thomaz (FEI)
91

17 Augmented Tattoo: a proposal of tattoo visualization in augmented reality
Jairo Calmon (UEFS), João Queiroz (UFJF), Claudio Goes (UEFS), Angelo Loula (UEFS)
97

18 Salient Region Detection Through Random Pixel Dissimilarities
Nelson Garcia De Paula (UTFPR), Mikhail A. Koslowski (UTFPR), Humberto R. Gamba (UTFPR), Gustavo B. Borba (UTFPR)
103

19 Image Segmentation applied on Registration Process for Remote Sensing
Juliano Marques Nunes (UFABC), Francisco de Assis Zampirolli (UFABC), Helena França (UFABC)
108

20 Comparison of interpolation Methods for Digital Images.
Wesley Dourado (UNESP), Aylton Pagamisse (UNESP), Marco A. Piteri (UNESP)
113

21 Off-line Signature Verification Using a Mixed Segmentation
William C. Farias (PUCPR), Edson J. R. Justino (PUCPR), Jacques Facon (PUCPR), Luiz E. S. Oliveira (UFPR), Robert Sabourin (ETS)
119

22 Recognizing Skin in digital images using Artificial Neural Networks
Igor Leonardo Oliveira Bastos (UFBA), Michele Fúlvia Angelo (UEFS)
125

23 Self-Organizing Traffic Lights: A Pedestrian Oriented Approach
Jessica S. Souza (UFMG), César A. M. Ferreira (UFMG), Cassio E. Santos Jr (UFMG), Victor H. C. Melo (UFMG), William Robson Schwartz (UFMG)
131

24 Development of an application for security based in face recongnition on Android plataform
Thaisa A. Correia (UNESP), Marco A. Piteri (UNESP), Almir O. Artero (UNESP), Francisco Assis da Silva (Unoeste/FIPP), Danillo Roberto Pereira (Unoeste/FIPP)
137

25 A multi-channel and multi-scale architecture for image segmentation based on morphological gradient
Karin S. Komati (IFES), Evandro O. T. Salles (UFES), Mario Sarcinelli Filho (UFES)
143

26 Weightings of Shannon Entropy's Additivity for Image Segmentation
Celso Dênis Gallão (FEI), Paulo Sérgio Rodrigues (FEI)
151

**27 A new approach to automate the seed vigor analysis on soybean seedling using digital images**
**Daniel Caio de Lima (USP-SC), Evandro L. L. Rodrigues (USP-SC), Lucio André de Castro Jorge (Embrapa)**
**157**

**28 The Impact of Compression Algorithms on Touchless-to-Touch Fingerprint Images**
**Alexandre Zaghetto (UNB), Pedro S. Franco (UNB), Daniel A. M. Sandoval (UNB)**
**162**

**29 Face Recognition Using 3DLBP Method Applied to Depth Maps Obtained from Kinect Sensors**
**João Batista Cardia Neto (UNESP), Aparecido Nilceu Marana (UNESP)**
**168**

**30 Evaluating LBP and CLBP in classifying of mammograms lesions**
**Yan A. S. Duarte (UFABC), Marcelo Z. do Nascimento (UFU), Sidon C. Duarte (UFABC), Leandro A. Neves (UNESP)**
**173**

**31 Traffic Sign Detection and Recognition using the AdaBoost and SIFT algorithm**
**Francisco A. da Silva (Unoeste/FIPP), Antônio C. P. Nascimento (Unoeste/FIPP), Maria Stela Veludo de Paiva (USP-SC), Almir Olivette Artero (UNESP), Marco Antônio Piteri (UNESP), Ricardo Luiz Barbosa (UNESP)**
**179**

**32 Skin Color Segmentation in Face Images: An Approach for False Positive Reduction**
**João Paulo Brognoni Casati (USP), Diego Rafael Moraes (USP-SC), Evandro Luis Linhari Rodrigues (USP-SC)**
**185**

**33 Multiscale Detection of Convexities and Concavities Based on Local Computation of Weights**
**Antonio Louro (UESC), Adilson Gonzaga (USP-SC)**
**190**

**34 Towards 3D point cloud from thermal images using colormap transformations**
**Andriy G. Krefer (UTFPR), Gustavo B. Borba (UTFPR), Mauren A. de Souza (UTFPR), Tania M. Centeno (UTFPR), Humberto R. Gamba (UTFPR)**
**196**

35 Application of texture analysis for differentiation of the greening from other pests
Patricia Pedroso Estevam Ribeiro (USP), Maria Stela Veludo Paiva (USP-SC), Lucio André de Castro Jorge (Embrapa)
202

36 An automatic facial expression recognition system evaluated by different classifiers
Caroline Silva (UFBA), Andrews Sobral (UFBA), Raissa Tavares Vieira (USP)
208

37 Facial expression recognition in static images by generalized procrustes analysis
Caroline Silva (UFBA), Andrews Sobral (UFBA), Raissa Tavares Vieira (USP)
213

38 Roads and Trajectories Detection in Agricultural Environment Using Catadioptric Vision System
Victor de Assis Rodrigues (UNESP), Vinicius Bergoli Trentini (UNESP), Luciano Cássio Lulio (USP), Mário Luiz Tronco (USP-SC)
218

39 Investigating the use of Block-Matching 3D Denoising Algorithm to Reduce Radiation Dose in Digital Mammography
Helder C. R. Oliveira (USP), Polyana F. Nunes (USP), Lucas R. Borges (USP), Marcelo A. da C. Vieira (USP-SC)
224

40 Human Iris Segmentation on Videos Obtained via Natural Lighting from SmartPhones
Hedenir Pinheiro (UFG), Ronaldo Costa (UFG), Renan Romero (UFG), Fabrizzio Alphonsus Alves de Melo Nunes (UFG), Leandro Oliveira (UFG), Gustavo Teodoro Laureano (UFG)
230

41 Methods to Musical Notes Recognition from Sheet Music: Preliminary Results
Caio Jin Aoqui (UNESP), Almir Olivette Artero (UNESP), Marco Antonio Piteri (UNESP), Danilo Medeiros Eler (UNESP)
237

42 Real-Time Hand Gesture Recognition Based on Sparse Positional Data
Tatiana Schmidt (UFMG), Felipe P. Araujo (UFMG), Gisele L. Pappa (UFMG), Erickson R. Nascimento (UFMG)
243

**43 Improving texture classification with nonextensive statistical mechanics**
**Lucas Assirati (USP), Núbia Rosa da Silva (USP-SC), Odemir Martinez Bruno (USP-SC)**
**249**

**44 Automatic Counting of Stomata in Epidermis Microscopic Images**
**Marcos William da Silva Oliveira (USP-SC), Núbia Rosa da Silva (USP-SC), Dalcimar Casanova (USP-SC), Luiz Felipe Souza Pinheiro (UNESP), Rosana Marta Kolb (UNESP), Odemir Martinez Bruno (USP-SC)**
**253**

**45 Video-Based Rendering Architecture for Remote Education via Web**
**Maikon Cismoski dos Santos (UNICAMP), Helio Pedrini (UNICAMP)**
**258**

**46 Lightness Constancy in Edge Detection: A Simple Approach using Luminance Ratios**
**Hugo Vieira Neto (UTFPR)**
**264**

**47 Using image analysis and processing for morphological characterization of bovine spermatozoa**
**Rebecca Espírito Santo da Cruz (UFU), Marcelo Emílio Beletti (UFU), Bruno Augusto Nassif Travençolo (UFU)**
**270**

**48 Comparison of illumination normalization techniques for face recognition**
**Jonatan Patrick Margarido Oruê (UFMS), Amaury A. Castro Jr. (UFMS), Wesley Nunes Gonçalves (UFMS)**
**275**

**49 Bone age estimation: Results obtained using Anacarp software versus Artificial Neural Networks methodology**
**André Henrique Silva (UNESP), Celso Olivete Júnior (UNESP), Ronaldo Celso Messias Correia (UNESP), Rogério Eduardo Garcia (UNESP)**
**281**

**50 Reconhecimento de Caminhos Táteis com Visão Computacional em Ambientes Controlados**
**Rodrigo Lima (Mackenzie), Maurício Marengoni (Mackenzie)**
**287**

**51 Grammatical Inference and SIFT for Scene Recognition**
**Lucas C. Ribas (UFMS), Marcelo Borth (IFMS), Amaury A. Castro Jr. (UFMS), Wesley N. Gonçalves (UFMS), Hemerson Pistori (UCDB)**
**293**

**52 An automated method for quantifying cytoplasmic gene expression level on early Drosophila embryos**
**Daniela Justiniano de Sousa (UFU), Maira Arruda Cardoso (UFRJ), Paulo Mascarello Bisch (UFRJ), Francisco José Pereira Lopes (UFRJ), Bruno Augusto Nassif Travençolo (UFU)**
**299**

**53 Local Method for Lens Distortion Correction Applied to Stereo Vision**
**Osmando Pereira Junior (IFTM), Joceli Mayer (UFSC)**
**305**

**54 Image Retrieval: Importance and Applications**
**João Augusto da Silva Junior (UFG), Rodiney Elias Marçal (UFG), Marcos Aurélio Batista (UFG)**
**311**

**55 Method of Inserting Poisson Noise in Digital Mammography Images to Simulate Radiation Dose Reduction**
**Lucas R. Borges (USP), Helder C. R. Oliveira (USP), Polyana F. Nunes (USP), Marcelo A. de C. Vieira (USP-SC)**
**316**

**56 Face Recognition Systems based on Wavelet Transform and SVM Ensembles**
**Daniel M. M. da Costa (USP), Felipe G. S. Teodoro (USP), Sarajane M. Peres (USP), Clodoaldo A. M. Lima (USP)**
**321**

**57 ECG Biometric Recognition based on fiducial features using Support Vector Machines**
**Felipe G. S. Teodoro (USP), Daniel M. M. da Costa (USP), Sarajane M. Peres (USP), Clodoaldo A. M. Lima (USP)**
**327**

# Measure Performance of Information Visualization

Eduardo C. Oliveira, Luciene C. Oliveira, Alexandre Cardoso, and Edgard A. Lamounier Jr.
*UFU*
*Uberlândia, Brasil*

***Abstract*—The information visualization is important as it makes the appropriate acquisition of the information through the visualization possible. The choice of the most appropriate information visualization method before commencing with the resolution of a given visual problem is primordial to obtaining an efficient solution. This article has as its objective to describe an information visualization meta-model classification approach based on treemap, which is able to identify the best information visualization model for a given problem. The actual state of the visualization field is described, and then the rules and criteria used in our research are shown, with the aim of presenting a proposal for meta-information visualization model based on treemap, inspired upon the periodic table meta-model.**

***Keywords—information visualization methods, performance analysis, knowledge visualization, visualization meta-models, treemap***

## I. INTRODUÇÃO

Atualmente, a quantidade de dados disponíveis nos sistemas de informação é incrementada de forma contínua e considerável, sendo uma elevada quantidade de elementos de dados, podendo cada elemento ter os mais variados atributos. O mesmo ocorre em muitos domínios do conhecimento, fazendo com que as aplicações dos modelos tradicionais para a visualização e análise de dados se tornem insuficientes, complexas e ineficientes [6].

A Visualização da Informação (VI) é composta por modelos e metamodelos. Um modelo de visualização é um processo sistemático, baseado em regras, representação externa, permanente e gráfica que retrata a informação de uma forma que é propícia a aquisição de conhecimentos, o desenvolvimento de uma compreensão elaborada, ou comunicar experiências [1].

A visualização fornece uma habilidade para compreender grande quantidade de informações e para resolver problemas com dados. Isso permite a percepção de padrões tais como características locais [2].

O processo de visualização inclui quatro estágios básicos, ilustrados na Fig. 1[2]:

- A coleção e armazenamento dos dados;
- O pré-processamento para transformar os dados em alguma informação que possa ser entendida;
- A visualização de informação através de algoritmos gráficos que produzem imagens na tela;
- O sistema cognitivo de percepção humana;

O cérebro humano, particularmente a sua percepção e cognição, é um dos modelos mais interessantes para o desenvolvimento de sistemas inteligentes [5].

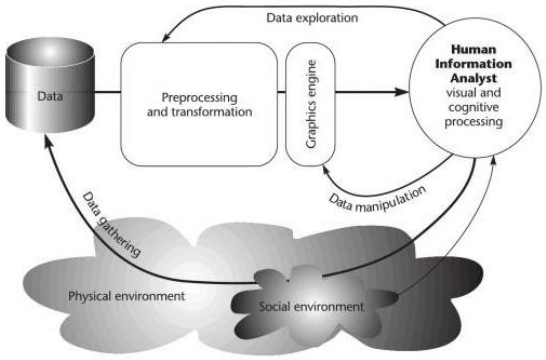



Fig. 1. Diagrama esquemático do processo de visualização [2]

Um processo de VI deve buscar a melhor representação da informação para possibilitar a rápida interpretação e o bom entendimento de uma grande quantidade de dados.

A quantidade de modelos para Visualização da Informação aumentou nos últimos tempos devido à busca de novas soluções para problemas de complexa visualização das informações e a grande quantidade de informações disponíveis. Com isso, há um crescimento da exigência pela qualidade dos modelos e da escolha mais adequada do modelo de visualização da informação para solucionar o dado problema.

O processo de busca pelo modelo a ser utilizado é de suma importância e consiste também em um processo sistematizado de Visualização da Informação dos modelos de Visualização da Informação. Desse modo, para se buscar um modelo apropriado para solução de um problema qualquer, é necessário entender os critérios de busca que possibilitem a Visualização da Informação mais adequada para os possíveis modelos de visualização já então existentes.

Um processo de VI tem como objetivo principal buscar a melhor forma de mapear as informações e limitar a quantidade de informações que serão apresentadas para para garantir a melhor percepção e cognição visual das informações pelos usuários. A melhor forma de mapeamento está relacionada ao objetivo esperado pela representação da informação, seja pela facilidade de busca da informação, seja através da ênfase de determinadas informações em relação a outras ou através da organização das informações. Esta representação da informação é realizada através de critérios e técnicas de visualização.

Os critérios da visualização da informação possibilitam a categorização, identificação e busca dos modelos de visualização. Esta categorização e identificação dos modelos de

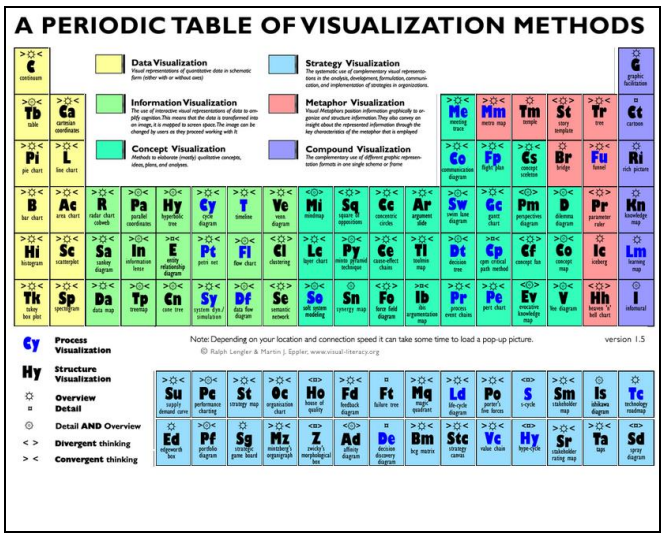



Fig. 2. Tabela Periódica dos métodos de visualização.

visualização devem possibilitar uma estrutura de busca rápida, interativa e de fácil manipulação.

Assim como existem modelos para Visualização da Informação, existem modelos para categorização dos modelos de VI, ou seja, modelos para Visualização da Informação da Visualização da Informação. Estes últimos são denominados metamodelos de Visualização da Informação.

Existem algumas tentativas desses metamodelos, como por exemplo, a Tabela Periódia da Visualização da Informação [1]. Entretanto, atualmente existe uma carência de metamodelos que possuem uma estrutura de busca rápida, interativa e de fácil manipulação para busca dos modelos mais adequados para solução dos problemas de VI.

Neste sentido, este trabalho propõe um *framework* genérico para visualização de informações baseado na lógica de Treemap, inspirada no modelo proposto pela Tabela Periódica. Este *framework* pode ser visto como um metamodelo que permite representar a visualização, interpretar e compreender as informações e possui uma estrutura de busca da informação simples e rápida.

Através dessa proposta, é possível analisar e comparar os critérios de visualização dos dois metamodelos, mostrar e demonstrar a diferença de performance na interpretação entre os dois.

## II. CRITÉRIOS PARA CLASSIFICAÇÃO DA VISUALIZAÇÃO DA INFORMAÇÃO

Os diversos modelos de visualização da informação se distinguem uns dos outros pelo uso de diferentes formas ou padrões de representação gráfica da informação, denominadas neste artigo como "critérios" para classificação e organização gráfica da visualização da informação. Esses critérios são usados na taxonomia da informação e também para classificação dos modelos nos metamodelos de visualização da informação.

Este tópico propõe a análise dos critérios utilizados no metamodelo da Tabela Periódica da Visualização da Informação e possíveis critérios que podem ser utilizados em um novo metamodelo denominado Treemap.

### *A. Tabela Periódica da Visualização da Informação*

O metamodelo constituído e apresentado em [1] classifica as técnicas de visualização e as organizaram em uma tabela periódica da informação. O modelo utilizado consiste em uma metáfora da tabela periódica da química.

Os princípios da estrutura organizacional que os autores utilizaram são:

- ***Complexity of Visualization*:** de baixo para cima, refere-se ao número de regras aplicadas para a utilização e / ou número de interdependências dos elementos a serem visualizadas.
- ***Main Application or Content Area* [como?, o que?]:** Dados, Informação, Conceito, Metáfora, Estratégia, Conhecimento Misturado. Além disso, os membros desse grupo também podem ser classificados de acordo com a intensidade do conhecimento, iniciando das explicitas, cujo conhecimento é objetivo (como ***Data Visualization***), para as mais implícitas, visualizações subjetivas do conhecimento (como ***Compound Knowledge Visualization***).
- ***Point of View* [quando?]:** *Detail* (com destaque aos itens individuais), *Overview* (figuras grandes), *Detail and Overview* (os dois ao mesmo tempo).
- ***Type of Thinking Aid* [por quê?]:** *Convergent* (reduzindo a complexidade) versus *Divergent* (adicionando complexidade).
- ***Type of Representation* [what?]:** *Process* (passo a passo cíclico no tempo e/ou sequência contínua), *Structure* (por exemplo, hierarquia ou redes causais).

A Fig. 2 mostra o modelo de visualização da tabela periódica da informação proposto por [1].

A tabela periódica disponibiliza 100 modelos de visualização da informação em uma única figura (ilustrada na tabela periódica), posicionando uma sigla para cada modelo e o seu nome logo abaixo da sigla. Ao posicionar o mouse no modelo, a tabela periódica de visualização da informação mostra um correspondente exemplo desse modelo. A tabela possui duas legendas para os critérios organizacionais dos modelos de visualização da informação utilizados na estrutura desse metamodelo.

Entretanto, a tabela não possibilita a pesquisa direta pelo nome de um determinado modelo. Por se tratar de uma única imagem para toda a tabela, a busca por um determinado modelo de visualização através do seu nome não é simples, pois o usuário deverá visualizar modelo por modelo e buscar o nome desejado. Não há interação para mostrar somente os modelos de visualização correspondentes a uma determinada categoria ou modelos que atendam um dado critério.

A organização da tabela periódica posiciona os modelos de visualização, que possuem características e finalidades

semelhantes, por proximidade. Tal estratégia facilita a categorização dos modelos de visualização de forma estática, principalmente no que diz respeito à classificação através de "Aplicação principal ou área de conteúdo" ("Main Application or Content Area"). Entretanto, a partir desse modelo, há dificuldades na busca da informação quando se deseja pesquisar por vários critérios simultaneamente, como por exemplo: obter todos os modelos de visualização que sejam "Concept Visualization", "Detail and Overview", "Visualization Structure" e "Thinking Divergent".

Assim, além de consistir em uma busca lenta, o modelo não disponibiliza uma forma interativa para obtenção do resultado, de forma a separá-lo visualmente das demais informações disponíveis no modelo.

A tabela periódica da visualização da informação se utiliza do critério de cor da área e posição para diferenciar as categorias de "Aplicação principal ou área de conteúdo" ("Main Application or Content Area"), cor da sigla para distinguir os princípios de "Tipo de Representação" ("Type of Representation") e símbolos para distinguir os princípios de "Ponto de vista" ("Point of View") e "Tipo de Pensamento de Ajuda" ("Type of Thinking Aid"). Várias técnicas de visualização foram utilizadas para classificar os modelos de visualização da informação segundo os critérios utilizados na metáfora da tabela periódica, mas o processo se torna complexo, uma vez que não facilita a busca dos modelos durante a visualização da informação.

Ao realizar consecutivas pesquisas de modelos através desses símbolos, o usuário pode se confundir nos critérios de pesquisa dos modelos, principalmente quando os símbolos dos modelos são similares, embora os critérios dos símbolos sejam distintos.

### *B. Visualização da Informação Baseado em Treemap*

Treemap consiste em uma forma da visualização da informação criada por Shneiderman [4] em 2006, capaz de organizar as informações em uma estrutura hierárquica, através dos princípios dessas próprias informações e apresentá-las num gráfico, a partir dessa estrutura.

Essa estrutura permite o acesso rápido da informação, de forma visual, separando as informações mais importantes ou mais desejadas após a sua busca, em um processo fortemente interativo.

O usuário pode separar as informações segundo vários critérios visuais, como por exemplo: posição, área, cor e textura. A Fig. 3 mostra um exemplo de Treemap cuja estrutura está orientada pelos critérios de posição, cor e área da "Concept Visualization". É possível determinar uma estrutura na forma de árvore para cada critério de busca. No caso da imagem da Fig. 4, a estrutura da árvore mostra uma organização orientada pelo critério de busca através de grupo da Treemap.

A estrutura da Treemap possibilita disponibilizar diversas formas de visualização para separá-las segundo critérios de busca dos dados. O usuário pesquisa a informação, associando-na a um princípio desejado (por exemplo, "Point of View" do tipo "Detail") segundo um critério de busca também desejado (por exemplo, pela cor vermelha) e a treemap apresenta as informações conforme a preferência do usuário.

Com tal arranjo, a mesma facilita a interação, de acordo com as necessidades visuais do usuário, no que diz respeito à diversidade de escolha de formas de visualização. Por exemplo, é possível destacar determinadas informações segundo cores ou, as mesmas informações segundo área, para usuários daltônicos.

Assim, a Treemap possui diversas formas de visualizações através de percursos otimizados, possibilitando maiores percepções e interações com os usuários.

## III. METAMODELO PROPOSTO PARA VISUALIZAÇÃO DA INFORMAÇÃO

Esta seção possui a finalidade de apresentação de uma proposta de metamodelo para Visualização da Informação que posua uma estrutura ideal para busca do melhor modelo de visualização a ser utilizado em determinada manipulação de informação.

### *A. Treemap da Visualização da Informação*

O metamodelo da Tabela Periódica da Visualização da Informação proposto em [1], foi convertido neste trabalho em um metamodelo de Treemap para Visualização da Informação, mantendo os princípios da estrutura organizacional.

É mantido o processo interativo de apontamento e seleção em um modelo para mostar um exemplo de visualização da informação correspondente a um dado modelo.

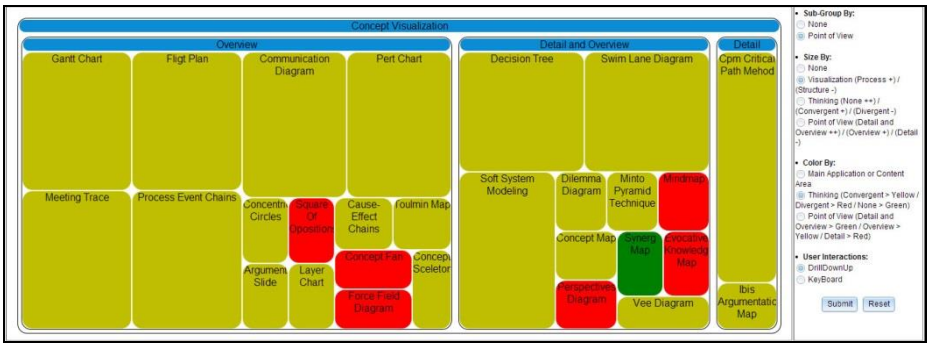



Fig. 3. *DrillDownUp* de "*Concept Visualization*"

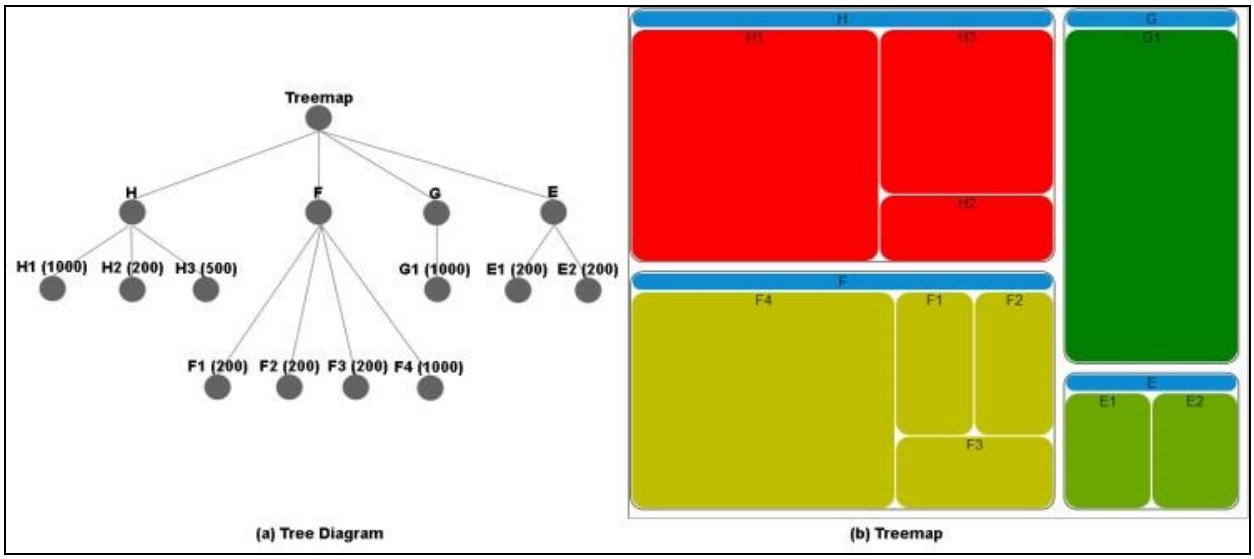



Fig. 4. Exemplo de estrutura de treemap

As legendas foram substituídas por uma busca interativa com o usuário, que permite a escolha do critério que deve gerar a forma de visualização da Treemap. A busca interativa considera os critérios através dos princípios correlacionados na Tabela 1. Observe que o critério de símbolo usado no metamodelo da Periodic Table, não é mais utilizado.

A escolha dos critérios que definem as formas de visualização das informações é importante para definição da busca mais adequada e com a percepção esperada pelo usuário. Neste metamodelo, alguns critérios podem ser utilizados para definir a busca da informação de mais de um princípio utilizados para categorizar os modelos de visualização da informação. Por exemplo, é possível buscar pelo princípio de "Point of View" através de qualquer um dos critérios de busca, mas o critério de posição é suficiente para organizar o princípio de "Main Application or Content Area".

TABELA 1
CRITÉRIOS QUE DEFINEM AS FORMAS DE VISUALIZAÇÃO NO METAMODELO TREEMAP.

| | Main Application or Content Area | Point of View | Type of Representation | Type of Thinking Aid |
|---|---|---|---|---|
| Position (or Group) | X | X | | |
| Area | | X | X | X |
| Color | X | X | | X |

Para os princípios que possuem vários tipos, é necessário associar a variação no critério. Por exemplo, critério de cor para o princípio "Type of Thinking Aid" possuem as variações de amarelo para "Convergent", vermelho para "Divergent" e verde para "None". O critério de área para o princípio "Type of Representation" possuem as variações de maior área para "Process" e menor área para "Structure".

Além disso, outra possível interação que facilita a visualização da informação consiste no mecanismo de *DrillDownUp* ou *Keybord*. Em casos de Treemaps, que contenham muitas informações (dezenas, centenas ou milhares), o recurso possibilita mostrar somente o grupo de informações de interesse, conforme ilustrado nas Fig. 3 e Fig. 5.

A Fig. 6 apresenta o modelo proposto neste trabalho da Treemap de Visualização da Informação. O modelo agrupa, por padrão, os modelos por "Main Application or Content" e possibilita a interação com o usuário. No exemplo dessa figura, o usuário solicitou a separação por sub-grupos por "Detail e Overview", tamanho por "Visualization", na qual os modelos caracterizados com visualizações por processos possuem tamanhos maiores que os classificados por estruturas. O critério de cor foi utilizado para separar as informações de "Thinking", na qual os modelos de cores amarelas correspondem ao "Convergent", vermelho os "Divergent" e verde os "None".

Assim, para obter todos os modelos de visualização que sejam "Concept Visualization", "Detail and Overview", "Visualization Structure" e "Thinking Divergent", basta buscar no grupo de "Concept Visualization" e subgrupo "Detail and Overview", os modelos que possuem área e tamanho pequenos e cor vermelha. Em uma única visualização apresentam-se todos os modelos por meio dos critérios selecionados pela interação do usuário.

## IV. VISÃO GERAL DE ANÁLISE DE ALGORITMOS

Esta seção possui a finalidade de analisar e comparar os processos de busca da informação nas visualizações dos metamodelos apresentados neste artigo. O fundamento utilizado nesta análise consiste na análise de algoritmos, com a hipótese que, assim como esses algoritmos de computadores, que podem ser percorridos logaritmicamente, linearmente e exponencialmente, a percepção humana de uma visualização da informação também pode demorar tempos logarítmicos,

lineares e exponenciais dependendo da estrutura e dos critérios de busca visuais disponibilizados para o usuário no modelo de visualização da informação, mas também da ordem de magnitude do número de operações fundamentais que a percepção humana precisa realizar para perceber, entender e abstrair os dados da forma visual.

O número de modelos de visualização da informação são pequenos e a técnica de análise de algoritmos geralmente é feita quando envolve grandes quantidades de informações. Entretanto, o processo cognitivo de um usuário comum pode ser lento para a leitura de determinado modelo, assim como ocorre nos metamodelos da Treemap e Periodic Table, o que viabiliza a análise de algoritmos para quantidades pequenas de informações.

### *A. Análise de Algoritmos*

A análise de algoritmos é um ramo da ciência da computação que estuda as técnicas de análise e projeto de algoritmos. A análise é um mecanismo capaz de orientar a melhor forma de implementar um algoritmo. Sendo assim, não é necessário que o algoritmo esteja implementado em uma linguagem de programação para realizar a sua análise.

A análise de algoritmos computacionais estipula a complexidade de tempo e de espaço através da complexidade do algoritmo. A complexidade do algoritmo reflete o esforço computacional requerido para executá-lo e complexidade de tempo e espaço refletem respectivamente a velocidade e quantidade de memória requeridas ao executar o algoritmo. Outra regra para determinar a complexidade de um algorítmo é levar em consideração os piores e os médios esforços dos subcálculos ou subprocedimentos.

Geralmente, o cálculo da complexidade concentra-se em determinar a ordem de magnitude do número operações fundamentais na execução do algoritmo e é representado por uma notação, por exemplo, a notação "*the Big oh*" é representada por 'O' e é usado junto com o "*order of growth*" para classificar a eficiência do algoritmo. A Tabela 2 [3] mostra os possíveis order of growth utilizados na representação da complexidade de tempo de algoritmo. Ela também apresenta as classes de eficiências de algoritmos e suas características.

TABELA 2
TABELA COM AS COMPLEXIDADES DE TEMPO DE ALGORITMOS [3]

| Nome da Classe de Eficiência | Ordem de Crescimento | Descrição | Exemplo |
|---|---|---|---|
| Constante | 1 | Unidade base de complexidade. | Obter tamanho de uma lista. |
| Logarítimico | log n | Se a complexidade é logarítimica, então o algorítmo nunca percorre todas as suas entradas, pois o problema é dividido em partes menores que serão iteradas. | Operação de busca binária |
| Linear | n | O tempo de execução do algoritmo depende do tamanho da entrada *n*. | Operação de busca sequêncial. |
| Quadrático | $n^2$ | Essa complexidade existe quando é necessário percorrer a entrada *n* de forma duplicada para resolver o problema. | Percorrer uma matriz de elementos. |
| Exponencial | $2^n$ | Essa complexidade existe quando uma taxa muito grande de crescimento de validações existe no algoritmo. | Geração de todos os subconjuntos de *n* elementos |
| Fatorial | n! | Esse tipo de complexidade existe quando é necessário permutar todas as combinações para resolver o problema. | Geração de todas as permutações de um problema combinatório. |

Para um determinado problema como ordenação de uma lista de números inteiros, é possível desenvolver algoritmos com complexidade O($n^2$) que requerem maior complexidade de tempo, mas também é possivel desenvolver algorítimos para o mesmo problema com complexiadades O(log n) que requerem menor complexidade de tempo. Caso o algoritmo seja desenvolvido com tempo de O(log n + c), na qual 'c' seja uma constante, o pior caso é considerado e o tempo passa a ser O(log n).

### *B. Análise de Algoritmos de Visualização*

Este tópico apresenta como finalidade a análise dos algoritmos de visualização da informação de ambos metamodelos, Periodic Table e Treemap. Hipoteticamente, a análise de algoritmos de visualização da informação se fundamenta nos mesmos princípios da análise de algoritmos computacionais.

Considere os n=28 modelos de VI existentes na categoria "*Strategy Visualization*", segundo o metamodelo da Periodic Table, mostrados na Fig. 5. Para buscar todos os modelos com o princípio "*Visualization*" do tipo "*Process*" é suficiente contar os m=6 modelos segundo o critério de cor azul correspondente a esse princípio ("*life-cycle diagram*", "*s-cycle*", "*techmology roadmap*", "*decision discovery diagram*", "*value chain*" e "*hype-cycle*"). Assim, a complexidade de tempo para extrair essa informação é O(m), pois deve-se percorrer visualmente esses "m" elementos. Esse critério de busca através de cor é bem usado pela perecepção humana.

Entretanto, para buscar todos os modelos com o princípio "*Point of View*" do tipo "*Overview*" e ao mesmo tempo com o princípio "*Thinking*" do tipo "*Convergent*" é necessário percorrer todos os modelos da "*Strategy Visualization*" para verificar se o símbolo de cada modelo corresponde ao da legenda dos critérios a serem buscados. Após verificar para o primeiro critério, deve-se percorrer todos os modelos do conjunto desse critério para delimitar quais também atende ao segundo critério. Assim, é necessário percorrer todos os n=28 modelos, delimitando o=17 modelos do tipo "*Overview*" e, logo em seguida, percorrer os o=17 modelos para verificar l=14 contém também o princípio "*Thinking*" do tipo "*Convergent*". A complexidade de tempo desse algoritmo é de O(m) + O(n). Como m>n, ou seja, "m" é o pior caso, a complexidade pode ser simplificada para O(m).

Fig. 5. Analise de algoritmo da *Strategy Visualization* da Treemap.

O mesmo estudo de caso utilizado na análise do metamodelo da Periodic Table será utilizado na análise do metamodelo da Treemap, a seguir. Entretanto, os critérios de visualização dos metamodelos são diferentes, pois enquanto a Periodic Table utiliza o critério de símbolo para buscar por determinados princípios, a Treemap utiliza o critério de posição, área e cor.

Neste metamodelo da Treemap, utilizando o critério de cor para "*Point of View*" e posição ou área para "*Thinking*" como mostrado na Fig. 5, basta contar o número de áreas amarelas com tamanhos intermediários correspondente a l=14. A complexidade de tempo desse algoritmo é de O(l). Como cada critério está organizado em árvore, "l" corresponde a uma parte logarítmica de "n" e a complexidade de tempo de algoritmo de visualização é menor no metamodelo Treemap comparado ao metamodelo Periodic Table, para a busca desse exemplo proposto, pois l<m.

### *D. Análise de Algoritmos de Visualização dos Metamodelos*

Calcular a complexidade de tempo de algoritmo de visualização de cada metamodelo é correspondente a calcular a complexidade de tempo correspondente a cada critério utilizado nas possíveis buscas pelos modelos existentes.

Como o metamodelo da Tabela Periódica utiliza os critérios de grupo ou posição (para "*Main Application or Content Area*"), cor (para "*Type of Representation*") e símbolo (para "*Type of Thinking Aid*" e "*Point of View*"), então a complexidade de tempo para os dois primeiros critérios é logarítmca, mas para o critério de símbolo é linear. Seja "x" o número médio de itens de cada conjunto de elementos que contém o critério de grupo, "y" o número médio de itens de cada conjunto de elementos que contém o critério de cor e "z1" e "z2" os números médios de itens de cada conjunto de elementos que contém os critérios de símbolo, então a complexidade de tempo para busca neste metamodelo é O(z1 + z2 + log(x) + log(y)).

Já o metamodelo da Treemap utiliza os critérios de grupo ou posição (para "*Main Application or Content Area*" e "*Point of View*"), cor (para "*Main Application or Content Area*", para "*Point of View*" e para "*Type of Thinking Aid*") e área ou tamanho (para para "*Point of View*", para "*Type of Representation*" e para "*Type of Thinking Aid*"), mostrado na Tabela 1. Seja "x" o número médio de itens de cada conjunto de elementos que contém o critério de grupo, "y" o número médio de itens de cada conjunto de elementos que contém o critério de cor e "z" o número médio de itens do conjunto de elementos que contém o critério de área, então a complexidade de tempo para busca neste metamodelo é O(log(x) + log(y) + log(z)). A complexidade de tempo do metamodelo da Treemap é menor que a complexidade de tempo do metamodelo da Periodic Table e, portanto, apresenta melhor desempenho de qualquer busca de dados na visualização da informação.

## V. CONCLUSÃO

Os critérios da visualização da informação possibilitam a identificação e a busca de informações através da estrutura organizacional dos modelos de visualização.

As análises de algoritmos apontam que a estrutura da inovadora proposta da Treemap permitiu uma busca mais rápida, interativa e de fácil manipulação para realizar a pesquisa dos modelos mais adequados para solução dos problemas de visualização da informação.

Alguns critérios de visualização são mais perceptíveis que outros. Assim, é possível determinar a complexidade de tempo de algoritmo tanto dos metamodelos quanto dos modelos de visualização de informação, a partir dos critérios de visualização utilizados por eles. Isso é medido através de um modelo criado neste trabalho denominado análise de algoritmos de visualização, análogo ao modelo de análise de algoritmos computacionais. Este modelo possibilita antecipar problemas de desempenho e expressividade na escolha do modelo mais adequado para visualização da informação antes da sua implementação, ou seja, durante a sua análise.

De acordo com os experimentos realizados, podemos observar que o metamodelo de visualização baseado em TreeMap possui maior eficácia e expressividade que o metamodelo baseado em Periodic Table, considerando os critérios de visualização de grupo e cor. Além disto, os

resultados dos experimentos mostram que o modelo de análise de algoritmos pode ser aplicado na visualização da informação.

O metamodelo proposto possui os critérios de posição e tamanho que foram avaliados com bons resultados de capacidade de abstração e distinção de percepção humana. O critério de visualização através de símbolo que o metamodelo Periodic Table propõe, teve piores resultados em relação aos outros critérios, pois os usuários confundem muito os símbolos na busca pela informação.

Conclui-se que a estrutura proposta permte que os usuários tenham facilidade de compreender e buscar as representações e informações que elas expressam, utilizando o metamodelo proposto de visualização baseado em TreeMap.

## REFERÊNCIAS


[1] R. Lengler and M.J. Eppler, "Towards A Periodic Table of Visualization Methods for Management," *GVE '07 Proceedings of the IASTED International Conference on Graphics and Visualization in Engineering.* ACTA Press Anaheim USA, pp. 83-88, 2007. doi: 10.1.1.95.6639.

[2] C. Ware ."Information Visualization – Perception for Design". San Francisco: Morgan Kaufmann. 2º Ed. Elsevier, 2004.

[3] A. Puntambekar. "Design And Analysis Of Algorithms". Technical Publications, pp 380, 2010. ISBN: 8184317786.

[4] B. Shneiderman, "The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations", *In IEEE Symposium on Visual Languages*. pp. 336-343, 1996.

[5] A.M. Gupta Solo and M.M Gupta. "Perspectives on Computational Perception and Cognition under Uncertainty". *In Proceedings of IEEE International Conference on Industrial Technology 2000*. vol. 1. pp 221-224, 2000.

[6] C.G. Healey. "Building a Perceptual Visualisation Architecture". *Behaviour & Information Technology*. vol. 19, nº 5, pp. 349-366, 2000.

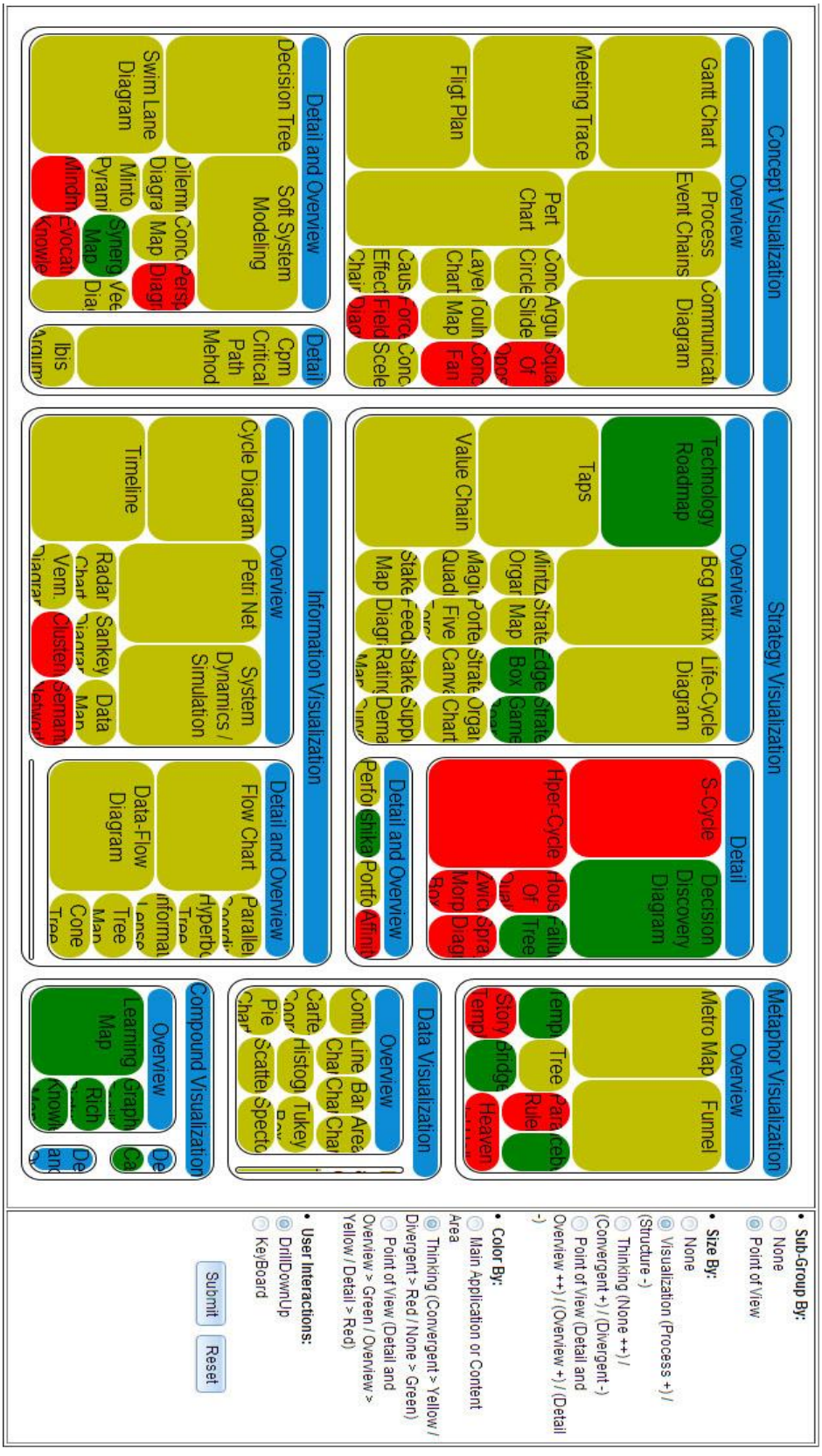



Fig. 6.Treemap para visualização da informação.

# SDRF +: A Face Recognition System

Adriano da Silva Marinho, Ed Porto Bezerra, Leonardo Vidal Batista
Programa de Pós-Graduação em Informática – Departamento de Informática
Universidade Federal da Paraíba
João Pessoa, Brasil
adriano.marinho@ci.ufpb.br, edporto@di.ufpb.br, leonardo@ci.ufpb.br

***Abstract*— Identification systems have become key components for many applications. Since traditional methods of authentication can be manipulated in order to bypass the systems, biometric authentication methods have been receiving more attention in recent years. The face is an example of a biometric trait. It is possible to perform face recognition using Discrete Cosine Transform (DCT). In order to adjust a face recognition system to uncontrolled environments, in this paper we developed an image normalizing module in relation to rotation and scale. The new system (SDRF+) was tested on the UFPB-Fotos and ORL face databases. The normalizing module is efficient, but the SDRF+ is still not suitable for uncontrolled environments.**

***Keywords*— *Biometric Authentication, Face Recognition, DCT.***

## I. INTRODUÇÃO

A proliferação de serviços que necessitam de autenticação gerou uma demanda por novos métodos para estabelecer a identidade de usuários [1]. O problema de estabelecer a identidade de um indivíduo pode ser dividido em dois: autenticação e identificação. No primeiro, um indivíduo alega uma identidade e o sistema deve confirmar ou negar. No segundo, o sistema deve estabelecer uma identidade para um indivíduo desconhecido [2] [3]. Métodos tradicionais para estabelecer a identidade de um usuário incluem mecanismos baseados em conhecimento (por exemplo, senhas) e mecanismos baseados em *tokens* (por exemplo, cartões de identidade). Porém, tais mecanismos podem ser perdidos, roubados ou até mesmo manipulados com o objetivo de burlar o sistema. Neste contexto, a autenticação e a identificação por Biometria [1] surgem como alternativas.

A autenticação biométrica oferece um mecanismo de autenticação mais confiável utilizando traços físicos ou comportamentais que permitem identificar usuários baseados em suas características naturais. Assim, é possível estabelecer a identidade de um usuário baseado em quem ele é. A face é um dos traços biométricos menos intrusivos.

A área de análise de faces pode ser dividida em diversas subáreas, como reconhecimento de face, detecção/localização de face, reconhecimento de expressões faciais e análise de poses [4].

O problema do reconhecimento de faces em vídeo e foto não é trivial, pois existem muitos fatores que influenciam na detecção e no reconhecimento, tais como: iluminação, posição da face, imagem ao fundo e diferentes expressões faciais. Assim, é comum encontrar na literatura sistemas de reconhecimento de faces em ambientes controlados [4] [5] [6].

Este artigo apresenta o SDRF+ (Sistema de Detecção e Reconhecimento de Faces +) e discute sobre seu desempenho. A seção II aborda o tema detecção e reconhecimento de faces, as seções III e IV apresentam, respectivamente, o SDRF e o SRDF+. Considerações finais e trabalhos futuros são apontados na seção V.

## II. DETECÇÃO E RECONHECIMENTO DE FACES

O reconhecimento facial é um caso específico de reconhecimento de padrões onde o padrão é uma face. É uma tarefa complexa para o computador, embora seja fácil para os seres humanos.

Algumas razões que ainda tornam o reconhecimento facial objeto de pesquisa são a preocupação crescente por segurança pública, a necessidade de verificar a identidade de um indivíduo para fins de acesso e a análise de dados para fins de entretenimento multimídia [7].

A Fig. 1 mostra a arquitetura-padrão de um sistema de reconhecimento facial [7].

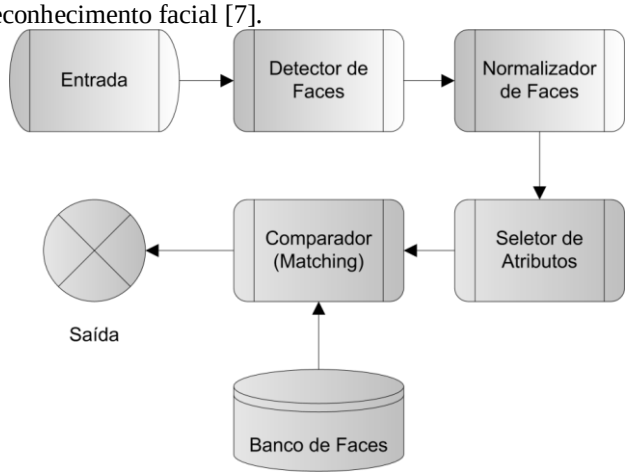



Fig. 1. Arquitetura de um Sistema de Reconhecimento Facial.

O sistema de reconhecimento facial recebe como entrada uma imagem contendo uma ou mais pessoas. A detecção ou localização de faces consiste na determinação da presença e posição espacial de cada face existente em uma imagem

A face detectada serve como entrada do módulo Normalizador de faces que pode operar padronizando as dimensões, a inclinação e a iluminação das imagens.

Uma vez normalizada, a imagem é passada para o módulo Seletor de Atributos, cujo funcionamento depende do método usado para o reconhecimento facial. O Seletor produz um vetor de características que é utilizado pelo módulo

Comparador para produzir a saída do sistema, que depende do tipo de reconhecimento que se está fazendo: identificação ou verificação. No processo de identificação facial, uma comparação 1:N é feita, de modo que a entrada para o sistema é uma face desconhecida que é associada a uma das faces previamente cadastradas no Banco de Faces. No processo de verificação, uma comparação 1:1 é feita, de modo que a entrada para o sistema é uma face desconhecida e uma identidade alegada pelo usuário: a saída é um valor de verdadeiro ou falso que aceita ou rejeita tal alegação [4].

O desempenho dos sistemas de reconhecimento facial depende de uma variedade de fatores, tais como iluminação, pose, expressões faciais, idade dos sujeitos, cabelo e acessórios. Assim, é comum encontrar na literatura sistemas de reconhecimento de faces em ambientes controlados [4] [5] [6].

## III. Um Sistema de Detecção e Reconhecimento de Faces Para Ambientes Controlados

O SDRF [8] efetua reconhecimento facial holístico utilizando a DCT e um seletor de baixas frequências com uma região elíptica. Ele pode receber como entrada um vídeo ou fotos com faces. No segundo caso, ele classifica as faces presentes nas fotos, produzindo como saída a porcentagem de faces corretamente classificadas. A arquitetura geral do sistema é apresentada na Fig. 2.

A arquitetura é composta por quatro módulos: Capturador, Detector de Face, Reconhecedor de Face e Exibidor. O Capturador realiza a obtenção das imagens de entrada, que podem ser obtidas de um arquivo de vídeo previamente capturado, lidas diretamente de algum dispositivo de captura de vídeo ou lidas de diretórios do disco rígido.

O SDRF não possui um Normalizador de Face. Logo, ele não realiza nenhum tipo de pré-processamento nas imagens antes de realizar o reconhecimento de face.

O Reconhecedor de Face possui um Seletor de Atributos, que utiliza como atributos os coeficientes das imagens DCT extraídos de uma região elíptica como a região mostrada na Fig. 3. Ele também possui um módulo de classificação das imagens que implementa um classificador de Vizinho Mais Próximo. Este utiliza como métrica de comparação a distância de Minkowski de ordem um e no qual a menor distância é utilizada para classificar uma face.

Os métodos para o reconhecimento de faces se dividem em métodos holísticos, métodos analíticos (não holísticos) e métodos híbridos [9] [10]. Os métodos analíticos analisam a face como um todo. O SDRF é um exemplo de sistema holístico. Os métodos analíticos [11] objetivam construir um espaço de características no qual a manipulação da face se torna mais simples. É possível utilizar a Transformada Cosseno Discreta (DCT) [12] para realizar reconhecimento facial holístico como visto em [13] [14].

O SDRF foi testado com dois Bancos de Faces: o banco de faces ORL [15] e o banco de faces UFPB-Fotos [8]. Ele apresentou bons resultados, acima de 90%, e se adequou a exigência comercial de não necessitar ser retreinado após a inserção de novos indivíduos no banco.

Para testar o reconhecimento de faces em fotos foi utilizado o procedimento da validação cruzada *leave-one-out* [8]. No banco UFPB-Fotos o reconhecedor atingiu uma taxa de 97,75% de acertos e no banco ORL 99,75% de acertos. Apesar de seus bons resultados em ambientes controlados, é necessário investigar se ele é adequado para qualquer ambiente.

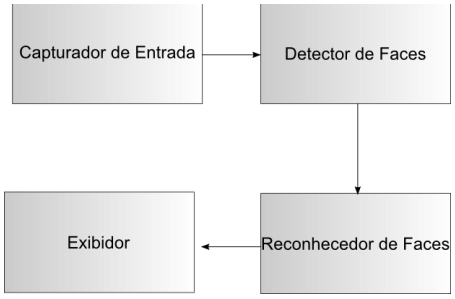



Fig. 2. Arquitetura Geral do SDRF (Fonte: OMAIA, 2009).

Fig. 3. Exemplo de região elíptica.

## IV. Um Sistema de Detecção e Reconhecimento de Faces para Ambientes Não Controlados

Lembramos que o SDRF não possui um módulo de normalização de imagens. Logo, não é possível adequar as imagens capturadas às mesmas condições das imagens utilizadas na etapa de treinamento do módulo Reconhecedor de Face. A Fig. 4 apresenta a arquitetura do SDRF+ e a Fig. 5 apresenta os submódulos do Normalizador de Imagens.

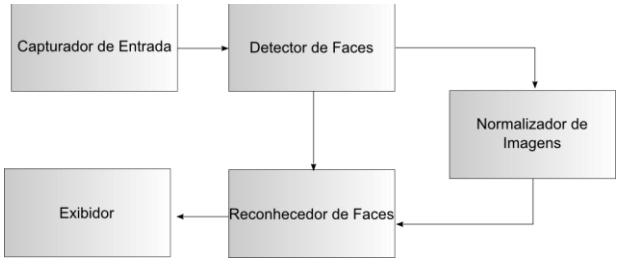



Fig. 4. Arquitetura do SDRF com o módulo de normalização de imagens.

O módulo normalizador recebe como entrada a imagem de face a ser normalizada e as coordenadas dos olhos da face presente nessa imagem. Então, as faces são normalizadas ou em relação à rotação ou em relação à escala. Os olhos das faces podem ser encontrados utilizando algoritmos de localização de olhos como o proposto em [16].

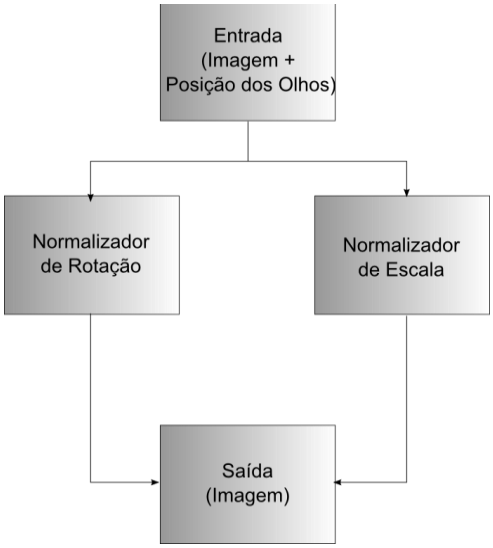



Fig. 5 Visão geral do normalizador de imagens.

.

### A. *O Submódulo Normalizador de Rotação*

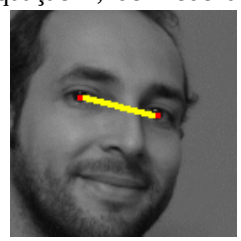
O submódulo normalizador de rotação tem como objetivo eliminar, se existir, a inclinação do segmento de reta que une os olhos de uma imagem de face. Escolhemos este tipo de normalização porque geralmente as pessoas não são filmadas com a cabeça inclinada para terem suas faces reconhecidas.

O processo de normalização da rotação é formado por 3 passos: o cálculo da inclinação do segmento de reta que une os olhos; a rotação da imagem; e o corte da imagem.

No passo 1 calculamos o arco tangente das posições dos olhos (Fig. 6 (a)). No passo 2, rotacionamos a imagem, de acordo com a Equação 1, de modo a eliminar a inclinação calculada no passo 1 (Fig. 6 (b)). Esta equação define a operação de rotação de um ângulo $\theta$ em torno de um ponto qualquer $(i_R, j_R)$ da imagem.

$$i' = (i - i_R)\cos\theta - (j - i_R)\sin\theta + i_R$$
$$j' = (i - i_R)\sin\theta + (j - i_R)\cos\theta + i_R \quad (1)$$

A operação de rotação pode deixar a imagem com espaços em branco no plano de fundo. Assim, no passo 3 utilizamos uma janela de tamanho 100 x 100 pixels para cortar a imagem, de modo que não existam espaços em branco nela (Fig. 6 (c)).



(a) Localização dos olhos.

(b) Aplicação de rotação na imagem.

(c) Aplicação de recorte na imagem.

Fig. 6. Passos do submódulo normalizador de posição.

### B. *O Submódulo Normalizador de Escala*

O submódulo normalizador de escala tem como objetivo deixar todas as imagens de um determinado banco de faces com a mesma distância entre os olhos, fazendo assim com que todas elas fiquem com a mesma distância para a câmera. Escolhemos esta normalização porque os erros do SDRF no teste em fotos eram causados por diferenças de escala.

O processo de Normalização de Escala é constituído de 4 etapas: o cálculo da distância entre os olhos padrão do banco de faces; o cálculo do fator de escala; o redimensionamento da imagem; e o recorte da imagem.

A etapa 1 consiste em calcular a distância entre olhos padrão de um dado banco de faces a partir da moda de todas as distâncias entre olhos desse banco. Na etapa 2, calculamos para a imagem a ser normalizada um fator de escala, de acordo com a Equação 2.

$$fatorDeEscala = D_{padrão} / D_k \quad (2)$$

Sendo $D_k$ a distância entre os olhos da imagem que está sendo normalizada.

Na etapa 3, redimensionamos a imagem utilizando o fator de escala e a operação de redimensionamento aplicada às linhas da imagem definida na Equação 3.

$$i' = fatorDeEscala \times (i - i_R) + i_R \quad (3)$$

Sendo:

$fatorDeEscala$ o fator de redimensionamento da imagem. Se $fatorDeEscala > 1$ a imagem será expandida, se $0 < fatorDeEscala < 1$ a imagem será reduzida.

$i$ a coordenada do pixel original.

$i'$ a coordenada do pixel redimensionado.

$i_R$ centro de redimensionamento, isto é, após a operação, todos os pontos das linha se afastam ou se aproximam de $i_R$.

A operação de redimensionamento deixa as imagens do banco de faces com tamanhos diferentes, mas o SDRF necessita que elas sejam do mesmo tamanho para o teste do reconhecimento em fotos. Assim, é necessário recortar as imagens (etapa 4) para um mesmo tamanho.

## V. RESULTADOS

Nesta seção, são descritos os resultados alcançados após o desenvolvimento do módulo normalizador de rotação e escala. O sistema foi testado com os bancos UFPB-Fotos e ORL [15].

### A. *Banco de Faces ORL*

O banco de faces ORL (do inglês, *Olivetti Research Lab*) foi desenvolvido nos laboratórios da Olivetti em Cambridge, Inglaterra. Ele é composto de 400 fotos de faces de 40 pessoas diferentes, em que cada pessoa possui 10 fotos em diferentes poses, com pequenas variações de iluminação, expressões faciais, e acessórios. Todas as imagens foram fotografadas contra um fundo homogêneo, e as faces estão todas verticais e frontais, com pequenas variações de angulação. Todas as imagens possuem tamanho 92 x 112 pixels e estão em escala de cinza com 256 níveis de cinza.

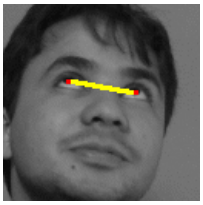
### B. *Banco de Faces UFPB-Fotos*

O banco de faces UFPB-Fotos [8] foi criado para validar o SDRF. Ele possui 20 imagens de 40 pessoas diferentes totalizando 800 imagens de faces. Todas foram capturadas frontalmente, com pouca rotação, tamanho 128 x 128 pixels e estão em escala de cinza com 256 níveis de cinza.

### C. *Normalização dos Bancos de Faces*

Normalizamos ambos os bancos primeiramente em relação à rotação e depois em relação à escala. Para isso, uma lista das posições corretas dos olhos de cada face dos bancos foi utilizada junto com as imagens. Para avaliar a necessidade de um Normalizador no SDRF, optamos por não utilizar um localizador automático de olhos, pois uma localização automática errada gera um erro na normalização.

Normalizamos os bancos em relação à rotação seguindo os passos da Seção IV.A. Na Fig. 7 é apresentado um exemplo de imagem do banco usada nesta normalização.



(a) Detecção de posição correta dos olhos.

(b) Normalização de (a).

Fig. 7. Exemplos de imagens utilizadas na normalização de rotação.

Em seguida normalizamos os bancos em relação à escala seguindo os passos da Seção IV.B. Na Fig. 8 são apresentados exemplos de imagens do banco ORL usadas nesta normalização.

(a) Imagem original.

(b) Imagem após normalização.

Fig. 8. Exemplos de imagens utilizadas na normalização de escala.

As imagens dos bancos ORL e UFPB-Fotos resultantes de ambas as normalizações possuem tamanho 100 x 100 pixels, ou seja, possuem menos informação que as versões originais. Além disso, os bancos originais mantêm a região acima da testa, como mostra a Fig. 9. Essa região contém, muitas vezes, parte do cabelo, que pode variar muito de uma captura para outra da mesma pessoa, ou seja, contem informação que não faz parte da face em si. Para contornar estes fatos e testar a necessidade de um módulo normalizador no SDRF geramos uma versão do banco UFPB-Fotos e uma versão do banco ORL de tamanho 100x100, a partir do mesmo corte 100x100 utilizado no normalizador.

(a) Imagem original

(b) Imagem normalizada.

(a) Imagem original

(b) Imagem normalizada.

Fig. 9. Diferença de informação entre imagem original e imagem normalizada

### D. *Execução do SDRF nos Bancos Normalizados*

Executamos o SDRF sobre as versões do banco UFPB-Fotos, tendo como resultado 96,62% de acertos no banco UFPB-Fotos 100 x 100, 97% no banco UFPB-Fotos normalizado em relação à rotação e 97,12% no banco UFPB-Fotos normalizado em relação à escala. Em ambos os casos a taxa de acertos para os bancos normalizados subiu, comprovando a necessidade de um módulo normalizador para o SDRF.

Em seguida, executamos o SDRF sobre as versões do banco ORL, obtendo como resultados 92,5% de acertos para o banco ORL 100x100, 92,5% de acertos para o banco ORL normalizado em relação à rotação e 94% de acertos para o banco ORL normalizado em relação à escala. A taxa de acertos para o banco normalizado em relação à rotação não foi alterada, enquanto que a taxa de acertos para o banco ORL normalizado em relação à escala subiu 1,5 ponto percentual. Novamente, isto confirma a necessidade de um módulo normalizador para o SDRF.

## VI. Considerações Finais e Trabalhos Futuros

Este trabalho teve como objetivo geral adequar o SDRF a ambientes não controlados. Para tanto, desenvolvemos um módulo de normalização de imagens em relação à rotação e escala.

Normalizamos os bancos UFPB-Fotos e ORL em relação à rotação e escala utilizando detecção automática de olhos.

Cortamos os bancos UFPB-Fotos e ORL para que se adequassem ao nível de informação dos bancos normalizados. Fizemos isto para que a necessidade de um módulo normalizador de imagens para o SDRF fosse devidamente averiguada. Na Tabela 1 é apresentado um resumo do desempenho, em termos de porcentagem de acertos, dos bancos UFPB e ORL cortados e normalizados no SDRF original.

Nota-se que a taxa de acertos nos bancos normalizados foi igual ou superior que os bancos sem normalização, o que confirma a utilidade do módulo normalizador para o SDRF. Além disso, confirma o averiguado em [8] no que diz respeito à diferença de distância para câmera no banco ORL.

TABELA 1 RESUMO DO DESEMPENHO DOS BANCOS UFPB-FOTOS E ORL NO SDRF.

| | Variação | | |
|---|---|---|---|
| Banco | Cortado (100 x 100) | Normalizado em relação à rotação | Normalizado em relação à escala |
| UFPB-Fotos | 96,62% | 97% | 97,12% |
| ORL | 92,5% | 92,5% | 94% |

Desenvolver um módulo normalizador para o SDRF foi uma escolha natural porque um módulo desse tipo é citado como parte integrante dos sistemas de reconhecimento facial e porque está bem documentado na literatura que normalizar imagens em sistemas de classificação, de modo que elas fiquem nas mesmas condições que as imagens de treinamento, leva à melhores resultados. Por isso a sua adição ao SDRF influenciou positivamente no desempenho do reconhecimento facial.

O SDRF+ ainda não é um sistema voltado para ambientes não controlados, mas é um sistema com mais opções de ambientes para uso.

As modificações desenvolvidas neste trabalho não são imunes a erros que podem ser corrigidos em trabalhos futuros.

## REFERÊNCIAS


[1] A. A. Ross, K. Nandakumar e A. K. Jain, Handbook of Multibiometrics, 1ª ed., Secaucus: Springer-Verlag New York, 2006.

[2] N. Thian, "Biometric Authentication System," Penang, Malásia, 2001.

[3] A. K. Jain, B. Klare e U. Park, "Face recognition: Some challenges in forensics," em *2011 IEEE International Conference on Automatic Face & Gesture Recognition and Workshops (FG 2011)*, 2011.

[4] W. Zhao, R. Chellappa, P. J. Phillips e A. P. Rosenfeld, "Face recognition: A literature survey," *ACM Comput. Surv.*, vol. 54, pp. 399-458, Dezembro 2003.

[5] H. Sellahewa e S. Jassim, "Image-Quality-Based Adaptive Face Recognition," *IEEE Transactions on Instrumentation and Measurement*, vol. 59, n. 4, Dezembro 2010.

[6] K. R. Singh, M. A. Zaveri e M. M. Raghuwanshi, "Illumination and Pose Invariant Face Recognition: A Technical Review," *International Journal of Computer Information Systems and Industrial Management Applications*, vol. 02, pp. 29-38.

[7] A. K. Jain e S. Z. Li, Handbook of Face Recognition, 2ª ed., Secaucus: Springer-Verlag New York, 2011.

[8] D. Omaia, "Um Sistema para Detecção e Reconhecimento de Face em Vídeo Utilizando a Transformada Cosseno Discreta," João Pessoa, 2009.

[9] W. Zhao e R. Chellappa, "Image-based Face Recognition: Issues and Methods," em *Image Recognition and Classification: Algorithms, Systems and Applications*, 1ª ed., CRC Press, 2002.

[10] S. T. Chaudhari e A. Kale, "Face Normalization: Enhancing Face Recognition," em *Third International Conference on Emerging Trends in Engineering & Technology (ICETET 2010)*, 2010.

[11] R. Brunelli e T. A. Poggio, "Face recognition: Features versus templates," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 15, n. 10, pp. 1042-1052, Outubro 1993.

[12] S. A. Khayam, "The Discrete Cosine Transform (DCT): Theory and Application," 2003.

[13] Z. M. Hafed e M. D. Levine, "Face Recognition Using the Discrete Cosine Transform," *International Journal of Computer Vision*, n. 3, pp. 167-188, 2001.

[14] F. M. d. S. Matos, "Reconhecimento de Faces Utilizando a Transformada Cosseno Discreta," João Pessoa, 2008.

[15] AT&T Laboratories, "The ORL Database of Face," 2002. [Online]. Available: http://www.cl.cam.ac.uk/research/dtg/attarchive/pub/data/att_faces.zip. [Acesso em Julho 2011].

[16] L. Xia, D. Yongqing, L. Su e H. Chao, "Rapid human-eye detection based on an integrated method," em *Communications and Mobile Computing (CMC), 2010 International Conference on*, 2010.

[17] R. O. Duda, P. E. Hart e D. G. Stork, Pattern Classification, 2ª ed., Wiley-Interscience, 2001.

[18] L. Xia, D. Yongqing, L. Su e H. Chao, "Rapid human-eye detection based on an integrated method," em *Communications and Mobile Computing (CMC), 2010 International Conference on*, 2010.

# An Effective Sequence of Operations for License Plates Recognition

Rafael Seiki Ikeizumi, Almir Olivette Artero, Marco Antônio Piteri
Unesp - Universidade Estadual Paulista
FCT - Faculdade de Ciências e Tecnologia
Presidente Prudente – SP, Brasil
rafaelseiki@gmail.com, {almir, piteri}@fct.unesp.br

Francisco Assis da Silva, Danillo Roberto Pereira
Unoeste - Universidade do Oeste Paulista
FIPP – Faculdade de Informática de Presidente Prudente
Presidente Prudente – SP, Brasil
chico@unoeste.br, dpereira@ic.unicamp.br

***Abstract* –This paper presents a new proposal for license plate recognition, defining an effective sequence of operations that is based on a combination of techniques found in the literature. Moreover, it also proposes some additional improvements that increase hit rates. Initially, the mathematical morphology is adopted in the plate segmentation step as a way to isolate the image region containing the plate. In order to segment the characters we adopt two techniques based on edge detection and projection. For the recognition step we suggest the supervised classification method implemented by Silva et al. [1]. In the experiments we use the image database provided by the Brazilian Center for Physics Research. The results obtained with the proposed sequence showed hit rates of 92.85%.**

***Keywords – License Plate Recognition, Optical Character Recognition, Morphology.***

## I. INTRODUÇÃO

O uso intenso dos veículos nos dias atuais tem gerado uma grande necessidade de monitoramento, seja para um controle de entrada e saída em diferentes locais, ou para o controle do tráfego em vias públicas. Apesar de existir uma grande quantidade de câmeras de monitoramento espalhadas ao longo das rodovias, que capturam as imagens dos veículos que passam por elas, a identificação das placas veiculares ainda continua sendo um problema sem uma solução definitiva, devido às dificuldades decorrentes de diferenças de iluminação, chuva, sujeira sobre a placa, etc. O tempo de reconhecimento também é um fator relevante, pois, de acordo com Dias [2], um usuário treinado é capaz de reconhecer uma placa veicular em aproximadamente cinco segundos. Entretanto, o grande fluxo de veículos observado nas rodovias exige um tempo de reconhecimento inferior a um segundo, o que o torna impraticável ao ser humano. Assim, a necessidade de se processar um grande volume de informação em um curto espaço de tempo constitui um fator importante para a automatização dessa tarefa.

Alguns países já têm adotado a tecnologia para tratar de problemas relacionados ao trânsito. Segundo Guingo [3], os Estados Unidos desenvolveram sistemas com a finalidade de monitorar o tráfego e aplicar multas. No Canadá, a identificação de veículos se destina, principalmente, aos pedágios, enquanto que na Inglaterra, o propósito principal é a fiscalização das vias públicas. No Brasil, o reconhecimento automático veicular se limita a algumas universidades e um número ainda muito pequeno de empresas, que buscam nas pesquisas acadêmicas, soluções para este problema. Assim, nacionalmente, o emprego desse tipo de tecnologia ainda é escasso, apesar da existência de muitas câmeras espalhadas pelas rodovias, ruas e também em diversos estabelecimentos comerciais.

Operadores humanos possuem uma grande habilidade para identificar e reconhecer as placas dos veículos, geralmente procurando-as nas regiões mais prováveis dos veículos, entretanto, esta heurística não é tão simples de ser implementada em sistemas computacionais, por causa dos diferentes formatos de veículos, excesso de informações presentes nos veículos (faróis, vidros, grades, pára-choque, propagandas, etc.). Além disso, as pessoas conseguem tratar muito facilmente as variações de iluminação nas imagens, que ocorrem entre dias ensolarados e nublados ou mesmo durante a noite.

A metodologia apresentada neste trabalho propõe uma combinação de abordagens encontradas na literatura, que foram melhoradas e dispostas em uma sequência que se mostrou bastante eficaz nos experimentos realizados. Para a etapa de localização da placa, foi utilizado o algoritmo de Carvalho [6], que é baseado em morfologia matemática, sendo acrescentada a esta fase a possibilidade de alteração do contraste, como uma alternativa automática para uma adaptação a situações problemáticas. Para a segmentação dos caracteres foram investigados dois métodos: projeção e detecção de contornos. Por fim, o reconhecimento dos caracteres foi efetuado usando o algoritmo de classificação supervisionada proposto por Silva et al. [1]. As melhorias introduzidas na sequência proposta, em complementação às técnicas existentes, foram suficientes para se obter um aumento razoável na taxa de acerto na localização das placas.

As demais seções deste trabalho estão organizadas da seguinte maneira: Na Seção II é realizada uma revisão bibliográfica das técnicas utilizadas na literatura nesta linha de pesquisa; Na Seção III é apresentada a metodologia proposta para o reconhecimento de placas veiculares; Na Seção IV são descritos alguns experimentos, objetivando avaliar o

desempenho da metodologia proposta; Finalmente, na Seção V são apresentadas as conclusões do trabalho.

## II. TRABALHOS RELACIONADOS

Os sistemas de reconhecimento da placa de licenciamento veicular (LPR – *License Plate Recognition*) possuem três etapas principais que são: localização da placa, segmentação dos caracteres e o reconhecimento. Dependendo do autor, algumas etapas adicionais podem ser acrescentadas ao processo. Os trabalhos de Dambros [4] e Dias [2] definem cinco fases que são: pré-processamento, localização da placa, enquadramento da placa, separação dos caracteres e o reconhecimento. O trabalho de Dieguez [5] inclui a fase de aquisição ou captura da imagem através de câmeras e sensores.

Dentre os trabalhos encontrados no âmbito de reconhecimento de caracteres podem-se citar os trabalhos de Suen [7], que efetua o reconhecimento por partes, subdividindo a imagem do objeto em 2, 4 e 6 regiões. O trabalho de Silva e Thomé [8] propõe que os atributos, ou características, sejam adquiridos calculando-se as distâncias, por meio de polígonos de referência, até o contorno da imagem. Park [9] sugere que a imagem que contém o caractere seja separada em $N$ por $M$ células equivalentes e, todas as células são utilizadas para gerar um vetor de atributos. Freitas [10] e Tay et al. [11] usaram, para formar os atributos, as características geométricas dos caracteres, como: centro de gravidade, dimensão, distâncias centro-zona, perfil em oito orientações e quantidade de transições horizontais e verticais na inclinação 45º e -45º. Montaña [12] efetua o reconhecimento de dígitos com o uso de uma rede neural *Perceptron Multilayer* e outra fundamentada no mapa de Kohonen. Também utilizando uma rede neural, Ferreira e Soares [13] reconheceram os caracteres que foram representados por matrizes de dimensões padronizadas. A solução de Polidório e Borges [14] propõe um reconhecimento por análise sintática, utilizando dois grupos de atributos para construir a gramática, que é baseada em: propriedades do traçado do esqueleto dos caracteres; posição do atributo-traço que é colocado em uma grade com quatro quadrantes. O algoritmo também leva em consideração a rotação dos caracteres dentro da grade. O trabalho de Trentini [15] realiza o reconhecimento de caracteres por um classificador baseado em árvores de decisão (*Random Trees*). Conci e Monteiro [16] empregam o reconhecimento através do cálculo do momentos invariantes.

A estratégia adotada neste trabalho, para o reconhecimento dos caracteres, usa o método apresentado por Silva et al. [1], que sugere uma abordagem simples e ágil para analisar o comportamento dos caracteres utilizando somente as transições que ocorrem entre os pixels vizinhos dos caracteres binarizados. Os caracteres são ajustados em uma malha de tamanho previamente definido, com o intuito de observar as transições entre níveis de cinza que ocorrem nos pixels adjacentes. Dessa maneira, uma malha com medidas $m$ x $n$ produz um vetor de transições, ou vetor de atributos, com $m$ x $n$ - 1 posições. Assim, uma lista é gerada com as transições que podem ser admitidas para cada caractere. As transições que ocorrem entre os atributos adjacentes de cada classe são anotadas, ou, ao invés disso, podem ser usadas as transições que não ocorrem em cada classe. Uma vez que as imagens dos caracteres são binárias, as prováveis transições que podem ocorrer entre dois pixels são: 00, 10, 01, 11. A Tabela I mostra um exemplo em que a classe 1, apresenta somente a transição 10 entre os atributos $a_1$ e $a_2$. Na classe 2, observa-se que não ocorrem as transições 10 e 00. Desta forma, para toda classe, são estabelecidas as transições que não ocorrem. Logo, partindo dessa informação, os caracteres a serem reconhecidos são colocados na classe que oferece maior grau de compatibilidade em relação às transições anotadas. Com isso, especifica-se o comportamento das sequências de pixels de cada classe, para que esta informação possa ser usada para classificar outros registros, os quais não se tem conhecimento da classe.

TABELA I. CONJUNTO DE DADOS CONTENDO 4 REGISTROS COM 13 ATRIBUTOS EM 2 CLASSES.

| | $a_1$ | $a_2$ | $a_3$ | $a_4$ | $a_5$ | $a_6$ | $a_7$ | $a_8$ | $a_9$ | $a_{10}$ | $a_{11}$ | $a_{12}$ | $a_{13}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| classe 1 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 0 |
| classe 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 |
| classe 2 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 0 |
| classe 2 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 |

## III. METODOLOGIA PROPOSTA

Este trabalho apresenta uma combinação de abordagens destinadas ao reconhecimento de placas veiculares, encontradas na literatura, dispondo-as em uma sequência eficaz e também incluindo melhorias no sentido de torná-las mais robustas em situações onde ocorrem variações de luminosidade, consistindo em um sistema adaptativo. A sequência é apresentada na Fig. 1.

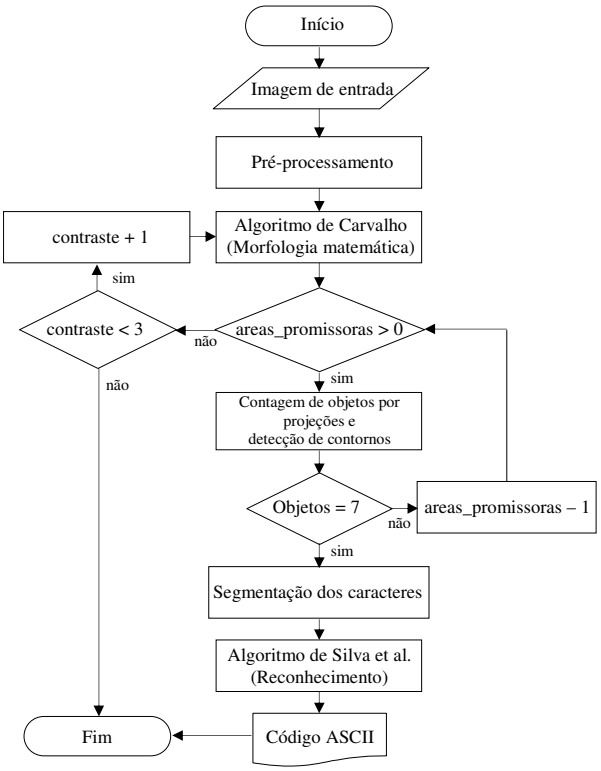



Fig. 1. Sequência de passos da metodologia proposta.

*A. Aquisição das Imagens*

As imagens usadas para avaliar a sequência proposta neste trabalho foram obtidas a partir do banco de imagens fornecido pelo Centro Brasileiro de Pesquisas Físicas (CBPF) [17]. Todas as imagens desta base foram coletadas com pouca variação na distância até os veículos, caracterizando um ambiente aproximadamente controlado.

*B. Pré-Processamento*

O pré-processamento adotado consiste em uma suavização por filtro gaussiano, usando uma janela 5x5, definida empiricamente, que se mostrou satisfatória na atenuação dos ruídos, sem comprometer a nitidez das bordas dos caracteres.

*C. Segmentação da Placa*

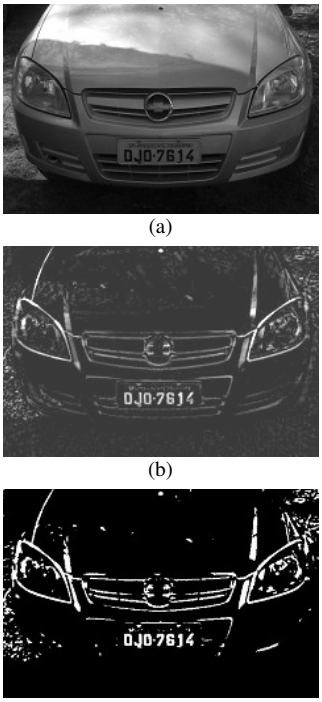
As placas dos veículos brasileiros possuem uma padronização deliberada pela Resolução 231 [18], na qual são definidas as dimensões obrigatórias, em milímetros, das placas dianteira e traseira como apresentado na Fig. 2. Além dos caracteres, outros objetos podem ser encontrados dentro de uma placa veicular, como os quatro furos destinados aos parafusos de fixação e a tarja com a identificação da UF e a cidade, o que pode dificultar o processo de localização e segmentação dos caracteres.

Fig. 2. Representação de uma placa veicular.

A etapa de localização da placa tem o objetivo de encontrar as regiões da imagem que possuem atributos relacionados a este objeto. As características mais exploradas são: o contraste dos caracteres com o fundo da placa, seu formato retangular com fundo homogêneo, a altura e o espaçamento constante dos caracteres, além da quantidade padrão de dígitos. Entre os métodos mais utilizados estão as propostas de morfologia matemática [6], variação tonal [19], detecção de bordas [4] e vizinhança de pixels [20].

Neste trabalho, o método adotado na etapa de localização da placa é o baseado na morfologia matemática [6], que se mostrou menos sensível às condições de luminosidade e conseguiu diminuir de forma expressiva o número de objetos irrelevantes da cena. O algoritmo considera o contraste dos caracteres com o fundo da placa e, através de características marcantes dos caracteres (espessura típica dos caracteres, cor do caractere e da placa, espaçamento entre os caracteres), consegue encontrar o local da placa. Os operadores morfológicos modificam a imagem original, gerando uma nova imagem, por meio de uma interação de um elemento estruturante que possui sua forma e tamanho determinados previamente de acordo com o conhecimento que se tem sobre a geometria do objeto procurado. Para que isto ocorra, as imagens precisam ser obtidas com escalas próximas, como é o caso das imagens disponíveis na base utilizada [17]. Durante a aplicação do filtro morfológico, os atributos geométricos da imagem que possuem dimensões próximas as do elemento estruturante são conservados, enquanto que os demais são suprimidos. Assim, as operações morfológicas tem a capacidade de diminuir o nível de detalhes da imagem, preservando as formas referentes à placa.

O algoritmo proposto por Carvalho [6] começa com uma operação *Top-hat* por fechamento, utilizando um elemento estruturante circular, com um diâmetro da largura do caractere, para que os dígitos sejam borrados (Fig. 3(a)). Logo após, a imagem resultante é subtraída da imagem original, gerando uma imagem com os caracteres realçados. Em seguida, é aplicado o método de binarização de Otsu [21], obtendo uma imagem binarizada, como mostra a Fig. 3(b).




(a) (b) (c)

Fig. 3. a) Imagem original; b) Imagem suavizada e processada pelo operador morfológico *Top-hat*; c) Imagem após a binarização de Otsu.

O próximo passo consiste em aplicar um fechamento com um elemento estruturante horizontal linear, com a largura equivalente à maior distância possível entre dois caracteres, que objetiva transformar o grupo de caracteres em um retângulo branco e, em sequência, são executados alguns passos para eliminar os objetos que não correspondem às características de uma placa. Aplicando-se uma abertura com um elemento estruturante vertical linear, os objetos com altura inferior a altura mínima dos caracteres são eliminados. Em seguida, outra abertura é realizada para eliminar os objetos com altura superior a altura máxima permitida para os caracteres. Por último, os elementos com largura inferior a uma placa veicular também são eliminados por meio de um elemento estruturante horizontal linear e os objetos restantes, as regiões de possível localização da placa, são dilatados. A sequência dos passos relacionados a estas operações é apresentada na Fig. 4.

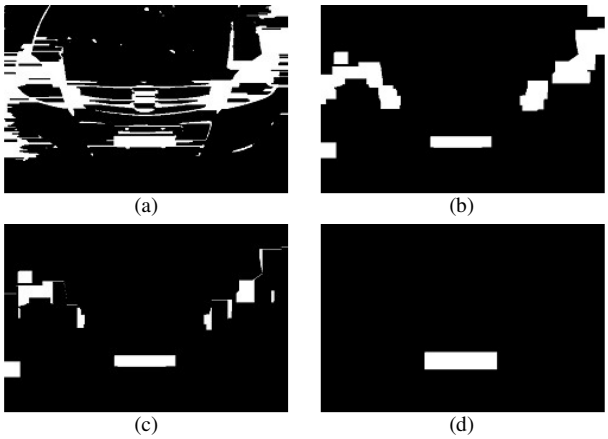


Fig. 4. Resutados das operações morfológicas. (a) fechamento horizontal linear; (b) abertura vertical linear; (c) eliminação de objetos com altura superior a altura máxima; (d) eliminação de objetos com largura inferior a de uma placa veicular e dilatação da área candidata.

Neste trabalho, complementando o processamento proposto por Carvalho [6], cada região candidata é extraída da imagem resultante da segmentação e redimensionada para um tamanho padrão de 330 x 100 pixels. Em seguida, uma nova operação de *Top-hat* é aplicada e uma outra binarização usando Otsu [21] é realizada localmente. Isto é necessário, pois a binarização aplicada em uma região específica produz um resultado melhor em relação à binarização aplicada na imagem inteira e, resultando em uma imagem como a exibida na Fig. 5.

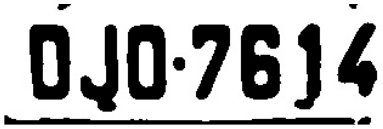


Fig. 5. Imagem após a binarização local de Otsu.

Outro acréscimo proposto neste trabalho trata da remoção de alguns ruídos e elementos indesejados na imagem (Fig. 5), alguns deles decorrentes dos parafusos de fixação e da região entre a placa e o veículo. Para esta finalidade foi aplicada uma nova filtragem morfológica resultando, na imagem apresentada na Fig. 6.

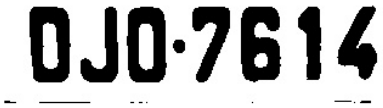


Fig. 6. Imagem após a redução dos elementos indesejados e ruídos.

No caso em que não se encontra uma área candidata, a metodologia proposta neste trabalho propõe uma alteração no contraste da imagem resultante do pré-processamento, e o algoritmo de Carvalho [6] é aplicado novamente. Se, após duas tentativas de alteração de contraste, nenhuma área candidata for encontrada, o processo de localização termina sem encontrar a região da placa.

### *D. Segmentação dos Caracteres*

Para verificar se a região candidata corresponde a uma placa veicular foi realizada uma verificação do número de objetos presentes nessa região por meio de duas técnicas de segmentação: a projeção e a detecção de contornos. Para este fim, foi adotado o método de projeção sugerido por Lu [22], que procura por vales da projeção vertical da imagem binarizada. A princípio, a cadeia de caracteres é enquadrada através das projeções horizontal e vertical (Fig. 7), para então serem separados de acordo com os mínimos regionais (vales) da projeção vertical da imagem resultante. As projeções horizontal e vertical podem ser representadas respectivamente pelas Equações 1 e 2 [23]. As projeções vertical e horizontal da imagem ilustrada na Fig. 6 são apresentadas na Fig. 7.

$$P_h(y) = \sum_{x=0}^{N-1} f(x, y) \tag{1}$$

$$P_v(x) = \sum_{y=0}^{M-1} f(x, y) \tag{2}$$

Após esse procedimento, é realizado o enquadramento dos caracteres pelas projeções vertical e horizontal. Assim, os caracteres são separados de acordo com os seus mínimos regionais (vales) da projeção vertical.

Fig. 7. Projeções dos caracteres apresentados na Fig. 6. (a) projeção vertical; (b) projeção horizontal.

Um dos grandes problemas da segmentação é causado por uma junção indesejada que ocorre por causa da proximidade entre dois caracteres e, a solução proposta neste trabalho para resolver este problema consiste em verificar as larguras dos objetos encontrados e, caso algum deles possua uma largura maior do que o máximo permitido para um caractere, esse objeto é dividido ao meio. Para cada pixel *p* de coordenadas (*x*, *y*), seus 8 vizinhos são analisados, à procura de seus elementos conexos. Essa conectividade está diretamente ligada à relação entre os pixels adjacentes. Assim, os objetos com altura e largura próximas as de um caractere são delimitados como mostra a Fig. 8.

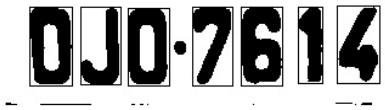
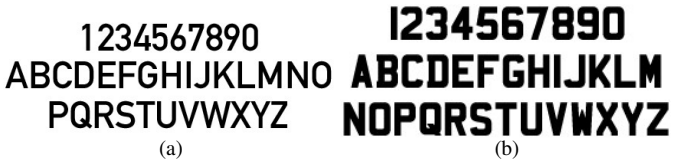


Fig. 8. Caracteres delimitados através da detecção de contornos.

Se forem encontrados sete objetos com dimensões próximas de um caractere, esta região é selecionada para a fase seguinte, em que os caracteres são recortados para o reconhecimento. Caso contrário, a próxima área candidata é escolhida para uma nova verificação. Esgotando-se as áreas candidatas uma alteração de contraste é realizada para uma nova tentativa de segmentação. Caso o limite de transformações de contraste seja atingido sem que os caracteres sejam segmentados, o sistema interrompe o processo finalizando a execução do sistema sem obter sucesso.

### E. Reconhecimento

O algoritmo adotado nesta esta etapa, proposto por Silva et al. [1], consiste em uma classificação supervisionada, sendo o seu treinamento feito com as duas fontes de caracteres adotadas nas placas brasileiras. Até o ano de 2008, a fonte mais adotada pelas placas veiculares era a fonte DIN Mittelschrift. A partir daquele ano a Resolução 231 [18] estabeleceu que as letras e números das placas nacionais deveriam estar em fonte Mandatory. Observando a Fig. 9 é possível comparar as diferenças entre estas duas fontes.



(a) (b)

Fig. 9. Exemplos de caracteres nas duas fontes. (a) caracteres em fonte DIN Mittelschrift; (b) caracteres em fonte Mandatory.

## IV. EXPERIMENTOS

Nesta seção são apresentados os resultados referentes à localização das placas, segmentação e reconhecimento dos caracteres. Conforme mencionado anteriormente, foi usada a base de imagens fornecida pelo CBPF [17], que possui 409 imagens em tons de cinza, de veículos na posição frontal, com placas de caracteres escuros, com uma resolução de 640x240 pixels (resolução espacial média horizontal e vertical de 1.6mm/pixel e 2.8mm/pixel, respectivamente). Na Fig. 10, são apresentadas duas imagens da base CBPF [17].

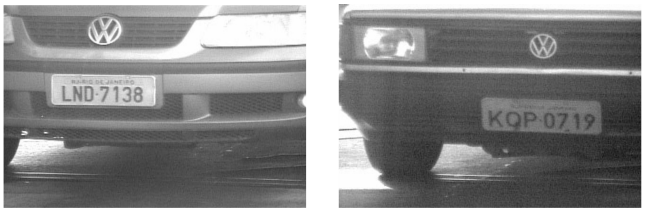



Fig. 10. Exemplos de imagens utilizadas no trabalho (Base CBPF [17]).

### A. Localização das Placas

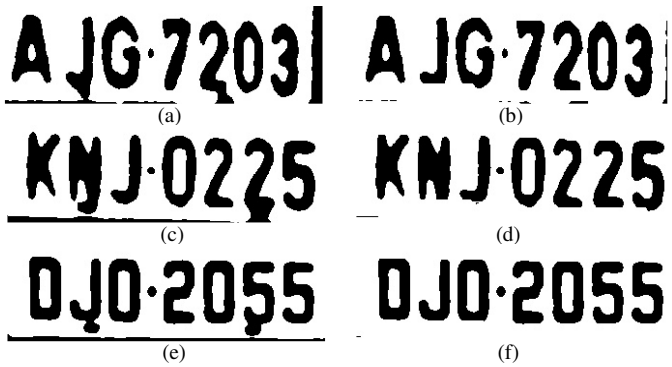
Para realizar a localização das placas foi utilizado o algoritmo de Carvalho [6], sem a alteração de contraste como inicialmente proposto pelo autor. Também foi realizada a localização das placas efetuando-se uma alteração de contraste, de modo a destacar alguns detalhes que não foram detectados com o contraste natural. Os resultados de localização das placas são exibidos na Tabela II, que mostra uma melhora substancial na taxa de acertos, após a inclusão da alteração de contraste.

TABELA II. RESULTADO DA LOCALIZAÇÃO DA PLACA SEM E COM ALTERAÇÃO DE CONTRASTE.

| | **Total** | **Acertos** | **Taxa de acerto** |
|---|---|---|---|
| **sem alteração de contraste** | 409 | 376 | 91,93% |
| **com alteração de contraste** | 409 | 394 | 96,33% |

### B. Segmentação dos Caracteres

A redução de ruídos e elementos indesejados proposta neste trabalho também contribui bastante para a qualidade da segmentação e a taxa de acerto. Três exemplos de placas binarizadas com e sem a aplicação da redução de ruídos e elementos indesejados são apresentados na Fig. 11.



(a) (b) (c) (d) (e) (f)

Fig. 11. Placas segmentadas e binarizadas. (a)(c)(e) placas antes de redução de ruídos; (b)(d)(f) placas após a redução de ruídos.

Após a redução de ruídos e elementos indesejados, foram aplicadas duas técnicas para segmentar os caracteres, que são a detecção de contornos e a projeção, sendo este último, o método que apresentou os melhores resultados. Também foi proposta uma terceira solução, utilizando os dois métodos juntos (híbrida), para melhorar a taxa de acerto nesta etapa. A Tabela III apresenta a porcentagem de acertos usando destas três soluções, em relação ao total de caracteres presentes nas duas bases, que é de 2.863.

TABELA III. RESULTADO DA SEGMENTAÇÃO DOS CARACTERES.

| Método | **Acertos** | **Taxa de acerto** |
|---|---|---|
| **Contornos** | 2.524 | 88,12% |
| **Projeção** | 2.681 | 93,64% |
| **Híbrido** | 2.705 | 94,50% |

### C. Reconhecimento dos Caracteress

Para a realização dos experimentos nesta etapa foram utilizados os 2.705 caracteres de tamanhos variáveis que foram extraídos da fase anterior (Tabela III - método híbrido). Esses caracteres foram enquadrados em uma malha de dimensões 70 x 35, sendo anotadas as transições entre dois pixels adjacentes, gerando-se um total de 2.449 transições (atributos). A fonte predominante dos caracteres é a DIN Mittelschrift, enquanto a minoria pertence ao tipo Mandatory. Diante disso, duas avaliações foram realizadas. Na primeira, o treinamento foi feito com os caracteres das duas fontes, gerando 36 classes, sendo dez delas para os números e 26 para as letras. Portanto, a classe dos caracteres 'A', por exemplo, gerou um vetor de características que reunia tanto as restrições da primeira fonte, quanto da segunda. Nesta primeira avaliação foi alcançada uma taxa de acerto de 91,84%.

Na segunda avaliação, as fontes foram treinadas separadamente, assim, o caractere 'A', por exemplo, passou a ser representado por dois vetores de transições. Com isso, foram geradas 72 classes, onde cada caractere alfanumérico é representado por duas classes, uma da primeira fonte e outra da

segunda. Usando essa abordagem, o resultado mostrou um aumento da taxa de acerto para 92,85%.

Os erros mais comuns ocorrem ao classificar caracteres distintos, mas que possuem tipografias semelhantes, como: 'O' e 'Q'; 'D' e 'O'; 'K' e 'X'; 'N' e 'H'; 'E' e 'F'; 'M' e 'N'. A Fig. 12 apresenta algumas amostras desses caracteres.

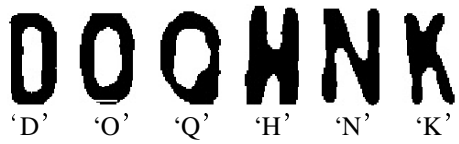



Fig. 12. Alguns caracteres prpícios à falhas devido à semelhança tipográfica.

Outros caracteres não puderam ser reconhecidos por apresentarem baixa qualidade da imagem, conforme mostra a Fig. 13. Nestes casos, os principais problemas encontrados estão relacionados aos ruídos, formas fora do padrão e segmentação parcial. O segundo, terceiro, quinto e sexto dígitos alfanuméricos apresentaram menor qualidade por estarem próximos de outros objetos como os furos da placa e o traço que separa as letras dos números.

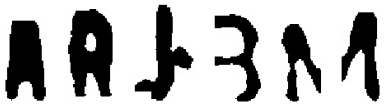


Fig. 13. Exemplos de caracteres com baixa qualidade.

Um trabalho anterior, usando esta mesma base de imagens [24], conseguiu extrair corretamente as placas em 89% das imagens, identificando 88% dos caracteres existentes nas placas e, obtendo uma taxa de reconhecimento de 78% das imagens da base.

## V. CONCLUSÕES

A tarefa de reconhecer as placas veiculares tem se mostrado de difícil solução e, uma solução definitiva, capaz de resolver todos os problemas ainda não é conhecida. Assim, toda e qualquer melhoria no processo, que seja capaz de elevar as taxas de acerto são muito significativas. Geralmente, o processo é realizado por meio de cinco ou seis etapas, que podem ser tratadas com uma grande quantidade de técnicas de processamento de imagens e visão computacional, assim, a determinação de uma sequência eficaz para a realização desta tarefa continua sendo alvo de pesquisas pela comunidade científica. Entre as contribuições deste trabalho merecem destaque: a inclusão do recurso automático de alteração do contraste, que aumentou a taxa de localização das placas de 91,93% para 96,33%; o método híbrido adotado na segmentação dos caracteres, que aumentou a taxa de 88,12% (contornos) e 93,64% (projeção) para 94,50% (híbrido); a utilização do treinamento separado das fontes de caracteres, na etapa de reconhecimento, que elevou a taxa de acertos de 91,84% para 92,85%.

## REFERÊNCIAS


[1] F.A. Silva, A.O. Artero, M.S.V. Paiva, R.L. Barbosa. "Um algoritmo rápido para o reconhecimento de caracteres". In: VII Workshop de Visão Computacional. UFPR,Curitiba, 2011, p. 149–154.

[2] F.G. Dias, "Melhorias para sistemas de reconhecimento da placa de licenciamento veicular. Dissertação – Univ. Est. de Campinas, 2005.

[3] B.C. Guingo, "Kapta - um sistema de reconhecimento automático de placas de veículos e suas aplicações potenciais". In: XVIII Congresso de pesquisa e ensino em transportes. [S.l.: s.n.], 2004.

[4] A.L. Dambros, "Sistema de Reconhecimento de Placas de Veículos Automotores". Trabalho de conclusão de curso, Centro Universitário Feevale, Novo Hamburgo, 2008.

[5] A.A. Dieguez, "Reconhecimento de Caracteres de Placa Veicular Usando Redes Neurais". Trabalho de conclusão de curso, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2010.

[6] J.E.R. Carvalho, "Uma Abordagem de Segmentação de Placas de Automóveis Baseada em Morfologia Matemática". Dissertação (Mestrado) — Universidade Federal Fluminense, Junho 2006.

[7] C.Y. Suen, J. Guo and Z. Li, "Analisis and recognition of alphanumeric handprints by parts". IEEE, Transactions on Systems, Man, and Cybernetics, v. 24, p. 614–631, Abril 1994.

[8] E. Silva e A.C.G. Thomé, "Reconhecimento de caracteres manuscritos utilizando time de redes neurais". In: IV Encontro Nacional de Inteligência Artificial. 2003. p. 13–16.

[9] J. Park, "An adaptative approach to offline handwritten word recognition". IEEE Transactions on Pattern Analysis and Machine Intelligence, v. 24, n. 7, p.920–931, Julho 2002.

[10] C.O.A. Freitas, "Uso de modelos escondidos de Markov para reconhecimento de palavras manuscritas". Curitiba-PR, 2001. Tese de Doutorado.

[11] Y.H. Tay, P.M. Lallican, M. Khalid, C. Viard-Gaudin and S. Knerr, "An offine cursive handwritten word recognition system". In: IEEE Region 10 Conference, Tencon 2001. p. 19–22.

[12] E.G. Montaña, "Digits recognition via neural networks". 2007. Acesso em: 20 de Nov. de 2013. Disponível em: <http://ociotec.com/wpcontent/uploads/2007/07/DigitsRecognition.pdf>.

[13] L.S. Ferreira e L.P. Soares, "Reconhecimento automático de placas veiculares". In: Workshop de Visão Computacional. Goiânia, 2012.

[14] A.M. Polidório e D.L. Borges, "Reconhecimento de caracteres alfanuméricos para identificação de placas de veículos". In: Anais do IX SIBGRAPI, 1996.

[15] V.B. Trentini, L.A.T. Godoy e A.N. Marana, "Reconhecimento automático de placas de veículos". In: Anais do WVC 2010 - VI Workshop de Visão Computacional. Pres. Prudente, 2010. p. 267–272.

[16] A. Conci e L.H. Monteiro, "Reconhecimento de placas de veículos por imagem". Dissertação (Mestrado) — UFF, Niterói, RJ, 2004.

[17] CBPF. Projeto de Reconhecimento de Placas de Veículos Brasileiros. Acesso em: 10 de Agosto de 2013. Disponível em: <http://www.cbpf.br/cat/pdsi/lpr/>.

[18] Brasil. Congresso. Senado. Resolução no 231. Coleção de leis da República Federativa do Brasil, Março, 2007.

[19] F.P.C. Souza, "Localização e leitura automática de caracteres alfanuméricos - Uma aplicação na identificação de veículos". Dissertação (Mestrado) – UFRS, 2000.

[20] I. Felau, "Estudo para Desenvolvimento de um Sistema de Extração de Características de Placas de Veículos Automotores". TCC, Centro Universitário Feevale, Novo Hamburgo, 2008.

[21] N. Otsu, "A threshold selection method from gray-level histograms". IEEE Trans. Sys. Man and Cybernnetics, SMC9, n. 1, p. 62–66, 1979.

[22] Y. Lu, "Machine printed character segmentation - An overview. Elsevier", 1995. 67-80 p.

[23] J.A. Tavares, I.S. Peretta, G.F.M. Lima, K. Yamanaka e M.S. Pais, "SLPTEO e SCORC: Abordagens para segmentação de linhas, palavras e caracteres em textos impressos". In: L.A.P Neves, H. Vieira Neto, A. Gonzaga, Avanços em Visão Computacional, Curitiba, PR: Omnipax, 2012. cap. 13, p. 239-263.

[24] D.S. Abreu, W.A.L. Alves, S.A. Araújo, A.F.H. Librantz, "Uma Abordagem para a Localização e o Reconhecimento de Placas de Licenciamento Veicular por meio de Operadores Morfológicos e Busca por Template", In: VII Workshop de Visão Computacional. UFPR,Curitiba, 2011, p. 79 - 84.

# Evolutionary Optimization Applied for Fine-Tuning Parameter Estimation in Optical Flow-based Methods

Danillo R. Pereira, Francisco Silva
FIPP
UNOESTE - Brazil
{danilopereira,chico}@unoeste.br

João P. Papa, Marco A. Piteri, Almir O. Artero
Dep. of Computing
Unesp - Brazil
papa@fc.unesp.br, {piteri,almir}@fct.unesp.br

Jose Delpiano
Faculty of Engineering
University of the Andes - Chile,
jd@miuandes.cl

***Abstract*—Optical flow methods are accurate algorithms for estimating the displacement and velocity fields of objects in a wide variety of applications, being their performance dependent on the configuration of a set of parameters. Since there is a lack of research that aims to automatically tune such parameters, in this work we have proposed an evolutionary-based framework for such task, thus introducing three techniques for such purpose: Particle Swarm Optimization, Harmony Search and Social-Spider Optimization. The proposed framework has been compared against with the well-known Large Displacement Optical Flow (LDOF) approach, obtaining the best results in four out eight image sequences provided by a public dataset. Additionally, the proposed framework can be used with any other optimization technique.**

***Keywords*—Social-Spider Optimization, Optical Flow, Evolutionary Optimization Methods**

## I. Introduction

Different Optical Flow (OF) techniques have been actively pursued in the last decades, since to estimate the displacement and the velocity of moving objects is a theme of great interest for general applications. New optical flow methods were developed in the last years, leading to some advancements in this area, as well as better matching results. However, the majority of these approaches requires as input a set of parameters for tuning their performance. Generally, these parameters are set empirically, which may be insurmountable for large image-based sequences. In addition, a manual tuning of the models requires a substantial human effort, being also more prone to errors. This can also limit the understanding of how optical flow methods does work in practice.

In the relative literature, most works do not consider the optimal selection of their set of parameters. They rather fine-tune the parameters manually, usually aiming to minimize some error rate metric. An exception can be found in the work of Salmen et al. [1], where the authors used a multi-objective (MO) evolutionary-based approach named "Covariance Matrix Adaption" to obtain more accurate optical flow algorithms. Very recently, Delpiano et al. [2] employed MO Genetic Algorithms for the same context. Although MO optimization algorithms can address multiple fitness functions, they might require more computational burden than mono-objective optimization algorithms, since the set of optimal solutions need to be constrained to the well-known Pareto Optimality.

In this context, it is possible to observe a relative lack of investigation of evolutionary optimization techniques applied for parameter tuning. The task of choosing proper values for OF methods can represent a large combinatorial problem that can be well-fitted by evolutionary algorithms, since such sort of techniques have been proposed in order to avoid traps from local optima. Therefore, this work aims to overcome this lack introducing three methods to the problem of mono-objective parameter tuning in the context of optical flow estimation: Particle Swarm Optimization (PSO) [3], Harmony Search (HS) [4] and Social-Spider Optimization (SSO) [5].

PSO is a method that simulates a swarm intelligence that finds a solution in a search space based on the dynamics of social behavior [3]. Each possible solution of the problem is modeled as a particle swarm that imitates its neighbours based on an objective function. Later on, HS was proposed and inspired from the improvisation process of musicians, more precisely jazz musicians [4]. This approach has been highlighted in the last decade due to the fast convergence and reduced number of calculations. Finally, based on the social dynamics of spiders, Cuevas et al. [5] proposed the Social-Spider Optimization, which considers both male and female spiders as well as their cooperative behaviour for solving optimization tasks. Such technique has demonstrated very promising results, being also so efficient as some state-of-the-art evolutionary-based approaches. The three evolutionary optimization techniques can be seen as heuristic approach that is based on a stochastic process.

Therefore, this paper aims to evaluate PSO, HS and SSO in the context of optical flow parameter estimation, since such techniques have never been employed for such purpose so far. The main contributions of this paper are two-fold: (i) to introduce PSO, HS and SSO for OF parameter estimation, as well as (ii) to compare them in such context. The remainder of this paper is organized as follows. Section II presents a brief theory background regarding optical flow and describes the evolutionary methods used in this work. Section III outlines the methodology employed in this work. Experiments are described in Section IV, and the conclusions are stated in Section V.

## II. Background Theory

### A. Optical Flow

Optical flow is a vector field representing "the distribution of apparent velocities of movement of brightness patterns in an image" [6]. The idea contains two basic assumptions: the "grey value constancy" and the "smooth flow of the intensity values"

between two successive images. Some articles still maintain the grey value constancy (as an example, see [7]), while other works report the necessity to loosen this assumption [8].

The OF constraint, given in Equation 1, is derived from the "grey value constancy" assumption. It relates the spatial and temporal derivatives of a 2D image $f = f(x, y, t)$ and the OF vector $\mathbf{w}$. It has a strong analogy with mass conservation in fluid mechanics, shown in Equation 2, where $\mathbf{w}$ is the fluid speed and $\rho$ is the fluid density. As fluid mass, image intensity is often supposed to remain constant under deformation and motion. However, Equation 1 and Equation 2 would only be equivalent exactly when $\nabla \mathbf{w} = 0$. This condition matches the smooth flow assumption, that is considered when regularizing the flow field:

$$\frac{\partial f}{\partial t} + \mathbf{w}\nabla f = 0, \tag{1}$$

$$\frac{\partial \rho}{\partial t} + \nabla(\rho \mathbf{w}) = 0. \tag{2}$$

The early work presented in [6] stated the need for an extra constraint to compute the optical flow field from an image sequence, and proposed one ad-hoc constraint based on the assumption of flow smoothness. Another early research work [9] proposed the consideration of the OF equation for several neighbouring pixels in order to avoid the need for an extra constraint. More than ten years later, Barron et al. [10] gave a comparison of several OF methods, mainly with respect to their average angular error (AAE) when applied to some image sequences. The tests in that study showed that the method in [9] was one of the most reliable methods at the moment. Several image data sets have been compiled recently for more precise evaluation and comparison of OF methods [11] [12]. Many problems of the original methods have been overcome and the accuracy of the OF methods on the top of the rankings has grown continuously. Several researchers have tried to preserve the discontinuity of natural motion fields [13], overcoming the original assumption of OF smoothness in [6]. After the comparison in [10], there have been further attempts to compare different methods. For example, in [14] the authors show a trade-off between computation time of an algorithm and angular error obtained, given a fixed density. They use operation curves of the OF algorithms to compare them. It is interesting to read a time comparison among OF algorithms given by [15], because they provide a picture of the computational load of OF algorithms. More recently, a group of researchers presented a series of real image sequences with ground truth OF obtained by tracking hidden fluorescent textures [11]. The authors also suggest a method for evaluation of OF algorithms. As a partial conclusion for OF, the accuracy of OF methods has improved and allows for usage in some common computer vision applications.

*Large Displacement Optical Flow:* Let $\mathbf{w} := (u, v)^T$ be the optical flow for a pair of consecutive frames $I_1, I_2$, being such frame pre-smoothed using a Gaussian filter with parameter $\sigma$. The large displacement optical flow (LDOF) method proposed by Brox and Malik [16] solves the energy functional given by:

$$\begin{aligned} E(\mathbf{w}) \quad = \quad & E_{color}(\mathbf{w}) + \gamma E_{gradient}(\mathbf{w}) \\ & + \alpha E_{smooth}(\mathbf{w}) + \beta E_{match}(\mathbf{w}, \mathbf{w_1}) \\ & + E_{desc}(\mathbf{w_1}), \end{aligned} \tag{3}$$

where the term $E_{color}$ represents the common assumption of grey value or color constancy; $E_{gradient}$ represents gradient constancy, which is invariant to an uniform illumination change; $E_{smooth}$ enforces regularity of the resulting optical flow; $E_{match}$ an energy related to point correspondences; minimization of $E_{desc}$ assures descriptor matching. The quantity $\mathbf{w_1}$ is an auxiliary variable which allows to integrate descriptor matching into a continuous approach. The implementation available for LDOF[1] allows the user to change the parameters $\sigma, \alpha, \beta, \gamma$. This set of parameters influence significantly on the accuracy (consequently on the error metrics) and at runtime.

### B. Particle Swarm Optimization

The PSO can be seen as a search algorithm based on stochastic processes [3], where the learning of social behavior allows each possible solution (particle) "fly" within that space (swarm) looking for other particles that have the best features, and thus minimizing or maximizing the objective function.

Each particle has a memory for storing its best local solution (local maxima or minima) and the global best solution (global maximum or minimum). Taking this into account, each particle has the ability to imitate others that provide the best positions in the swarm. This mechanism can be summarized in three principles: (i) evaluation, (ii) comparison and (iii) imitation. Each particle can evaluate others within your neighborhood through some objective function, it can compare with your own value, and finally decide whether it is a good choice to imitate it.

The swarm is modeled as a multidimensional space $\mathbb{R}^n$, where each particle $x_i = (p_i, v_i) \in \mathbb{R}^n$ has two main features: (i) position $p_i$ and (ii) velocity $v_i$. The best local $\widehat{p_i}$ and global $\widehat{s}$ solution (position in the swarm) are also known. After setting the size of the swarm (the number of particles), each particle is initialized with random values for velocity and position. Each particle is then evaluated with respect to some objective function and its local maxima/minima is updated. The global maximum/minimum value is updated with the particle that reached the best position in the swarm. This process is repeated until some convergence criterion. The position and velocity of the particle $x_i$ at time step $t + 1$ are updated by Equations 4 and 5, respectively:

$$p_i(t + 1) = p_i(t) + v_i(t), \tag{4}$$

and

$$v_i(t + 1) = \Psi v_i(t) + c_1 r_1(\widehat{p_i} - x_i(t)) + c_2 r_2(\widehat{s} - p_i(t)) \tag{5}$$

where $\Psi$ is the inertia force that controls the interaction power between particles, and $r_1, r_2 \in [0, 1]$ are random variables that give the idea of stochasticity for the PSO method. The constants $c_1$ and $c_2$ are also used to guide the particles (input parameters for the algorithm).

### C. Harmony Search

Harmony Search (HS) is a meta-heuristic method that is based on the improvisation process of musicians searching for a good harmony [4]. This method has a theoretical stochastic

[1] http://lmb.informatik.uni-freiburg.de/resources/software.php, accessed on April 7th, 2014.

background. Each iteration of this approach generates a new harmony vector $h_{new} = (h^1_{new}, h^2_{new}, ..., h^n_{new})$ based on memory considerations, small adjustments, and randomization, i.e., music improvisation ($n$ is the number of decision variables).

The idea of the memorization step is to model the process of creating songs, in which the musician can use his/her memories of good musical notes to create a new song. This process is modeled by the Harmony Memory Considering Rate (HMCR) parameter; mathematically:

$$h^j_{new} \leftarrow \begin{cases} h^j_{new} \in \left\{h^j_1, \ldots, h^j_m\right\} & \text{with probability HCMR} \\ h^j_{new} \in \mathcal{S}_j & \text{with probability (1-HCMR),} \end{cases} \quad (6)$$

where $m$ and $\mathcal{S}_j$ are the number of harmonies and the set of ranges for each decision variable j, respectively. Therefore, HMCR $\in [0, 1]$ is the probability of choosing one value from the historic values stored in the harmony memory, and $(1-$HMCR$)$ is the probability of randomly choosing one feasible value. Further, every component $j$ of the new harmony vector $h_{new}$ is examined to determine whether it should be pitch-adjusted, which is controlled by the Pitch Adjusting Rate (PAR) variable:

$$h^j_{new} \leftarrow \begin{cases} \text{Yes}, & \text{with probability PAR} \\ \text{No}, & \text{with probability (1-PAR).} \end{cases} \quad (7)$$

The pitch adjustment for each instrument is often used to improve solutions and to avoid local optima. This mechanism concerns shifting the neighboring values of some decision variable in the harmony. As such, if the pitch adjustment decision for the decision variable $h^j_{new}$ is Yes, then $h^j_{new}$ is replaced as follows:

$$h^j_{new} \leftarrow h^j_{new} + \psi_j b, \quad (8)$$

where $b$ is an arbitrary distance (bandwith) for the continuous design variable, and $\psi_j \in [0, 1]$ is an ad-hoc parameter.

In the recent years, several researches have focused on developing variants from the traditional HS [17], [18]. Some of them propose ways to set the HS parameters dynamically, and others propose new improvisation schemes. In our implementation, we employed the Novel Global Harmony Search (NGHS) [18], since this variation has shown to be the most accurate. The NGHS depends on a single parameter $P$ that denotes the probability of occurrence of an improvisation during a new particle's creation.

### *D. Social-Spider Optimization*

Social-Spider Optimization is based on the cooperative behavior of social-spiders [5], and it takes into account two genders of search spiders: males and females. Depending on the gender, each agent is conducted by a set of different operators emulating a cooperative behavior in a colony. The search space is assumed as a communal web and a spider's position represents an optimal (near optimal) solution. An interesting characteristic of social-spiders is the female-biased population. The number of male spiders hardly reaches 30% of the total colony members. The number of females $N_f$ is randomly selected within a range of 65-90% of the entire population $N$, being calculated as follows:

$$N_f = [(0.9 - \xi 0.25)N], \quad (9)$$

where $\xi$ is a random number between $[0, 1]$. The number of male spiders $N_m$ is given by:

$$N_m = N - N_f. \quad (10)$$

Each spider $i$ receives a weight $\phi_i$ according to the fitness value of the solution:

$$\phi_i = \frac{fitness_i - worst}{best - worst}, \quad (11)$$

where $fitness_i$ is the fitness value obtained by the evaluation of the $ith$ spider's position $i = 1, 2, \ldots, N$. The $worst$ and $best$ mean the worst fitness value and best fitness value of the entire population, respectively.

The communal web is used as a mechanism to transmit information among the colony members. The information is encoded as small vibrations and depends on the weight and distance of the spider which has generated them:

$$V_{i,j} = \phi_j e^{-d^2_{i,j}}, \quad (12)$$

where $d_{i,j}$ is the Euclidean Distance between the spider $i$ and $j$. We can consider three special relationships:

- Vibrations $V_{i,c}$ are perceived by the spider $i$ as a result of the information transmited by the member $c$ who is the nearest member to $i$, and possesses a higher weight $\phi_c > \phi_i$;
- The vibrations $V_{i,b}$ perceived by the spider $i$ as a result of information transmitted by the spider $b$ holding the best weight of the entire population;
- The vibrations $V_{i,f}$ perceived by the spider $i$ as a result of the information transmitted by the nearest female $f$.

Social-spiders perform cooperative interaction over other colony members depending on the gender. In order to emulate the cooperative behavior of the female spider, a new operator is defined in Equation 13. The movement of attraction or repulsion $\varphi_i$ of a female spider $i$ at time step $t+1$ is developed over other spiders according to their vibrations, which are emitted over the communal web:

$$\varphi_i(t+1) = \begin{cases} \varphi_i(t) + \alpha * V_{i,c} * (s_c - \varphi_i(t)) + \beta * V_{i,b} * \\ (s_b - \varphi_i(t)) + \gamma * (rand - \frac{1}{2}) \\ \text{if } r_m < PF; \\ \varphi_i(t) - \alpha * V_{i,c} * (s_c - \varphi_i(t)) - \beta * V_{i,b} * \\ (s_b - \varphi_i(t)) + \gamma * (rand - \frac{1}{2}) \\ \text{if } r_m \geq PF, \end{cases} \quad (13)$$

where $r_m$, $\alpha$, $\beta$, $\gamma$ and $rand$ are uniform random numbers between $[0, 1]$, and $s_c$ and $s_b$ represent the nearest member to $i$ that holds a higher weight and the best spider of the entire population, respectively.

The male spider population is divided into two classes: dominant and non-dominant. The dominant class spider has better fitness in comparison to non-dominant, and they are attracted to the closest female spider in the communal web. In the other hand, non-dominant male spiders tend to concentrate in the center of the male population as a strategy to take advantage of resources that are wasted by dominant males. The movement of male spiders is given by:

$$\delta_i(t+1) = \begin{cases} \delta_i(t) + \alpha * V_{i,f} * (s_f - \delta_i(t)) + \gamma * (rand - \frac{1}{2}) \\ \text{if } \phi_{N_f+i} > \tilde{\phi}; \\ \delta_i(t) + \alpha * \left( \frac{\sum_{h=1}^{N_m} \delta_h(t) * \phi_{N_f+h}}{\sum_{h=1}^{N_m} \phi_{N_f+h}} \right) \\ \text{if } \phi_{N_f+i} \leq \tilde{\phi}. \end{cases} \quad (14)$$

where $s_f$ represents the nearest female spider to the male spider $i$ and $\tilde{\phi}$ is the median weight of male spider population. Thus, the reader can observe we have distinct movement equations for male and female spiders. Mating is performed by dominant males and female members in a social-spider colony. Considering $r$ (calculated by Equation 15) as being the radius, when a dominant male spider locates female members inside $r$, it mates, forming a new brood:

$$r = \frac{\sum_{j=1}^{n} l_j^{high} - l_j^{low}}{2n}, \quad (15)$$

where $n$ is the dimension of the problem, and $l_j^{high}$ and $l_j^{low}$ are the upper and lower bounds, respectively. Once the new spider is formed, it is compared to the worst spider of the colony. If the new spider is better, the worst spider is replaced by the new one.

## III. Methodology

This section describes the experimental setup as well as the results concerning the optimization methods using image sequences and their respective ground truths from the Middlebury[2] dataset [11], which has been frequently used for the evaluation of different OF methods [11], [16]. The Middlebury dataset contains 8 synthetic and laboratory sequences with a dense ground truth, which can be observed in Figure 1.

We have employed the LDOF (Section II-A) together with our implementation of SSO, PSO and HS evolutionary optimization approaches. In order to compare the optimization methods, we computed the mean of End Point Error (EPE) [19] and Average Angular Error (AAE) [10] obtained by 5 executions for each of them. Such quantitative evaluation approaches are commonly used by the optical flow estimation community. As we are employing a single-objective optimization, we chose to optimize the value of EPE (we believe this approach is more feasible to be employed here).

The proposed methodology can be divided in two rounds, as depicted in Figure 2. In the first round, for each image sequence we used different parameters obtained by each of the evolutionary optimization methods, unlike the methodology used by LDOF baseline [16], that employed a single parameter set for all image sequences. In the second round, we executed the proposed optimization framework over LDOF for all eight sequences using the set of parameters that achieved the best results for each evolutionary optimization method obtained in the first round, being this set of parameters the one that has achieved the best overall performance (average EPE over all eight sequences).

The parameters used for each evolutionary optimization algorithm are displayed in Table I. Notice for all optimization algorithms we used 20 agents and 200 iterations. It is important to shed light over that such values have been set based on previous experiments [5], [20].

| Algorithm | Parameters |
|---|---|
| PSO | $c_1 = c_2 = 2.0$ and $\psi = 0.9$ |
| SSO | $PF = 0.5$ |
| HS | $P = 0.1$ |

TABLE I. The parameters used for each optimization algorithm.

## IV. Experiments

This section presents the results obtained by PSO, HS and SSO for optical flow parameter optimization purposes. We would like to stress we did not consider the runtime, since our goal is to minimize the EPE metric. Furthermore, the parameters to be optimized have a strong influence on both EPE and runtime. Thus, in order to find a tradeoff between the execution time minimization and EPE, it is necessary to apply multi-objective optimization methods. Additionally, we did not use statistical methods for ranking the tests, since their standard deviations are too small for that.

In regard to the first round of experiments, Table II shows the results obtained applying PSO, HS and SSO using a different set of parameters for each image sequence. In regard to EPE measure, it is possible to observe that SSO achieved the best results in four out eight Middleburry sequences, and PSO obtained the remaining top four best values. In addition, HS did not achieve the best result in any of the tests, and obtained the second best performance in only one of the sequences (Grove2). If one consider the baseline (Table III), the proposed approaches have achieved better results in five sequences (Dimetrodon, Hydrangea, RubberWhale, Urban2 and Venus). However, it is important to point out we have used a different set of parameters for each image sequence, while the baseline used a single set of parameters for all sequences.

Therefore, in order to make a fair comparison against with the baseline, we performed a second round of experiments, so that we used a unique set of parameters for all image sequences. Table III compares the results obtained by the proposed approach using PSO, HS and SSH against with the LDOF baseline. Figure 3 shows the histograms of the results obtained for EPE and AAE using the best selected set of parameters. Figure 3(b) plots the comparison between the evolutionary optimization methods and the LDOF baseline [16].

In the second round of experiments, we achieved better results than LDOF baseline in four sequences (Dimetrodon, Hydrangea, Urban2 and Venus). Is worth noting that we did not perform a global optimization using the eight sequences, but an individual optimization in each sequence and posteriorly for

[2]http://vision.middlebury.edu/flow/data/

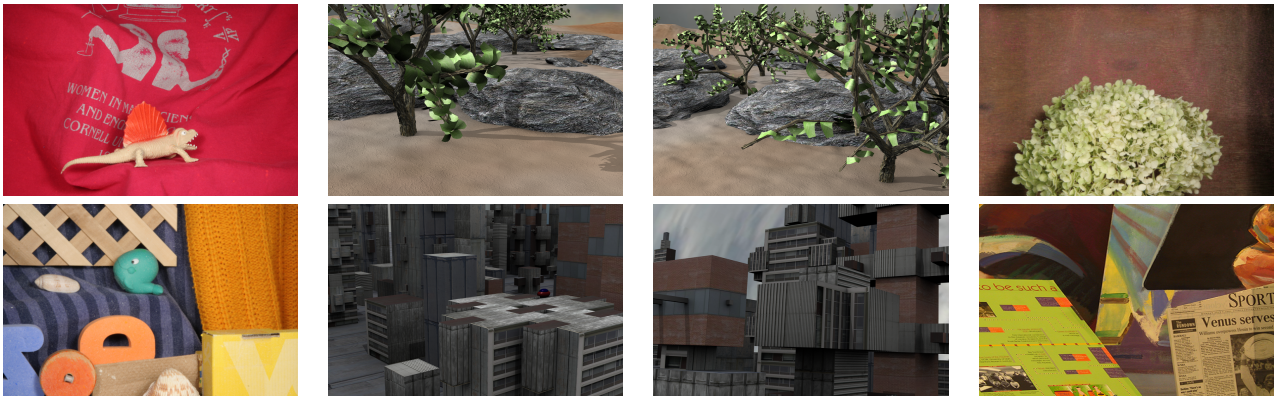


Fig. 1. Images from the Middlebury Dataset used in the experiments. From left to rigth: Dimetrodon, Grove2, Grove3, Hydrangea, Urban2, Urban3, RubberWhale and Venus.

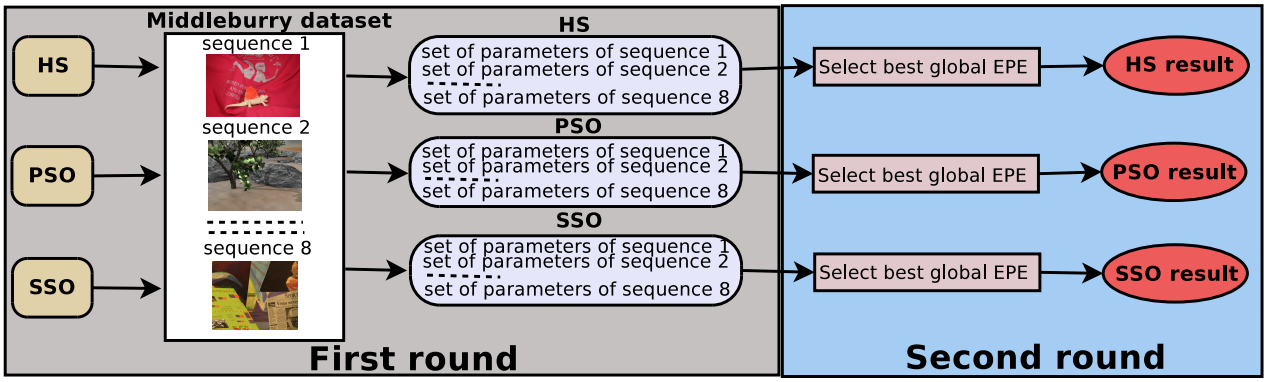



Fig. 2. Proposed methodology for optical flow parameters estimation: notice one can employ any other optimization technique.

| Sequence | PSO | | HS | | SSO | |
|---|---|---|---|---|---|---|
| | EPE | AAE | EPE | AAE | EPE | AAE |
| Dimetrodon | **0.08** ± 0.00 | 1.69 ± 0.00 | 0.09 ± 0.00 | 1.78 ± 0.05 | 0.09 ± 0.00 | 1.75 ± 0.01 |
| Grove2 | **0.17** ± 0.00 | 2.57 ± 0.01 | 0.17 ± 0.00 | 2.60 ± 0.00 | 0.17 ± 0.00 | 2.60 ± 0.00 |
| Grove3 | 0.67 ± 0.00 | 6.24 ± 0.04 | 0.69 ± 0.00 | 6.35 ± 0.06 | **0.66** ± 0.02 | 6.22 ± 0.09 |
| Hydrangea | 0.16 ± 0.00 | 2.03 ± 0.00 | 0.17 ± 0.00 | 2.08 ± 0.01 | **0.16** ± 0.00 | 2.01 ± 0.02 |
| RubberWhale | **0.10** ± 0.00 | 3.41 ± 0.04 | 0.11 ± 0.00 | 3.69 ± 0.03 | 0.10 ± 0.00 | 3.48 ± 0.02 |
| Urban2 | 0.30 ± 0.00 | 2.62 ± 0.02 | 0.34 ± 0.00 | 2.85 ± 0.08 | **0.29** ± 0.00 | 2.54 ± 0.01 |
| Urban3 | 0.48 ± 0.04 | 4.54 ± 0.27 | 0.63 ± 0.01 | 5.20 ± 0.28 | **0.44** ± 0.00 | 4.26 ± 0.01 |
| Venus | **0.28** ± 0.00 | 4.34 ± 0.01 | 0.34 ± 0.01 | 5.42 ± 0.40 | 0.29 ± 0.01 | 4.62 ± 0.43 |

TABLE II. QUANTITATIVE ANALYSIS FOR OPTICAL FLOW OPTIMIZATION USING PSO, HS AND SSO. THE BEST EPE VALUES OBTAINED FOR EACH SEQUENCE ARE IN BOLD.

| Sequences | HS | PSO | SSO | LDOF baseline |
|---|---|---|---|---|
| Dimetrodon | 1.78 | 2.21 | **1.71** | 1.82 |
| Grove2 | 3.12 | 3.04 | 2.87 | **2.09** |
| Grove3 | 6.99 | 6.82 | 6.78 | **5.59** |
| Hydrangea | 2.73 | 3.04 | **2.12** | 2.32 |
| RubberWhale | 4.29 | 4.59 | 4.15 | **3.77** |
| Urban2 | 3.03 | 2.94 | **2.70** | 3.99 |
| Urban3 | 4.92 | 4.27 | 5.00 | **2.28** |
| Venus | 5.76 | 4.47 | **4.19** | 5.19 |

TABLE III. THE BEST VALUES OF AAE OBTAINED BY THE LDOF BASELINE [16]. THE BEST AAE VALUES OBTAINED FOR EACH SEQUENCE ARE IN BOLD.

each evolutionary optimization approach we selected the best global hyper-parameter set. Therefore, the methodology employed in this work can be defined as a semi-global approach. Furthermore, we chose to optimize the EPE and not the AAE, as the LDOF baseline did. However, it is possible to observe the proposed evolutionary-based framework can outperform LDOF for some situations.

## V. CONCLUSIONS

In this work, we dealt with the problem of tuning parameters in optical flow-based environments, more precisely LDOF. The proposed framework consists of a bunch of evolutionary optimization techniques that model the problem of parameters' estimation as an optimization task. We have employed PSO, SSO and HS to address this problem, being the former two

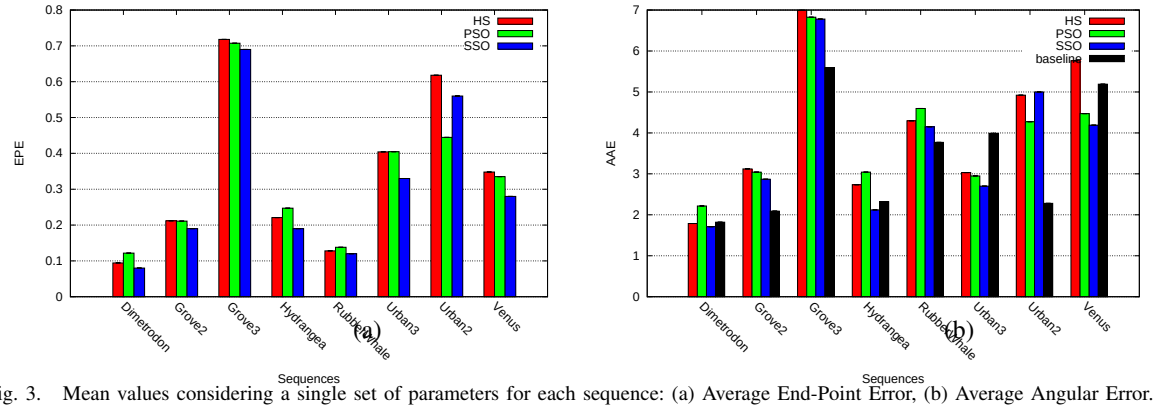



Fig. 3. Mean values considering a single set of parameters for each sequence: (a) Average End-Point Error, (b) Average Angular Error.

approaches the most accurate, which also outperformed the compared baseline (LDOF) in four out eight datasets. Given the fact that we did not use evolutionary methods in the global-way, i.e., we aimed to optimize each sequence separately and then considering the best overall performance, the results were quite satisfactory.

The methodology described in this work provide an automatic way for parameters estimation, thus avoiding the use of brute-force methods or even manually set them. Notice the proposed approach is mono-objective, which means only one objective function is optimized, i.e., EPE in this work. In our tests, we disregard the runtime due to the fact that the chosen parameters greatly influence both the runtime as well as the error rate. In order to establish a trade-off among computational load and error rate minimization, one must use multi-objective optimization methods, being that one our future work.

## ACKNOWLEDGMENTS

The authors are grateful to FAPESP grant #2013/20387-7, and CNPq grants #470571/2013-6 and #303182/2011-3.

## REFERENCES

[1] J. Salmen, C. Lukas, and C. Igel, "Real-time estimation of optical flow based on optimized haar wavelet features," in *EMO*, Ricardo H. C. Takahashi, Kalyanmoy Deb, Elizabeth F. Wanner, and Salvatore Greco, Eds. 2011, Lecture Notes in Computer Science, Springer.

[2] J. Delpiano, L. Pizarro, R. Verschae, and J. Ruiz del Solar, "Multi-objective optimization for characterization of optical flow methods," in *Proceedings of the International Conference on Computer Vision Theory and Applications*, 2014, pp. 566–573.

[3] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference on Neural Networks*, 1995, pp. 1942–1948.

[4] Z.W. Geem, *Music-Inspired Harmony Search Algorithm: Theory and Applications*, Springer Publishing Company, Incorporated, 2009.

[5] E. Cuevas, M. Cienfuegos, D. Zaldívar, and M. Pérez-Cisneros, "A swarm optimization algorithm inspired in the behavior of the social-spider," *Expert Systems with Applications*, vol. 40, no. 16, 2013.

[6] Berthold K. P. Horn and Brian G. Schunck, "Determining optical flow," *ARTIFICAL INTELLIGENCE*, vol. 17, pp. 185–203, 1981.

[7] A. Bruhn, J. Weickert, and C. Schnörr, "Lucas/kanade meets horn/schunck: Combining local and global optic flow methods," *International Journal of Computer Vision*, vol. 61, no. 3, 2005.

[8] N. Cornelius and T. Kanade, "Adapting optical-flow to measure object motion in reflectance and x-ray image sequences," in *Proc. Of the ACM SIGGRAPH/SIGART Interdisciplinary Workshop on Motion: Representation and Perception*, NY, USA, 1986, Elsevier North-Holland.

[9] B. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proceedings of the 7th international joint conference on Artificial intelligence*, 1981.

[10] J. L. Barron, D. J. Fleet, and S. S. Beauchemin, "Performance of optical flow techniques," *INTERNATIONAL JOURNAL OF COMPUTER VISION*, vol. 12, pp. 43–77, 1994.

[11] S. Baker, D. Scharstein, J. P. Lewis, S. Roth, M. Black, and R. Szeliski, "A database and evaluation methodology for optical flow," *Int. J. Comput. Vision*, vol. 92, no. 1, pp. 1–31, Mar. 2011.

[12] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[13] A. Bruhn and J. Weickert, "A multigrid platform for real-time motion computation with discontinuity-preserving variational methods," *International Journal of Computer Vision*, vol. 70, pp. 257–277, 2006.

[14] H. Liu, T. Hong, M. Herman, and R. Chellappa, "Accuracy vs. efficiency trade-offs in optical flow algorithms," in *COMPUTER VISION AND IMAGE UNDERSTANDING*. 1996, pp. 271–286, Academic Press.

[15] D. Gibson and M. Spann, "Robust optical flow estimation based on a sparse motion trajectory set.," *IEEE Transactions on Image Processing*, vol. 12, no. 4, pp. 431–445, 2003.

[16] T. Brox and J. Malik, "Large displacement optical flow: descriptor matching in variational motion estimation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, pp. 500–513, 2011.

[17] P. Yadav, Rajesh Kumar, S.K. Panda, and C.S. Chang, "An intelligent tuned harmony search algorithm for optimisation," *Information Sciences*, vol. 196, pp. 47–72, 2012.

[18] D. Zou, L. Gao, J. Wu, and S. Li, "Novel global harmony search algorithm for unconstrained problems," *Neurocomputing*, vol. 73, no. 216–18, pp. 3308–3318, 2010.

[19] M. Otte and H.-H. Nagel, "Optical flow estimation: Advances and comparisons," in *Computer Vision ? ECCV '94*, Jan-Olof Eklundh, Ed., vol. 800 of *Lecture Notes in Computer Science*, pp. 49–60. Springer Berlin Heidelberg, 1994.

[20] D. Zou, L. Gao, J. Wu, and S. Li, "Novel global harmony search algorithm for unconstrained problems," *Neurocomputing*, vol. 73, no. 16-18, pp. 3308–3318, 2010.

# A Real-time approach for hand-gesture recognition based on computer vision and artificial neural networks

Luan Caius Ramos, Flavio de Barros Vidal, Bruno Macchiavello, Alexandre Zaghetto
Departamento de Ciencia da Computação
Universidade de Brasília - Distrito Federal - Brasil
Email: luancaius@hotmail.com, fbvidal@cic.unb.br, bruno@image.unb.br, zaghetto@image.unb.br

***Abstract*—This paper proposes a real-time hand-gesture recognition method based on artificial neural networks. It is divided into three steps: (a) hand detection; (b) descriptors extraction; and (d) classification using an artificial neural network. Results show that the proposed method can correctly identify 10 different hand-gestures, presenting an F-measure of 0.972. In addition, our implementation achieved the rate of 29 classifications per second.**

## I. Introdução

A utilização de informações provenientes de imagens digitais tem se tornado cada vez mais requerida nos dias atuais, como em aplicações em codificação de imagens e vídeo, sistemas de video-monitoramento, imageamento remoto, biometria, imagens médicas, jogos, entre outras. Diversos são os desafios ainda a serem explorados, sendo que um dos maiores consiste em fazer o computador reconhecer adequadamente o que representa cada elemento dentro de uma imagem, desafio este que tradicionalmente é foco de diversos estudos na área de Visão Computacional [1]. Parte da dificuldade pode ser atribuída à elevada complexidade computacional dos algoritmos elaborados para essa finalidade [2].

Podemos considerar o reconhecimento de posturas para gestos manuais como um dos requisitos a ser atendido com o objetivo de se aperfeiçoar a interação homem-máquina. Em uma sociedade que cada vez mais dispensa o uso de dispositivos de entrada tradicionais, como o teclado e o mouse, ter um sistema confiável e robusto que possa adequadamente reconhecer posturas para gestos manuais é um passo importante para a consolidação de sistemas computacionais modernos [3]–[8].

Assume-se ainda, que as posturas devam ser realizadas da maneira mais natural possível, ou seja, sem a utilização de luvas ou marcações especiais que poderiam tornar a experiência desconfortável [9] ao usuário. Isso dificulta em muito a tarefa de classificação do gestos. A mão humana tem, por exemplo, a cor e textura muito próxima da cor do punho e do braço, e para se reconhecer uma postura, é essencial antes delimitar a região de interesse para que um classificador possa ser, então, aplicado. Se a região de interesse não for bem escolhida, é provável que o classificador falhe e, portanto, responda de forma equivocada. Diversas pesquisas surgiram utilizando técnicas diferentes de segmentação, visando à obtenção da cor de pele de forma rápida e precisa [10].

Outro grande desafio é utilizar os dados contidos na região de interesse, que idealmente contém os *pixels* da mão, de forma a conseguir classificar a postura gestual representada. Para isso, faz-se necessário definir um conjunto de descritores que seja suficiente na distinção dentre posturas.

É muito importante ressaltar que o objetivo não é propor um método genérico e sem conhecimento *a priori* do gestos manuais, que seja capaz de classificar todo e qualquer conjunto de posturas manuais. Ao invés disso, está sendo proposto um algoritmo capaz de realizar a classificação de posturas manuais em tempo real, com recursos computacionais de baixo custo. Assumimos que as posturas manuais e seus respectivos significados podem ser definidos *a priori*, de forma a evitar um número excessivo de classificações equivocadas. Vários tipos de classificadores poderiam ter sido utilizados, mas para avaliar a capacidade do processo de classificação proposto em tempo, utilizou-se uma rede neural artificial do tipo *perceptron* multicamada [11].

Para se alcançar os objetivos propostos neste trabalho, as demais Seções foram organizadas como: Na Seção II é apresentado o método desenvolvido e suas principais características; nas Seções III e IV são descritos os resultados obtidos, métricas para validação utilizadas e as conclusões e trabalhos futuros, respectivamente.

## II. Método Proposto

O método proposto pode ser dividido em 3 etapas (Fluxograma da Figura 1): (II-A) Detecção da mão; (II-B) Extração de descritores; e; (II-C) Classificação por meio de redes neurais artificiais. Essas três etapas serão detalhadas nas seções que se seguem.

### A. Detecção da Mão

A primeira etapa consiste em se localizar a mão dentro de cada *frame* do vídeo adquirido por meio de uma câmera de baixa resolução ($640 \times 480$ *pixels*). Essa etapa pode ser subdividida em 7 passos: (II-A1) segmentação por cor de pele; (II-A2) operação morfológica de abertura; (II-A3) rotulagem e reclassificação de componentes conexos; (II-A4)

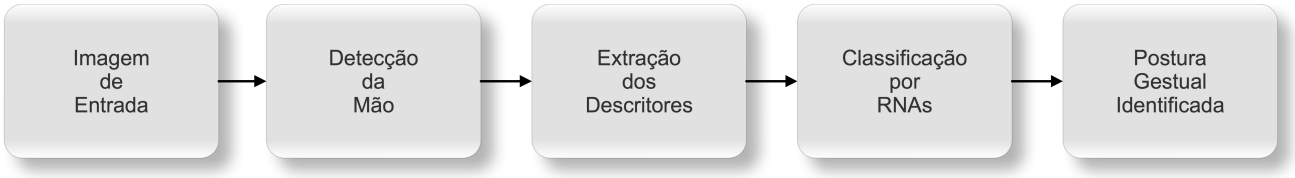



Fig. 1. Fluxograma da metodologia proposta.

preenchimento de buracos; (II-A5) localização da mão; (II-A6) definição da janela de interesse; e (II-A7) definição da lateralidade da mão.

*1) Segmentação por Cor de Pele:* A Fig. 2 ilustra um *frame* de vídeo típico. Com o objetivo de se localizar a região de interesse, inicialmente realiza-se a segmentação do *frame* por cor de pele [10].

Fig. 2. *Frame* de vídeo típico.

Considerando o espaço de cores RGB [12], são classificados como pele apenas os *pixels* que satisfazem aos seguintes critérios de cor:

$$R > 95 \text{ e } G > 40 \text{ e } B > 20 \text{ e}$$
$$max(R,G,B) - min(R,G,B) > 15 \text{ e}$$
$$|R - G| > 15 \text{ e } R > G \text{ e } R > B.$$

Os demais *pixels* serão classificados como fundo. A imagem da Fig. 3 mostra o resultado da segmentação. Trata-se de uma imagem binária, onde a cor branca representa regiões candidatas a conterem a mão e a cor preta representa o fundo.

*2) Operação Morfológica de Abertura:* Em seguida, realiza-se uma operação de abertura morfológica ou seja, uma erosão seguida de uma dilatação, visando à diminuição de ruídos presentes na imagem binária segmentada. O resultado dessa operação morfológica é mostrado na Fig. 4. Pode-ser perceber que pequenos conjuntos de *pixels* brancos foram eliminados.

Fig. 3. Imagem Segmentada.

Fig. 4. Operação de Abertura.

*3) Rotulagem e Reclassificação de Componentes Conexos:* Nesse passo, rotula-se os componentes conexos e realiza-se uma filtragem por área, ou seja, reclassifica-se como fundo os componentes conexos cujas áreas sejam menores que um determinado limiar mínimo. Aqui foi empiricamente estabelecido como critério de área mínima o valor de 800 *pixels*.

*4) Preenchimento de Buracos:* Realiza-se o preenchimento de regiões consideradas como buracos, que são regiões no interior de um componente conexo. A Fig.5 mostra simultaneamente o resultado da reclassificação de componentes conectados e do preenchimento de buracos.

Note que agora temos apenas componentes conexos sem buracos e suficientemente grandes em relação à imagem. O próximo passo é buscar dentre cada componente conexo restante, aquele que representa uma mão.

Fig. 5. Resultado da reclassificação de componentes conexos e preenchimento de buracos.

*5) Localização da Mão:* No presente trabalho foi proposto como protocolo de aquisição; assumimos que a mão a ser detectada tenha cinco dedos, esteja aberta e que seu dedo médio seja o dedo com maior altura. Se o dedo médio é o dedo mais alto, então a coordenada de altura inicial do componente conexo é ele próprio. Com isso é possível fazer uma busca horizontal na imagem para verificar se existe um dedo indicador e um dedo anelar. Caso esses dedos existam e suas larguras estejam na faixa de $\pm 30\%$ da largura do dedo médio, então considera-se que uma mão foi encontrada. Uma vez que a mão foi localizada, torna-se necessário extrair algumas de suas características básicas, a saber, sua altura e sua lateralidade (esquerda ou direita).

*6) Definição da Janela de Interesse:* Pode ocorrer de o punho, ou até mesmo o braço, se juntarem com a mão formando um único grande componente conexo. Nesse caso, a altura da mão iria da ponta do dedo médio até a parte mais baixa desse componente conexo. Para evitar que isso ocorra, a altura da mão é definida como sendo 2.25 vezes a altura do dedo médio. Tal valor é suficiente para reduzir a altura da janela de interesse a uma tamanho que seja suficiente para os propósitos da classificação.

*7) Definição da Lateralidade de Mão:* O último passo na detecção da mão é definir sua lateralidade (direita ou esquerda). Para isso, identifica-se os dois pontos extremos $y_1(i_1, j_1)$ e $y_2(i_2, j_2)$, mostrados na imagem da Fig. 6. Se $i_1 < i_2$ então essa é uma mão direita, caso contrário, trata-se de uma mão esquerda.

Simultaneamente, os componentes conexos restantes são classificados como fundo e a mão é enquadrada em uma janela menor para que as próximas operações utilizem menos memória e processamento.

Fig. 6. Detecção da mão direita.

### *B. Extração de Descritores*

Através da janela obtida no passo anterior, calcula-se as projeções horizontais e verticais da mão binária. Tais projeções serão utilizadas como descritores da postura. Para isso:

1) defini-se um tamanho $N$ para um vetor de projeções verticais $P(j)_v$, com $j = 1..N$, e divide-se as colunas da imagem binária em $N$ subintervalos de tamanho $\delta w$, de forma a gerar $N$ subimagens com $h$ linhas e $\delta w = w/N$ colunas;
2) calcula-se cada elemento do vetor $P(j)_v$ a partir do somatório dos *pixels* brancos contidos em cada subimagem $I_j(m, n)$, de acordo com a Eq. 1. O fator $1/(\delta h \times w)$ normaliza os elementos de $P(j)_v$;
3) defini-se um tamanho $N$ para um vetor de projeções horizontais $P(i)_h$, com $i = 1..N$, e divide-se as linhas da imagem binária em $N$ subintervalos de tamanho $\delta h$, de forma a gerar $N$ subimagens com $\delta h = h/N$ linhas e $w$ colunas; e
4) calcula-se cada elemento do vetor $P(i)_h$ a partir do somatório dos *pixels* brancos contidos na subimagem $I_i(m, n)$, de acordo com a Eq. 2. O fator $1/(\delta h \times w)$ normamiza os elementos de $P(i)_h$.

$$P(j)_v = \frac{\sum_{m=1}^{h} \sum_{n=1}^{\delta w} I_j(m, n)}{h \times \delta w} \quad (1)$$

$$P(i)_h = \frac{\sum_{m=1}^{\delta h} \sum_{n=1}^{w} I_i(m, n)}{\delta h \times w} \quad (2)$$

A imagem da Fig. 7 ilustra o processo acima descrito, considerando $N = 5$.

A partir dos vetores da projeção horizontal e vertical anteriormente descritos, define-se um novo vetor $P$, de tamanho $2N$, constituído pela concatenação de ambas as projeções. Esse novo vetor formará a entrada da rede e será usado tanto para treinamento quanto para operação.

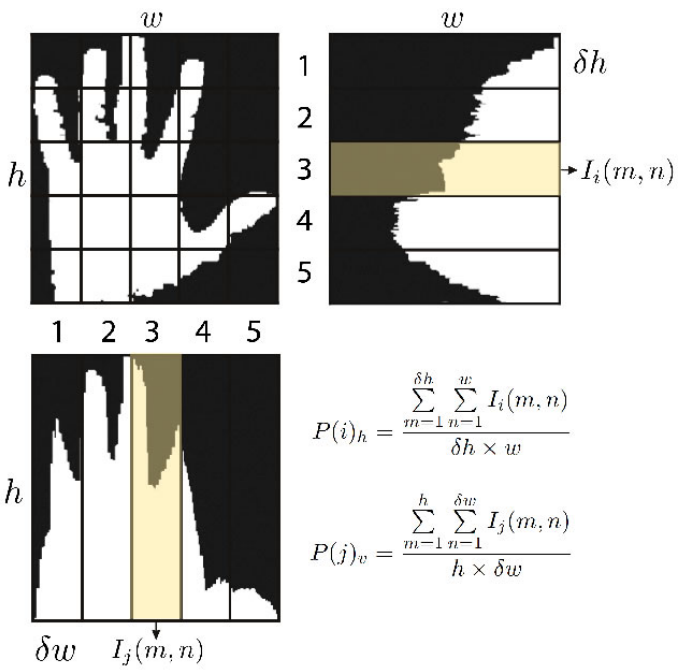



Fig. 7. Ilustração do processo de geração das projeções vertical e horizontal, considerando $N = 5$.

### C. Classificação por meio de Redes Neurais Artificiais

Conforme dito anteriormente, a classificação das posturas manuais será realizada por meio de redes neurais artificiais do tipo perceptron multicamada. Em nossos experimentos definimos vetores de projeções horizontal e vertical de tamanhos 10 e 20, o que implica em uma camada de entrada de tamanho 20 e 40, respectivamente. A quantidade de neurônios na camada escondida variou de 2 a 10, e a quantidade de neurônios da camada de saída foi fixado em 2. Na camada de entrada os neurônios não realizam qualquer computação. Já nas camadas oculta e de saída, foi utilizada a função de ativação *log-sigmóide*, definida pela Eq. 3 o que resulta em saídas entre 0 e 1. O algoritmo de treinamento utilizado é o *Levenberg-Marquardt* [13]. A performance da rede é medida de acordo com o erro médio quadrático.

$$f(x) = \frac{1}{1 + e^{-x}} \tag{3}$$

Escolhemos 10 posturas manuais que coincidem com as letras *A, B, C, D, H, I, L, V, W e Y*, conforme definidas pela *American Sign Language*. Sendo assim, as posturas em questão apresentam um caráter estático e não dinâmico [14]. Como nosso objevito não é o reconhecimento específico desse alfabeto, vamos chamar cada uma das postura de $X_n$, com $n = 1..10$. Para cada uma das letras, treinou-se uma rede neural diferente, resultando em 10 redes denominadas $R_n$, $n = 1..10$. Dessa maneira, as $R_1, R_2, ..., R_n$ foram treinadas de forma a reconhecerem $X_1, X_2, ...X_n$, respectivamente, e rejeitarem as demais posturas. Cada rede $R_n$ é treinada para responder com o vetor alvo $T = [1\ 0]$, caso o padrão apresentado a ela seja a postura $X_n$ e com $T = [0\ 1]$ caso contrário. Ou seja, cada rede reconhece apenas o padrão para o qual foi treinada e indica como desconhecidos os 9 padrões restantes, conforme ilustrado da Fig. 8. É importante ressaltar que na operação os neurônios de saída não respondem exatamente com 1's e 0's, mas apenas com valores aproximados. Por isso, é necessário incluir uma etapa de competição para decidir qual dos dois neurônios de saída apresenta o maior valor de resposta (neurônio vencedor) e, a partir dessa informação, definir a resposta final da rede como [1 0] ou [0 1].

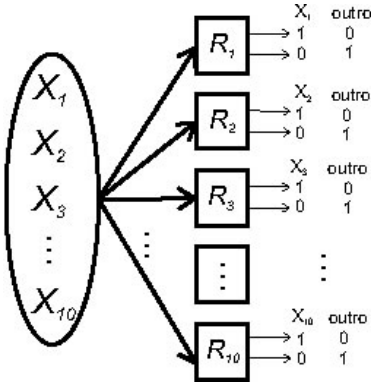



Fig. 8. Ilustração do classificador proposto: cada rede reconhece com $T = [1\ 0]$ apenas o padrão para o qual foi treinada e indica como desconhecidos, com $T = [0\ 1]$, os 9 padrões restantes.

## III. Resultados Experimentais

No treinamento e validação, foram adquiridos consecutivamente conjuntos com 500, 1000 e 2000 vetores por postura. Durante o procedimento de captura, procurou-se variar a postura com o objetivo de se introduzir ruído no treinamento. Após a obtenção dos conjuntos utilizados na etapa de treinamento dos classificadores, foram adquiridos novos conjuntos para a fase de testes, totalizando mais 2000 vetores para cada postura. Treinaram-se, então, as redes, escolhendo sempre a de melhor desempenho entre 10 criadas para identificar cada postura. Como foram definidas 10 redes neurais, uma por postura, utilizando-se entradas com 20 ou 40 elementos, com 500, 1000 ou 2000 amostras por postura e com 2 a 9 neurônios na camada oculta, obteve-se um total de 480 ($10 \times 2 \times 3 \times 8$) redes neurais.

Para se definir as condições ideais de operação, utilizou-se a *F-measure*, definida pela Eq. 4,

$$F\text{-}measure = 2 \cdot \frac{precision \times recall}{precision + recall}, \tag{4}$$

onde *precision* e *recall* são definidas por,

$$\begin{aligned} precision &= \frac{tp}{(tp+fp)} \\ recall &= \frac{tp}{(tp+fn)}, \end{aligned} \tag{5}$$

e $tp$, $fp$ e $fn$ são as taxas de verdadeiros positivos (*true positive*), falsos positivos (*false positive*) e falsos negativos

(*false negative*), respectivamente. De uma maneira geral, a medida *precision* de um sistema de reconhecimento de padrões representa a relação entre a quantidade de classificações verdadeiramente positivas ($tp$) retornadas pelo sistema e a quantidade total de classificações retornadas como positivas ($tp + fp$). Em outras palavras, representa a probabilidade de uma classificação retornada pelo sistema como positiva ser uma ocorrência verdadeiramente positiva. Já a medida *recall* representa a relação entre a quantidade de classificações verdadeiramente positivas retornadas pelo sistema ($tp$) e a quantidade de classificações que deveriam ter sido retornadas como positivas ($tp + fn$). Ou seja, representa a probabilidade de uma ocorrência verdadeiramente positiva ser classificada como tal pelo sistema. A *F-measure* é, dessa forma, a média harmonica entre *precision* e *recall*.

As curvas da Fig. 9 mostram as *F-measures* para entradas de tamanho 20, conjuntos de treinamento com 500, 1000 e 2000 amostras por postura e número de neurônios na camada escondida variando de 2 a 9. Já as curvas da Fig. 10 mostram as *F-measures* para as mesmas condições anteriormente descritas, com exceção dos vetores de entrada, que passam a ser de tamanho 40. Independentemente do tamanho do conjunto de treinamento, os testes foram realizados com outros 2000 vetores por postura, desconhecidos pelos classificadores.

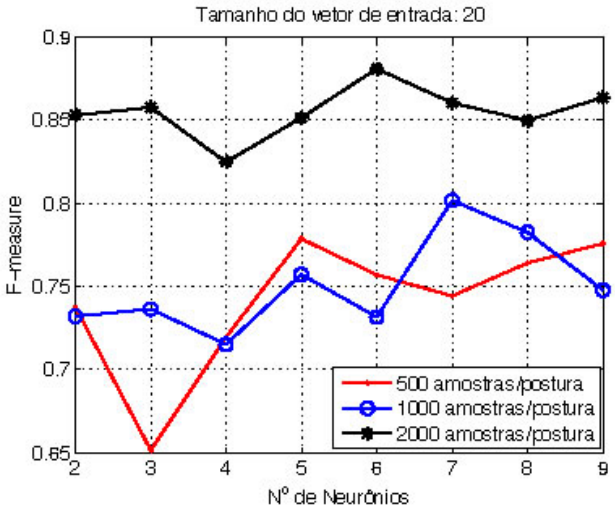



Fig. 9. As curvas mostram as *F-measures* para entradas de tamanho 20, conjuntos de treinamento com 500, 1000 e 2000 amostras por postura e número de neurônios na camada escondida variando de 2 a 9. A maior *F-measure* observada foi 0.881, para 6 neurônios

Os melhores resultados foram obtidos com conjuntos de treinamento formados por 2000 vetores por postura manual. Com relação aos vetores de entrada, aqueles com tamanho 40 resultaram em melhores índices, quando os conjuntos de treinamento com 500 e 1000 amostras por postura foram utilizados. Já para conjuntos de treinamento de tamanho 2000 por postura, a diferença de desempenho entre vetores de entrada com tamanho 20 e 40 não foi tão significativa. Sendo assim, optou-se por trabalhar com vetores de entrada de tamanho 20 e conjuntos de treinamento de 2000 amostras por postura. Nessas condições, o melhor

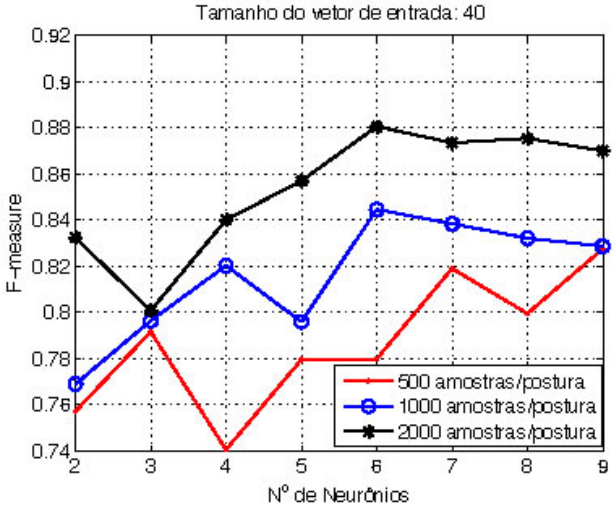



Fig. 10. As curvas mostram as *F-measures* para entradas de tamanho 40, conjuntos de treinamento com 500, 1000 e 2000 amostras por postura e número de neurônios na camada escondida variando de 2 a 9. A maior *F-measure* observada foi 0.880, para 6 neurônios

resultado nos testes foi obtido utilizando-se 6 neurônios na camada escondida (*F-measure* observada foi 0.881).

Considerando que os gestos são capturados por meio de uma câmera a 30 *frames* por segundo, propomos que a classificação dos gestos não seja realizada *frame* a *frame*, ou seja, a cada 1/30 segundo, conforme os experimentos anteriores, mas uma vez a cada conjunto de 20 *frames*. Vamos considerar que o reconhecimento do gesto foi realizado em tempo real se conseguirmos realizar uma classificação em menos de 1 segundo. Nesse experimento, cada conjunto de 2000 amostras por postura é reduzido a 100 conjuntos de 20 amostras. Como são 10 posturas, o total de conjuntos é 1000. Cada uma das 20 amostras de um conjunto é apresentada às 10 redes $R_n$. Cada rede que realiza uma classificação positiva recebe um voto. Depois das 20 amostras terem sido apresentadas, a postura $X_n$ reconhecida será aquela referente à rede $R_n$ que recebeu a maior quantidade de votos.

Os resultados desse novo experimento, variando-se a quantidade de neurônios de 2 a 9, podem ser visto na Fig. 11. Também nesse caso, o melhor desempenho foi obtido com 6 neurônios na camada escondida. A *F-measure* observada foi 0.972, ou seja, aproximadamente 10% superior ao caso em que os classificadores operam sem o procedimento de votação sugerido.

Com relação à complexidade computacional, estimou-se o tempo gasto pela implementação desde a captura da imagem até à classificação da postura manual. Calculou-se a quantidade média de classificações que o programa foi capaz de realizar, atingindo-se a marca de aproximadamente 29 classificações por segundo. Isso cumpre com o objetivo de se operar em tempo real. O computador utilizado nos testes possui um processador Core I7 (2630, 2.0 GHz), e memória RAM DDR3 de 8 GB. O programa foi implementado em linguagem de programação C/C++.

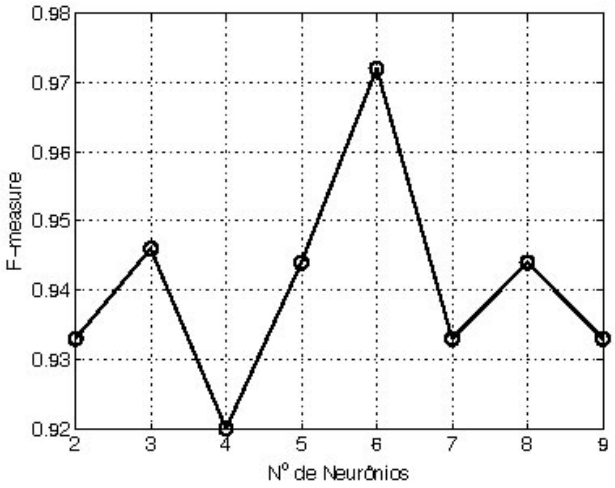



Fig. 11. Desempenho do sistema utilizando votação, variando-se o número de neurônios na camada escondida de 2 a 9. Também nesse caso, o melhor resultado foi obtido para 6 neurônios na camada oculta ($F\text{-}measure = 0.972$).

## IV. Conclusões e Trabalhos Futuros

O presente trabalho propôs um método pra realizar o reconhecimento de posturas manuais utilizando redes neurais artificiais. Procurou-se desenvolver um método que fosse eficiente com relação ao reconhecimento e, ao mesmo tempo, apresentasse baixa complexidade computacional. Os resultados demonstraram que para as condições consideradas esses objetivos foram alcançados.

O reconhecimento de posturas em tempo real sem a utilização de marcas ou luvas especiais, constitui um desafio. É evidente que o bom desempenho global do sistema dependente do bom desempenho de cada uma de suas etapas, porém, pode-se considerar a etapa que realiza a extração dos descritores da mão como sendo a mais crítica, pois quanto mais adequados forem tais descritores, melhor será o desempenho dos classificadores. A abordagem aqui sugerida utiliza descritores que se mostraram suficientes para se realizar a identificação das 10 postura manuais consideradas nos testes. No entanto, para um conjunto maior e mais complexo de posturas, talvez sejam necessários descritores adicionais ou outro conjunto completamente diferente.

Os testes demonstraram o bom desempenho do método proposto, principalmente após a introdução do procedimento de votação ($F\text{-}measure = 0.972$). Para trabalhos futuros, pretende-se ampliar o número de posturas manuais, aprimorar os descritores utilizados, realizar uma avaliação subjetiva da complexidade computacional, realizar a avaliação do desempenho do classificador para diferentes algoritmos de segmentação por de pele, e, por fim, verificar a viabilidade de se implementar o sistema em um dispositivo móvel.

## References

[1] L. G. Shapiro, G. C. Stockman, L. G. Shapiro, and G. Stockman, *Computer Vision*. Prentice Hall, Jan. 2001.

[2] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, 2nd ed. The MIT Press, Sep. 2001.

[3] D.-H. Lee and K.-S. Hong, "Game interface using hand gesture recognition," in *Computer Sciences and Convergence Information Technology (ICCIT), 2010 5th International Conference on*, Nov 2010, pp. 1092–1097.

[4] S. Jin, Y. Li, G. ming Lu, J. xun Luo, W. dong Chen, and X.-X. Zheng, "Som-based hand gesture recognition for virtual interactions," in *VR Innovation (ISVRI), 2011 IEEE International Symposium on*, March 2011, pp. 317–322.

[5] S.-H. Lee, M.-K. Sohn, D.-J. Kim, B. Kim, and H. Kim, "Smart tv interaction system using face and hand gesture recognition," in *Consumer Electronics (ICCE), 2013 IEEE International Conference on*, Jan 2013, pp. 173–174.

[6] T. Ayshee, S. Raka, Q. Hasib, M. Hossain, and R. Rahman, "Fuzzy rule-based hand gesture recognition for bengali characters," in *Advance Computing Conference (IACC), 2014 IEEE International*, Feb 2014, pp. 484–489.

[7] G. Khurana, G. Joshi, and J. Kaur, "Static hand gestures recognition system using shape based features," in *Engineering and Computational Sciences (RAECS), 2014 Recent Advances in*, March 2014, pp. 1–4.

[8] R. Elakkiya, K. Selvamani, and S. Kanimozhi, "A framework for recognizing and segmenting sign language gestures from continuous video sequence using boosted learning algorithm," in *Issues and Challenges in Intelligent Computing Techniques (ICICT), 2014 International Conference on*, Feb 2014, pp. 498–503.

[9] T. A. C. Bragatto, G. I. S. Ruas, A. P. Martins, and M. V. Lamar, "Um modelo de máquinas de vetores de suporte estruturada em Árvore binária probabilística aplicado ao reconhecimento de posturas manuais em tempo real," *Anais do XXVII Simpósio Brasileiro de Telecomunicações*, 2009.

[10] Z. Xu and M. Zhu, "Color-based skin detection: survey and evaluation," *Proceeding of the 12th International Multi-Media Modelling Conference*, 2006.

[11] S. Haykin, *Neural Networks: A Comprehensive Foundation (2nd Edition)*, 2nd ed. Prentice Hall, Jul. 1998.

[12] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2001.

[13] M. T. Hagan and M. Menhaj, "Training feedforward networks with the marquardt algorithm," *IEEE Transactions on Neural Networks*, vol. 5, pp. 989–993, 1994.

[14] A. H. Ozaki, F. Mendonça, and D. L. Borges, "Segmentacão, deteccão e acompanhamento automático de gestos em vídeo usando visão por computador," *Anais do XVIII Congresso Brasileiro de Automática*, 2010.

# Feature description based on Mean Local Mapped Pattern

Carolina Toledo Ferraz*, Osmando Pereira Junior†, Adilson Gonzaga‡
Department of Electrical Engineering - EESC/USP
University of São Paulo
Av. Trabalhador São Carlense, 400
13560-590, São Carlos, SP, Brasil
*caroltoledoferraz@gmail.com, †osmandoj@gmail.com, ‡agonzaga@sc.usp.br

***Abstract*—Local feature description has gained a lot of interest in many applications, such as texture recognition, image retrieval and face recognition. This paper presents a novel method for local feature description based on gray-level difference mapping, called Mean Local Mapped Pattern (M-LMP). The proposed descriptor is robust to image scaling, rotation, illumination and partial viewpoint changes. Furthermore, this descriptor more effectively captures the nuances of the image pixels. In our experiments, the descriptor is compared to the Center-Symmetric Local Mapped Pattern (CS-LMP) and the Center-Symmetric Local Binary Pattern (CS-LBP). The results show that our descriptor performs better compared to these two methods.**

## I. Introduction

In image processing, the local feature description plays an important role in applications such as texture recognition, content-based image retrieval and face recognition. The objective is to build a feature vector providing a representation that allows efficient matching of local structures between images.

One of the most widely known key feature descriptors published in the literature is the scale-invariant feature transform (SIFT) [1]. It was introduced by Lowe in 1999 [2]. It is characterized by a 3D histogram of gradient locations and orientations and stores the bins in a vector of 128 positions. Many other descriptors like SIFT have been proposed, such as Principal Components Analysis SIFT (PCA-SIFT) [3], Gradient Location and Orientation Histogram (GLOH) [4], Speeded Up Robust Features (SURF) [5], Colored SIFT (CSIFT) [6], Center-Symmetric Local Binary Pattern (CS-LBP) [7] and Kernel Projection Based SIFT (KPB-SIFT) [8]. The GLOH descriptor is very similar to SIFT, however, it uses a Log-Polar location grid instead of a Cartesian one. SURF approximates SIFT using the Haar wavelet response and using integral images to compute the histograms bins. It uses a descriptor vector of 64 positions, providing a better processing speed. Jin, Liu, Lu and Tong [9] proposed the Improved Local Binary Pattern (ILBP) that is an improvement of the LBP feature and is used for face detection [9]. It compares all of the pixels (including the central pixel) with the mean intensity. On the other hand, the Mean Local Binary Pattern (M-LBP) descriptor [10] does not consider the central pixel in this calculation. Center-Symmetric Local Binary Pattern (CS-LBP) [7] combines the strengths of the well-known SIFT descriptor and the Local Binary Pattern (LBP) texture operator. In [7], it is shown that the construction of the CS-LBP descriptor is simpler than SIFT, but it generates a feature vector of 256 positions, which is twice the SIFT vector size.

In the Local Mapped Pattern (LMP) approach [11], the authors consider the sum of the differences of each gray-level of a given neighborhood to the central pixel as a local pattern that can be mapped to a histogram bin using a mapping function.

The Center-Symmetric Local Mapped Pattern (CS-LMP) [11] combines the desirable properties of the CS-LBP and the Local Mapped Pattern (LMP). This approach is based on the sum of the differences of each gray-level of a given neighborhood to the center-symmetric pairs of pixels as a local pattern that can be mapped to a histogram bin using a mapping function.

In this paper, we propose a new interest region descriptors, denoted as the Mean Local Mapped Pattern (M-LMP). The M-LMP descriptor is based on the CS-LBP, ILBP and M-LBP using the LMP methodology. This new descriptor captures the small transitions of pixels in the image more accurately, resulting in a greater number of correct matches than the CS-LMP and CS-LBP.

The rest of this paper is organized as follows. In Section II, we briefly describe the CS-LMP, CS-LBP, ILBP, M-LBP and LMP. Section III gives details on the proposed approach. The experimental evaluation and the results are presented in Section IV. Finally, we conclude the paper in Section V.

## II. Local Descriptors

In this section, five previously published local descriptors are shown as the theoretic material for our approach: the CS-LBP, CS-LMP, ILBP, M-LBP and the LMP.

### A. Center-Symmetric Local Binary Pattern (CS-LBP)

The CS-LBP is a modified version of the LBP texture feature and SIFT descriptor. In [7], the authors state that the LBP produces a rather long histogram and therefore is difficult to use in the context of an image descriptor. To solve this problem, they modified the scheme of how to compare the pixels in the neighborhood. Instead of comparing each pixel with the center one, the method compares center-symmetric pairs of pixels.

For the construction of the descriptor, the image regions are divided into cells with a location grid, and for each cell, a

CS-LBP histogram is built. The CS-LBP feature extraction for each pixel of the region is accomplished according to Equation 1:

$$CS-LBP_{R,N,T}(x,y) = \sum_{i=0}^{(N/2)-1} s(n_i - n_{i+(N/2)})2^i, \quad (1)$$

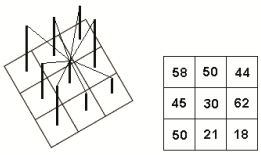
where $n_i$ and $n_{i+(N/2)}$ correspond to the gray values of center-symmetric pairs of pixels of N equally spaced pixels on a circle of radius R and

$$s(x) = \begin{cases} 1, & \text{if} \quad x > T, \\ 0, & \text{otherwise.} \end{cases}$$

$T$ is set to 0.01 [12]. For a $4 \times 4$ Cartesian grid, the final feature vector contains 256 positions.

### B. Local Mapped Pattern (LMP)

The LMP methodology assumes that each gray-level distribution within an image neighborhood is a local pattern. This pattern can be represented by the gray-level differences around the central pixel [11]. Figure 1 shows an example of a $3 \times 3$ neighborhood and a $3D$ graphic of the respective local pattern.



| | | |
|---|---|---|
| 58 | 50 | 44 |
| 45 | 30 | 62 |
| 50 | 21 | 18 |

Fig. 1. Gray-level differences as a local pattern

Each pattern defined by a $W \times W$ neighborhood will be mapped to a histogram bin $h_b$ using Equation 2, where $f_{g(i,j)}$ is the mapping function, $P(k,l)$ is a weighting matrix of predefined values for each pixel position within the neighborhood, and $B$ is the number of the histogram bins. This equation represents the weighted sum of each gray-level difference between the neighboring pixels and the central one, mapped onto the [0,1] interval by a mapping function and rounded to $B$ possible bins.

$$h_b = round\left(\frac{\sum_{k=1}^{W}\sum_{l=1}^{W}(f_{g(i,j)}P(k,l))}{\sum_{k=1}^{W}\sum_{l=1}^{W}P(k,l)}\right)(B-1), \quad (2)$$

For texture analysis, the LMP approach uses a common sigmoid curve as in Equation 3.

$$f_{g(i,j)} = \frac{1}{1+e^{\frac{-[A(k,l)-g(i,j)]}{\beta}}}, \quad (3)$$

where $\beta$ is the curve slope and $[A(k,l)-g(i,j)]$ are the gray-level differences within the neighborhood centered in $g(i,j)$. The proposed weighting matrix is:

$$P(k,l) = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

### C. Center-Symmetric Local Mapped Pattern (CS-LMP)

The CS-LMP is based on the CS-LBP, but using a mapping function in the LMP method [11].

Equation 4 shows the feature extraction for each pixel of the interest region, defined in a neighborhood $W \times W$:

$$CS-LMP_{R,N,f}(x,y) = round\left(\frac{\sum_{i=0}^{N/2-1} f(i)P(i))}{\sum_{i=0}^{N/2-1} P(i)}\right) b \quad (4)$$

where $b$ is the number of histogram bins minus one, $P(i)$ is the $ith$ element of the weight matrix $P$ defined in the Equation 5, and $f(i)$ is the sigmoid mapping function defined in Equation 6.

$$\mathbf{P} = [p_0 p_1 p_2 ... p_{N/2-1}]^t \quad (5)$$

where

$$p_i = 2^i$$

$$f(i) = \frac{1}{1+e^{\frac{-[n_i - n_{i+(N/2)}]}{\beta}}} \quad (6)$$

where $n_i$ and $n_{i+(N/2)}$ correspond to the gray level values of the center-symmetric pairs of $N$ equally spaced pixels on a circle of radius $R$ and $\beta$ is the sigmoid curve slope. If the coordinates of $n_i$ on the circle do not correspond to the image coordinates, the gray-value of this point is interpolated.

The Hessian-Affine detector was adopted. Afterwards, the regions are mapped to a circular region of constant radius and rotated in the direction of the dominant gradient, as described in [7] and [1]. Each region is divided into cells with a grid size of $4 \times 4$. The features are extracted for each pixel of the interest region using a CS-LMP descriptor and a histogram is built for each cell.

The parameters $R$, $N$ and $\beta$ were empirically established, and will be showed in Section IV.

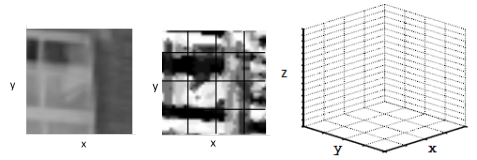
As it was presented in [12], a bilinear interpolation is made over the $x$ and $y$ dimensions to share the weight of the feature between the four nearest cells. Thus, a $3D$ histogram of the CS-LMP feature locations and values is built as shown in Figure 2. The descriptor is made by concatenating the feature histograms computed for the cells, obtaining a vector of 256 positions($4 \times 4 \times 16$). The descriptor is then normalized as in the CS-LBP presented in Section II-A.

### D. The ILBP and M-LBP methods

ILBP proposed an improved LBP that compares all of the pixels (including the central pixel) with the mean intensity of the pixels in the patch [13].

The ILBP formulation is shown in Equation 7:

$$ILBP_{N,R}(x,y) = \sum_{p=0}^{N-1} s(g_p - g_{mean})2^P + s(g_c - g_{mean})2^N, \quad (7)$$



(a) Input region (b) 4×4 cells' division (c) 4×4×16=256 bins of concatenated histogram

Fig. 2. CS-LMP descriptor

where

$$g_{mean} = \frac{1}{N+1}\left(g_c + \sum_{p=0}^{N-1} g_p\right) \tag{8}$$

and

$$s(x) = \begin{cases} 1, & \text{if} \quad x > T, \\ 0, & \text{otherwise.} \end{cases}$$

The ILBP considers $N$ sample points evenly distributed on a circle with radius $R$ from the central pixel.

The Mean Local Binary Pattern (M-LBP) is similar to the ILBP, but does not consider the central pixel. In this method, the mean of a region is calculated as the threshold instead of using the central pixel gray-value [10]. An example is shown in Figure 3.

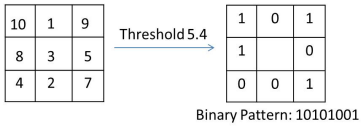



Fig. 3. M-LBP

## III. Mean Local Mapped Pattern (M-LMP)

As the original LBP is determined just by the gray level of the center pixel, some noise can be embedded in the final result. To avoid the noise and make the feature more robust, the average of the sub-region is used instead of gray level of a single pixel [10]. Thus, M-LMP is determined by the comparison of the pixels of a circular neighborhood to the average of this neighborhood, as shown in the Equation 9.

$$M - LMP_{R,N,f}(x,y) = round\left(\frac{\sum_{i=0}^{N-1} f(i)P(i)}{\sum_{i=0}^{N-1} P(i)}\right) b \tag{9}$$

where $b$ is the number of histogram bins minus one, $P(i)$ is the $ith$ element of the weight matrix $P$ defined in the Equation 10, and $f(i)$ is the sigmoid mapping function defined in the Equation 11.

$$\mathbf{P} = [p_0 p_1 p_2 ... p_{N-1}]^t \tag{10}$$

where

$$p_i = 2^i$$

$$f(i) = \frac{1}{1 + e^{\frac{-[n_i - n_{mean}]}{\beta}}} \tag{11}$$

where $n_i$ corresponds to the gray-level values of a neigborhood of $N$ equally spaced pixels on a circle of radius $R$, $\beta$ is the sigmoid curve slope and $n_{mean}$ is defined by Equation 12.

$$n_{mean} = \frac{1}{N}\sum_{i=0}^{N-1} n_i \tag{12}$$

Figure 4 shows an example of the M-LMP code generation for a circular neighborhood of radius $R = 2$ and $N = 8$ pixels. The number of bins defined for this example is 16, and thus $b = 15$ in Equation 9, generating codes in the range $[0, 15]$.

The construction of the descriptor was made in the same way as the CS-LMP described in Section II-C. The parameters $R$, $N$ and $\beta$ were empirically established as well for the CS-LMP as is shown in Section IV.

## IV. Experimental Evaluation

In this section, we present the performance of the M-LMP, CS-LMP and CS-LBP descriptors applied to the matching of image pairs using nearest-neighbor similarity.

The image database is available on the internet [1] [2]. It has 16 sets of images with 6 different types of transformations: viewpoint change, rotation change, scale, ilumination, JPEG compression and blur. The training images used in this paper are shown in Figure 5 and the test images are shown in Figure 6.

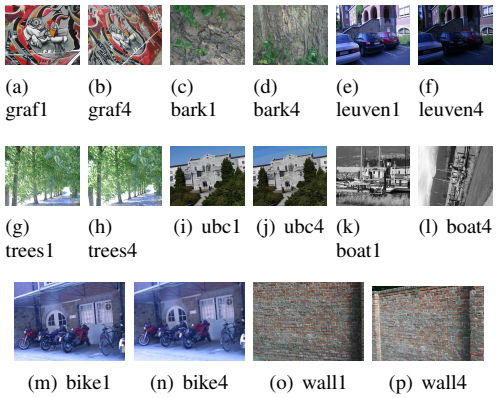



Fig. 5. Training Images: graf (viewpoint change), bark (scale and rotation change), trees (blur image), leuven (ilumination change), ubc (JPEG compression), boat (scale and rotation change), bike (blur image), wall (viewpoint change)

Table I presents the evaluation of the parameter $\beta$ for the M-LMP method applied to the image "bark". We tested some values of the $\beta$ parameter and we generated the recall $\times$ precision curves, measuring its respective area under curve (AUC). For most of the test images, the best value of $\beta$ was near 1.2 for M-LMP. For CS-LMP, the best value was 0.6 [11].

[1] http://www.robots.ox.ac.uk/ vgg/research/affine/

[2] http://lear.inrialpes.fr/people/mikolajczyk/

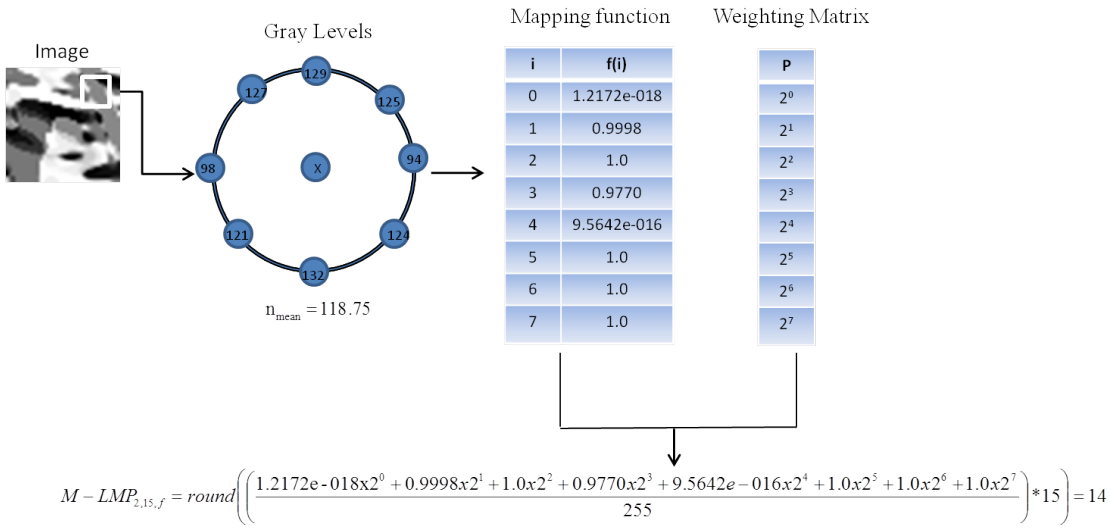



Fig. 4. M-LMP features for a neighborhood of 8 pixels

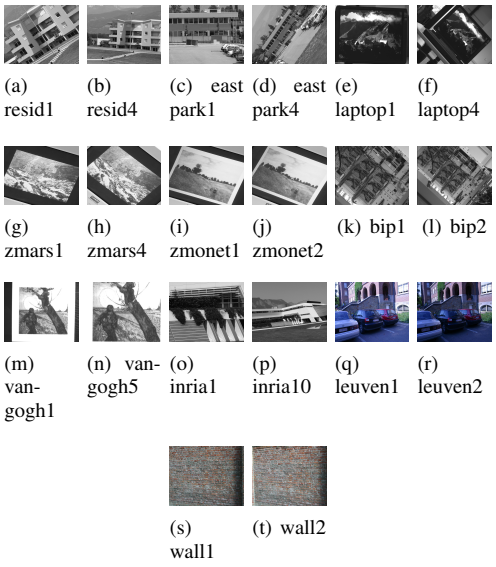



Fig. 6. Test Images: resid, east park, east south, ensigmag and laptop (scale and rotation), zmarks, zmonet, new york (rotation change), axterix, belledone, bip, crolle, lap and vangogh (scale change), inria (scale and rotation), leuven (ilumination change), wall (viewpoint change)

Table I. $\beta$ EVALUATION EXAMPLE FOR THE IMAGE "BARK"

| $\beta$ | 1.0 | 1.1 | **1.2** | 1.3 | 1.4 | 1.5 | 1.6 | 1.7 |
|---|---|---|---|---|---|---|---|---|
| **AUC** | 0.8862 | 0.8862 | **0.9032** | 0.8862 | 0.8791 | 0.8960 | 0.8980 | 0.8862 |

The evaluation criteria is explained in the next subsection.

### A. *Evaluation criteria*

The performance of the M-LMP descriptor was evaluated and compared with CS-LMP and CS-LBP, using an evaluation criterion based on Heikkilä *et al.* [7]. This criterion consists of the number of correct and false matches for a couple of images, knowing a priori the set of true matches (i.e., the ground truth). Heikkilä *et. al.* [7] determine the ground truth by the overlap error of the two regions. The overlap error determines how a region A of image 1 overlaps region B of image 2. Image 2 is projected on image 1 by a homography matrix H. The overlap error is defined by the ratio of the intersection and the union of the regions given by Equation 13:

$$\epsilon_S = 1 - (A \cap H^T BH)/(A \cup H^T BH) \tag{13}$$

A match is assumed to be correct if $\epsilon_S < 0.5$.

To evaluate the performance of our descriptors, we present the results with recall $\times$ precision curves [14] and their respective (AUC).

Precision represents the number of correct matches over the total number of possible matches returned (falses+corrects) and is defined by Equation 14.

$$precision = \frac{n_c}{n} \tag{14}$$

where $n$ is the number of possible matches returned and the Euclidean distance between their descriptors is below a threshold $t_d$. The number of correct matches $n_c$ includes the matched pairs whose Euclidean distance between their descriptors is below a threshold $t_d$ and $\epsilon_S < 0.5$

Recall represents the ratio between the correct matches and the total number of true matches (the ground truth) and is defined by Equation 15.

$$recall = \frac{n_c}{M}, \tag{15}$$

where $M$ is the number of elements in the set of true matches.

To construct the recall $\times$ precision curve, each point is defined by choosing $n \in [1, N]$, where $N$ represents the total number of possible matches. Analyzing this curve, a perfect descriptor would give a precision equal to 1 for any recall.

### B. Matching results

The interest region detector used to compute the descriptors was the Hessian-Affine detector. As discussed in the previous subsection, the parameters set for the M-LMP were $\beta = 1.2$, $b = 15$, $R = 2$, and $N = 8$. For CS-LMP were $\beta = 0.6$, $b = 15$, $R = 2$, and $N = 8$, and for the CS-LBP we used the parameters $R = 2$, $N = 8$, and $T = 0.01$ [12].

We generated the recall $\times$ precision curves for all images of the test set and calculated the AUC for each of them. However, for reasons of clarity, we present only two of them in Figure 7. Note that there is a good precision for low recall for the three descriptors, i.e., the first region pairs returned are correct matches; however, as the number of region pairs returned increases, the precision decreases. We can see that the reduction of precision is more accentuated by CS-LBP. As the number of returned matches increases $(n)$, M-LMP achieves a greater number of correct matches $(n_c)$ compared to CS-LMP and CS-LBP.

The AUC results are shown in Table II for the M-LMP, CS-LMP and CS-LBP descriptors. Note that for all images, CS-LMP has an AUC value greater than CS-LBP and that the M-LMP descriptor has AUC values grater than the two other descriptors.

Table II. AREA UNDER CURVE (AUC)

| **Image** | M-LMP | CS-LMP | CS-LBP |
|---|---|---|---|
| east park | **0.8801** | 0.8758 | 0.7842 |
| laptop | **0.9374** | 0.9219 | 0.8891 |
| inria | **0.8963** | 0.8748 | 0.8110 |
| zmars | **0.9702** | 0.9659 | 0.9499 |
| zmonet | **0.9669** | 0.9659 | 0.9364 |
| resid | **0.9090** | 0.9043 | 0.8332 |
| leuven | **0.9556** | 0.9479 | 0.9242 |
| wall | **0.9684** | 0.9639 | 0.9498 |

Table III presents the $n$ best returned correspondences by the three descriptors. For example, for the image named "resid", taking into account the first 1027 possible matches returned, we have 583 correct matches for the M-LMP, 580 for CS-LMP and 545 for the CS-LBP descriptor. The image "east park" presents the difference of 56 correct matches favorable to the M-LMP and for the image "zmars", our proposed approach outstands by 28 correct matches over the CS-LBP.

Figure 8 presents the correct matches achieved between the image pair called "laptop". It was performed to check the images test results when applying the descriptors M-LMP, CS-LMP and CS-LBP.

Table III. CORRECT MATCHES

| | **Image** | **Correct Matches** |
|---|---|---|
| M-LMP | resid | **583/1027** |
| CS-LMP | | 580/1027 |
| CS-LBP | | 545/1027 |
| M-LMP | east park | **645/1307** |
| CS-LMP | | 641/1307 |
| CS-LBP | | 589/1307 |
| M-LMP | bip | **441/767** |
| CS-LMP | | 436/767 |
| CS-LBP | | 429/767 |
| M-LMP | zmars | **1688/22001** |
| CS-LMP | | 1681/22001 |
| CS-LBP | | 1660/22001 |
| M-LMP | zmonet | **430/539** |
| CS-LMP | | 428/539 |
| CS-LBP | | 424/539 |
| M-LMP | vangogh | **237/342** |
| CS-LMP | | 236/342 |
| CS-LBP | | 226/342 |
| M-LMP | inria | **87/292** |
| CS-LMP | | 85/292 |
| CS-LBP | | 81/292 |
| M-LMP | laptop | **298/323** |
| CS-LMP | | 294/323 |
| CS-LBP | | 282/323 |
| M-LMP | leuven | **688/916** |
| CS-LMP | | 681/916 |
| CS-LBP | | 668/916 |
| M-LMP | wall | **820/1144** |
| CS-LMP | | 815/1144 |
| CS-LBP | | 803/1144 |

## V. CONCLUSION

In this paper, we proposed a novel method for feature description based on local pattern analysis. The method combines the CS-LBP, ILBP and M-LBP using the LMP methodology. It compares the pixels to the neighborhood average.

M-LMP features has many properties that make it well-suited for this task: a relatively short histogram and tolerance to rotation and scale change. Furthermore, it does not require setting many parameters. The performance of the M-LMP descriptor was compared to the CS-LMP and CS-LBP descriptor in terms of recall $\times$ precision curves. As M-LMP captures the gray-level nuances of the images better, its the recall $\times$ precision curves show a better performance than CS-LMP and CS-LBP methodology. We can conclude that M-LMP is a promising descriptor, with better performance over the CS-LMP and CS-LBP descriptor. Among the 10 tested images, the recall $\times$ precision curves were greater for all images for M-LMP than CS-LMP and CS-LBP. Moreover, the number of correct correspondences outperformed the CS-LMP and CS-LBP.

## ACKNOWLEDGMENT

The authors would like to thank the São Paulo Research Foundation (FAPESP), grant #2011/18645-2, and National Council for Scientific and Technological Development (CNPQ) for their financial support of this research.

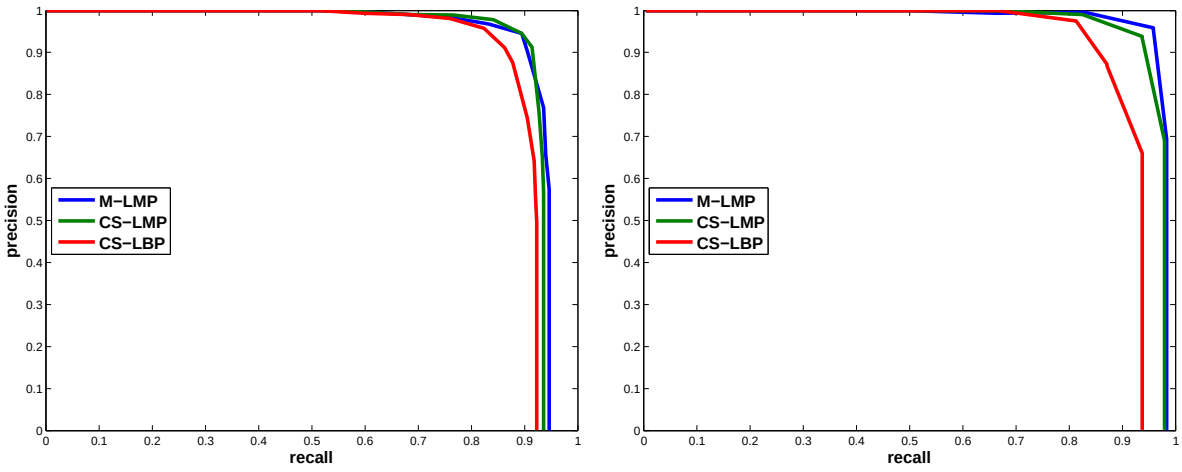
## REFERENCES

[1] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, pp. 91–110, 2004.

[2] D. Lowe, "Object recognition from local scale-invariant features," *International Conference on Computer Vision*, pp. 1150–1157, 1999.

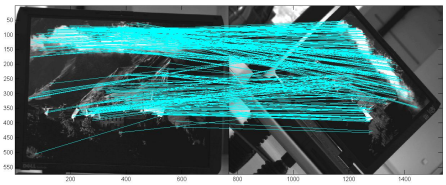
[3] Y. Ke and R. Sukthankar, "Pca-sift: A more distinctive representation for local image descriptors," *Computer Vision and Pattern Recognition, 2004.*, vol. 2, pp. 506 – 513, 2004.

[4] K.Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *IEEE Trans. Pattern Anal. Mach Intell.*, vol. 27(10), pp. 1615–1630, 2005.

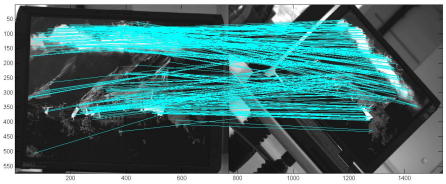



(a) bip (b) vangogh

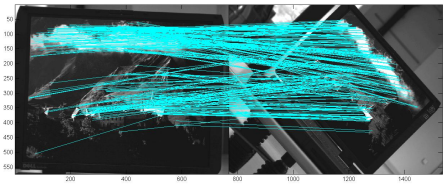
Fig. 7. Results of the *recall* × *precision* curves for CS-LMP, M-LMP and CS-LBP descriptors.



(a) M-LMP - 298 correct matches



(b) CS-LMP - 294 correct matches



(c) CS-LBP - 282 correct matches

Fig. 8. Matches achieved between the image pair called "laptop". Total of 323 correspondences.

[5] H. Bay, T. Tuytelaars, and L. V. Gool, "Surf: speeded up robust features," *European Conference on Computer Vision*, vol. 1, pp. 404–417, 2006.

[6] A. E. Abdel-Hakim and A. A. Farag, "Csift: A sift descriptor with color invariant characteristics," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR06)*, vol. 2, pp. 1978 – 1983, 2006.

[7] M. Heikkilä, M. Pietikäinen, and C. Schmid, "Description of interest regions with local binary patterns," *Pattern Recognition*, vol. 42, pp. 425–436, 2009.

[8] G. Zhao, L. Chen, G. Chen, and J. Yuan, "Kpb-sift: a compact local feature descriptor," *MM '10 Proceedings of the international conference on Multimedia*, pp. 1175–1178, 2010.

[9] H. Jin, Q. Liu, H. Lu, and X. Tong, "Face detection using improved lbp under bayesian framework," *Proceedings of the Third International Conference on Image and Graphics (ICIG04)*, pp. 306–309, 2004.

[10] G. Bai, Y. Zhu, and Z. Ding, "A hierarchical face recognition method based on local binary pattern," *CISP08 Proceedings of the 2008 Congress on Image and Signal Processing*, vol. 2, pp. 610–614, 2008.

[11] C. T. Ferraz, O. P. Jr., and A. Gonzaga, "Feature description based on center-symmetric local mapped patterns," in *Proceedings of the 29th Symposium on Applied Computing*, 2014, pp. 39–44.

[12] M. Heikkilä, M. Pietikäinen, and C. Schmid, "Description of interest regions with center-symmetric local binary patterns," *Springer-Verlag Berlin Heidelberg*, 2006.

[13] D. Huang, C. Shan, M. Ardabilian, Y. Wang, and L. Chen, "Local binary patterns and its application to facial image analysis: A survey," *IEEE Transactions on Systems, man, and cybernetics - part c: Applications and reviews*, vol. 41(6), pp. 765–781, 2011.

[14] R. Baeza-Yates and B. Ribeiro-Neto, *Modern Information Retrieval*, A. Press, Ed. New York, Addison-Wesley, 1999.

# FACE RECOGNITION USING COMPLEX WAVELET TRANSFORM, CLUSTERING AND FUZZY INTEGRAL

André Luiz Nogueira
Instituto Federal de Sergipe
Lagarto, Sergipe
Email: andreln27@yahoo.com

Junior Leal do Prado
Instituto Federal de Sergipe
Aracaju, Sergipe
Email: jrprado@gmail.com

***Resumo*—Face recognition is a problem of verification or identification of a face from an image. In identification, an input face is compared with faces from a database, which purpose is to identify the input face. In this article, the proposed algorithm is compared with other technique based on a complex wavelet transform. The proposed method is basically divided into four steps. The first uses a Complex Wavelet Transform for feature extraction of faces. The second step is the application of PCA to images obtained previously. Next Fuzzy Clustering is applied that determines the prototypes of each class. Finally, the last step is to use a Fuzzy Integral responsible for the fusion of classifiers. The experimental results were made using the Yale and Essex databases and show that the proposed algorithm is superior in performance compared to other technique.**

## I. INTRODUÇÃO

O reconhecimento de faces é uma subárea do reconhecimento de padrões, que vem recebendo uma atenção especial devido a suas aplicações comerciais, de segurança e militares.

Os métodos para o reconhecimento de faces [15] dividem-se basicamente em três categorias:

1) métodos baseados em características (locais);
2) métodos holísticos (globais);
3) métodos híbridos (utilizam as duas abordagem anteriores).

Os métodos baseados em características, primeiro processam a face para identificar e depois extrair características faciais como olhos, nariz, boca, etc. Em seguida, calculam relações geométricas entre estes pontos, reduzindo a face de entrada a um vetor com características geométricas.

Algoritmos baseados em características podem ser invariantes ao tamanho, orientação e/ou iluminacção, entretanto existem desvantagens como a detecção automática de características e escolha de quais características são importantes.

Os métodos holísticos descrevem as faces utilizando características globais, ou seja, sem tentar localizar pontos individuais. Estas técnicas utilizam análise estatística, redes neurais e transformações lineares ou não. Tem como vantagem utilizarem toda face, sem destruir nenhuma informação processando exclusivamente determinados pontos. Entretanto esses métodos são sensíveis a variação de posição, escala, etc, o que restringe seu uso às imagens obtidas em ambientes controlados.

A combinação dos métodos baseados em características e holísticos dá origem aos métodos híbridos.

O algoritmo proposto é um método híbrido, pois combina um método baseado em característica, a Transformada Wavelet Complexa e um método holístico ou estatístico, a PCA.

Uma imagem de $q \times p$ pixels pode ser representada por um vetor em um espaço de dimensão $qp$. Essa dimensão pode ser muito alta comprometendo a robustez e a velocidade do reconhecimento.

Um modo comum para resolver esse problema é utilizar técnicas para redução e seleção de características. Uma das mais populares é a Análise de Componentes Principais [2], [11].

A PCA projeta os dados originais em um subespaço de dimensão reduzida, preservando a maior parte da informação original. Além da PCA, outra abordagem vem sendo empregada no reconhecimento de faces envolve a Transformada Wavelet [8], [9], [11]. Nos trabalhos iniciais, a Transformada Wavelet era aplicada na compressão, redução de ruído e detecção de características de imagens. Recentemente, foi proposta a Transformada Wavelet Complexa de árvore dupla (TWC-AD) que supera a Transformada Wavelet (real), pois é aproximadamente invariante a deslocamentos e apresenta um desempenho superior na detecção de bordas de imagens [4], [5].

No reconhecimento de padrões, outra técnica utilizada a Análise de Agrupamentos Fuzzy [6], [7], cujo objetivo é classificar os dados de acordo com a similaridade entre eles e organizá-los em grupos.

Considerando as diferentes performances dos classificadores, é desejável que sua fusão produza classificadores com desempenhos melhores do que aqueles obtidos utilizando os classificadores individualmente. Baseado em [10], a Integral Fuzzy pode ser usada para fusão de classificadores.

O algoritmo proposto utiliza para extração de características das imagens a TWC e a PCA, já para a classificação as técnicas empregadas são Agrupamento Fuzzy e Integral Fuzzy.

Diante de uma revisão bibliográfica recentemente realizada, foi constatado que a fusão dessas abordagens supracitadas em um único algoritmo, visando ao reconhecimento de faces,

ainda não foi muito explorada. Comprova-se esse fato devido a uma reduzida quantidade de publicações no meio científico que utilizam essas técnicas integradas. Nesse ínterim, o presente artigo vem contribuir com alguns passos, nesta direção, e evidenciar a superioridade do algoritmo proposto quando comparada ao método apresentado em [3].

Nas próximas seções são apresentadas as ferramentas utilizadas no algoritmo proposto neste artigo.

### A. PCA

A Análise de Componentes Principais [1], [2], [11] é usada para determinar um subespaço, cujos vetores da base correspondem à direção de máxima variância no espaço original.

A PCA é capaz de transformar um espaço de dimensão $M$ em um espaço de dimensão $N$ $(M >> N)$, preservando a maior parte dos dados originais. O algoritmo PCA é descrito a seguir.

A base de vetores PCA é calculada a partir de um conjunto $I$, de imagens de treinamento. Primeiramente, as linhas das matrizes associadas às imagens são concatenadas formando um vetor e, feito isso, calcula-se a média desse conjunto. A seguir, subtrai-se a média de cada um dos vetores de $I$, obtendo $I_1, I_2, ..., I_n$.

Esses vetores são arranjados em uma matriz X, um vetor por coluna.

$$X = [I_1, I_2, ..., I_n]. \quad (1)$$

onde, $M = XX^T$ a matriz de covariância do conjunto de treinamento. Os componentes principais são calculados resolvendo a equação:

$$R^T(XX^T)R = \Lambda. \quad (2)$$

onde $\Lambda$, é a matriz diagonal de autovalores e $R$ é a matriz dos autovetores. O autovetor associado ao maior autovalor indica a característica de maior variância; o autovetor associado ao segundo maior autovalor é ortogonal ao primeiro e, assim sucessivamente. Os autovalores são ordenados e apenas $N$ autovetores associados aos maiores autovalores são usados para definir o subespaço (PCA) que, consequentemente, terá dimensão $N$.

No algoritmo proposto, a PCA é executada nas subimagens obtidas após a aplicação da TWC.

### B. Transformada Wavelet Complexa

A Transformada Wavelet Real [11], [13] é uma poderosa ferramenta na resolução de problemas das ciências e engenharia. Entre eles podemos destacar: redução de ruído em sinais de áudio e imagens, compressão de sinais, detecção de objetos e reconhecimento de imagens. A TWR é sensível a oscilações e não é invariante a deslocamentos.

A Transformada Wavelet Complexa [4], [5] surge como uma alternativa que supera a TWR, já que é menos sensível a oscilações e é aproximadamente invariante a deslocamentos.

Na Figura 2, tem-se coeficientes obtidos após a aplicação da TWR na função apresentada na Figura 1. Nota-se que no gráfico da Figura 2, tem-se uma oscilação o que dificulta a localização de singularidades. Todavia, quando aplicamos a mesma função à TWC e geramos um gráfico com a magnitude dos coeficientes Wavelets, temos o gráfico da Figura 3, cujo ponto de singularidade é bem mais fácil de ser determinado, já que o gráfico não apresenta oscilações.

Na Figura 4, os dois primeiros gráficos são da mesma função: um deslocado em relação ao outro. Os dois últimos gráficos da Figura 4 foram obtidos aplicando a TWR aos dois primeiros gráficos. Pode ser observado que os dois últimos gráficos da Figura 4 são diferentes, ou seja, representações diferentes para a mesma função, o que não é desejável quando se está trabalhando com reconhecimento de padrões. Os dois gráficos da Figura 5, foram obtidos aplicando a TWC aos dois primeiros gráficos da figura 4. Nesse caso, observa-se que os gráficos são praticamente os mesmos, logo, um pequeno deslocamento não pertuba a representação do sinal após a aplicação da Transformada.

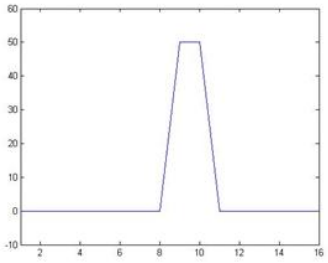



Figura 1. Sinal Original

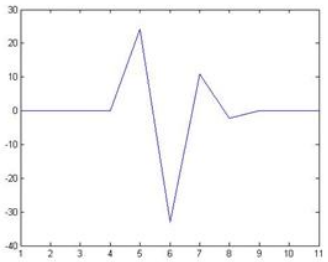



Figura 2. Sinal obtido após aplicação da TWR

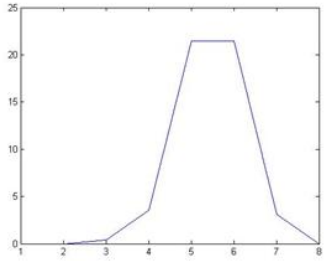



Figura 3. Sinal obtido após aplicação da TWC

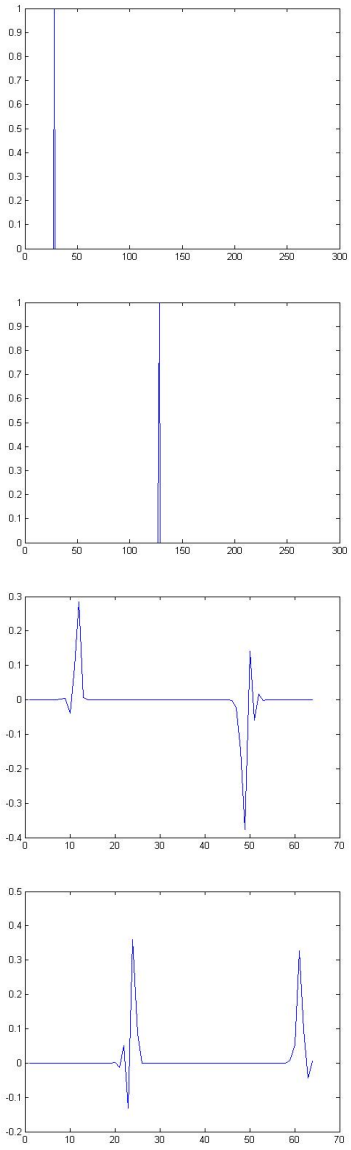
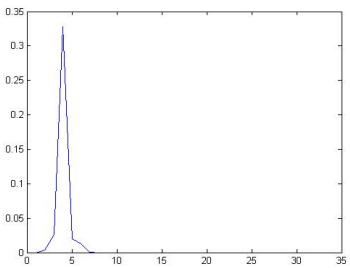


Figura 4. Sinais Original (Originais), transladado e sinais obtidos após aplicação da TWR



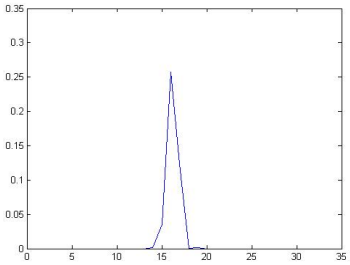


Figura 5. Sinais obtidos após aplicação da TWC

Uma abordagem para implementar a TWC é a chamada TWC de árvore dupla (TWC-AD) que emprega duas Transformadas Wavelets Discretas Reais (TWDR). A primeira determina a parte real e, a segunda, a parte imaginária da TWC-AD, ou seja, $\psi(t) = \psi_h(t) + j\psi_g(t)$, $j^2 = -1$ .

A TWC-AD-2D é caracterizada por seis Wavelets:

$$\begin{aligned}
\psi^1(x,y) &= \phi(x)\psi(y) \\
\psi^2(x,y) &= \psi(x)\psi(y) \\
\psi^3(x,y) &= \psi(x)\bar{\phi}(y) \\
\psi^4(x,y) &= \bar{\psi}(x)\bar{\phi}(y) \\
\psi^5(x,y) &= \bar{\psi}(x)\phi(y) \\
\psi^6(x,y) &= \phi(x)\bar{\psi}(y)
\end{aligned} \tag{3}$$

onde $\psi(x) = \psi_h(x) + j\psi_g(x)$ e $\phi(x) = \phi_h(x) + j\phi_g(x)$ são funções Wavelet e escala em uma dimensão.

Cada nível da Transformada está associada a oito subimagens, duas subimagens aproximações e seis subimagens detalhes, cujos coeficientes são correspondentes aos ângulos -75, -45,-15,75,45,15 conforme pode ser observado na Figura 7.

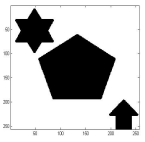
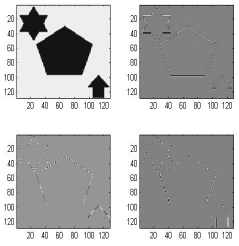


Figura 6. Imagem Original



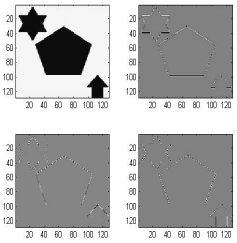


Figura 7. Subimagens obtidas após aplicação da TWC

A TWC foi utilizada neste trabalho com objetivo de reduzir efeitos indesejáveis causados por variação de iluminação, o que poderia comprometer o reconhecimento. Outra característica importante é sua seletividade direcional que permite a detecção de bordas em diferentes resoluções.

### C. Agrupamento Fuzzy

O agrupamento é uma técnica de classificação não supervisionada de padrões (dados) em grupos (clusters). Intuitivamente, padrões semelhantes devem estar no mesmo grupo e padrões distintos devem estar em grupos diferentes.

O algoritmo c-means [6], [7] é utilizado para determinar agrupamentos e seus centros. A relação entre o dado e o agrupamento (centros) é feita através de um grau de pertinência que pode variar em $[0, 1]$, em que $0$ indica exclusão total desse dado do agrupamento e $1$ indica inclusão total do dado no agrupamento. Quanto mais próximo do(s) centro(s) o dado estiver, maior será seu grau de pertinência.

Os dados $X = \{x_k | k = 1, ...n\}$ são classificados minimizando uma função objetivo baseada em uma norma e nos centros dos agrupamentos. A função objetivo é uma soma ponderada dos erros quadráticos entre os grupos que é definida por

$$J_m(X, U, V) = \sum_{k=1}^{n} \sum_{i=1}^{c} \mu_{ik}^m ||x_k - v_i||_A^2 \tag{4}$$

sendo que $1 \leq m \leq \infty$, onde

$$V = \{v_1, v_2, ..., v_c\}, v_i \in R^n \tag{5}$$

é o conjunto dos centros dos agrupamentos e $U = (\mu_{ik})$ é uma matriz em que cada elemento $\mu_{ik}$ representa o grau de pertinência de $x_k$ no agrupamento $i$.

Se $||x_k - v_i||_A^2 > 0 \ \forall i, k$ então $U, V$ podem minimizar $J_m$ apenas quando $m > 1$ e

$$v_i = \frac{\sum_{k=1}^{n} (\mu_{ik})^m x_k}{\sum_{k=1}^{n} (\mu_{ik})^m}, \quad 1 \leq i \leq c, \tag{6}$$

$$\mu_{ik} = \frac{1}{\sum_{j=1}^{c} \left( \frac{||x_k - v_i||_A^2}{||x_k - v_j||_A^2} \right)^{\frac{1}{m-1}}} \tag{7}$$

$1 \leq i \leq c$ e $1 \leq k \leq n$.

O algoritmo c-means foi utilizado para se determinar os centros de cada classe (ou grupo) que, posteriormente, foram usados no cálculo da função de pertinência $\mu_{ik}$.

### D. Integral Fuzzy

Integrais Fuzzy [10], [12] são funcionais não-lineares definidos com relação a uma Medida Fuzzy. Um conjunto de funções:

$$g : P(Y) \rightarrow [0, 1] \tag{8}$$

($P(Y)$ denota a família de todos os subconjuntos de $Y$) é chamado uma Medida Fuzzy se as seguintes condições forem satisfeitas:

1) $g(\emptyset) = 0, g(Y) = 1$;
2) $g(A) \leq g(B), A \subset B, \forall A, B \in P(Y)$;
3) $\lim_{i \to \infty} g(A_i) = g(\lim_{i \to \infty} A_i)$.

se $\{A_i\}_{i=1}^{\infty}$ é uma sequência de conjuntos mensuráveis, uma Medida $g_\lambda$ Fuzzy satisfaz a propriedade adicional:

$$g(A \cup B) = g(A) + g(B) + \lambda g(A) g(B)$$

para todo $A, B \subset Y$, $A \cap B = \emptyset$ e para algum $\lambda > -1$.

O valor de $\lambda$ pode ser determinado a partir da condição $g(\lambda) = 1$, resolvendo

$$g_\lambda(Y) = \frac{1}{\lambda} \left( \prod_{i=1}^{n} (1 + \lambda g^i) - 1 \right), \lambda \neq 0$$

onde $\lambda \in (-1, \infty)$ e $g^i$ é o valor da função densidade. O cálculo da Integral Fuzzy com relação à Medida $g_\lambda$-Fuzzy com os valores da função densidade $g^i$. Estes valores podem ser interpretados como grau de importância de cada um dos classificadores.

De acordo com [10], a Integral Fuzzy da função $h$ calculada sobre $Y$ com relação Medida Fuzzy $g$ é definida por:

$$\int_Y h(y) \circ g(.) = sup_{\alpha \in [0,1]} [min[\alpha, g(\{y | h(y) \geq \alpha\})]]. \tag{9}$$

Quando os valores de $h(.)$ são ordenados em uma sequência decrescente, a Integral Fuzzy de Sugeno é calculada como segue abaixo:

$$\int_Y h(y) \circ g(.) = max_{i=1}^{n} [min(h(y_i), g(A_i))]. \tag{10}$$

onde $A_i = \{y_1, ..., y_n\}$ denota um subconjunto de elementos do universo de discurso.

Os valores de $g(A_i)$ assumidos pela Medida Fuzzy sobre o subconjunto de elementos correspondentes é determinado por:

1) $g(A_1) = g(\{y_1\}) = g^1$
2) $g(A_i) = g^i + g(A_{i-1}) + \lambda g^i g(A_{i-1}), \quad 1 \leq i \leq n.$

A equação (10) não verifica a propriedade de extensão da Integral de Lebesgue, isto é, quando a medida é aditiva a expressão acima não retorna à integral no sentido de Lebesgue.

Para contornar isso, [14] propuseram a chamada Integral de Choquet calculada da seguinte maneira:

$$\int_Y h(y)dg(.) = \sum_{i=1}^{n}[h(y_i) - h(y_{i+1})]g(A_i)h(y_{n+1}). \quad (11)$$

onde $A_i = \{y_1, ..., y_n\}$ denota um subconjunto de elementos do universo de discurso.

No algoritmo proposto, os valores $h(y_{ik})$ da classe $k$ no classificador $i$ são calculados da seguinte forma:

$$h(y_{ik}) = \sum_{\mu_{ij} \in C_k} (\mu_{ij})/N_k. \quad (12)$$

Os valores de $g^i$ foram obtidos a partir de

$$g^i = \frac{w_i p_i}{\sum_i p_i} \quad (13)$$

onde $p_i$ denota a taxa de reconhecimento de cada classificador para um conjunto de treinamento, os valores $w_1, w_2, ..., w_6$ são pesos associados a cada classificador e foram obtidos experimentalmente.

A seguir é descrito o algoritmo proposto em dois estágios.

## II. Algoritmo Proposto

O algoritmo proposto para Reconhecimento de Faces Humanas baseado na TWC, PCA, Agrupamento Fuzzy e Integral Fuzzy divide-se em dois estágios:

### A. Estágio 1: Treinamento

Primeiro são selecionadas $C$ imagens de cada classe (pessoa) e, se a base de imagens possui $k$ classes, são selecionadas $Ck$ imagens. A cada imagem selecionada é aplicada TWC, obtendo-se seis subimagens $I1, I2, ..., I6$. Seja $CI1$ o conjunto as subimagens $I1$, $CI2$ o conjunto das subimagens $I2$ e assim por diante. Depois de concatenar cada uma das linhas das matrizes associadas às subimagens obtidas na etapa anterior, é aplicada a PCA a cada um dos conjuntos $CI1, ..., CI6$, obtendo-se os subespaços $S1, S2, ..., S6$. Na última etapa, o algoritmo c-means é aplicado a cada um dos subespaços gerados anteriormente, obtendo-se os centros de cada grupo.

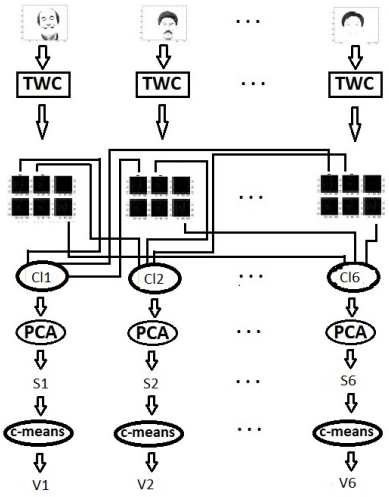



Figura 8. Estágio 1

### B. Estágio 2: Teste

Após a escolha da imagem que será submetida ao reconhecimento, aplica-se a TWC, obtendo-se seis subimagens $T1, T2, ..., T6$. Cada subimagem é projetada no subespaço correspondente, isto é, $T1$ é projetada em $S1$, $T2$ em $S2$ e assim sucessivamente. O algoritmo c-means fornece a função de pertinência associada ao vetor em cada uma das classes e em cada subespaço. A Integral Fuzzy é responsável pela fusão das funções de pertinência. Na Figura 9 tem-se o diagrama do estágio 2.

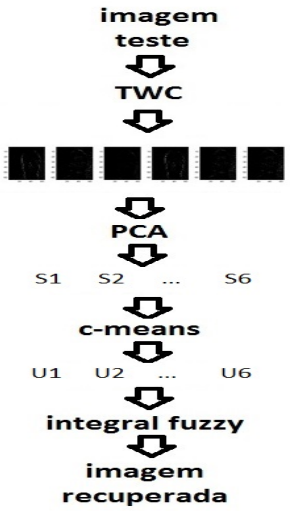



Figura 9. Estágio 2

A seguir, temos o diagrama do algoritmo apresentado em [3].

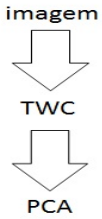



Figura 10. Diagrama do Algoritmo TWC+PCA

## III. Resultados Experimentais

O algoritmo proposto foi implementado utilizando-se o software Matlab 7.6 (R2008a)@, em um notebook LG, Intel Core i5, com 4G de memria RAM.

Foram realizados testes utilizando-se 165 imagens da base Yale (divididas em 15 classes, com 11 imagens cada classe) e 1320 imagens da base Essex (divididas em 132 classes, com 10 imagens cada classe). Na Figura 11, tem-se uma amostra da base Yale e na Figura 12, uma amostra das imagens da base Essex. As imagens das duas bases foram redimensionadas para a resolução $128 \times 128$.

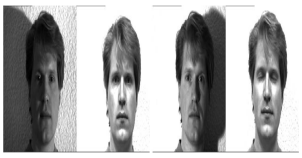


Figura 11. Base de imagens Yale

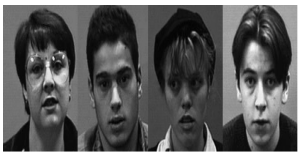


Figura 12. Base de imagens Essex

No algoritmo proposto, para cada imagem submetida ao reconhecimento, primeiro é aplicada a TWC, que gera as subimagens que serão projetadas no subespaço PCA; e para cada vetor, no subespaço PCA, é calculado o grau de pertinência em relação aos agrupamentos. Por fim, é realizada a fusão dos classificadores. Haverá sucesso se a imagem submetida ao reconhecimento e a imagem recuperada forem do mesmo indivíduo. Caso contrário, o reconhecimento terá falhado.

Os resultados na tarefa de Reconhecimento de Faces Humanas foram obtidos selecionando alguns parâmetros como a dimensão do subespaço gerado pela PCA (d), a resolução da imagem obtida após aplicação da Transformada Wavelet (r), o número de centros de cada agrupamento (c) e o número de imagens do conjunto de treinamento (C). O algoritmo proposto é comparado com o algoritmo apresentado na Figura 10, nos dois algoritmos a distância utilizada foi a euclidiana.

Nas Tabelas, 1 e 2 os parâmetros utilizados foram $d = 5, 10, 15$, $r = 32 \times 32$, $c = 2$, $C = 3, 4, 5$ e, dessa forma foram submetidas ao reconhecimento $8, 7, 6$ imagens. Já nas Tabelas 3 e 4 temos $d = 10, 20, 30, 40$, $r = 64 \times 64$, $c = 2$ e $C = 3, 4, 5$.

Tabela. 1.Taxa de Reconhecimento do Algoritmo Proposto

| dimensão | **Conjunto de Treinamento** | | |
|---|---|---|---|
| | 3 | 4 | 5 |
| 5 | 74,1 % | 76,1 % | 73,3 % |
| 10 | 81,6 % | 84,7 % | 81,9 % |
| 15 | 77,5 % | 81,9 % | 86,6 % |

Tabela. 2.Taxa de Reconhecimento do Algoritmo TWC+PCA

| dimensão | **Conjunto de Treinamento** | | |
|---|---|---|---|
| | 3 | 4 | 5 |
| 5 | 68,3 % | 61,9 % | 66,6 % |
| 10 | 72,5 % | 74,2 % | 78,8 % |
| 15 | 76,6 % | 79 % | 82,2 % |

Tabela. 3.Taxa de Reconhecimento do Algoritmo Proposto

| dimensão | **Conjunto de Treinamento** | | |
|---|---|---|---|
| | 3 | 4 | 5 |
| 10 | 78,3 % | 84,7 % | 83,3 % |
| 20 | 79,1 % | 87,6 % | 91,1 % |
| 30 | 82,5 % | 90,4 % | 93,3 % |
| 40 | 86,6 % | 90,4 % | 94,4 % |

Tabela. 4.Taxa de Reconhecimento do Algoritmo TWC+PCA

| dimensão | **Conjunto de Treinamento** | | |
|---|---|---|---|
| | 3 | 4 | 5 |
| 10 | 73,3 % | 74,2 % | 78,8 % |
| 20 | 78,3 % | 84,7 % | 84,4 % |
| 30 | 80 % | 86,6 % | 84,4 % |
| 40 | 80 % | 86,6 % | 84,4 % |

Os resultados apresentados nas Tabelas 5 e 6 foram obtidos utilizando a base de imagens Essex, com resoluo $16 \times 16$, c=2, d=10,20,30,40,50, C=3,4,5.

Tabela. 5.Taxa de Reconhecimento do Algoritmo Proposto

| dimensão | **Conjunto de Treinamento** | | |
|---|---|---|---|
| | 3 | 4 | 5 |
| 10 | 97,6 % | 98,4 % | 98,4 % |
| 20 | 98,3 % | 99,2 % | 99,2 % |
| 30 | 98,9 % | 99,2 % | 99,2 % |
| 40 | 98,9 % | 99,2 % | 99,2 % |
| 50 | 98,9 % | 99,1 % | 99,2 % |

Tabela. 6.Taxa de Reconhecimento do Algoritmo TWC+PCA

| dimensão | **Conjunto de Treinamento** | | |
|---|---|---|---|
| | 3 | 4 | 5 |
| 10 | 90,5 % | 89,8 % | 91,3 % |
| 20 | 93 % | 94 % | 95 % |
| 30 | 93,9 % | 94,8 % | 94,5 % |
| 40 | 93 % | 94,3 % | 95,6 % |
| 50 | 94,4 % | 95,2 % | 95,1 % |

## IV. CONCLUSÃO

O reconhecimento de faces humanas é dividido em três partes: aquisição dos dados, extração de características e tomada de decisão. Para extração de características, o algoritmo proposto utiliza a decomposição da imagem através da TWC, que permite descrever cada uma das faces em seis direções e a PCA, reponsável pela redução da dimensão dos dados obtidos após a aplicação da TWC. Já na tomada de decisão utiliza a função de pertinência obtida a partir do algoritmo c-means Fuzzy, que determina os representantes de cada classe, de forma que a distância entre os elementos de uma mesma classe e seus representantes seja minimizada. A integral fuzzy é responsável pela fusão dos classificadores.

Em todos os testes realizados independente da base utilizada Yale ou Essex, da resolução da imagem (16 $\times$ 16, 32 $\times$ 32 ou 64 $\times$ 64), da dimensão do subespaço PCA (5,10,15 ou 10,20,30,40,50) e do conjunto de treinamento (3,4,5). O algoritmo proposto superou o algoritmo apresentado em [3].

## REFERÊNCIAS


[1] A. M. Martinez and A. C. Kak, *PCA versus LDA*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 228-233, 2001.

[2] B. A. Draper, K. Baek, M. Stewart and J. R. Bevedge, *Recognition faces with PCA and ICA*, Computer Vision and Image Understanding, 115-137, 2003.

[3] B. Dharmendar, B. K. Naveen, M. Sunitha and K. Sunanda, *Dual Tree Complex Wavelet Transform for Face Recognition Using PCA Algorithm*, International Journal of Research and Computational Technology, 58-70, 2013.

[4] C. C. Liu and D. Q. Dai, *Face Recognition Using Dual-Tree Complex Wavelet Features*, IEEE Transactions on Image Processing, 1726-1739, 2009.

[5] I. W. Selesnick, R. G. Baraniuk and N. G. Kingsbury, *The Dual-Tree Complex Wavelet Transform*, IEEE Signal Processing Magazine, 123-151, 2005.

[6] J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*, Plenum Press, 1-13, 1981.

[7] J. C. Bedzdek and J. C. Dunn, *Optimal Fuzzy Partitions: A heuristic for estimating the parameters in a mixture of normal distributions*, IEEE Transactions on Computers, 835-838, 1975.

[8] J. C. Goswami and A. K. Chan, *Fundamentals of Wavelets: Theory, Algorithms and Applications*, Wiley, 2011.

[9] J. L. Water, *A Primer on Wavelets and Their Scientific Applications*, Chapman e Hall/CRC, 1999.

[10] K. C. Kwak and W. Pedrycz, *Face Recognition Using Fuzzy Integral and Wavelet Decomoposition Method*, IEEE Tansaction on Systems, 1666-1675, 2004.

[11] K. Kim, S. Y. Oh and H.C. Choi, *Facial Feature Extraction Using PCA and Wavelet Multi-Resolution Image*, Proceedings of the 6th IEEE International Conference on Automatic Face and Gesture Recognition, 439-444, 2004.

[12] M. Grabisch, T, Murofushi and M. Sugeno, *Fuzzy Measures and Integrals: Theory and Applications*, Phy-Verlag, 2000.

[13] P. C. Yuen, D. Q. Dai and G. C. Feng, *Wavelet based PCA for Human Face Recognition*, Image Analysis and Interpretation,IEEE Southwest Symposium, 223-228, 1998.

[14] T. Murofushi, M. Sugeno, *Measures and the Choquet Integral as an integral with respect to a Fuzzy Measure*, Fuzzy Sets Syst.,201-227, 1988.

[15] W. Zhao, R. Chellappa, P.J. Phillips, and A. Rosenfeld, Face Recognition: A Lit- erature Survey, ACM Computing Suverys, 399-458, 2003.

# Cachaça type identification using color information and computer vision

Bruno Urbano Rodrigues*, Ronaldo Martins da Costa*, Rogerio Salvini*, Anderson da Silva Soares*,
Flávio Alves da Silva†, Márcio Caliari†, Karla Cristina R. Cardoso† e Tânia Isabel M. Ribeiro‡
*Instituto de Informática
Universidade Federal de Goiás
Email: brunourb@gmail.com, ronaldocosta@inf.ufg.br,
rogeriosalvini@inf.ufg.br, anderson@inf.ufg.br
†Escola de Agronomia e Engenharia de Alimmentos
Universidade Federal de Goiás
Email: flaviocamp@ufg.br, macaliari@ig.com.br, karlagropan@hotmail.com
‡Escola Superior Agrária
Instituto Politécnico de Bragança
Email: tania.im.ribeiro@gmail.com

***Abstract*—Cachaça is a type of distilled drink from sugarcane with great economic importance. Its classification includes three types: aged, premium and extra premium. These three classifications are related to the aging time of the drink in wooden casks. Besides the aging time, it is important to know what kind of wood were used in the barrel storage in order to inform the correct properties for each drink. This paper shows a method for automatic recognition of wood type using color information from a computer vision system. The results showed that information color have relevant information to identify a cachaça. It's a better method when compare chemical features method. In the study case using two classifiers, LDA (Linear Discriminant Analysis) and QDA (Quadratic Discriminant Analysis). Comparing both classifiers, the LDA and QDA had 98.13% and 99.03% of accuracy, respectively, using just color information.**

## I. Introdução

A cachaça é o destilado mais consumido entre as bebidas alcoólicas brasileiras. É um tipo de bebida especial produzida a partir da cana-de-açucar (*Saccharum sp*) similar ao rum. Tem como diferencial o uso de diferentes tipos de madeira em seu processo de envelhecimento.

O envelhecimento consiste em armazenar a cachaça em barris ou tonéis de madeira por um tempo determinado. Tal processo produz alterações na composição química, no aroma, no sabor e na cor da bebida [1]. A legislação classifica a cachaça em três tipos: cachaça envelhecida, cachaça *premium* e cachaça *extra premium*. A diferença entre os tipos está relacionado ao quantitativo de armazenagem e ao tempo de armazenamento. A cachaça envelhecida e a cachaça *premium* tem período de envelhecimento de no mínimo 1 ano. Na composição da cachaça envelhecida 50% deve ser envelhecida e os outros 50% utiliza-se de cachaça não-envelhecida através do processo chamado *blend*, onde é realizada mistura de cachaças. Por sua vez a cachaça *premium* possui 100% de cachaça envelhecida. A cachaça *extra-premium* tem período de envelhecimento de no mínimo 3 anos e contém somente cachaça envelhecida. Diferentemente das bebidas mais famosas do mundo como uísques, *brandies* e vinhos que são armazenados somente em barris de carvalho, a cachaça é a única que usa diferentes madeiras para o processo de envelhecimento[1][2]. Cada madeira confere à bebida uma análise sensorial que envolve a medição, interpretação e compreensão das respostas humanas para as propriedades percebidas pelos sentidos (paladar - sabor, olfato - aroma, visão - cor). O conhecimento detalhado da composição química e sensorial da cachaça, bem como o tempo de maturação constituem fatores importantes no controle de qualidade da bebida e avaliação das alterações que possam contribuir para a melhoria de processos de produção.[1]

### A. Trabalhos correlatos

Nesta seção apresentamos alguns trabalhos correlatos a classificação e caracterização de destilados que abrangem os seguintes temas: cromatografia gasosa, espectrometria, visão computacional, redes neurais, algoritmos genéticos, análise discriminante e análise de componentes principais.

O trabalho de De Souza[3] utilizou cromatografia gasosa - olfatometria - para separar e caracterizar os odores presentes na cachaça e rum. Estes dois produtos de cana de açúcar foram comparados e os padrões identificados a partir de uma análise sensorial descritiva. A desvantagem da utilização deste método é manutenibilidade, pois possui alto custo. Augusti[4] demostrou a diferenciação entre cachaça artesanal e o rum utilizando espectrometria de massa por ionização. Utilizou a análise de componentes principais (PCA), abordagem estatística na qual os dados são representadas por um subconjunto de seus autovetores, observando tipo de madeiras (amburana -*Amburana cearensis* e jequitibá - *Cariniana legalis*). Seu trabalho contribui para que novos estudos possam utilizar desta técnica para a identificação de cachaças artesanais e industriais bem como a detecção de adulterações pela adição de caramelo e outras substâncias como corantes.

Wan[5] utilizou a visão computacional combinada com redes neurais artificiais. Foi utilizada uma abordagem à nível estrutural e microscópico dos vinhos a serem classificados, por meio da análise da microestrutura e características da textura, fator que influencia na atribuição da cor à amostra.

Partindo da idéia de que os diferentes vinhos têm variação na microestrutura(microscopia) e micrografias(partículas), o estudo objetivou a extração de características em comum para delimitar um padrão. Para tal, utilizaram redes neurais para a classificação das amostras. Os resultados apresentados confirmam que é possível a classificação do vinho através da sua micrografia, possibilitando a utilização das característica em outros contextos.

Manuel[6] utiliza método baseado na análise discriminante quatrática(QDA) combinado com a transformada quadrádica de Fisher para a classificação de vinhos. A análise discriminantes quadrática é utilizada como esquema de classificação e para fins de comparação foi usado outros três métodos de extração de características: transformada linear de Fisher, transformada de Fourier e transformada Wavelet. Em seus resultados foi apresentado ganho na utilização da discriminante quadrática como método de classificação combinada com a transformada de Fisher para a extração de características do vinho.

Boisier[7] utiliza o $\Delta E$ baseado no espaço de cores CIELab nas amostras e demostra o agrupamento de acordo com as tonalidades classificadas. O $\Delta E$ mede a distância entre duas cores utilizando a métrica euclidiana. O objetivo proposto foi representar as cores dos vinhos com número limitado de cores que foram chamadas de *nuances*. A aplicação do $\Delta E$ objetivou realizar um comparativo com o modelo de cor HVS (*hue*, *saturation* e *value*), observando o brilho, cromaticidade e saturação. Assim, analisar o espectro de cor, classificar e agrupar de acordo com a tonalidade. Os resultados obtidos são encorajadores na medida em que permitem uma caracterização exata e reprodução da cor do vinho. O modelo de cor RGB é um sistema de cores aditivas formado por Vermelho (Red), Verde (Green) e Azul (Blue). Cores aditivas são cores emitidas ou projetadas. A cor é gerada pela mistura de vários cumprimentos de onda luminosa, provocando uma sensação de cor quando atinge o olho. Formatos RGB, também conhecidos por *true-color*, usam 8-bits por canal. O modelo de cor CIELab é um sistema de cores subtrativo. CIELab descreve as cores básicas em três qualidades: L* é luminosidade, a* e b* contem a informação de croma. L* é luminância, medida da densidade e intensidade de uma luz refletida numa dada direção. As informações a* e b*, referem-se ao valor das cores[8]. Qiongshuai[9], em sua análise demonstra o ganho da utilização de algoritmos genéticos na preleção e classificação de vinhos, combinada com visão computacional. Kruzlicova[10], avalia dados por meio de uma rede neural artificial e para método comparativo utiliza a análise de variância (ANOVA). Cozzolino[11] propôs investigar a relação entre a análise sensorial, visibilidade (VIS) e espectroscopia de infravermelho (NIR) para avaliar as propriedades sensoriais de variedades comerciais de vinhos australianos utilizando PCA.

Os métodos descritos nos artigos relacionados não abordam a multiplicidade de madeiras que podem ser utilizadas, bem como o uso agregado dos valores de dados químicos, valores obtidos pelo colorímetro e não há relação, até então, de fotografias digitais das amostras utilizando-se do modelo de cor RGB. Ora os trabalhos utilizam apenas os dados químicos, ora utilizam apenas os dados do colorímetro (modelo de cor CIELab). Quando utilizam dados químicos e dados do colorímetro, não observam o modelo de cor RGB. O colorímetro é um equipamento que caracteriza amostras de cores para obter uma medida objetiva das características da cor. Este equipamento geralmente está disponível somente em laboratórios de pesquisa e indústrias. Um ponto relevante é tornar acessível esta tecnologia independente do produtor. Observando os métodos instrumentais, o custo, a manutenibilidade e manuseio é feito por um especialista. Com os métodos computacionais pode-se trazer resultados otimizando tempo e recursos para quaisquer usuário seja ele um especialista ou produtor.

Portanto, neste trabalho é proposto um método de classificação de amostras de cachaça armazenadas em tipos de madeiras diferentes a fim de definir a identidade da cachaça baseado em suas características de cor. Sendo assim, catalogar cores a partir da madeira na qual a cachaça foi armazenada. Será realizado cruzamento das informações obtidas na análise química, nas informações extraídas de colorímetros bem como dados obtidos através da aplicação de algoritmos de processamento digital de imagens, executados sobre as fotografias digitais de amostras da cachaça. Foi utilizada a técnica de Análise de Componentes Principais (PCA), para tratar problemas de mal condicionamento de variáveis e são utilizados 2 classificadores: Análise Discriminante Linear(LDA) e Análise Discriminante Quadrática(QDA).

## II. Materiais e Métodos

### A. Amostras

Foram avaliadas amostras de cachaça com até 36 meses de envelhecimento em tonéis de Amburana (*Amburana cearensis*), Carvalho (*Quercus spp*) e Castanheira (*Bertholletia excelsa H.B.K*) . O tempo de envelhecimento foi acompanhado a cada 4 meses e as amostras avaliadas são de 4 tonéis para cada tipo de madeira. Desta maneira o total de amostras que serão analisadas são de 36 amostras por madeira conforme é demostrado na Tabela I.

| Madeira | Amostras (meses) | Total |
|---|---|---|
| amburana (*Amburana cearensis*), | 4 - 8 - 12 - 16 - 20 - 24 - 28 - 32 - 36 | 36 |
| carvalho (*Quercus spp*) | 4 - 8 - 12 - 16 - 20 - 24 - 28 - 32 - 36 | 36 |
| castanheira (*Bertholletia excelsa H.B.K*) | 4 - 8 - 12 - 16 - 20 - 24 - 28 - 32 - 36 | 36 |
| Total de amostras | | 108 |

Tabela I: Amostras de cachaça analisadas no período de 36 meses.

As análises físicas e químicas foram realizadas nos laboratórios de tecnologia de bebidas e de análise físico-químicas, da Escola de Agronomia da Universidade Federal de Goiás. Estas análises seguem parâmetros estabelecidos pela Instrução Normativa nº 13 do MAPA (Ministério da Agricultura, Pecuária e Abastecimento)[12] que define diretrizes sobre o consumo e comercialização.

A determinação do pH, densidade, teor alcoólico real a 20ºC, acidez volátil fixa e total, extrato seco, compostos fenólicos, cor e atividade antioxidante foram realizadas nos tempos 0, 2, 4, 6, 8, 10 e 12, ou seja, de 2 em 2 meses de armazenamento para que fossem observadas as alterações,

durante o período de envelhecimento. Na análise foram levantados os atributos a serem utilizados no modelo. Os atributos são descritos na Tabela II.

| Descrição dos atributos químicos | |
|---|---|
| Atributo 1 | Álcool Aparente |
| Atributo 2 | Álcool Real |
| Atributo 3 | Esteres Totais |
| Atributo 4 | Acetato Etila |
| Atributo 5 | Lactato Etila |
| Atributo 6 | Aldeídos |
| Atributo 7 | Alcoois Totais |

| Descrição dos atributos químicos | |
|---|---|
| Atributo 8 | n-propil |
| Atributo 9 | Isobutanol |
| Atributo 10 | Isoamílico |
| Atributo 11 | 1-Butanol |
| Atributo 12 | 2-Butanol |
| Atributo 13 | Álcool Metílico |
| Atributo 14 | Furfural |

Tabela II: Atributos químicos extraídos da análise das amostras

### B. Sistema de visão computacional

Por seguinte, as amostras foram fotografadas pela câmera digital *Canon EOS REBEL XS* com configuração de ISO 100, abertura a 4.0mm e formato de imagem configurada para *RAW* que contém a totalidade dos dados da imagem tal como captada pelo sensor da câmera fotográfica. A luz do ambiente para fotografar as amostras foi controlada através de um dispositivo que permite a incidência da luz na posição contrária à lente da câmera fotográfica. Um filtro especial foi utilizado para evitar reflexos no líquido e permitir a captura de uma imagem digital adequada para o processamento.

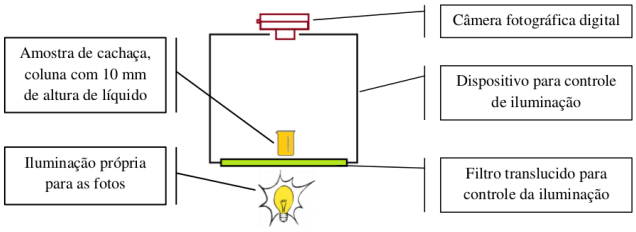



Figura 1: Sistema para captura das fotografias digitais

A Figura 1 apresenta o projeto do dispositivo desenvolvido que foi utilizado neste trabalho. A técnica foi inspirada por Sun[13] em seu trabalho de classificação de carne bovina. O objetivo do dispositivo é controlar o ambiente da fotografia digital para uma melhor absorção das cores do objeto alvo, neste contexto a cachaça, com o propósito de observar uma correlação entre as características de cores obtidas pelo colorímetro (modelo CIELab L*a*b* - *Lightness, redness and yelowness*) e da câmera digital (modelo RGB) (*Red, Green and Blue*).

| Descrição dos atributos para CIELab | |
|---|---|
| Atributo 15 | *Lightness* |
| Atributo 16 | *Redness* |
| Atributo 17 | *Yelowness* |

| Descrição dos atributos para RGB | |
|---|---|
| Atributo 18 | *Red* |
| Atributo 19 | *Green* |
| Atributo 20 | *Blue* |

Tabela III: Representação dos atributos para os modelos de cores CIELab(a) e RGB(b) com seus respectivos números e descrição.

Acima é apresentado as propriedades utilizadas no modelo CIELab e RGB apresentadas. Assim como na análise química, as propriedades do modelo de cor CIELab e RGB foram nomeadas e separadas para serem utilizadas no classificador.

### C. Validação cruzada

Para validação do modelo obtido usou-se procedimento de validação cruzada (*cross-validation*), técnica que avalia a capacidade de generalização de um modelo, a partir de um conjunto de dados.[14] Devido ao número limitado de amostras, a técnica de validação cruzada foi utilizada para se medir a acurácia dos classificadores. Envolve o uso de uma única observação a partir da amostra original, como os dados de validação e dados de treinamento. Neste trabalho utilizou-se o método chamado *leave-and-out*, onde *n* é igual ao número total de dados(N). Isto é, são construídos *n* classificadores, onde *n* é o número de exemplos de treinamento e aproveita-se ao máximo os dados e não envolve sub-amostragem aleatória.

### D. Análise de Componentes Principais

O método para extração de características utilizado foi a a análise de componentes principais (PCA). Técnica da estatística multivariada que consiste em transformar um conjunto de variáveis originais em outro conjunto de variáveis de mesma dimensão denominadas de componentes principais A análise de componentes principais é associada à redução de um conjunto de variáveis, com menor perda possível da informação, onde os componentes principais é uma combinação linear de todas as variáveis originais. São independentes entre si e estimados com o propósito de reter, em ordem de estimação, o máximo de informação, em termos da variação total contida nos dados.[15]

### E. Classificadores utilizados

Para a classificação das amostras de cachaça foi utilizada a Análise Discriminante Linear (LDA - *Linear Discriminant Analysis*) e Análise Discriminante Quadrática (QDA - *Quadratic Discriminant Analysis*). O objetivo de ambas consite em obter funções matemáticas capazes de classificar um objeto em uma de suas classes, com base nas características consideradas, buscando minimizar a probabilidade de classificação errada.

Na análise discriminante linear considera que a diferença entre as classes está relacionada apenas com o vetor média das classes, visto que a matriz de covariância é igual para todas as classes. A fronteira de decisão fica definida por uma função linear e uma razoável acurácia pode ser obtida, desde que haja uma diferença razoável entre os vetores média de cada classe. Se os vetores médias forem muito próximos ou quase iguais, o classificador linear fornecerá resultados pobres. Por sua vez, a análise discriminante quadrática estima que cada classe tem vetor média e matriz de covariâncias próprios, resultando em funções de decisão quadráticas[16][17]. A métrica utilizada foi a distância de *Mahalanobis* que difere da distância Euclidiana por levar em consideração a correlação entre os conjutos de dados.

### F. Resultados e discussão

Foram realizados experimentos para verificar quais informações são mais importantes para o problema de classificação do tipo de madeira para o envelhecimento de cachaça. As informações obtidas pelo estudo foram divididas em duas partes: dados químicos e dados de informações do modelo de cores (CIELab e RGB).

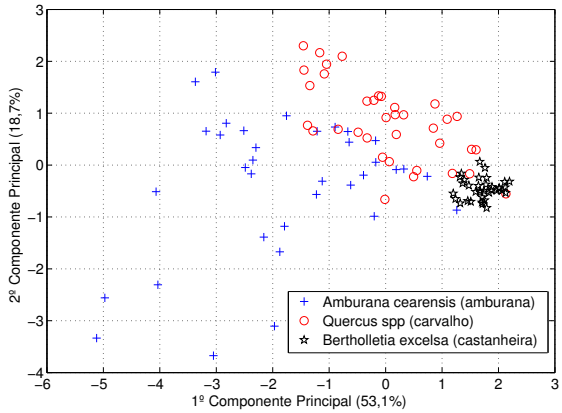



Figura 2: Gráfico de dispersão de objetos utilizando as duas componentes principais de maior variância

A Figura 2 apresenta a dispersão dos objetos das 3 classes consideradas projetadas sobre as 2 primeiras componentes principais de maior variância. As componentes principais foram calculadas sobre todas as variáveis disponíveis, isto é, dados químicos e dados de informação de cor referenciados na Tabela II e Tabela III. É possível observar que o uso de componentes principais não permite uma boa discriminização das classes, visto que há uma sobreposição entre os objetos de diferentes classes. Para os dois grupos de dados medidos, foi calculada a discriminabilidade de Fisher univariada. Tal medida indica o quanto uma variável discrimina as classes consideradas. O resultado é apresentado na Figura 3.

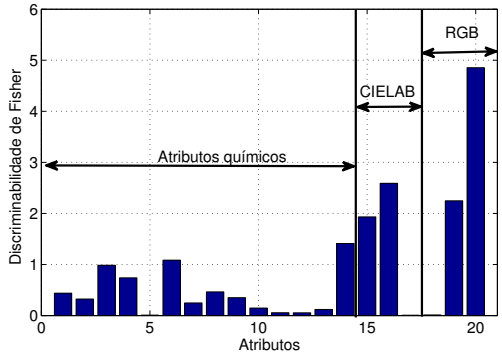



Figura 3: Discriminabilidade de Fisher utilizando atributos químicos, modelo de cores CIELab e RGB para o problema de classificação do tipo de madeira para envelhecimento de cachaça.

Como se pode notar os atributos relacionados a informação de cor tem maior discriminabilidade para o problema de classificação do tipo de madeira. Vale ressaltar que a discriminabilidade de Fisher considera o atributo de forma univariada, assim o uso dos atributos de maior discriminabilidade não implica em um bom modelo de classificação.

Identificados os atributos de maior discriminabilidade (Atributos 16 e 20), conduziu-se os experimentos utilizando classificador discriminante linear e quadrático. Para cada classificador foi utilizado atributos químicos e informações dos modelos de cores. Ao analisar a taxa de acerto das análises discriminantes na Tabela IV, foi verificado que a análise discriminante quadrática (QDA), obteve o melhor resultado (99,06%) utilizando dados de informações do modelo de cores (CIELab e RGB).

| Usando atributos químicos | | |
|---|---|---|
| | LDA | QDA |
| Taxa de acerto(%) | 93,45% | -* |
| número de erros | 7 | - |
| **Usando modelo de cores** | | |
| | LDA | QDA |
| Taxa de acerto(%) | 98,13% | 99,06% |
| número de erros | 2 | 1 |

Tabela IV: Taxa de acerto para as análises discriminantes linear e quadrática

*Não foi possível obter os resultados do classificador de análise discriminante quadrática (QDA) utilizando atributos químicos em razão da presença de colinearidade dos atributos considerados e que não permitiu a inversão da matriz de covariância que é utilizada na construção do classificador.

Uma das vantagens do uso de classificadores como de análise discriminante é a possibilidade de extrair o conhecimento acerca de como a classificação é realizada. Neste caso em particular é possível visualizar as fronteiras de decisão geradas pelo classificador através de hiperplanos. Os hiperplanos dividem o espaço em diversas regiões e estas divisões determinam a fronteira de decisão que distingue as classes. Para que os hiperplanos consigam separar as classes corretamente, não pode haver pontos de classes distintas numa mesma região.

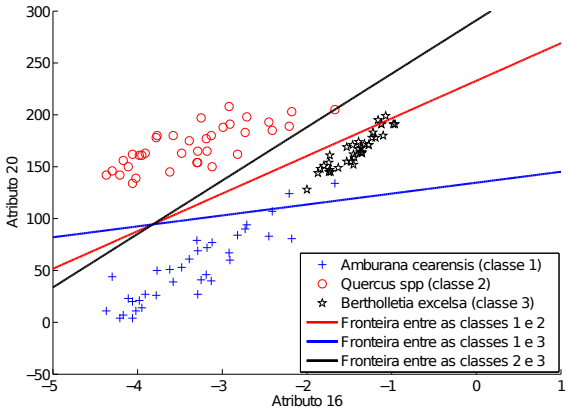



Figura 4: Gráfico de dispersão de objetos com hiperplano de separação de classes obtidos por discriminante linear

Na Figura 4 é apresentado 3 hiperplanos e 3 classes que correspondem as madeiras Amburana (classe 1 - *Amburana cearensis*), Carvalho (classe 2 - *Quercus spp*) e Castanheira (classe 3 - *Bertholletia excelsa H.B.K*). O hiperplano de cor vermelha faz a separação entre as classes 1 e 2, o hiperplano de cor azul faz a separação entre as classes 1 e 3, e o hiperplando de cor preta faz a separação entre as classes 2 e 3. Nota-se que os hiperplanos mostram que as classes são linearmentes separáveis, pois não há sobreposição das classes identificadas nas fronteiras de decisão.

Posteriormente foi realizado experimento utilizando a análise discriminante quadrática. Com a análise discriminante quadrática, pode-se utilizar além de hiperplanos, elipsóides determinando regiões de classes mais precisas. Na Figura 5, são identificadas três regiões segmentadas por hiperplanos, cada qual apresenta as classes identificadas. O hiperplano de cor vermelha faz a distinção entre as classes 1 e 2, seu traçado é linear pois o seu componente quadrático é igual a zero. Diferente dos hiperplanos de cor azul e preta. O hiperplando azul distingue as classes 1 e 3, e neste cenário há uma maior delimitação da região que identificada a classe 3. No hiperplano de cor preta a distinção está para as classes 2 e 3.

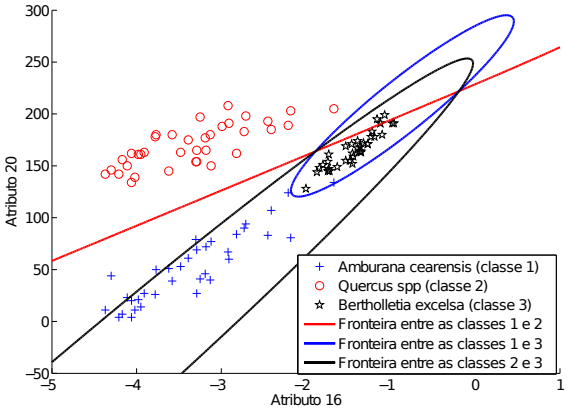



Figura 5: Gráfico de dispersão de objetos com hiperplano de separação de classes obtidos por discriminante linear quadrática

Fazendo um comparativo entre as fronteiras observadas nas figuras 4 e 5, nota-se que o classificador de discriminante quadrático gerou uma fronteira de área menor para a classe 3. Tal resultado é explicado em razão da discriminante quadrática utilizar matrizes de covariâncias individuais para cada classe o que permite que a dispersão dos objetos seja considerada de forma individual, ao contrário do discriminante linear.

### *G. Conclusão*

O artigo proposto demostra que a utilização de informações de cores para o problema de identificação do tipo de madeira para classificação de amostra de cachaça apresenta melhores resultados do que a classificação utilizando dados químicos. O custo inerente a utilização de dados químicos é alto. A mão-de-obra deve ser especializada e o seu alto custo é restrito a laboratórios de pesquisa e desenvolvimento. Assim a utilização de um método que observa o modelo de cores (RGB e CIELab) é factível na substituição dos atributos químicos, seja pelo custeio e taxa de acerto. Como todo processo de produção sua aplicação deve ser em ambiente controlado para evitar falhas e fraudes.

## AGRADECIMENTOS

Os autores da pesquisa agradecem a CAPES, FAPEG e Escola de Agricultura do Instituto Politécnico de Bragança pelo suporte financeiro provido para a pesquisa.

## REFERENCES

[1] F. W. B. Aquino, R. F. Nascimento, S. Rodrigues, and A. R. S. Casemiro, "Determinação de marcadores de envelhecimento em cachaças," *Food Science and Technology (Campinas)*, vol. 26, pp. 145 – 149, 03 2006. [Online]. Available: http://dx.doi.org/10.1590/S0101-20612006000100024

[2] J. B. Faria, D. W. Franco, and J. R. Piggott, "The quality challenge: cachaça for export in the 21st century," in *Distilled spirits: tradition and innovation 2004*. Nottingham University Press, Nottingham, UK, 2004, pp. 215–221.

[3] M. D. C. A. de Souza, P. Vásquez, N. L. del Mastro, T. E. Acree, and E. H. Lavin, "Characterization of cachaça and rum aroma," *Journal of Agricultural and Food Chemistry*, vol. 54, no. 2, pp. 485–488, 2006, pMID: 16417309. [Online]. Available: http://pubs.acs.org/doi/abs/10.1021/jf0511190

[4] P. P. de Souza, D. V. Augusti, R. R. Catharino, H. G. L. Siebald, M. N. Eberlin, and R. Augusti, "Differentiation of rum and brazilian artisan cachaça via electrospray ionization mass spectrometry fingerprinting," *Journal of Mass Spectrometry*, vol. 42, no. 10, pp. 1294–1299, 2007. [Online]. Available: http://dx.doi.org/10.1002/jms.1197

[5] Y. Wan, X. Sun, and R. Guo, "Shape and structure features based chinese wine classification," in *Computational Intelligence and Natural Computing, 2009. CINC '09. International Conference on*, vol. 2, 2009, pp. 39–43.

[6] M. A. Duarte-Mermoud, N. H. Beltrán, and M. A. Bustos, "Chilean wine varietal classification using quadratic fisher transformation," *Pattern Anal. Appl.*, vol. 13, no. 2, pp. 181–188, 2010.

[7] B. Boisier, A. Mansouri, P. Gouton, and P. Trollat, "Wine color characterization and classification for nuances reproduction," in *Signal-Image Technology Internet-Based Systems (SITIS), 2009 Fifth International Conference on*, 2009, pp. 93–98.

[8] R. C. Gonzalez and R. E. Woods, *Digital Image Processing (3rd Edition)*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 2006.

[9] L. Qiongshuai and W. Shiqing, "A hybrid model of neural network and classification in wine," in *Computer Research and Development (ICCRD), 2011 3rd International Conference on*, vol. 3, 2011, pp. 58–61.

[10] D. Kruzlicova, J. Mocak, B. Balla, J. Petka, M. Farkova, and J. Havel, "Classification of slovak white wines using artificial neural networks and discriminant techniques," *Food Chemistry*, vol. 112, no. 4, pp. 1046 – 1052, 2009. [Online]. Available: http://dx.doi.org/10.1016/j.foodchem.2008.06.047

[11] D. Cozzolino, G. Cowey, K. Lattey, P. Godden, W. Cynkar, R. Dambergs, L. Janik, and M. Gishen, "Relationship between wine scores and visible-near-infrared spectra of australian red wines," *Analytical and Bioanalytical Chemistry*, vol. 391, no. 3, pp. 975–981, 2008. [Online]. Available: http://dx.doi.org/10.1007/s00216-008-2071-3

[12] M. Ministério da Agricultura, Pecuária e Abastecimento, "Instrução normativa nº 13, de 29 de junho de 2005," [online], Jan. 2014, http://goo.gl/1DjmwL.

[13] X. Sun, H. J. Gong, F. Zhang, and K. J. Chen, "A digital image method for measuring and analyzing color characteristics of various color scores of beef," in *Image and Signal Processing, 2009. CISP '09. 2nd International Congress on*, 2009, pp. 1–6.

[14] R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection." Morgan Kaufmann, 1995, pp. 1137–1143.

[15] W. Härdle and L. Simar, *Applied Multivariate Statistical Analysis*. Springer, 2012.

[16] M. Erbert, "Uso da análise discriminante regularizada (rda) no reconhecimento de padrões em imagens digitais hiperespectral de sensoriamento remoto," Master's thesis, Universidade Federal do Rio Grande do Sul - UFRGS, 2001.

[17] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Annals of Eugenics*, vol. 7, no. 7, pp. 179–188, 1936.

# Neural Filter Applied To Fissure Detection

Edson Cavalcanti Neto*, Paulo César Cortez*, Valberto E. R. da Silva Filho†,
Tarique S. Cavalcante* and Pedro P. Rebouças Filho†
*Departamento de Teleinformática
Universidade Federal do Ceará, Fortaleza, Ceará, Brazil
Email: edsoncavalcanti@hotmail.com, cortez@lesc.ufc.br, tarique@lesc.ufc.br
†Instituto Federal do Ceará, Fortaleza, Ceará, Brazil
Email: valbertoenoc@gmail.com and pedrosarf@ifce.edu.br

***Abstract*—Cancer is a class of diseases characterized by out-of-control cell growth. There are over 100 different types of cancer, and each is classified by the type of cell that is initially affected. Thus, the initial treatment for this disease is the lobectomy. To plan the lobectomy, it is important to possess precise information about fissures, that determine the limits of the lobes. In order to obtain a more effective pulmonary fissure segmentation, and to do that independently from other lung structures. This paper proposes to perform the pulmonary fissure detection in 3D CT chest scans by the use of Local Binary Patterns (LBP) and Artificial Neural Networks (ANN). We provide a system with enhancement and post-processing on the 3D image. In this paper the focus is on develop a supervised fissure enhancement filter and is evaluated on images from 5 CT exams from LOLA11. The results of the proposed algorithm is compared with two others papers. The results from the MLP classifying the dataset are the following values coefficient of accuracy = 94%, specificity = 94.05% and sensitivity = 83.18%.**

## I. Introduction

Cancer is currently the disease which has been more prominent in the academic community, leading the statistics of death worldwide with 8.2 million deaths in 2012 [1]

Cancer is a class of diseases characterized by out-of-control cell growth. There are over 100 different types of cancer, and each is classified by the type of cell that is initially affected. Cancer is not just one disease but many diseases. Most cancers are named for the organ or type of cell in which they start - for example, cancer that begins in the colon is called colon cancer; cancer that begins in melanocytes of the skin is called melanoma.

Lung cancer has been the most common cancer in the world. There are estimated to be 1.8 million new cases in 2012. It is the most common cause of death from cancer worldwide, estimated to be responsible for 1.59 million deaths. Of all the kinds of pulmonary tumors, bronchogenic carcinomas represents approximately 90% to 95% of them. Among these, adenocarcinoma, squamous cell carcinoma, small cell carcinoma, and large cell carcinoma are the most common. Thus, the initial treatment for this disease is the lobectomy [2], [3]

For the surgeon to plan the lobectomy correctly, it is of utmost importance for him to possess precise information about lung anatomy to perform an actual procedure. Typical lung anatomy is composed of five distinct regions called lobes. Fissures determine the limits of these lobes. The right lung contains three divisions, therefore has two fissures, while the left lung only two lobes and one fissure [4].

Usually, the lobe segmentation is obtained through fissure segmentation in Computerized Tomography images (CT). The automatic segmentation of the lobes is quite challenging, because the density of the fissures is very similar to the density of the other structures, such as small blood vessels. Besides that, some parts of the fissure may not be completely visible in the CT scan image because of the Partial Volume Effect (PVE), which makes fissures incomplete. To overcome these difficulties, several works on the literature propose to make use of additional anatomic information to help on the segmentation of pulmonary fissure.

Zhang [5] make use of an anatomical atlas of the lung to initiate lobe segmentation. They also use a fuzzy logic system to identify the final fissure. Pu [6] on the other hand make use of computational geometry for initial fissure segmentation, followed by a Laplacian smoothing filter and processing of the final Guassian image with statistics measures. Sapthagirivasan [7] used fissure adaptive sweep and wavelet transform based on complex 2D double tree to segment fissures. Lassen [8] make use of blood vessel segmentation and an enhancement filter using the hessian matrix eigenvalues to segment the fissures.

The work proposes to detect fissures of lungs using Local Binary Patterns (LBP), Artificial Neural Networks (ANN), Gaussian Filter, and Morphology.

This is paper is organized as follows: In section II with a brief of Local Binary Pattern(LBP) as well as Neural Network Multilayer Perceptron (MLP) and the methodology used on this paper. In Section III, a quantitative analysis of the results will be given and discussed, and finally in Section IV we draw conclusions.

## II. Materials and Methods

This section will formulate the Local Binary Pattern (LBP) used, as well as give a brief description of Multilayer Perceptron (MLP) and present the methodology on the tests and result extraction.

The images used in this work is part of the CT scan dataset available for academic purposes LOLA11 at http://lola11.com/.

### A. Local Binary Pattern

Local Binary Pattern is a very robust algorithm for texture classification in terms of gray-scale variations since the operator is invariant against any monotonic transformation of the grayscale. Furthermore, it is computationally simple and might be processed with a few operations in a small neighbourhood and a lookup table [9].

A simple analysis of each pixel in a small image neighborhood generates a binary code called LBP code. The variables $P$ and $R$ are used compute LBP which correspond to the total number of involved neighbours and the radius of the neighbourhood respectively. If the coordinate of central pixel are $(0; 0)$, then the coordinates of neighbour pixel are given by ( $-R\sin(2\pi p/P)$, $R\cos(2\pi p/P)$ ). The neighbours gray values that are not in the image grids can be estimated by interpolation.

Therefore, to generate the LBP code, the algorithm compares the pixel with its neighbours. The use of the center pixel as threshold for its neighbors automatically creates the binary pattern with the neighborhood pixels [10].

$$LBP_{P,R} = \sum_{p=0}^{P-1} s(g_p - g_c)2^P, s(x) = \begin{cases} 1, x \geq 0 \\ 0, x < 0 \end{cases} \quad (1)$$

where $g_c$ is the gray value of the central pixel, $g_p$ is the value of its neighbours [9].

After the identification of each pixels LBP pattern, we compute the $U$ value. The $U$ value of an LBP pattern is defined as the number of spatial transitions (bitwise 0/1 changes) in that pattern [10].

$$U(LPB_{P,R}) = \begin{array}{l} |s(g_{P-1} - g_c) - s(g_0 - g_c)| \\ + \sum_{p=1}^{P-1} |s(g_p - g_c) - s(g_{p-1} - g_c)| \end{array} \quad (2)$$

The uniform LBP patterns refer to the patterns which have limited transition (U $\leq$ 2) in the binary pattern [9]. In this case, we mapped from $LBP_{P,R}$ to $LBP^{u2}_{P,R}$, The superscript $u2$ means uniform patterns with (U $\leq$ 2), wich has $P \times (P-1) + 3$ distincts patterns that are uniform.

Finally, a histogram is built to represent the texture image.

$$\begin{array}{l} H(k) = \sum_{i=1}^{I} \sum_{j=1}^{J} f(LBP_{P,R}(i,j), k), k \in [0, K] \\ f(x,y) = \begin{cases} 1, x = y \\ 0, otherwise \end{cases} \end{array} \quad (3)$$

where $K$ is the maximal LBP pattern value, $I$ and $J$ are the image dimensions.

In this paper, we use $LBP^{u2}_{8,2}$ and to compute the pattern of the voxels for classification are considered sub-images of size $11 \times 11$.

### B. Neural Network Multilayer Perceptron

The MLP is a Single-layer Neural Networks (SLNN) organized in cascade and subdivided in input layer, one or more hidden layers and output layer [11], [12]. Figure 1 represents those subdivisions.

According to Man [13], SLNN do not represent linearly separable functions. This problem is solved by the use two or more neurons with adaptive weights. However, it is necessary to use a training algorithm to adjust weights in these layers [14], which is one that performs error back propagation to compute errors of hidden layers [12], [15].

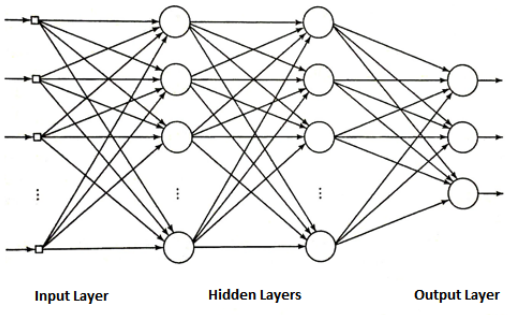



Figure 1. Representation of typical MLP network with two hidden layers

One output layer with nonlinear neurons and one or more intermediate layers composed of neurons that represent the network activation function is the composition of a MLP network [16], [17]. The signal is always forward propagated, layer-by-layer.

The MLP network has been useful in various problems through its supervised training back propagation algorithm). The error adjustment is the focus of this algorithm. Figure 1 illustrates a generic model of a multilayer network.

The following equations represents a MLP network of one hidden layer and $n$ neurons and an output layer of $q$ neurons, respectively.

$$w_i(t) = \begin{pmatrix} w_{i0}(t) \\ \vdots \\ w_{ij}(t) \end{pmatrix} \quad (4)$$

which $j$ represents the number of inputs, and $i$ the i-th neuron from the output layer. Similarly, the weights which connect the neurons from the hidden layer with the neurons from the output layers are defined by

$$m_k(t) = \begin{pmatrix} m_{k0}(t) \\ \vdots \\ m_{ki}(t) \end{pmatrix} \quad (5)$$

which $i$ represents the number of neurons on hidden layer, and $k$ the k-th neuron on the output layer.

The MLP is a supervised network, that is, for the adjustment of its weights happens by the comparison of the network output with the actual system output [17]. Accordingly, it is common to represent the training phase dividing it in two stages. The first one is called the forward signal, where the data flows from the input layer to the output layer, generating the output value of the network. While the other one called backward signal, and goes in the opposite direction to the forward signal using the error previously produced to adjust the intermediate layers weights [18], [19].

The first stage, of the direct sequence of data, involves the computation of the activation functions and the output of all the neurons. Starting from the input, passing through the hidden layer, to the output layer. To calculate the output of the hidden layers neurons, it is necessary to compute its output activation by the following process [12].

$$u_i(t) = \sum_{j=0}^{p} w_{ij}(t)x_j(t) = w_i^T(t)x(t) \tag{6}$$

$$z_i(t) = \phi_i(u_i(t)) = \phi_i(w_i^T(t)x(t)) \tag{7}$$

where $u_i(t)$ is the product of the input data $x(t)$ and their weights vector $w_i(t)$ of the hidden layer and $z_i(t)$ is the result of the hidden layer activation function $\phi_i$. The activation function $\phi_i$ used in this case was the logistic sigmoid, which is:

$$\phi_i(u_i(t)) = \frac{1}{1 + \exp[-u_i(t)]} \tag{8}$$

The next stage is the repetition of the equations 6 and 7 for the output layer

$$u_k(t) = \sum_{i=0}^{q} m_{ki}(t)z_i(t) = m_i^T(t)z(t) \tag{9}$$

$$y_k(t) = \phi_k(u_k(t)) = \phi_k(m_k^T(t)z(t)) \tag{10}$$

where $u_k(t)$ is the result from the product of the input data of output layer by their weights and $y_k(t)$ is the result from the activation function from the output layer.

In the following stage the data flow in the opposite direction, computing the last layer and the back propagated error to the hidden layers. Accordingly, the first step is to compute the error of each neuron in the output layer.

$$e_k(t) = d_k(t) - y_k(t) \tag{11}$$

where $d_k(t)$ is the actual value of each of the $k-th$ neuron of the output layer. To compute the hidden layer error, the backward propagation of the error play it role bringing the error from the output layer through the local gradient of the neuron, such that

$$\delta_k(t) = e_k(t)\phi'(u_k(t)) \tag{12}$$

where $e_k(t)$ is the output layer error and $\phi'_k(u_k(t))$ activation function derivative of the output layer.

The logistic sigmoid function derivative is

$$\phi'_k(u_k(t)) = \frac{d\phi_k(u_k(t))}{du_k(t)} = y_k(t)[1 - y_k(t)] \tag{13}$$

In possession of the computed error, it is possible to update the layers weight.

$$w_{ij}(t+1) = w_{ij}(t) + \alpha\delta_i(t)x_j(t) \tag{14}$$

where $\alpha$ is the learning rate, $\delta_i$ corresponds to the local gradient of i-th neuron.

For the output layer, the update equation for the weights is given by

$$m_{ki}(t+1) = m_{ki}(t) + \alpha\delta_k(t)z_i(t) \tag{15}$$

where $\alpha$ is the learning rate, $\delta_k$ corresponds to the local gradient of k-th neuron.

The filter applied to the fissure in this work makes use of the parameters defined by the algorithm of Section II-A as the input of the neural network. The configuration of the network is consisted of 59 features on the input, one hidden layer with 100 neurons and 2 neurons on the output layer, on to represent fissure and other to indicate that is not a fissure. For the training stage of the network, a set of 28000 features were used from 100 images of LOLA11 dataset and validated by the cross-validation technique, making use of 70% of data for training, 20% for testing and 10% to validate the training.

Therefore the database was created by the data from the LPB described on section II-A, calculated to each voxel considering subimage dimension. The used network was trained by 150 epochs with a learning rate of $\eta = 0,4$.

### C. Methodology

The developed algorithm in this work, see Figure 2, is divided in lung segmentation, 3D Gaussian filter, features extraction with LBP, fissure detection with MLP and in the end an post-processing whith component connected analysis and mathematical morphology.

The lung segmentation area was processed based on algorithm from [20]. The method is started with a analysis of low resolution images from CT to identify the seed and then is done a region growing with a fixed threshold.

Afterward the image is filtered with a 3D gaussian filter with the purpose to smooth the components of high frequency on the image. In this paper the gaussian filter applied used $\mu$=1mm and mask of $3 \times 3 \times 3$.

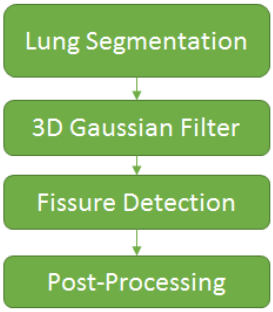



Figure 2. algorithm fluxogram.

On fissure detection step, the LBP features were calculated as described on section II-A, and the MLP used follows the procedure described on section II-B.

The Post-Processing was applied to remove the noise and the false positive generated by the MLP. To obtain the image after the post-processing was used the component conected analysis technique to remove structures with small volume with a threshold of 700 voxels. After the component analysis it was done an erosion in the iamge then the image was thresholded with the HU level between fissures densities. In the end, it was applied a closing on the image with a $3 \times 3 \times 3$ structuring element with the goal to connect small gaps on the fissures.

## III. Results and Discussion

The results presented on this section are based on voxels defined as fissure after the post-processing step. Therefore each voxel is classified as one of these categories:

- True Positive (TP) - Voxels from the database labeled as fissure and classified as fissure by the algorithm.
- True Negative (TN) - Voxels from the database labeled as not fissure and classified as not fissure by the algorithm.
- False Positive (FP) - Voxels from the database labeled as not fissure and classified as fissure by the algorithm
- False Negative (FN) - Voxels from the database labeled as fissure and classified as not fissure by the algorithm.

Based on these categories, it is possible to calculate the quantitative measures Accuracy (ACC), Specificity (SPC) e Sensitivity (S) to evaluate the algorithm performance. These metrics are given by:

$$ACC = \frac{TP + TN}{(TP + TN + FP + FN)} \tag{16}$$

$$SPC = \frac{TN}{(FP + TN)} \tag{17}$$

$$S = \frac{TP}{(TP + FN)} \tag{18}$$

The ACC measures the algorithm hit rate, in other words the degree of similarity between the result of the algorithm and the desired scenario. The SPC measures the ratio of false examples that were classified correctly, in other words the amount of voxels which are not fissure and were classified as one. The S quantifies the ratio of true examples that were classified correctly. In this case the amount of fissures that were classified as one.

To compute the results it was created one database with 49,807,360 examples with the LBP features calculated from the voxels of images selected from LOLA11. The examples from the database was classified by the MLP, which gave the following values as results: 27687 TP classifications, 2754038 FP, 47017035 TN and 8600 FN. Based on these values it was obtained the following rates: $ACC = 94\%$, $SPC = 94.05\%$ and $S = 83.18\%$

We highlight the excellent performance of the algorithm by observing the rates of $ACC$ and $SPC$. Based on $ACC$, we realized that the obtained result is close to the desired scenario, now the relation of $SPC$ we realized that the algorithm defines well the structures which are not fissure.

The sensitivity obtained the lowest rate, however it shows a satisfactory rate, once the classification is done at voxel level. Therefore, the fissure detected by the algorithm does not have voxels, it is possible to represent well the lung fissure.

Table I. Comparison between algorithms

| | Accuracy (%) |
|---|---|
| Proposed Algorithm | 94 |
| Kuhnigk [21] | 70 |
| Rikxoort [22] | 77 |

Comparing the results with Rikxoort [22] and Kuhnigk [21], the results from the proposed algorithm on this paper has a better result. On table I is shown the Comparison between algorithms, which Rikxoort [22] obtain an Accuracy of 71%, Kuhnigk [21] the rate of 87% and the proposed algorithm achieves the rate of 94%.

On Figure 3 are shown the examples of the obtained results. The Figures 3a and 3d represents the original images from CT, while on Figures 3b and 3e are shown the results based on MLP and LBP and on Figures 3c and 3f the final result is shown after the post-processing step. From the images on Figures 3c and 3f it is possible to see that the fissures is well caracterized by the algorithm.

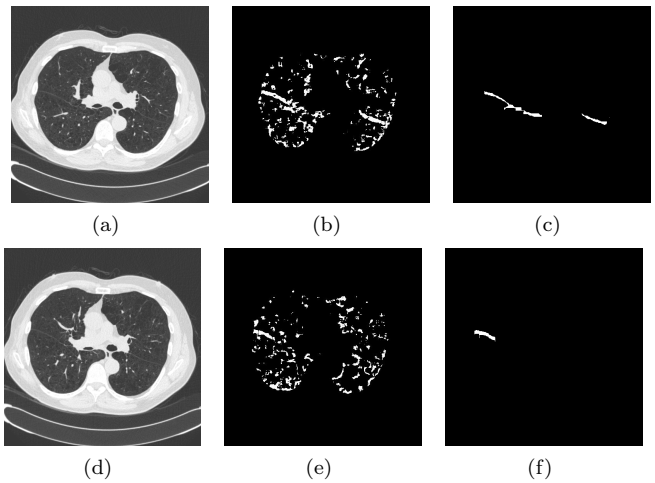


Figure 3. result from the algorithm: a) and d) Original Images, b) and e)LBP filtered Images, c) and f)Post-Processed Images.

## IV. Conclusion

This paper presents an algorithm for automatic detection of pulmonary fissures in thorax CT images. The main idea of this algorithm is the analysis of the texture patterns generated by LBP, the classification of these patterns, and the filtering of this classification based on connected component analysis and mathematical morphology.

This paper uses the LOLA11 dataset of thorax CT images to measure the effectiveness of the proposed method. The comparison of the classification results in TP, TN, FN, and FP, along with the estimation of similarity, sensitivity, and specificity coefficients represents the quantification method used.

Finally, the conclusion for the tested images, the automatic lung fissure detection, presented satisfactory performance with an 94.05% of accuracy, 94% specificity, and 83.18% of sensitivity. It is worth nothing that the sensibility coefficient does not have any major effects on the result. That is because the coefficient represents a value per each voxel, and even if some voxel are not in the set, it is possible to obtain a clear perception of the structure.

## Acknowledgment

The authors would like to thank National Council for Scientific and Technological Development (CNPq) for the support.

## References

[1] C. R. UK. (2014, March) Cancer research uk. [Online]. Available: http://www.cancerresearchuk.org/

[2] D. M. S. Almasslawi and E. Kabir, "Using non-rigid image registration and thin-plate spline warping for lung cancer progression assessment," in *IEEE International COnference on COmputer Science and Automation Engineering*, 2011.

[3] C. Wang, X. Li, L. Dai, G. Zhang, L. Gao, and Q. Zhao, "Analysis of lung cancer gender differences and age structure in the high prevalence areas of xiamen," in *IEEE Conference on International Spatial Data Mining and Geographical Knowledge Services*, 2011.

[4] S. N. Kumar. and V. Kavitha, "Automatic segmentation of lung lobes and fissures for surgical planning," in *International Conference on Emerging Trends in Electrical and Computer Technology*, 2011.

[5] L. Zhang, E. A. Hoffman, and J. M. Reinhardt, "Atlas-driven lung lobe segmentation in volumetric x-ray ct images," *IEEE Transactions on Medical Imaging*, vol. 25, pp. 1–15, 2006.

[6] J. Pu, J. K. Leader, B. Zheng, F. Knollmann, C. Fuhrman, F. C. Sciurba, and D. Gur, "A computational geometry approach to automated pulmonary fissure segmentation in ct examinations," *IEEE Transactions on Medical Imaging*, vol. 28, pp. 710–719, 2009.

[7] V. Sapthagirivasan and V. Mahadevan, "Denoising and fissure extraction in high resolution isotropic ct images using dual tree complex wavelet transform," in *International Conference on Software Technology and Engineering*, 2010.

[8] B. Lassen, E. M. van Rikxoort, M. Schimidt, S. Kerkstra, B. van Ginneken, and J.-M. Kuhnigk, "Automatic segmentation of the pulmonary lobes from chest ct scans based on fissures, vessels, and bronchi," *IEEE Transactions on Medical Imaging*, vol. 32, pp. 210–222, 2013.

[9] T. Ojala, M. Pietikäinen, and T. Mäenpää, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, pp. 971–987, 2002.

[10] Z. Guo, L. Zhang, and D. Zhang, "A completed modeling of local binary pattern operator for texture classification," in *IEEE Transactions on Image Processing*, 2010.

[11] D. W. Ruck, S. K. Rogers, M. Kabrisky, M. E. Oxley, and B. W. Suter, "The multilayer perceptron as an approximation to a bayes optimal discriminant function," *IEEE Transactions on Neural Networks*, vol. 1, 1990.

[12] S. O. Haykin, *Neural Networks and Learning Machines*. Pearson Prentice Hall, 2008.

[13] Z. Man, K. Lee, D. Wang, Z. Cao, and S. Khoo, "Robust single-hidden layer feedforward network-based pattern classifier," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, pp. 1974–1986, 2012.

[14] E. dos Santos Alves, J. B. O. S. Filho, R. M. Galliez, and A. Kritski, "Specialized mlp classifiers to support the isolation of patients suspected of pulmonary tuberculosis," in *2013 BRICS Congress on Computational Intelligence & 11th Brazilian Congress on Computational Intelligence*, 2013.

[15] A. Rahideh and M. H. Shaheed, "Cancer classification using clustering based gene selection and artificial neural networks," in *2nd International Conference on Control, Instrumentation and Automation*, 2011.

[16] T. V. Nguyen, J. C. Patra, and E.-L. Ang, "Blind image extraction from nonlinear mixtures using mlp-based ica," in *International Conference on Multimedia and Expo*, 2003.

[17] M. A. Arbib, *The handbook of Brain Theory and Neural Networks*. M.I.T Press, 2003.

[18] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.

[19] B. Widrow and R. Winter, "Neural nets for adaptative filtering and adaptative pattern recognition," *IEEE Computer*, vol. 21, pp. 25–39, 1988.

[20] J.-M. Kuhnigk, "Quantitative analysis of lung morphology and function in computed tomographic images," Ph.D. dissertation, 2008.

[21] J.-M. Kuhnigk, V. Dicken, S. Zidowitz, L. Bornemann, B. Kuemmerlen, S. Krass, H.-O. Peitgen, S. Yuval, H.-H. Jend, W. S. Rau, and T. Achenbach, "New tools for computer assistance in thoracic ct-part i: Functional analysis of lungs, lung lobes, and bronchopulmonary segments," *Radio Graphics*, vol. 25, pp. 525–536, 2005.

[22] E. M. van Rikxoort, B. de Hoop, S. van de Vorst, M. Prokop, and B. van Ginneken, "Automatic segmentation of pulmonary segments from volumetric chest ct scans," *IEEE Transactions on Medical Imaging*, vol. 28, pp. 621–630, 2009.

# True-Motion Estimation and Compensation with Multi Temporal Block Matching Search

André L. Martins, Evandro L.L. Rodrigues
Laboratório de Visão Computacional e Sistemas Microprocessados,
Departamento de Engenharia Elétrica, Escola de Engenharia de São Carlos
Universidade de São Paulo
Av. Trabalhador São-carlense, 400 – Cep 13566-590 - São Carlos, SP
E-mails: andremartins66@usp.br, evandro@sc.usp.br

***Abstract*—Improvements in de-interlacing algorithms are fundamental to explore image quality potential on modern TV screen technologies. This paper presents a true motion vector verification algorithm based on multi temporal block matching strategy, applied to video de-interlacing. During field scanning, a block of pixels is selected, then a block with the same reference coordinates is extracted but in the future field. A searching process runs in the previous field, looking for the most similar block; the best fit block and their coordinates are extracted from the previous field, and a new searching process is done but in the future field. If the searching process in the future field results back to the original block coordinates, the motion vector is validated; in other case, a correction over the motion vectors is done. Error calculation showed that the proposed algorithm presents image quality improvement, when compared to a classical motion compensation algorithm.**

***Palavras-chave—upscaling, de-interlacing, estimação de movimento, motion adaptive, motion compensation.***

## I. INTRODUÇÃO

As tecnologias de telas de TV que se baseiam em dispositivos com plasma, LCD ou LED, necessitam de sinal de vídeo com padrão de varredura progressiva. Isto se deve a detalhes inerentes à tecnologia destes tipos de tela. Apesar disso, conforme a norma ABNT NBR 15604 [1], por questões de compatibilidade com o parque instalado de aparelhos de TV do tipo Tubo de Raios Catódicos (TRC), as emissoras de TV digital ainda devem gerar, além do padrão progressivo, transmissão de vídeo com varredura por campos entrelaçados. No padrão entrelaçado, cada quadro (*frame*) de vídeo é dividido em dois conjuntos distintos de linhas, chamados "campos" (*fields*). As linhas ímpares de um quadro são organizadas em um campo chamado "campo ímpar", e as linhas pares são organizadas em outro campo, chamado "campo par". Para reproduzir a imagem na tela, os aparelhos de TV tipo TRC apresentam os campos de varredura entrelaçada na sequência em são transmitidos, ou seja, primeiro todas as linhas do campo ímpar, e depois todas as linhas do campo par. Este padrão de varredura foi criado para minimizar problemas de desconforto visual causados por cintilação da imagem (que tornou-se mais intensa com o advento de aparelhos com TRC de alto brilho) sem que fosse necessário aumentar a banda de transmissão. A solução trouxe melhorias importantes à época de sua invenção.

Mesmo para o formato de transmissão de vídeo com definição conhecida como "Full-HD" (1920x1080), a norma ABNT NBR 15604 [1] contempla apenas a varredura em campos entrelaçados. Deste modo, é necessário que os aparelhos de TV (ou dispositivos externos, como a *set-top box*) realizem o processo de conversão do sinal de vídeo do padrão entrelaçado para o padrão progressivo; tal processo é conhecido como *de-interlacing*.

O processo de *de-interlacing* não se resume simplesmente em reorganizar a sequência das linhas antes de apresentá-las na tela; este método, embora simples, gera distorções na imagem, conhecidos como "artefatos de vídeo". Grande parte das distorções geradas pelo processo de *de-interlacing* é causada por objetos que se movimentam na imagem. Diversos algoritmos foram desenvolvidos na tentativa de corrigir estas distorções, mas ainda há muito a ser explorado.

Em muitos casos, é necessário aplicar processamentos semelhantes aos utilizados nos algoritmos de *de-interlacing* na adequação de formatos de vídeo, para apresentação em telas de maior resolução, com a menor geração de artefatos possível. Tal adequação é conhecida como *video upscaling*. Por sua complexidade, certos algoritmos de *video upscaling* são implementados em hardware (pelo menos parcialmente), como descrito por Ramadevi [2].

Durante os processos de *de-interlacing* e *upscaling*, é necessário estimar linhas inexistentes nos campos originais da varredura entrelaçada. É uma tentativa de reconstruir detalhes da imagem que não foram transmitidos devido à limitação da banda, ou que nem sequer foram capturadas pela câmera. Conforme Bellers [3], o processo de *de-interlacing* "...duplica a densidade vertical de amostragem".

Algoritmos mais avançados de *de-interlacing* são capazes de identificar regiões da imagem onde há movimento, e aplicar a chamada "compensação de movimento" (*motion compensation*); tais algoritmos requerem a determinação de vetores de movimento real (*true motion vectors*). Neste artigo é apresentado o desenvolvimento de um algoritmo de verificação de vetores de movimento pela estratégia conhecida por *block matching search*, que identifica movimento pela

busca de estruturas (macroblocos) equivalentes entre os campos. O método proposto se baseia no algoritmo de estimação de movimento proposto por De Haan [4], e tem como contribuição a verificação do vetor de movimento por meio da busca no sentido temporal futuro, após a busca no sentido temporal passado, utilizada para determinação do bloco de maior compatibilidade.

Na próxima seção, é apresentada uma classificação e uma descrição básica dos métodos de *de-interlacing* utilizados como base para o desenvolvimento do algoritmo proposto.

## II. MÉTODOS DE *DE-INTERLACING*

### A. *De-interlacing simples*

No método de *de-interlacing* simples, ocorre basicamente uma reordenação das linhas; as linhas do campo ímpar (1,3,5,…) são intercaladas com as linhas do campo par (2,4,6,…), formando um quadro progressivo completo (com as linhas na sequência numérica (1,2,3,4,5,6,…). Quando este quadro é apresentado numa tela de TV de plasma, LCD ou LED, surgem artefatos de vídeo nas regiões em que existe efeito de movimento de um campo para outro. Estas falhas surgem pelo fato de que os dois campos que geraram o quadro foram adquiridos em instantes diferentes, e esta diferença temporal é maior do que seria se a imagem fosse capturada em varredura progressiva e transmitida com o dobro da banda. Além da diferença temporal da captura dos campos, a sequência das linhas do quadro progressivo gerado não corresponde à ordem temporal em que elas foram capturadas pela câmera.

Existe, portanto, uma incompatibilidade temporal entre os campos, e se houver movimento de objetos (ou movimento global geral devido à movimentação da câmera) entre os instantes de aquisição dos dois campos, a reorganização das linhas gera um efeito de dupla imagem, ou "serrilhamento" de partes de objetos ou pessoas que se movem de um campo para outro. Este efeito é pior quanto maior a velocidade dos objetos, e é pouco perceptível em telas do tipo TRC, pela própria tecnologia em si, e também pelo fato da menor resolução deste tipo de tela não permitir uma comparação direta de qualidade imagem em todas as resoluções disponíveis nas outras tecnologias mais modernas.

### B. *De-interlacing pela média intracampo (média espacial) ou intercampos (média temporal)*

No caso de estimação de linhas pela média intracampo, as linhas inexistentes são reconstruídas com base nas linhas vizinhas do mesmo campo (vizinhança espacial), ou até entre as linhas vizinhas, mas no campo anterior e/ou posterior (vizinhança temporal). Em ambos os casos, ocorre borramento da imagem, pois este tipo de procedimento produz os efeitos de um filtro passa-baixas. Os artefatos de imagem gerados podem ser piores que o *de-interlacing* simples, pois o borramento ocorre apenas em metade das linhas, uma vez que a outra metade é composta por linhas originais do campo.

### C. *De-interlacing por estimação de movimento*

O conceito de estimação de movimento foi inicialmente desenvolvido para identificação de padrões de repetição em vídeo, como descrito por Mounts [5] e para estimação de deslocamento de um quadro para outro, como descrito por Netravali [6], mas com objetivo final de reduzir a banda de transmissão de imagens de TV. A ideia básica é transmitir apenas informações referentes a regiões da imagem onde há movimento, e o restante da imagem é construída simplesmente pela repetição dos pixels do campo anterior.

A utilização de estimação de movimento com o objetivo de tratar sinais de vídeo surgiu mais tarde, tendo como destaque o trabalho de Netravali [7], que desenvolveu e patenteou um método para realizar interpolação por estimação de movimento entre campos consecutivos.

Houve um salto na evolução nos algoritmos de *upscaling* e *de-interlacing* de vídeo desde o surgimento da tecnologia de *true-motion estimation*, desenvolvido para reduzir os erros na estimação de vetores de movimento. Muitos destes algoritmos estimadores de vetores de movimento são baseados em busca por blocos similares (*block matching search*), proposto por Vandendorpe [8]. Neste tipo de algoritmo, o campo atual é dividido em blocos de pixels; cada bloco é selecionado e os campos passado e/ou futuro são varridos (dentro de uma vizinhança predefinida) em busca de um bloco com máxima similaridade.

Os processos modernos de *de-interlacing* e *upscaling* requerem algoritmos de estimação de movimento como necessidade fundamental. Tais algoritmos utilizam estimação de movimento real para a produção de linhas inexistentes no sinal de vídeo original e que completam cada quadro de vídeo.

### D. *Motion adaptive e motion compensation*

O conceito de algoritmos *motion adaptive* se refere a processamentos que identificam áreas da imagem que possuem movimento e realizam uma interpolação espacial nestas regiões. Nas regiões sem movimento é feita interpolação temporal. Estes algoritmos tendem a borrar a imagem nas regiões de movimento.

Já os algoritmos ditos de *motion compensation* interpolam temporalmente as áreas com movimento, utilizando para isso os vetores de movimento. Para que isto seja possível, é necessária a determinação de *true motion vectors*, para identificar as coordenadas corretas de um objeto em movimento em cada campo. O processo de interpolação é mais controlado, e deve considerar as alterações de coordenadas do bloco equivalente de um campo para outro devido ao movimento detectado.

Existem diversos casos em que o desenvolvimento de algoritmos de *de-interlacing* baseados em *motion estimation* e *motion compensation* resultaram na criação de circuitos integrados dedicados, com o objetivo de aumentar seu desempenho. Um destes desenvolvimentos é descrito por De Haan [9].

Na próxima seção, são apresentados os detalhes do algoritmo proposto e a estratégia utilizada para verificação de desempenho.

## III. METODOLOGIA

### A. *O algoritmo base*

O método de compensação de movimento de De Haan [4] foi utilizado como base por diversos pesquisadores. Neste método, cada campo de vídeo é subdividido em macroblocos, de tamanho tal que garanta que os objetos da imagem sejam maiores que o tamanho do macrobloco. Na Fig. 1 é apresentado um campo hipotético, que foi subdividido em macroblocos de tamanho arbitrário.

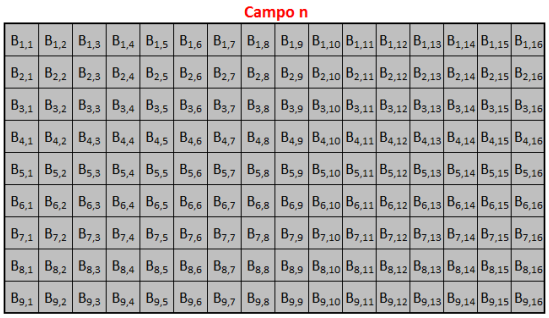



**Figura 1:** Um campo hipotético de vídeo dividido em macroblocos.

Supondo que o campo atual seja ímpar, as linhas pares terão que ser estimadas de alguma maneira, de modo que seja montado o quadro completo. Para isso, o campo é varrido bloco a bloco, e para cada bloco ocorre um processo de busca pelo bloco compatível no campo anterior (n-1).

São necessários dois campos para estimar as linhas faltantes (além do campo sendo varrido atualmente). O procedimento é feito da seguinte forma: a varredura dos blocos é feita no campo atual n; selecionam-se as coordenadas (x,y) do bloco, e é extraído o valor dos pixels do campo das mesmas coordenadas mas no campo futuro (n+1), conforme Fig. 2.

**Figura 2:** Extraindo valor dos pixels de um bloco no campo futuro (n+1).

Uma vez extraído o valor dos pixels do bloco de coordenadas (x,y) no campo futuro (n+1), retorna-se ao campo anterior (n-1), define-se uma vizinhança de busca (em número de pixels), e é realizada a busca pelo bloco com maior semelhança, conforme Fig. 3.

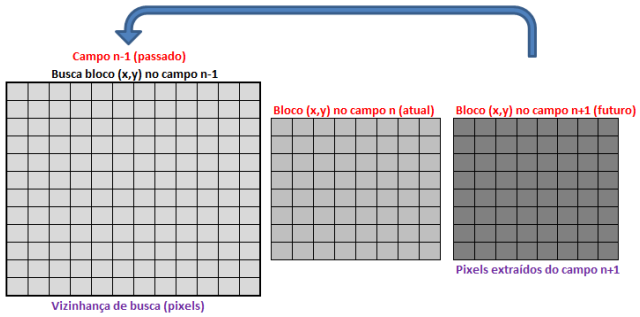



**Figura 3:** Busca do bloco (x,y) no campo anterior (n-1).

A vizinhança de busca comporta certo número de blocos candidatos, o que define o "conjunto de candidatos" (*candidate set*). Existem diversos métodos de busca, cada um deles com diferentes estratégias para minimizar tempo de processamento. Neste trabalho foi utilizada a busca exaustiva, e como critério de *matching* foi utilizado o erro quadrático médio (MSE), conforme descrito por Bellers [3]. O cálculo do MSE é apresentado em (1).

$$MSE = \sum_{\vec{x} \in B(\vec{X})} (F(\vec{x}, n+1) - F(\vec{x} - \vec{C}, n-1))^2 \quad (1)$$

Em que:

- *MSE = erro quadrático médio entre o bloco no campo n+1 e o bloco candidato no campo n-1;*
- $\vec{x}$ *= coordenadas do pixel no campo n;*
- *B = bloco selecionado no campo n;*
- $\vec{X}$ *= coordenadas do bloco no campo n;*
- *F = nível do pixel;*
- *n = número do campo atual;*
- $\vec{C}$ *= vetor de movimento do bloco candidato no campo n-1;*

Ao final da busca, identifica-se o bloco equivalente como aquele que apresenta o menor MSE, e utiliza-se as coordenadas deste bloco como o vetor de movimento de todos os pixels do bloco de coordenadas (x,y) atual da varredura. No caso em que o vetor de movimento aponta para as mesmas coordenadas do bloco original, considera-se que não houve movimento, e os pixels faltantes no campo n são estimados pela repetição dos pixels no campo n-1.

Conforme indicado por Bellers [3], quando é identificado movimento, realiza-se o processo de *motion compensation* por meio da estimação dos pixels das linhas faltantes pela mediana entre as duas linhas vizinhas do campo atual n (vizinhança espacial) e a linha correspondente no campo anterior (n-1), identificada pelas coordenadas do vetor de movimento. Este cálculo é apresentado em (2).

$$F_i(\vec{x}, n) = MED\{F(\vec{x} - \vec{u}_y, n), F(\vec{x} + \vec{u}_y, n), F(\vec{x} - \vec{C}, n-1)\} \quad (2)$$

Em que:

- $F_i$ *= valor do pixel estimado;*

- $\vec{x}$ = *coordenadas do pixel;*
- *n = número do campo atual;*
- *MED = cálculo da mediana;*
- *F = nível do pixel de coordenadas* $\vec{x}$ *;*
- $\pm\vec{u}_y$ = *vetor que aponta para o pixel na mesma coluna do pixel* $\vec{x}$ *, mas na linha abaixo/acima;*
- $\vec{C}$ = *vetor de movimento do bloco candidato no campo n-1;*

Este processo de estimação está representado na Fig. 4.

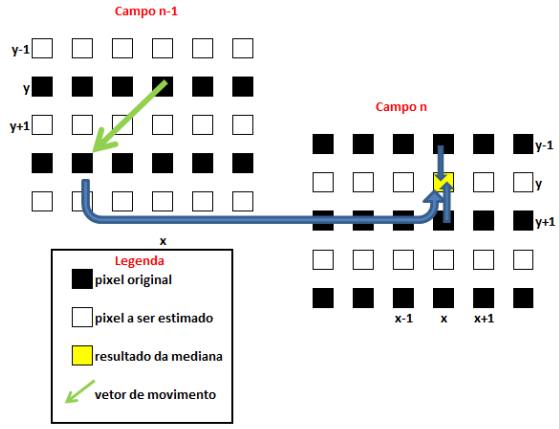



**Figura 4:** Estimação de um pixel pela mediana da vizinhança, considerando o vetor de movimento real.

## B. *O algoritmo proposto*

No algoritmo proposto, o bloco semelhante encontrado no campo anterior (n-1) é extraído normalmente, como realizado no método de De Haan [4], e então ocorre uma nova busca, mas agora no campo posterior (n+1), de onde foi extraído o bloco que foi buscado no campo anterior. Esta nova busca tem por objetivo confirmar o vetor de movimento, verificando se o vetor da nova busca aponta ou não para a posição original do bloco no campo posterior (x,y). Este processo é representado na Fig. 5.

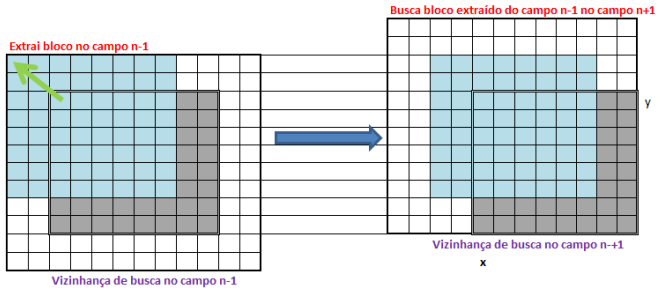



**Figura 5:** Busca do bloco extraído do campo anterior (n-1) no campo posterior (n+1) para confirmação do vetor de movimento.

Por diversas vezes, pode acontecer de o bloco extraído no campo anterior (n-1) ter maior similaridade com outro bloco que não seja o bloco das coordenadas de origem (x,y). Caso as coordenadas do bloco encontrado nesta nova busca sejam as mesmas coordenadas do bloco de origem, o vetor de movimento é considerado consistente, sendo então validado. A estimação das linhas faltantes do quadro atual é então realizada pelo cálculo da mediana das linhas vizinhas espacial e temporalmente. Na situação em que a busca pelo bloco no campo n+1 resulta em coordenadas diferentes das do bloco de origem, ocorre invasão parcial da região de outros macroblocos, conforme representado na Fig. 6.

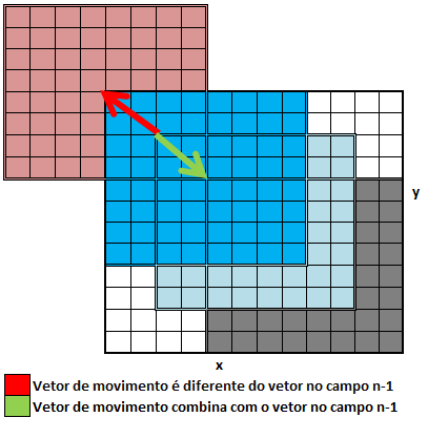



**Figura 6:** Busca pelo bloco no retorno ao campo n+1 pode resultar em vetor diferente do original; o bloco encontrado invade parcialmente outros blocos.

Caso esta situação de inconsistência ocorra, o vetor de movimento original não é confirmado, e os pixels do bloco similar da busca de retorno ao futuro têm seu vetor de movimento corrigido. Desta forma, a estimação das linhas faltantes do quadro atual é realizada pelo mesmo método da mediana, mas considerando o vetor de movimento corrigido.

Este processo ocorre de bloco em bloco, até que todo o campo de vídeo seja varrido e todas as linhas faltantes sejam estimadas.

## C. *Avaliação de desempenho do algoritmo proposto*

Para calcular os erros das linhas estimadas e fazer a comparação entre os algoritmos, foram gerados campos entrelaçados artificiais a partir de arquivos de vídeo. O procedimento adotado foi o seguinte: seleciona-se um arquivo de vídeo e eliminam-se as linhas ímpares dos quadros pares, e as linhas pares dos quadros ímpares, produzindo os campos artificiais. Estes campos foram submetidos aos algoritmos de estimação de movimento e *de-interlacing*, que geraram quadros desentrelaçados. Os quadros gerados pelos algoritmos foram comparados com os quadros originais do vídeo, determinando-se o erro pela soma das diferenças absolutas (SAD). O cálculo da SAD entre o quadro original e o quadro estimado é apresentado em (3).

$$SAD = \sum \left| F(\vec{x}, n) - Fi(\vec{x}, n) \right| \quad (3)$$

Em que:

- *SAD = soma das diferenças absolutas;*
- *F= nível do pixel no quadro original;*
- $\vec{x}$ = *coordenadas do pixel;*
- *n = número do quadro atual;*

- *Fi= nível do pixel no quadro estimado;*

O algoritmo proposto foi implementado na plataforma MatLab [10], e seu desempenho foi comparado aos seguintes métodos: *de-interlacing* simples, média espacial, média temporal e algoritmo de compensação de movimento de De Haan [4]. O tamanho do macrobloco foi definido como parâmetro, assim como a dimensão da vizinhança de busca. O algoritmo também permite definir vizinhanças de tamanhos diferentes na busca no campo anterior e na busca no campo posterior (confirmação). Os erros foram comparados em diversas combinações de tamanho de bloco, tamanho da vizinhança da busca no campo anterior e tamanho da vizinhança da busca no campo posterior.

Para todos os vídeos em que o algoritmo proposto foi aplicado, primeiramente foram gerados os quadros entrelaçados artificiais. Como ilustração do processo, são apresentados quadros de um dos vídeos utilizados, denominado "viplane.avi", arquivo nativo da plataforma Matlab; trata-se de um vídeo 80 quadros, 168 linhas e 360 colunas. A Fig. 7 mostra um quadro do vídeo original.

**Figura 7:** Quadro original do video "viplane.avi", nativo da plataforma MatLab.

O vídeo é processado e convertido, de modo que cada quadro original gera um único campo artificial. São utilizadas somente as linhas ímpares do quadro ímpar, formando um campo ímpar artificial com a metade das linhas originais. O mesmo acontece com os quadros pares, sendo que neste caso são extraídas as linhas pares, formando o campo par artificial. A Fig. 8 apresenta um quadro gerado pelo método de *de-interlacing* simples.

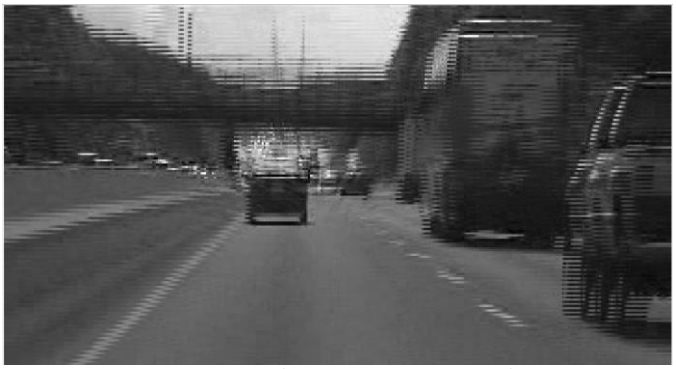


**Figura 8:** Quadro formado pelos campos artificias.

Podem-se notar os artefatos de vídeo gerados pelo *de-interlacing* simples, como efeitos de dupla imagem e "serrilhamento", em regiões de movimento de objetos.

Uma vez gerados os campos artificiais, aplica-se sobre eles o algoritmo de *de-interlacing*; no caso dos algoritmos com estimação de movimento, primeiramente são aplicados os algoritmos de estimação de movimento e geração dos vetores; a partir dos vetores de movimento obtidos, aplica-se o algoritmo de *de-interlacing*. Como a informação das linhas originais é conhecida (eliminadas para gerar os campos artificiais), pode-se determinar o erro exato da estimação das linhas faltantes, que foram geradas pelo algoritmo de *de-intelacing*.

Na próxima seção, são apresentados os resultados dos testes realizados e a comparação com o algoritmo base de De Haan [4].

## IV. RESULTADOS DE TESTES

Dentre os diversos testes realizados, são apresentados aqui os resultados da aplicação do algoritmo proposto em cinco arquivos de vídeo. Inicialmente, o algoritmo foi configurado para macroblocos de tamanho 8x8, conforme indicado por De Haan [4]; durante o desenvolvimento, variou-se o tamanho dos macroblocos de 3x3 a 16x16, e a vizinhança variou de 1 a 5 pixels maior que o tamanho dos macroblocos. A mudança destes parâmetros teve grande influência no tempo de processamento, mas pouca influência nos resultados em termos de qualidade de imagem, o que estimula uma melhor investigação posterior. A baixa sensibilidade do algoritmo ao tamanho dos macroblocos, desde que eles sejam menores que os objetos da imagem, foi evidenciada por De Haan [4]. A Fig. 9 mostra um quadro gerado pelo algoritmo proposto, aplicado ao vídeo "viplane.avi".

**Figura 9:** Quadro gerado pelo algoritmo proposto.

Pode-se observar na Fig. 9 que o algoritmo proposto foi capaz de corrigir diversos artefatos de vídeo, mas ainda restaram regiões com o efeito de "serrilhamento".

Para identificação das regiões da imagem onde o algoritmo proposto atua efetivamente, um quadro de vídeo extra foi gerado indicando os macroblocos onde ocorreu "não confirmação" dos vetores originais.

A Fig. 10 mostra as regiões da imagem nas quais o algoritmo proposto realizou correção dos vetores de movimento.

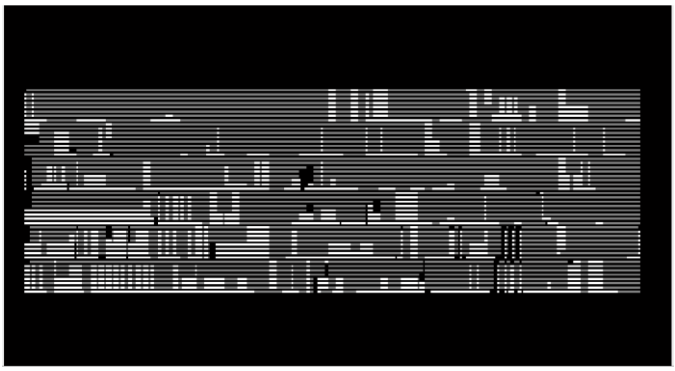


**Figura 10:** Regiões da imagem cujos vetores de movimento foram corrigidos.

Além do método de De Haan [4] e do método proposto, também foram implementados os métodos de *de-interlacing* simples, o método de estimação pela média espacial e o método de estimação pela média temporal (todos na plataforma Matlab).

Para medir o desempenho e comparar erros de estimação de linhas entre diversos algoritmos, foi realizado o cálculo da SAD, conforme (3), de cada quadro estimado em relação ao quadro do vídeo original. Foi criada uma variável que acumula este valor quadro a quadro, de modo que ao final do processamento do vídeo completo, este acumulador conterá o valor somado de diferenças de todos os quadros. A Tabela 1 mostra a SAD acumulada em relação ao vídeo original, para cinco vídeos diferentes, submetidos aos cinco métodos de *de-interlacing* implementados.

| Soma Acumulada de Diferenças Absolutas | | | | | | |
|---|---|---|---|---|---|---|
| Video | Entrelaçado | Média Espacial | Média Temporal | MC Haan | MC proposto | % MC's |
| viplane.avi | 70.685 | 65.800 | 57.684 | 44.855 | 43.836 | -2,27 |
| viptrain.avi | 492.600 | 1.203.100 | 380.680 | 393.290 | 391.720 | -0,40 |
| viplanedeparture.avi | 243.890 | 234.170 | 180.890 | 136.820 | 130.720 | -4,46 |
| viptraffic.avi | 35.916 | 54.118 | 29.855 | 21.285 | 20.591 | -3,26 |
| vipwarnings.avi | 479.700 | 472.490 | 396.510 | 314.090 | 308.730 | -1,71 |

**Tabela 1:** Erro comparativo dos métodos de *de-interlacing* implementados.

A coluna da direita da Tabela 1 apresenta o erro percentual do algoritmo proposto em relação ao algoritmo de De Haan [4].

Outra implementação também realizada e testada refere-se à varredura do campo e seleção dos macroblocos. Ao invés de varrer o campo com deslocamento igual à dimensão do macrobloco, realizou-se a varredura com deslocamentos iguais à metade do tamanho dos macroblocos. Esta estratégia não apresentou diminuição significativa dos erros, além de ter a desvantagem de aumentar o tempo de processamento.

## V. Conclusões

Ficou demonstrado que o algoritmo de verificação de vetores de movimento proposto apresentou ganhos de qualidade de imagem em relação ao método de De Haan [4], contribuindo para o avanço dos métodos de *de-interlacing* e *upscaling*.

Outras estratégias podem ser incorporados ao método proposto para reduzir erros dos vetores de movimento, diminuir o tempo de processamento do algoritmo de busca e algoritmo de *de-interlacing*, como o proposto por Chen [11]. Chen apresenta métodos para identificar regiões da imagem que, mesmo após a aplicação do algoritmo de *motion compensation*, ainda apresentam imperfeiçoes visíveis, como o "serrilhamento", descrito em seu trabalho como "*feathering effect"*.

Para trabalhos futuros, será realizada uma análise de tempo de processamento em relação a outros métodos. Também serão feitos testes com a inclusão de correções iterativas, realizando diversas buscas do macrobloco nos campos anterior e posterior, até que ocorra a convergência dos vetores de movimento. Também é necessário incluir proteções contra erros de estimação dos vetores de movimento, conforme citado por Bellers [3].

A influência do tamanho da janela de busca na qualidade de imagem, tanto no campo anterior quanto no campo posterior, também requer investigação mais aprofundada.

Também para trabalhos futuros é importante investigar o desempenho do algoritmo em vídeos com maior definição, e situações de oclusão.

## Referências


[1] Associação Brasileira de Normas Técnicas. Sobre a norma ABNT NBR 15604.Website da ABNT. http://www.abnt.org.br/imagens/Normalizacao_TV_Digital/ABNTNBR 15604_2007Vc_2008.pdf (2014).

[2] G. Ramadevi, T.Mallikarjuna, "Real time artifact-free image upscaling", Journal of Electronics and Communication Engineering, vol. 3, Issue 1, pp.12-19, Sep-Oct 2012.

[3] E.B. Bellers and G. De Haan, "*De-interlacing: A key technology for scan rate conversion*," Elsevier, Amsterdam, 2000.

[4] G. De Haan, P. W. A. C. Biezen, H. Huijgen, and O. A. Ojo, "Truemotion estimation with 3-D recursive search block matching," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 3, no. 5, pp. 368–379, Oct. 1993.

[5] F.W. Mounts, "A video encoding system with conditional picture-element replenishment", The Bell System Technical Journal, 48, pp. 2545-2554, Sept. 1969.

[6] A.N. Netravali and J.D. Robbins, "Motion compensated television coding: Part I", The Bell System Technical Journal, 58, pp. 631-670, Aug. 1978.

[7] Netravali et al, "Video signal interpolation using motion estimation", US Patent 4,383,272, May, 1983.

[8] L. Vandendorpe et al., "*Motion-compensated conversion from interlaced to progressive formats*," Signal Process.: Image Commun., vol. 6, pp. 193-211, 1994.

[9] G. De Haan, j. Kettenis, A. Loning and B.de Loore, "IC for motion-compensated 100Hz TV with natural-motion movie-mode", IEEE Transactions on Consumer Electronics, vol. 42, pp. 165-174, May,1996.

[10] The MathWorks, Inc. Sobre a plataforma MatLab. Mathworks web site. http://www.mathworks.com/products/matlab/. (2014).

[11] Y.-R. Chen and S.-C. Tai, "True motion compensated de interlacing algorithm," IEEE Trans. On Cir. & Syst. For Video Tech, vol. 19-, no. 10, pp. 1489-1498, Oct. 2009.

# Color Texture Classification under Varying Illumination

Tamiris Trevisan Negri
Department of Electrical Engineering and Computing
EESC/USP - University of Sao Paulo
Sao Carlos, SP - Brazil
Email: tamirisnegri@gmail.com

Adilson Gonzaga
Department of Electrical Engineering and Computing
EESC/USP - University of Sao Paulo
Sao Carlos, SP - Brazil
Email: agonzaga@sc.usp.br

***Abstract*—Color texture descriptors have gained a lot of interest in computer vision applications. Methods for grayscale texture analysis have been extended to color images. This paper presents a new descriptor for color texture analysis based on the Local Mapped Pattern (LMP) methodology called Color Local Mapped Pattern (CLMP). For each color channel, C-LMP considers the sum of the differences of each pixel of a given neighborhood to the central pixel as a local pattern that can be mapped to a histogram bin by using a mapping function. The histograms obtained from the color channels are concatenated in a color texture descriptor. The classification performance of the C-LMP is performed over Outex 14 texture database, considering three illumination sources. In our experiments, three different color spaces were considered: RGB, $L^*a^*b^*$ and HSV. Our results show that the C-LMP has better performance under illumination variances than the best results reported.**

***Keywords*: Color texture, Local Mapped Pattern, LBP, illumination invariance.**

## I. Introduction

Texture analysis in digital images plays an important role in many applications such as robotics [1], biometry [2], document processing [3], medicine [4], remote sensing [5] and visual inspection [6]. Methods of texture analysis have usually been applied to gray-scale images. However, new approaches using both color and textures information has been introduced in order to improve the characterization of color textures [7] [8] [9] [10].

The already established methods for grayscale texture analysis as Local Binary Pattern (LBP) and its variations [11] [12], and Gabor filter [13] can be extended to color images. The descriptor is applied to each color channel separately, and the feature vectors obtained from different channels are concatenated into a color texture descriptor [7].

Jain and Healey [14] used the Gabor filter outputs to compute unichrome and opponent features for color texture. Based on the opponent processes in the human visual system, opponent features capture the differences between filtered color channels. In [7], the LBP operator is extended to color texture in a similar manner. Maenppa and Pietikainen [7] also presented four different methods of combining separate color and texture measures. The color and texture features are combined on similarity measure level or classifier level.

Unichrome and opponent features are also considered in the Color Local Binary Pattern (CLBP) [15]. CLBP combines both color and grayscale features within a feature-level fusion framework for face recognition.

Color histograms were proposed by Swain and Ballard [16] as a feature descriptor. They showed that color histograms are robust to changes in view and to partial occlusion. However, they are not stable over changes in the color of scene illumination [7]. Funt and Finlayson [17] extended Swain's method to be illumination independent. The color constant color indexing method proposed by them is easy to implement and it works very well at different lighting conditions.

Variation of illumination is one of the challenges in the classification of textures [7]. Natural textures are generally subject to changes in illumination and it is wanted that the texture descriptor is able to cope with these changes.

Maenpaa e Pietikainen [7] compared the performance of several methods to color texture classification on two image databases: Vistex [18] and Outex [19]. In their experiments, five different color spaces were considered: RGB, rg chromaticity coordinates, $I_1I_2I_3$ [20], $L^*a^*b^*$ and HSV. In experiments with static illumination all methods presented good performance: the hit-rates reached $95.4\%$. However, the performance of methods was not so good when there were changes in illumination. The best score was $69.5\%$ and it was achieved by the LBP operator applied in grayscale texture. In their paper, the authors concluded that grayscale texture measures work better than color texture measures under varying illumination.

In a recent work, Ferraz et al. [21] presented a new descriptor called Local Mapped Pattern (LMP). The LMP methodology assumes that each gray-level distribution within an image neighborhood is a local pattern that can be mapped to a histogram bin using a mapping function. In their proposed application the LMP methodology had better performance under varying illumination than the LBP methodology.

In this paper, we propose a novel descriptor for color texture based on the Local Mapped Pattern (LMP) methodology, called Color Local Mapped Pattern (C-LMP). The performance of our proposed color texture descriptor was evaluated over the Outex $14$ database [19] and it was compared to the LBP performance and some other methods proposed in [7]. The Outex $14$ database provides textures imaged under three different illumination sources, allowing to evaluated if the C-LMP is robust to illumination variances. In our experiments, three different color spaces were considered: RGB, $L^*a^*b^*$ and HSV.

The remainder of this paper is divided into four sections. In Section II, we briefly describe LBP and LMP methods. Section III give details on the proposed approach C-LMP. The experimental evaluation and the results are presented in Section IV. Finally, we conclude the paper in Section V.

## II. Local Binary Pattern and Local Mapped Pattern

The LBP provides a way for describing local patterns in a texture as binary codes. The $3 \times 3$ pixels neighborhood shown in Fig. 1(a) is thresholded by the gray-level of the central pixel. The pixel values in the threshold neighborhood, as shown in Fig. 1(b), are multiplied by the respective weights in Fig. 1(c). The result, for this example, is shown in Fig. 1(d). Finally, the values of the eight values are summed to obtain the number of this texture code. For this example:

$$LBP_code = 1 + 0 + 0 + 8 + 0 + 32 + 0 + 128 = 169$$

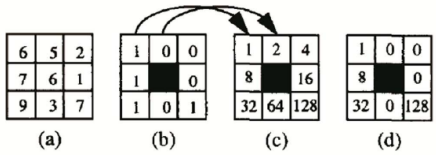



Fig. 1: The Local Binary Pattern [11] a) Image gray-level b) Thresholded neighborhood c) Weighting matrix d) Generated pattern.

Ojala et al. [12] proposed the local binary pattern (LBP) operator where the eight gray-level differences within a pixel neighborhood are operated by Eq. (1) and the result is recorded in an 8-bit number.

$$LBP = \sum_{i=1}^{8} S(g_0, g_i)2^{i-1}, S(g_0, g_i) = \begin{cases} 1, \text{if} & g_i \geq g_0 \\ 0, \text{if} & g_i < g_0 \end{cases} \quad (1)$$

The center pixel neighborhood is defined as:

| | | |
|---|---|---|
| $g_1$ | $g_2$ | $g_3$ |
| $g_4$ | $g_0$ | $g_5$ |
| $g_6$ | $g_7$ | $g_8$ |

That is, the LBP approach maps a pixel gray-level structure, within a neighborhood, to a histogram of 256 bins. This formulation is limited to a $3 \times 3$ pixels' neighborhood. Intending to expand the LBP method to operate in a general $W \times W$ neighborhood and mapping the resulting code to a histogram of B bins, Ferraz et al. [21] proposed to equate the LBP as in Eq. (2) where the function $f_g$ is the Heaviside function shown in Eq. (3), $B$ is the number of histogram bins and $P(i)$ is the weighting matrix shown in Fig. 1(c).

$$LBP = round\left(\frac{\sum_{i=1}^{W} f_g P(i)}{\sum_{i=1}^{W} P(i)}\right)(B-1) \quad (2)$$

$$f_g = H[g_i - g_0] = \begin{cases} 1, \text{if} & g_i - g_0 \geq 0 \\ 0, \text{if} & g_i - g_0 < 0 \end{cases} \quad (3)$$

By using the equation (3), the Heaviside function became the "mapping function", of a pixel gray-level structure in a neighborhood, to a specific code in the histogram and the method was called Local Mapped Pattern (LMP).

The basic LBP is not well suited for rotated textures due its weighting matrix based on the power of two [22]. New researches were published by the same authors, aimed to solve this problem [12] with the introduction of the $LBP^{riu2}_{P,R}$ (rotation invariant uniform). In this new version, the gray levels in the pixel neighborhood are sampled in a circular chain. Samples that are not located in the pixels center are interpolated. It is necessary to set the number of samples $(P)$ and the neighborhood radius $(R)$.

The LMP operator considers the sum of the differences of each gray-level of a given neighborhood to the central pixel as a local pattern that can be mapped to a histogram bin by using a mapping function. In the LMP methodology, Heaviside step function used in Eq. (2) can be replaced with any function. Furthermore, the weighting matrix can be chose according to the application and the $B$ parameter can be optimized.

For texture analysis, the authors suggest to use a smooth approximation to the Heaviside step function by a logistic curve, or a common sigmoid curve as in Eq. (4)

$$f_g = \frac{1}{1 + e^{\frac{-[g_i - g_0]}{\beta}}} \quad (4)$$

where $\beta$ is the curve slope. So that the descriptor is robust to changes of rotation, the proposed weighting matrix is:

$$P(i) = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad (5)$$

Thus, each pattern defined by a W x W neighborhood will be mapped to a histogram bin $h_b$ using Eq. (6). Fig. 2 shows a comparison between the LBP and LMP code generated for the same neighborhood of one texture sample, considering $B = 256$.

$$h_b = round\left(\frac{\sum_{i=1}^{W}\left(\frac{1}{1+e^{\frac{-[g_i - g_0]}{\beta}}}P(i)\right)}{\sum_{i=1}^{W} P(i)}\right)(B-1) \quad (6)$$

Sigmoid function, whose graph is "S-shaped" curve, appears in a great variety of contexts, such as the transfer functions of many neural networks. Their ubiquity is no accident: these curves are among the simplest nonlinear curves, striking a graceful balance between linear and nonlinear behavior.

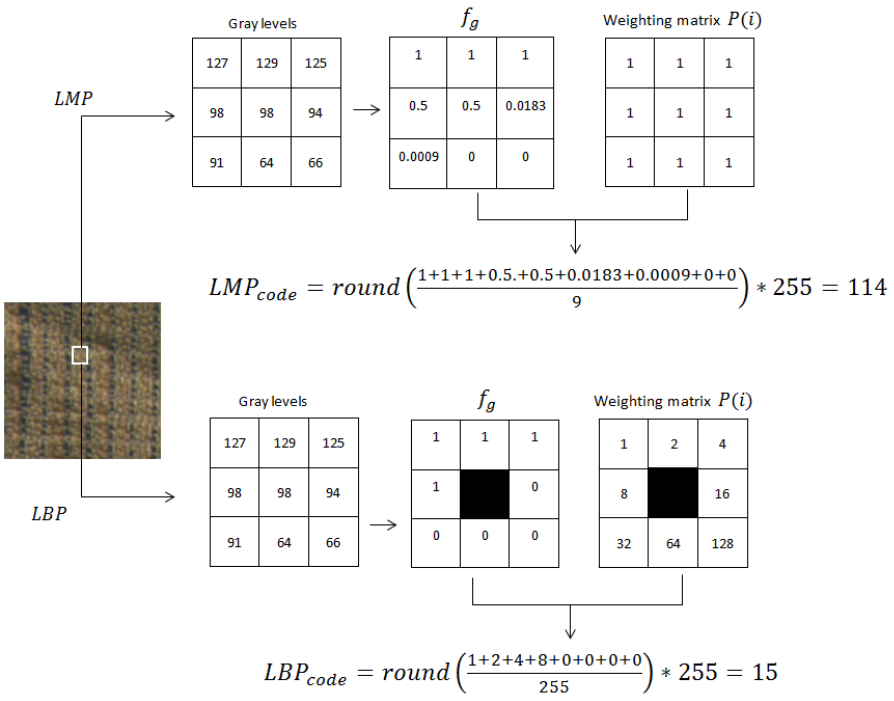



Fig. 2: Comparison between the LBP and LMP codes.

## III. Color Local Mapped Pattern

Based on the color texture descriptor proposed by Maenppa and Pietikainen [7], we investigated the use of the LMP descriptor in color textures. In this paper, we propose the Color Local Mapped Pattern (C-LMP) by applying the LMP method to each color channel.

As well as in the LMP, the sum of the differences of each pixel value of a given neighborhood to the central pixel is considered as a local pattern that can be mapped to a histogram bin by using a mapping function. The histograms obtained from the different color channels are concatenated, generating the color texture descriptor C-LMP. Fig. 3 shows an example of the C-LMP methodology applied in a RGB image.

## IV. Experimental Evaluation

### A. Experimental setup

To evaluate the performance of the C-LMP, we have performed an experiment on the Outex database, which is a public framework provided by the University of Oulu, Finland, composed of a bank of textures and test suites. In our experiment, the test suit Outex_TC_00014 (Outex 14) [7] was used. Outex 14 texture database contains 68 textures of size $746 \times 538$ pixels, each one imaged under three different illumination sources. The reference illumination source was a 2856 K incandescent CIE A light. The other illumination sources are 2300 K horizon sunlight and 4000 K fluorescent TL84. All images were imaged with a 100 dpi resolution and $0^\circ$ rotation. Fig. 4 shows an example of the each texture in the reference illumination. Fig. 5 shows the same texture imaged under three different illumination sources.

The 68 Outex textures in the reference illumination were used as training data. The textures were split into sub-images of $128 \times 128$ pixels. Since the size of the original images was $746 \times 538$ pixels, 20 sub-images were obtained of each texture, producing 1360 samples. Half of the samples from each texture were used in training, that is, 680 samples. To perform the division of the sub-images, a checkerboard pattern was used considering the upper left sub-image being the first training sample.

The 136 differently illuminated images were also split into 20 sub-images of $128 \times 128$ pixels each, producing 2720 samples. Again, a checkerboard pattern was used to split the set in two. Thus, there were 1360 samples in the test data.

### B. Texture Classification

For each texture sample, and considering each pixel in a neighborhood of 3x3 pixels, the C-LMP histogram bins were mapped by using the sigmoid function (Eq. 4) and the unit weighting matrix (Eq. 5).

The optimal value of $\beta$ (curve slope) was tunned by using a training set. The comparison between two samples is performed by employing a distance metric to evaluate the best fit between the two histograms. A query or "model" sample is considered to be correctly classified if it has the shortest distance from the same class of the training sample as illustrated in Figure 6.

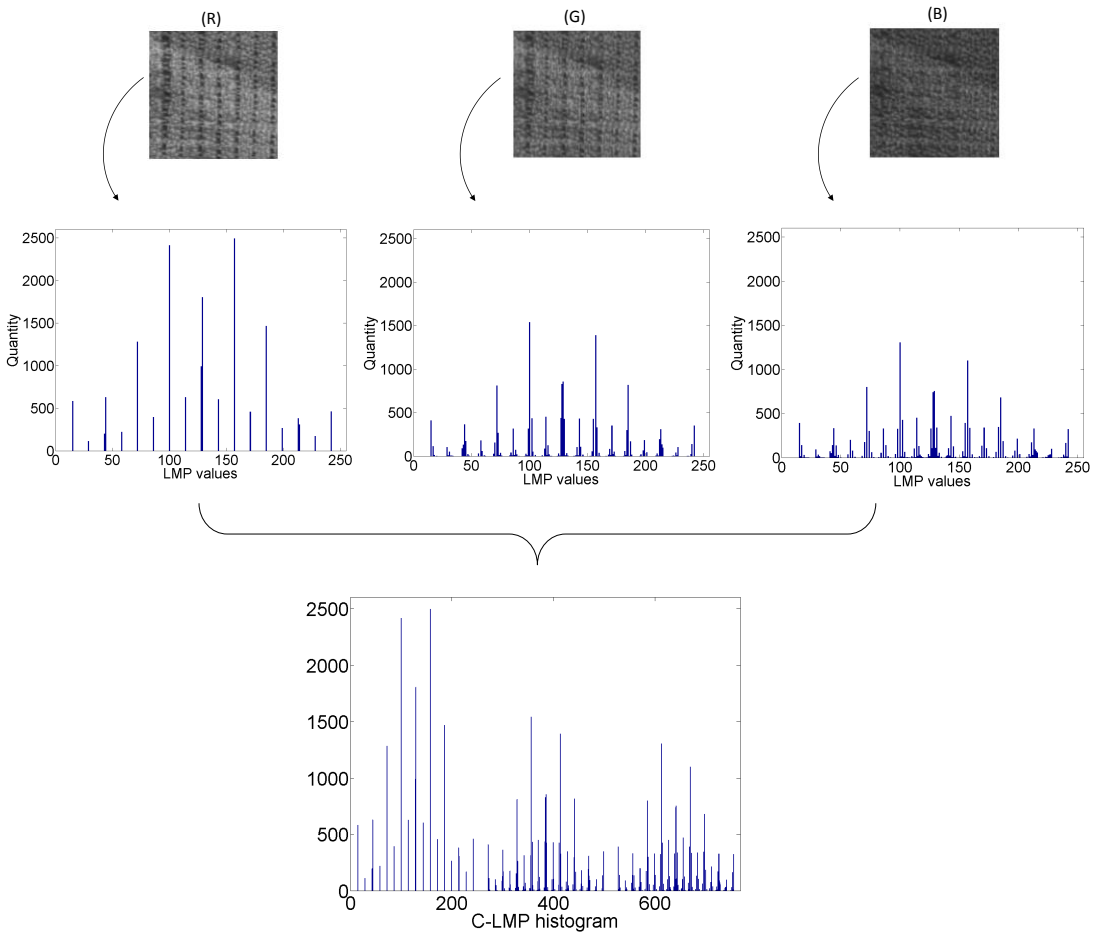



Fig. 3: The C-LMP methodology applied on RGB color space.

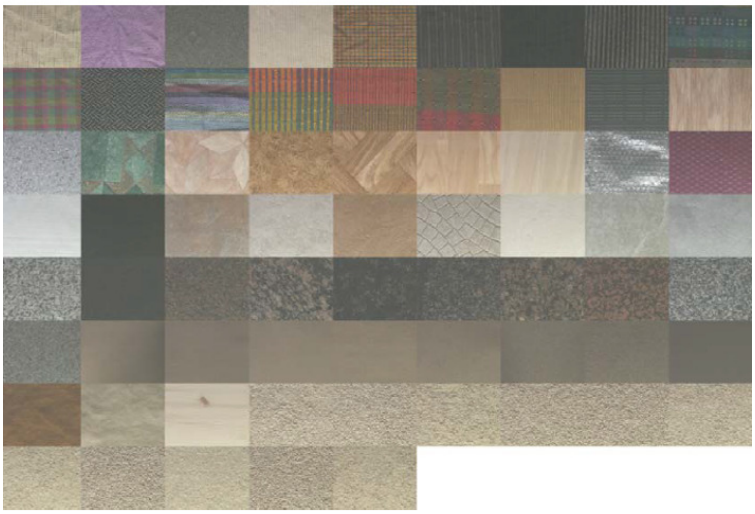
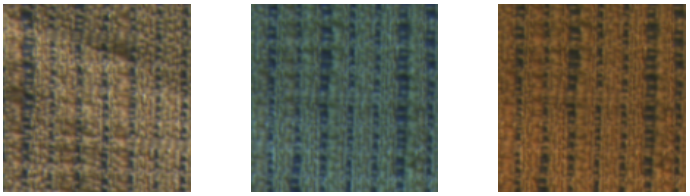


Fig. 4: Outex textures ([7]).



(a) Reference illumination.

(b) 2300 K horizon sunlight.

(c) 4000 K fluorescent TL84.

Fig. 5: The same texture imaged under three different illumination sources.

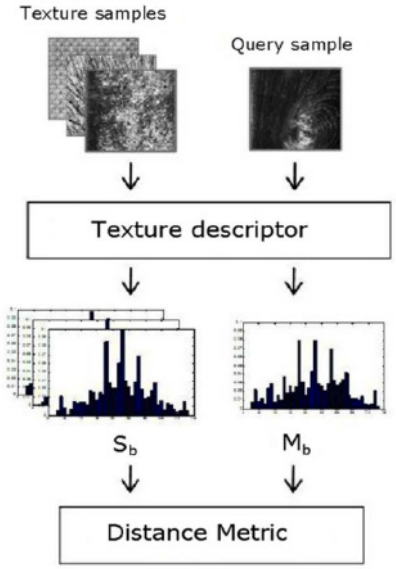



Fig. 6: Method for the texture classification [23].

The method performance was evaluated using the chi-square ($\chi^2$) distance metric given by

$$\chi^2(S, M) = \frac{1}{2} \sum_{b=1}^{B} \frac{(S_b - M_b)^2}{(S_b + M_b)} \quad (7)$$

where $S$ is the image sample, $M$ is the query image (or model image), $B$ is the number of bins of the compared histograms, $S_b$ is the number of sample values in each bin $b$, and $M_b$ is the number of model probabilities in each bin $b$.

Then, the confusion matrices are generated with true positives ($TP$), or the number of correctly classified samples, and false negatives ($FN$), or the wrongly classified samples, for all of the query samples. The hit-rate ($H$) or sensitivity can be calculated by

$$H = \frac{TP}{TP + FN}. \quad (8)$$

As C-LMP is a parametric model, we need to adjust the sigmoid curve slope ($\beta$) and the number of the histogram bins ($B$). The best values for the $\beta$ parameter are reached as soon as the sensitivity ($H$) is maximum using all the texture samples from the training set, with leave-one-out cross validation. In this work, we chose $B = 256$ without any optimization. The $\beta$ parameter was adjusted for each color channel separately on the three color spaces investigated: RGB, $L^*a^*b^*$ and HSV. The best values to the $\beta$ curve slope are presented in Table I.

TABLE I: Best values to $\beta$ curve slope

| Color Spaces | Channels | $\beta$ |
|---|---|---|
| RGB | R | 0.0990 |
| | G | 0.3500 |
| | B | 0.3890 |
| $L^*a^*b^*$ | $L^*$ | 0.0600 |
| | $a^*$ | 0.0149 |
| | $b^*$ | 0.0359 |
| HSV | H | 0.0005 |
| | S | 1.0014 |
| | V | 0.3700 |

The database used in our study as well as the methodology of classification is the same used by Maenppa and Pietikainen [7] in order to allow the comparison of the performance of descriptors. In [7], several experiments were performed. Color histograms [16] and the color constant color indexing method [17] were tested in various color spaces and the opponent color texture method [14] was applied using LBP operator and Gabor filters. LBP operator and Gabor filters [11] [12] [13] were also applied in color texture and gray-scale texture. In addition, methods combining separate color and texture features were tested. The best score was $69.5\%$ and it was reached by a variation of LBP operator applied in gray-scale texture.

Among all the methods proposed by Maenppa and Pietikainen applied to color texture analysis (color histograms, color constant color indexing, LBP operator and Gabor filters applied to color texture analysis, opponent color texture and methods combining separate color and texture features) in different color spaces (RGB, rg chromaticity coordinates, $I_1I_2I_3$ [20], $L^*a^*b^*$ and HSV) the best results were obtained by the $LBP(_{8,1}+^{u2}_{16,3}+^{u2}_{24,5})$ in the $L^*a^*b^*$ space ($67.8\%$ of sensitivity). Table II shows these results compared to the C-LMP performance.

TABLE II: Comparison of the C-LMP performance and the best results reported for Outex 14 [7].

| Method | Sensitivity |
|---|---|
| $LBP(_{8,1}+^{u2}_{16,3}+^{u2}_{24,5})$ gray-scale texture [7] | 69.50 |
| $LBP(_{8,1}+^{u2}_{16,3}+^{u2}_{24,5})L^*a^*b^*$ [7] | 67.80 |
| C-LMP (RGB) | 80.14 |
| C-LMP ($L^*a^*b^*$) | **83.16** |
| C-LMP (HSV) | 75.36 |

The C-LMP applied in any of the three color spaces investigated overcame all methods presented in [7]. The best result was achieved in the $L^*a^*b^*$ color space ($83.16\%$), increasing $19.65\%$ the hit-rate compared to $LBP(_{8,1}+^{u2}_{16,3}+^{u2}_{24,5})$ applied in the gray-scale texture.

The evidence that grayscale images are better for texture analysis than color textures in varying illumination presented in [7] is clearly not true. The results presented in this paper show that the C-LMP descriptor works well with color texture, in addition to being robust to illumination variances.

For future works, other experiments can be performed to investigate the use of other color spaces and test the recent variations of LMP descriptor presented in [21].

## V. Conclusion

In this paper, we proposed a novel method for color texture classification based on the Local Mapped Pattern (LMP) methodology. The proposed descriptor was evaluated in textures imaged under different illumination sources. The results were compared to the performance of other methodologies over the same image database. C-LMP approach presented better results than all methods compared, showing that the proposed methodology is robust to changes in illumination and it works well for color texture classification.

## Acknowledgment

The authors would like to thank the Sao Paulo State Fundation for Supporting Research (FAPESP) for their financial support of this research, process number $2011/18645-2$.

## References

[1] M. Blas, M. Agrawal, A. Sundaresan, and K. Konolige, "Fast color/texture segmentation for outdoor robots," in *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, Sept 2008, pp. 4078–4085.

[2] D. Huang, C. Shan, M. Ardabilian, Y. Wang, and L. Chen, "Local binary patterns and its application to facial image analysis: A survey," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 41, no. 6, pp. 765–781, Nov 2011.

[3] A. Busch, W. Boles, and S. Sridharan, "Texture for script identification," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 27, no. 11, pp. 1720–1732, Nov 2005.

[4] Q. Chang, B. Zhang, and R. Liu, "Texture analysis method for shape-based segmentation in medical image," in *Image and Signal Processing (CISP), 2011 4th International Congress on*, vol. 3, Oct 2011, pp. 1146–1149.

[5] F. Pla, G. Gracia, P. Garca-Sevilla, M. Mirmehdi, and X. Xie, "Multispectral texture characterisation for remote sensing image segmentation," in *Pattern Recognition and Image Analysis*, ser. Lecture Notes in Computer Science, H. Arajo, A. M. Mendona, A. J. Pinho, and M. I. Torres, Eds. Springer Berlin Heidelberg, 2009, vol. 5524, pp. 257–264.

[6] X. Xie, "A review of recent advances in surface defect detection using texture analysis techniques," *Electronic Letters on Computer Vision and Image Analysis*, vol. 7, no. 3, 2008. [Online]. Available: http://elcvia.cvc.uab.es/article/view/268

[7] T. Maenpaa and M. Pietikainen, "Classification with color and texture: jointly or separately?" *Pattern Recognition*, vol. 37, no. 8, pp. 1629 – 1640, 2004.

[8] C. Palm, "Color texture classification by integrative co-occurrence matrices," *Pattern Recognition*, vol. 37, no. 5, pp. 965 – 976, 2004.

[9] A. Drimbarean and P. Whelan, "Experiments in colour texture analysis," *Pattern Recognition Letters*, vol. 22, no. 10, pp. 1161 – 1167, 2001.

[10] S. Chindaro, K. Sirlantzis, and F. Deravi, "Texture classification system using colour space fusion," *Electronics Letters*, vol. 41, no. 10, pp. 589–590, May 2005.

[11] T. Ojala, M. Pietikainen, and D. Harwood, "A comparative study of texture measures with classification based on featured distributions," *Pattern Recognition*, vol. 29, no. 1, pp. 51 – 59, 1996.

[12] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 24, no. 7, pp. 971–987, Jul 2002.

[13] B. Manjunath and W. Ma, "Texture features for browsing and retrieval of image data," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 18, no. 8, pp. 837–842, Aug 1996.

[14] A. Jain and G. Healey, "A multiscale representation including opponent color features for texture recognition," *Image Processing, IEEE Transactions on*, vol. 7, no. 1, pp. 124–128, Jan 1998.

[15] J.-Y. Choi, Y.-M. Ro, and K. Plataniotis, "Color local texture features for color face recognition," *Image Processing, IEEE Transactions on*, vol. 21, no. 3, pp. 1366–1380, March 2012.

[16] M. J. Swain and D. H. Ballard, "Color indexing," *International Journal of Computer Vision*, vol. 7, pp. 11–32, 1991.

[17] B. V. Funt and G. D. Finlayson, "Color constant color indexing," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 17, no. 5, pp. 522–529, May 1995.

[18] "VisTex: Vision texture database," 1995. [Online]. Available: http://whitechapel.media.mit.edu/vismod/imagery/VisionTexture/vistex.html

[19] T. Ojala, T. Maenpaa, M. Pietikainen, J. Viertola, J. Kyllonen, and S. Huovinen, "Outex - new framework for empirical evaluation of texture analysis algorithms," in *Pattern Recognition, 2002. Proceedings. 16th International Conference on*, vol. 1, 2002, pp. 701–706 vol.1.

[20] Y.-I. Ohta, T. Kanade, and T. Sakai, "Color information for region segmentation," *Computer Graphics and Image Processing*, vol. 13, no. 3, pp. 222 – 241, 1980.

[21] C. T. Ferraz, O. P. Junior, and A. Gonzaga, "Feature description base on center-symmetric local mapped patterns," in *Symposium on Applied Computing, ACM-SAC 2014*, Gyeongju, Korea, March 24 - 28 2014, pp. 39–44.

[22] T. Ojala, M. Pietikainen, and D. Harwood, "Performance evaluation of texture measures with classification based on kullback discrimination of distributions," in *Pattern Recognition, 1994. Vol. 1 - Conference A: Computer Vision amp; Image Processing., Proceedings of the 12th IAPR International Conference on*, vol. 1, Oct 1994, pp. 582–585 vol.1.

[23] C. Chierici *et al.*, "A new approach for analyzing rotated textures," in *Workshop of Computer Vision (WVC 2013)*, 2013.

# *Image Enhancement Algorithms for Infrared Cameras*

*Lucas Rotava*
Laboratório de Visão Computacional, Departamento de Engenharia Elétrica e de Computação
EESC – USP
São Carlos, SP, Brazil
lucas_rotava@yahoo.com.br

*Evandro Luís Linhari Rodrigues*
Laboratório de Visão Computacional, Departamento de Engenharia Elétrica e de Computação
EESC – USP
São Carlos, SP, Brazil
evandro@sc.usp.br

***Infrared cameras are largely used in security, industrial and military areas. In many applications, it is important to detect and recognize objects in the viewed images. A parameterized filter that can enhance details in the infrared image can be very useful, helping the observer to detect and identify important objects in the image in different operation conditions, as the user can change the parameter's value to achieve better image quality. Another important characteristic in infrared images is contrast, as good contrast makes it easier to find objects in the image. This work presents a set of image processing algorithms used in an infrared camera, focusing on the parameterized filter and a piecewise algorithm to improve contrast, both implemented into an FPGA. All algorithms were implemented and runs in a camera with resolution of 640 x 512, and 30 frames per second. The results obtained with the algorithms are very interesting, as they successfully improve the image detail level.***

***Keywords—infrared imaging; FPGA; spatial filter; contrast enhancement***

## I. INTRODUÇÃO

Câmeras de infravermelho, ou câmeras térmicas, são dispositivos que detectam a radiação infravermelha de uma cena e a transforma em uma imagem visível. Qualquer corpo acima de 0K emite radiação infravermelha, assim, as câmeras térmicas podem detectar objetos em ambientes completamente escuros, sem luz visível. Devido à essa característica, essas câmeras são muito interessantes em diferentes áreas, como sistemas de segurança, de defesa e aplicações médicas. Além disso, a quantidade de radiação infravermelha emitida depende da temperatura do corpo: quanto mais quente o corpo, mais radiação é emitida. Isso pode ser utilizado em inspeções industriais, elétricas, em detecção de vazamento de gazes e até em testes de placas eletrônicas.

A importância das câmeras térmicas, principalmente na área de segurança e de defesa, implica no desenvolvimento de novos materiais e métodos voltados para a construção de câmeras e seus subsistemas. Os detectores de infravermelho estão evoluindo em rapidez e em resolução. Além disso, as câmeras estão cada vez menores e mais leves, portanto é necessário que o processador de imagens seja pequeno e seja rápido o suficiente para o processamento, como os ASIC (*Application-Specific Integrated Circuit*) e os FPGA (*Field-Programmable Gate Array*) podem ser.

Um FPGA é, basicamente, um *hardware* digital reconfigurável, que pode ser configurado a partir de um código escrito em linguagem de descrição de *hardware*. Por isso, pode ser configurada especificamente para uma aplicação, de forma rápida. Algoritmos paralelos, como os executados no processador de imagens térmicas, podem ser implementados eficientemente num FPGA. Em [1], alguns algoritmos de melhoramento de imagens, como ajuste de brilho, ajuste de contraste e equalização de histograma, são implementados em FPGA.

Já com relação aos algoritmos voltados para câmeras térmicas, em [2], o autor mostra que uma característica de grande importância é o contraste. Como a radiação infravermelha detectada é a soma das componentes refletida, transmitida e emitida, objetos muito quentes, ou muito frios, no plano de fundo da cena podem reduzir o contraste da imagem. Equalização de histograma e controle automático de ganho são algoritmos básicos que visam o melhoramento do contraste. Em [3] é apresentada uma comparação entre diferentes algoritmos de melhoramento de contraste.

Outro ponto importante em imagens térmicas é que os detectores geralmente possuem profundidade de pixel de 12 ou 14 bit, já os dispositivos de exibição possuem 8 bit. O algoritmo de compressão, ou redução, é o responsável por reduzir a profundidade dos pixels do detector para a profundidade dos *displays*. Caso existam pequenos detalhes na imagem detectada, é possível que durante a compressão eles sejam removidos. Então, um filtro de melhoramento de detalhes pode ser interessante para evitar essas perdas, já que ele aumenta a intensidade dos detalhes da imagem antes do algoritmo de compressão ser executado.

Nesse trabalho, o FPGA deve processar um vídeo com resolução de 640 x 512 em 30 fps. Além do FPGA, existem três memórias para armazenamento dos *frames* processados por cada grupo de algoritmos, assim, é possível que pelo menos três algoritmos sejam executados paralelamente no FPGA. Os algoritmos básicos de processamento de imagens termais serão brevemente apresentados, a fim de apresentar o sistema completo de processamento de imagens da câmera utilizada. Além dos algoritmos básicos, serão apresentados dois outros

algoritmos que foram desenvolvidos para atuar no realce dos detalhes e na redução da faixa dinâmica da imagem, a fim de manter detalhes importantes na imagem exibida ao usuário.

## II. ALGORITMOS BASE

O conjunto básico de processamento de imagens para câmeras térmicas é composto por correção de não uniformidade (*non-uniformity correction*, ou NUC), substituição de *bad-pixels* (*bad-pixel replacement*, ou BPR) e redução da faixa dinâmica, [4].

Os métodos mais comuns de NUC são os baseados em calibração. Nesses métodos é necessária a utilização de uma fonte uniforme de radiação, como um corpo negro, que é um equipamento que emite uma área uniforme de radiação, com intensidade controlada e conhecida. O corpo negro é posicionado, de forma a cobrir toda a visada da câmera, e é esquentado até uma temperatura conhecida. Como a cena observada pela câmera é uniforme, o sinal de saída do detector pode ser utilizado para calcular os parâmetros de correção para cada pixel da matriz. Se esse procedimento é realizado para dois valores de temperatura diferentes, o método é chamado de calibração de dois pontos e retorna dois parâmetros de correção, o ganho e o *offset*. A imagem corrigida é obtida multiplicando o pixel de entrada pelo ganho e somando com o valor do *offset*. Caso o corpo negro não esteja disponível é possível utilizar um *shutter*, que cobre todo o detector e tem a temperatura razoavelmente uniforme, porém desconhecida. Utilizando o *shutter*, a calibração é feita para apenas um valor de temperatura, então apenas o parâmetro *offset* é calculado, resultando numa correção mais simples. Essa é a correção de um ponto, e é necessária pois a não uniformidade do detector pode variar com o tempo de utilização da câmera e com a variação da temperatura ambiente. Esses métodos baseados em calibração apresentam parâmetros de correção bastante eficientes, em troca de um procedimento de calibração demorado e no qual a câmera deve ficar inoperante por um período de tempo, [5].

Existem outros algoritmos de NUC que não necessitam de equipamentos externos, como o corpo negro. Esses são os algoritmos baseados em cena e, como o nome diz, eles utilizam a cena observada, basicamente a variação dessa cena aliada a elementos estatística, para calcular os parâmetros de correção. A desvantagem é que os resultados obtidos nesse tipo de algoritmo são piores que os dos algoritmos baseados em calibração.

Mesmo após a correção de não uniformidade, alguns *pixels* ainda podem apresentar valores diferentes, quando observando uma cena uniforme. Esses *pixels* não respondem corretamente ao estímulo de entrada, portanto são considerados *bad-pixels*. Eles aparecem principalmente durante o processo de fabricação do detector e, para melhorar a qualidade da imagem observada, eles devem ser corrigidos, ou substituídos. A substituição dos *bad-pixels* é feita em duas etapas. Primeiro, utilizando o corpo negro, os *bad-pixels* são identificados, [6], e armazenados numa tabela. Então, durante a utilização da câmera, os *bad-pixels* são substituídos pelo processo de BPR, como em [7]. Existem, basicamente, três tipos de *bad-pixel*: aqueles que nunca mudam de valor; aqueles que são muito ruidosos, [6]; e aqueles que apresentam um valor muito deslocado da média dos *pixels*, [8]. Esses últimos são considerados defeituosos pois eles podem saturar antes dos outros *pixels*, diminuindo a sensibilidade da câmera.

Completando os algoritmos básicos de uma câmera térmica, a redução da faixa dinâmica é necessária pois a profundidade de *pixel* dos detectores de infravermelho é geralmente 12 ou 14 bits, já a dos *displays*, ou monitores, é de 8 bits. Uma forma fácil de fazer essa redução é mapear os valores de entrada para os valores na faixa de saída. A Fig. 1 é um exemplo desse mapeamento, no qual a entrada é de M bits e a saída é de N bits, [9].

É importante que esse algoritmo seja cuidadosamente desenvolvido para minimizar as perdas na imagem. Um bom começo é encontrar o histograma válido da imagem de entrada e, então, mapear somente a faixa de valores do histograma encontrado para a nova faixa dinâmica. Dessa forma, os valores de intensidade não utilizados para a cena observada são descartados, e somente a informação útil é passada para a imagem de saída.

Esse trabalho apresenta um algoritmo de redução linear por partes (em tradução livre de *piecewise linear*), um método simples que divide o histograma de entrada e faz o mapeamento de cada parte separadamente. Esse método é baseado no algoritmo de realce de contraste por partes, assim, além de reduzir a faixa dinâmica, esse algoritmo também faz o tratamento do contraste da imagem. Basicamente, utilizando uma lista de parâmetros, ele encontra alguns níveis no histograma de entrada e mapeia esses níveis, linearmente, para valores pré-determinados no histograma de saída.

Também é apresentado um filtro espacial de realce parametrizado, que é utilizado para realçar detalhes na imagem antes da redução de faixa dinâmica, dessa forma esses detalhes podem ser preservados, mesmo após a redução. O filtro é baseado no filtro de realce de bordas, com a adição do parâmetro que permite a seleção do nível de realce desejado. Além disso, foi adicionada uma componente de filtro da média, para aumentar a flexibilidade do filtro parametrizado. Na Fig. 2

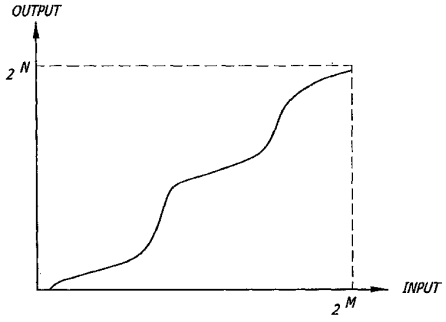



Fig. 1. Exemplo de mapeamento para algoritmo de redução de faixa dinâmica, de [9]

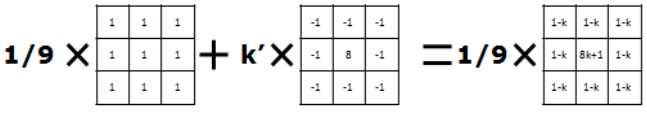



Fig. 2. Derivação do filtro parametrizado proposto

é representada a derivação do filtro proposto.

## III. IMPLEMENTAÇÃO EM FPGA

Este trabalho propõe a implementação desses algoritmos em FPGA, para que o vídeo capturado pelo detector seja processado, em tempo real, e exibido para o observador com baixa latência. Os processos de NUC e BPR serão brevemente apresentados pois o foco principal do trabalho são os algoritmos de relace de contraste e do filtro de realce de detalhes. O objetivo da implementação é conseguir realizar os processamentos paralelamente de forma rápida e com baixa latência, para que o tempo entre a cena real e o vídeo exibido pela câmera seja pequeno.

Para o algoritmo de NUC, tanto a calibração de dois pontos quanto a de um ponto são realizadas. A de dois pontos é feita em laboratório, anteriormente à utilização da câmera, e define os parâmetros de ganho da correção. Os parâmetros de *offset* são determinados durante a calibração de um ponto, que é feita por um processo no FPGA durante a inicialização da câmera. A calibração de um ponto é feita em cinco etapas, iniciando após o fechamento do *shutter*. Durante as três primeiras etapas é feito o cálculo da média entre 128 *frames*, para que o ruído aleatório da imagem seja reduzido. Esse valor é escolhido pois apresenta boa filtragem de ruído e também para que a divisão seja feita facilmente em *hardware*, apenas deslocando o vetor de bits para a direita 7 vezes. Na quarta etapa o *pixel* médio é calculado. Por fim, é na quinta etapa que o parâmetro *offset* de cada *pixel* é calculado.

Com os parâmetros calculados, o FPGA simplesmente faz uma multiplicação e uma adição para corrigir a imagem. O *pipeline* desse processo, como mostrado na Fig. 3, é formado pela leitura dos parâmetros, multiplicação do valor de entrada pelo ganho, soma com o *offset* e armazenamento do *pixel* corrigido.

O método de BPR implementado é o de simples substituição pelo valor de um *pixel* vizinho imediato. O índice relativo do vizinho que será utilizado é determinado e armazenado juntamente com o índice do *bad-pixel*, no mapa de *bad-pixels*. Então, o processo simplesmente faz a leitura de uma entrada no mapa de *bad-pixels*, lê o valor do *pixel* substituto e sobrescreve o *bad-pixel*.

Nesse trabalho, o método de redução de faixa dinâmica proposto é baseado no algoritmo de realce de contraste linear por partes, portanto, é necessário criar o histograma da imagem de entrada. Para melhorar o desempenho, serão utilizadas duas áreas de memória para armazenamento do histograma, assim, enquanto uma das áreas é atualizada com o histograma da imagem atual, o histograma formado da imagem anterior, na outra área de memória, é analisado. Esse processo é ilustrado na Fig. 4. As áreas de memória utilizadas são do tipo *on-chip*, internas ao FPGA.

Para criar o histograma, o FPGA utiliza o valor do *pixel*

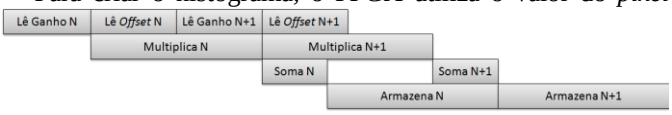



Fig. 3. *Pipeline* do algoritmo de NUC

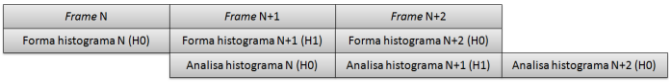



Fig. 4. Pipeline dos algoritmos de criação e análise do histograma

corrente como endereço na memória de histograma, faz a leitura do valor atual, contagem de *pixels*, incrementa esse valor e sobrescreve com o valor incrementado. Quando fizer isso para todos os *pixels* do *frame*, o histograma estará completo.

Para realizar a análise do histograma formado é necessário definir dois grupos de dados, os *thresholds* e os *targets*. Os valores no vetor de *thresholds* correspondem às porcentagens de *pixels* que vão delimitar as divisões no histograma, para que cada divisão seja transformada de forma diferente. Os valores no vetor de *targets* representam os valores de saída, para os quais os níveis encontrados no histograma de entrada, pela porcentagem de *pixels*, são mapeados. Para exemplificar, são definidos os dois vetores:

$$Threshold = [p_0\ p_1\ \dots\ p_{n-1}]$$

$$Target = [T_0\ T_1\ \dots\ T_{n-1}]$$

O processo vai então varrer o histograma, acumulando o valor total de *pixels*. O primeiro nível no qual o total de *pixels* acumulados é maior que $p_i\%$ do total de *pixels* da imagem, será mapeado para $T_i$. Dessa forma, ao definir N valores de *threshold*, o algoritmo irá encontrar N pontos no histograma, $[P_0\ P_1\ \dots\ P_{n-1}]$, e irá mapear esses pontos para os valores de *target* respectivos, resultando nas seguintes equações lineares:

$$\begin{cases} F_t(n) = T_0;\ F_o(n) < P_0 \\ F_t(n) = F_o(n) * g_i - o_i;\ P_i < F_o(n) < P_{i+1}, 0 \le i \le n-1 \\ F_t(n) = T_{n-1};\ F_o(n) \ge P_{n-1} \end{cases} \quad \text{Eq. 1}$$

$F_t(n)$ é o *pixel* n do *frame* de saída, $F_o(n)$ é o *pixel* n do *frame* de entrada. Como os pontos $P_i$ são mapeados para $T_i$, os valores de $g_i$ e $o_i$ são dados por:

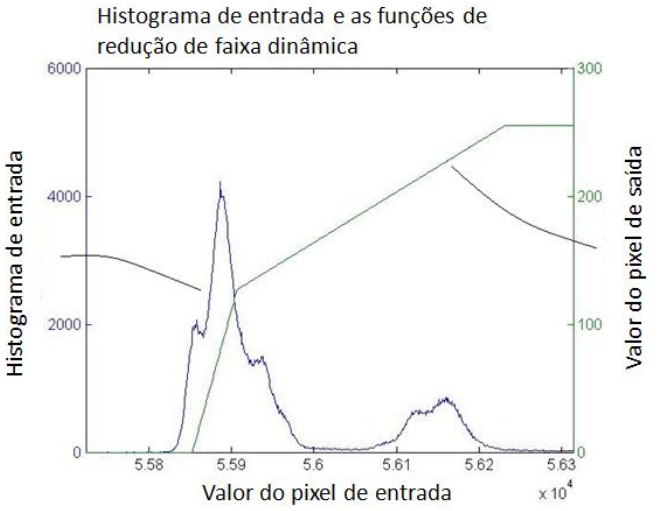



Fig. 5. Exemplo de um histograma de entrada e as funções de redução de faixa dinâmica com 3 pontos

$$g_i = \frac{T_{i+1} - T_i}{P_{i+1} - P_i} \quad \text{Eq. 2}$$

$$o_i = P_i * g_i \quad \text{Eq. 3}$$

Na Fig. 5 estão os gráficos de um histograma de entrada e as curvas para uma compressão de três pontos com:

$$Threshold = [5\ 50\ 95]$$

$$Target = [0\ 127\ 255]$$

Com esses parâmetros, os 50% de *pixels* que são mais escuros na imagem são mapeados para a metade inferior do histograma de saída, de 0 até 127, e os 50% de *pixels* mais claros da imagem são mapeados para a metade superior do histograma de saída, 128 até 255. Com mais parâmetros, a distribuição de *pixels* é acentuada.

Para minimizar as perdas ocasionadas durante a redução da faixa dinâmica, foi implementado o filtro parametrizado de realce de detalhes. É utilizado um filtro espacial $3x3$, portanto, é necessário formar a matriz de vizinhança do *pixel* filtrado. Como são nove *pixels* na matriz, seriam necessárias nove leituras da memória, o que pode consumir muito tempo. A fim de reduzir esse tempo, foi observado que as matrizes de vizinhança de *pixels* consecutivos possuem seis *pixels* em comum, portanto, basta armazenar temporariamente esses seis *pixels* e fazer a leitura apenas dos três restantes. Na Fig. 6, o sinal **neighborhood** é a matriz de vizinhança do *pixel* atual, o **last_pixel** armazena os *pixels* em comum para o próximo *pixel* a ser filtrado e o **last_line** contém os *pixels* em comum para o primeiro *pixel* da próxima linha.

Preliminarmente o filtro foi implementado em Matlab para a avaliação do seu comportamento. Verificado o funcionamento do filtro, foi feita a implementação em FPGA.

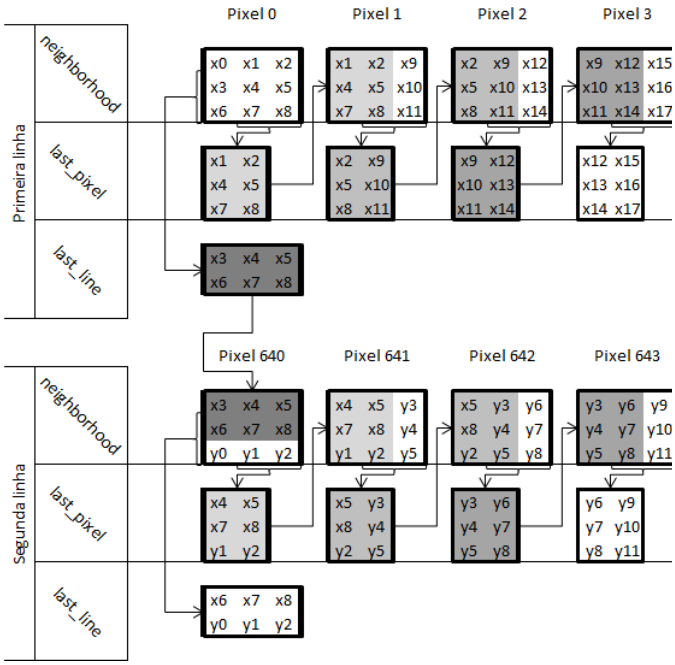



Fig. 6. Relação entre matrizes de vizinhança de *pixels* vizinhos

## IV. RESULTADOS

O algoritmo de redução de faixa dinâmica é composto por dois processos. O primeiro é o que faz a criação e a análise do histograma, determinando os valores das transformações lineares que são passados para o segundo processo, que faz a redução em si. Respectivamente, as Tabela I e Tabela II são os recursos utilizados pelas duas partes do algoritmo de redução, para redução com dois e quatro pontos, a fim de se comparar duas configurações diferentes. Todos eles foram otimizados para utilizar *clock* de 150 MHz. Na Fig. 7 são mostrados os resultados da redução de faixa dinâmica com 3, 7, 11 e 15 pontos, igualmente espaçados. Essa figura serve para ilustrar o efeito do algoritmo, que como é por partes, tende a distribuir os *pixels* no histograma de saída. Pelos histogramas de cada imagem, é possível ver que aumentando o número de pontos na redução, a distribuição dos *pixels* é mais bem feita.

A avaliação do filtro foi feita com uma imagem simples, que consiste de quatro quadrados escuros nos cantos e um claro

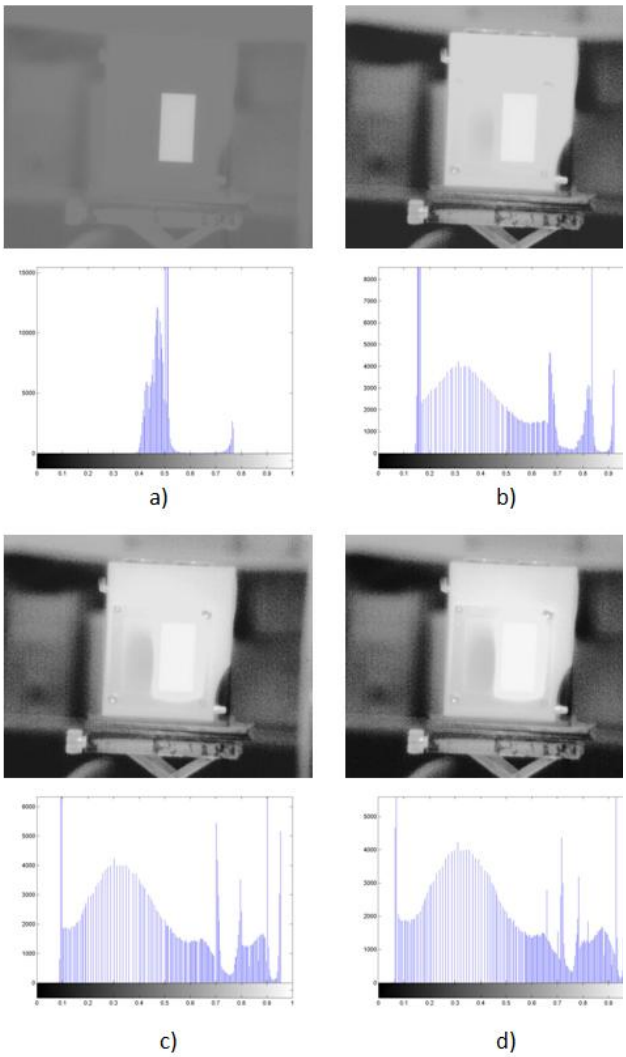



Fig. 7. Comparativo entre os resultados do algoritmo de redução de faixa dinâmica com: a) 3; b) 7; c) 11; e d) 15 pontos

TABELA I. RECURSOS UTILIZADOS PELO PROCESSO DE FORMAÇÃO E ANÁLISE DE HISTOGRAMA

| Compressão de 2 pontos | |
|---|---|
| Total de elementos lógicos: | 2252 |
| Total de bits de memória: | 1245184 |
| Blocos DSP 18x18: | 2 |
| **Compressão de 4 pontos** | |
| Total de elementos lógicos: | 4460 |
| Total de bits de memória: | 1245535 |
| Blocos DSP 18x18: | 4 |

TABELA II. RECURSOS UTILIZADOS PELO PROCESSO DE REDUÇÃO DE FAIXA DINÂMICA

| Compressão de 2 pontos | |
|---|---|
| Total de elementos lógicos: | 329 |
| Total de bits de memória: | 0 |
| Blocos DSP 18x18: | 1 |
| **Compressão de 4 pontos** | |
| Total de elementos lógicos: | 844 |
| Total de bits de memória: | 0 |
| Blocos DSP 18x18: | 3 |

no centro. Dentro dos quadrados foram desenhadas linhas, com níveis ligeiramente diferentes. O algoritmo de redução de faixa dinâmica utilizado possui apenas dois pontos, 1% e 99%, mapeados para 0 e 255, respectivamente. O parâmetro do filtro foi incrementado e os resultados armazenados. Na Fig. 8 são mostrados os resultados para quatro valores do parâmetro. Note que com o parâmetro colocado em 30, é possível ver todas as linhas internas aos quadrados.

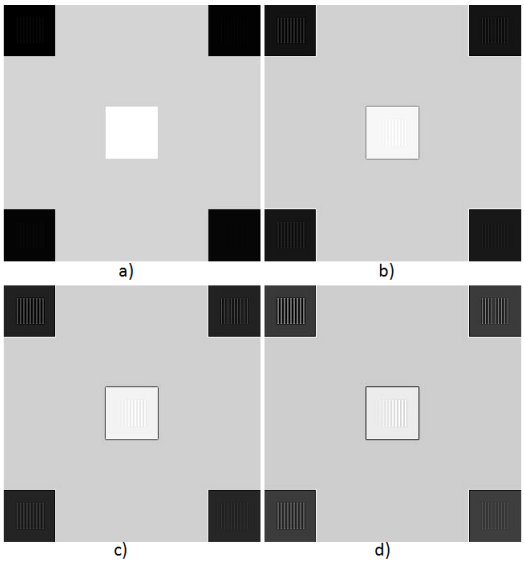



Fig. 8. Imagem de teste filtrada com o parâmetro configurado para: a) 1; b) 8; c) 15; e d) 30

O algoritmo implementado em FPGA também apresentou resultados interessantes. Nas Fig. 9 e Fig. 10 são mostradas possíveis aplicações reais do filtro. Na primeira é possível identificar os números na placa do carro somente após a filtragem da imagem. Na segunda, o caminhão não é facilmente reconhecido na imagem não filtrada.

Os recursos utilizados pelo algoritmo do filtro são listados na Tabela III, e esse algoritmo também funciona com *clock* de 150 MHz.

## V. CONCLUSÕES

Todos os algoritmos foram implementados no FPGA e possuem desempenho suficiente para processar o vídeo com resolução de 640 x 512 *pixels* rodando em 30 *frames* por segundo. O *clock* de *pixel* do detector é de 20 MHz, menor que o de 150 MHz utilizado nos algoritmos.O algoritmo de redução de faixa dinâmica, que também faz ajuste de contraste, é interessante pois possui implementação bastante simples e pode ser configurado de diferentes formas. Se somente dois pontos são definidos, ele trabalha como um simples algoritmo de AGC. Porém, se mais pontos forem definidos o algoritmo distribui os *pixels* da imagem pelo histograma de saída, semelhante à equalização de histograma, com a diferença que com o algoritmo proposto, é possível mapear, por exemplo, os

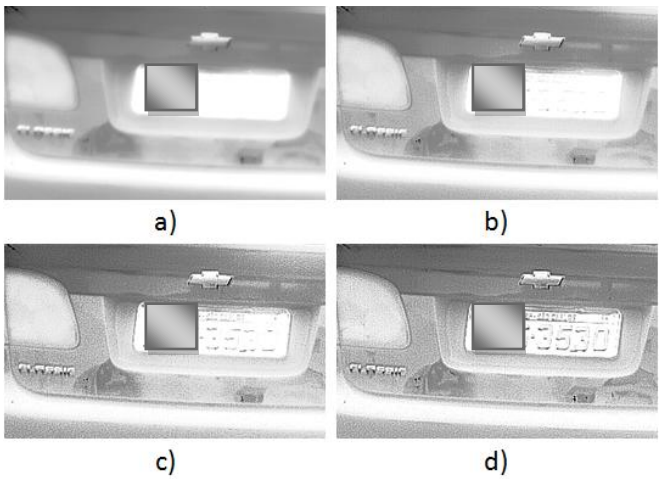



Fig. 9. Imagem real após filtragem com parâmetro configurado para: a) 1; b) 5; c) 10; e d) 15

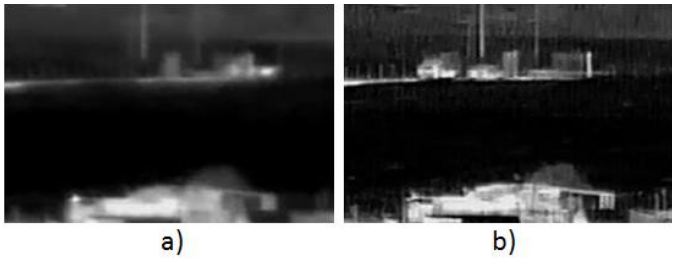



Fig. 10. Imagem real após filtragem com parâmetro configurado para: a) 1 e b) 10

TABELA III. RECURSOS UTILIZADOS PELO PROCESO DE FILTRO

| | |
|---|---|
| Total logic elements: | 3759 |
| Total memory bits: | 148 |
| Embedded DSP 18x18: | 9 |

10% dos *pixels* mais claros para a metade superior do histograma de saída, dando mais ênfase para esses *pixels* mais claros. Um estudo mais detalhado a respeito da definição da quantidade de pontos, os *thresholds*, e seus valores finais, os *targets*, ainda deve ser feito. É estudada ainda uma forma de se definir os valores dos *targets* automaticamente, baseada em informações do histograma de entrada.

Os resultados do filtro parametrizável foram promissores, tendo em vista que mais detalhes continuam visíveis na imagem quando o valor do parâmetro é aumentado. A técnica utilizada para realizar a convolução da imagem com o filtro espacial se mostrou bastante eficiente para o sistema proposto, já que aproveita os valores comuns de matrizes de vizinhanças de pixels vizinhos, reduzindo o tempo gasto para a filtragem. Dessa forma, o algoritmo de filtragem é mais rápido que a captura da imagem pelo detector.

Como visto no resumo dos recursos utilizados pelo algoritmo, é utilizado um multiplicador 18x18 por *pixel* da matriz de vizinhança e alguma quantidade de memória *on-chip* para o divisor. Os blocos DSP são recursos pouco utilizados no sistema completo e a quantidade de memória utilizada é uma parte muito pequena do total disponível. O número de elementos lógicos mostrados na tabela é o total de lógica utilizada tanto pelo filtro quanto pela interface do filtro com a memória de armazenamento do *frame*. Essa quantidade é menor que 4% do total disponível no FPGA, considerando a utilização da Altera Cyclone IV EP4CE115.

## REFERÊNCIAS


[1] Sowmya S., Paily R. FPGA Implementation Of Image Enhancement Algorithms. International Conference Communications and Signal Processing (ICCSP), pp. 584 – 588, Feb. 2011.

[2] Rogalski, A. Infrared detectors, status and trends. Progress in Quantum Electronics, vol. 27, issue 2-3, pp. 59-210, 2003.

[3] Zuo C., Chen Q., Liu N., Ren J., and Sui X. Display and detail enhancement for high-dynamic-range infrared images, *Opt. Eng*. 50(12), 127401 (July 11, 2011 October 12, 2011 October 21,2011 November 18, 2011).

[4] Holst, G. C. Testing and Evaluation of Infrared Imaging Systems. Florida: JCD; Bellingham: SPIE, 2008. ed. 3.

[5] Hayat, M. M.; Ratliff B. M.; Tyo J.S.; Agi K.; "Generalized Algebraic Algorithm for Scene-based Nonuniformity Correction," procspie, 5556,122-136, 2004

[6] López-Alonso J M, Alda J. Bad pixel identification by means of principal components analysis. Optical Engineering, 2002, 41(9): 2152–2157

[7] Fischer A D, Downes T V, Leather R. Median spectral-spatial bad pixel identification and replacement for hyperspectral SWIR sensors. In: Shen S S, Lewis P E, eds. Proceedings of SPIE, 2011, 6565: 65651E

[8] Cedip. BPR Procedure. Available in http://www.lmgc.univ-montp2.fr/~servex/B_Parc_materiel_et_Maintenance/Documents_machine/01014_Camera_IR_TITANIUM/Documentations/Logiciels/S9807_Cirrus_Software/DP003U-A_BPR_Procedure.pdf. Accessed in 18/03/2014

[9] Cheung, F. N. G. Data Translation System and Method. Patent US2007/013532, 2007

# Evaluation of Machine Learning and Bag of Visual Words Techniques for Pollen Grains Classification

Carolini Nascimento Martins Rodrigues, Ariadne Barbosa Gonçalves, Gercina Gonçalves da Silva, Hemerson Pistori
Universidade Catolica Dom Bosco – UCDB
Avenida Tamandare, 6000. Jardim Seminario. Campo Grande -MS
E-mail: {carolinirodrigues28, ariadne.gon, gercina.goncalves}@gmail.com; pistori@ucdb.br.

***Abstract— This work presents some new results regarding the automation of pollen grains classification using computer vision. A technique based on the Bag of Visual Words and Supervised Machine Learning algorithms is proposed and evaluated. A dataset of pollen grain images taken from 9 different pollen types was created and used in the experiments.***

***Keywords— Histograma de Palavras Visuais, Aprendizagem Supervisionada, Pólen.***

## I INTRODUÇÃO

A identificação da origem do mel e outros produtos apícolas, como o pólen e a própolis, pode ajudar no desenvolvimento da produtividade apícola, bem como impulsionar a economia da região. Na produção de produtos apícolas as abelhas operárias de *Apis mellifera* saem da colmeia e percorrem um raio de voo de 3 km [18] a procura de flores recursos florais. Nesse processo, grãos de pólen ficam aderidos à abelha, e são carregados para dentro da colmeia. Assim, ao analisar amostras de produtos apícolas é possível identificar os grãos de pólen presentes neles, pois o pólen é uma marca indelével da procedência botânica do produto, o que influência diretamente na sua denominação de origem.

A determinação da origem botânica de produtos apícolas é feita pela classificação visual dos tipos polínicos presentes no produto, sendo que ela pode ser muito morosa e cansativa, passível de erros devido ao desgaste físico e mental do especialista. Ressalta-se que a área que realiza o estudo dos grãos de pólen é a Palinologia. O bioma Cerrado possui uma flora muito rica, e consequentemente com grande variabilidade na morfologia dos polens. No entanto, alguns polens podem ser muito semelhantes o que dificulta a classificação e requer acurácia do especialista nesta tarefa. Além disso, existem semelhanças intraespecíficas dificultando a classificação tornando a automação da identificação polínica por visão computacional uma maneira precisa e rápida de classificação dos grãos de pólen.

A técnica de Histograma de Palavras Visuais ou Bag of Words (BOW) refere-se a um vetor de atributos que é extraído de determinada imagem ou conjunto de imagens. Para a extração do atributo é necessário o uso de algoritmos auxiliares como os de detecção de pontos de interesse SIFT ou SURF, e em conjunto com o BOW são utilizados algoritmos de aprendizagem automática. O objetivo do trabalho é a utilização da técnica BOW para classificação automática de grãos de pólen.

O trabalho está organizado em 6 seções. Na Seção II será apresentada a base teórica referente a trabalhos correlatos utilizando o BOW e de trabalhos com classificação e contagem de grãos de pólen. Na Seção III é descrita a técnica e as utilidades do BOW. A Seção IV apresenta a metodologia utilizada para o experimento. A Seção V apresenta os resultados e discussões sobre os experimentos, e as conclusões em conjunto com propostas para trabalhos futuros estão expostos na Seção VI.

## II TRABALHOS CORRELATOS

O problema da identificação de padrões visuais em coleções de imagens na histologia é abordado em [5]. O método proposto representa um conteúdo visual usando o BOW. O método consiste em um mecanismo de interpretação de dados associados às regiões das imagens correspondentes, para tanto dois conjuntos de imagens foram criados. O primeiro com imagens histológicas, e o segundo foi composto por imagens microscópicas usadas para diagnósticos.

Diferentes representações de palavras visuais e codebooks [10] foram testadas, e o desempenho foi avaliado qualitativamente e quantitativamente. O método explora a representação do BOW para realizar a mineração de dados visuais a partir de uma perspectiva mais ampla, em que o foco seja o conjunto de imagens e não as imagens individuais. Os resultados demostraram que o BOW é uma boa alternativa para representar o conteúdo visual de imagens histológicas.

A abordagem do BOW para detecção de melanoma maligno com base em imagens microscópicas de epiluminescência foi aplicada em [15]. Os algoritmos utilizados para classificação foram o SVM e o Naive Bayes. O conjunto de 100 imagens utilizadas foram de pele humana lesionada, representadas através de histogramas. Através deste conjunto de imagens os algoritmos foram testados e treinados, sendo 70 imagens com lesões provocadas por motivos adversos e 30 imagens com lesões provenientes do melanoma. Nas imagens foram aplicados filtros, como o Wavelet e o "Gabor-like". Os resultados indicam que o algoritmo SVM teve melhor desempenho em relação ao conjunto de imagens.

Um protótipo de um sistema para classificação automática de imagens de câncer colorretal captadas pela técnica NBI (Imagiologia de banda estreita) foi apresentado em [17]. O sistema consiste em uma representação BOW, com o uso do detector de ponto de interesse SIFT e o classificador

SVM. Os experimentos foram feitos em um conjunto de 907 imagens de tumores colorretais, sendo que o tamanho do tumor varia entre 100 x 300 a 900 x 800 e nesse conjunto há imagens que compartilham o mesmo tumor. As imagens foram divididas em 10 grupos, onde 9 foram usadas para treinamento e o restante para teste. A classificação obteve uma taxa de reconhecimento máximo de 94,1%.

No trabalho [7], é feita a classificação de três tipos diferentes de grãos de pólen pertencentes à família Urticaceae. Devido à reação alérgica no ser humano, as práticas clínicas mostram grande interesse na diferenciação desses grãos de pólen. Informações morfológicas são utilizadas por especialistas para distinguir diferentes tipos de grãos de pólen. O objetivo do trabalho de [7] foi de desenvolver um sistema automático de reconhecimento de padrões para classificar os grãos de pólen. Nas imagens foram aplicadas as operações de erosão e dilatação para obter o tamanho do grão de pólen. O trabalho concentra-se em um conjunto de formas baseadas em características de modo a conseguir a classificação. Um conjunto de trinta imagens foi usado com 10 imagens de cada tipo de grão de pólen. Uma vez que todos os atributos foram calculados, os dados foram passados para um classificador de distância mínima. A distância utilizada foi a de Mahalanobis para medir a similaridade entre um caso de consulta e as classes do conjunto de imagens. Se a imagem é corretamente classificada soma um ao contador. O processo repete-se N vezes, cada vez com exclusão de uma imagem diferente. Obteve-se uma percentagem de classificação correta de 97%.

A classificação automática de grãos de pólen foi proposta em [8] onde são utilizados dois métodos para classificação. O primeiro é o modelo de contorno ativo (snake), e o segundo é o Paradise Network (rede paraíso). O pólen tem um limite de espessura, ou de dois gumes quando visto através de um microscópio óptico, logo o modelo snake é usado para detectar esses dois gumes, se eles estiverem presentes.

Uma análise de grãos de pólen foi realizada por [4] com o objetivo de contar o número de pólen nas imagens. Foi aplicada a técnica de watershed para o processamento da imagem e em seguida o detector de partículas para realizar a contagem. O resultado foi similar à contagem humana. O trabalho concluiu a possibilidade de obter uma alternativa eficaz para a contagem de grão de pólen.

## III HISTOGRAMA DE PALAVRAS VISUAIS

A técnica de Histograma de Palavras Visuais (*Bag of Visual Words* – BOW) é composta de um histograma com números dos descritores de padrões encontrados em uma imagem. Para a classificação de imagens é aplicado o mesmo procedimento. Contudo a definição do vocabulário é um procedimento um pouco mais complexo. São usadas técnicas auxiliares denominadas SIFT e SURF [2]. Essas técnicas definem pontos de interesse na imagem (Figura 1). Os pontos são descritos por valores numéricos, os vetores. A quantidade de pontos em cada imagem é variável, o que impossibilita a utilização da técnica por algoritmos de aprendizagem baseados em vetores de atributos, por isso o BOW utiliza um algoritmo de agrupamento chamado K-médias (K-Means) em todos os pontos de interesse da imagem agrupados, e para cada grupo é calculada uma média. As médias ou número de grupos do K-Médias definem a quantidade de palavras visuais do vocabulário em que cada grupo representa uma palavra. Após esse processo as médias também são utilizadas na extração de atributos de cada imagem.

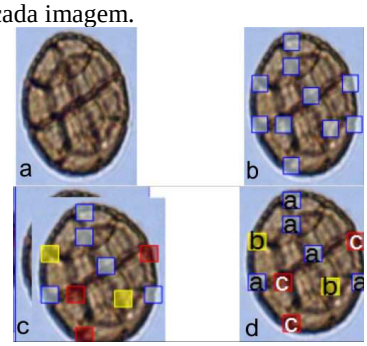



Figura 1: Atributos extraídos pelo BOW. a) Imagem original; b) detecção dos pontos de interesse pelo algoritmo Surf; c) Agrupamento dos vetores pelo K-means. d) Cada agrupamento é transformado em uma palavra pelo BOW.

O conjunto de treinamento é formado por imagens referentes ao problema, divididas em classes. Por exemplo, na classificação de pólen, as imagens são divididas de acordo com cada espécie. Os pontos de interesse das imagens são associados a cada palavra visual, de acordo com a menor distância entre o ponto analisado e a média de cada grupo. É criado um histograma com a quantidade de pontos de interesse de cada grupo. A frequência do histograma é usada como característica da imagem, que é associada a sua respectiva classe. Ao analisar uma nova imagem o processo é o mesmo da análise de pontos de interesse das imagens que são associados a cada palavra visual, de acordo com a menor distância entre o ponto analisado e a média de cada grupo.

## IV MATERIAIS E MÉTODOS

Neste trabalho foram analisados somente os méis florais, pois possuem diversidade de grãos de pólen. As amostras de mel necessitam de processamento para extração dos grãos de pólen que estão contidos no mel, para que seja possível a sua observação em microscópio, pois não é possível a visualização dos detalhes do grão de pólen a olho nu. Para o isolamento dos grãos de pólen nas amostras de mel foi utilizado o método de acetólise [11]. Para tanto é necessário pesar uma alíquota de 10 g de mel, que foi diluído em 10 ml de água destilada morna, e adicionar 50 ml de álcool etílico 95% em cada amostra. A mistura foi centrifugada a 2500 RPM para que os grãos de pólen fossem isolados, sendo que na parte superior do recipiente ficou um material líquido (sobrenadante) que foi descartado, e o sedimento polínico que é o material que contém a concentração de grãos de pólen ficou aderido no fundo do recipiente (decantação). O sedimento recebeu 2 ml de ácido acético para retirada da água excedente da amostra e então foram acrescentados 2 ml da solução acetolítica. Os tubos foram colocados em banho-maria, para que ocorresse a reação da solução acetolítica com o calor, pois este processo retira o conteúdo líquido do interior do grão de pólen, deixando apenas o esqueleto do pólen. Após este processo, o sobrenadante foi descartado e 2 ml de glicerina foram adicionadas à amostra, que ficou em repouso por uma hora, pois este processo reidrata o pólen fazendo com que retorne a sua forma original. Uma demonstração do uso do processo de acetólise pode ser visto na Figura 2, que apresenta imagem do

tipo *Protium*. As lâminas polínicas foram preparadas da seguinte maneira: um cubo de gelatina glicerinada com dimensão 3 mm por 3 mm foi cortado e posicionado no fundo do recipiente para que os grãos de pólen pudessem se aderir ao cubo. O cubo foi colocado em uma placa de vidro de formato retangular (lâmina e lamínula) para então ser analisado no microscópio. As imagens dos tipos polínicos foram capturadas através de um microscópio digital que possui uma Monotela LCD, da marca LCD Micro Bresse com aumento de até 40X. Foi feita a captura das imagens, de diferentes ângulos dos grãos de pólen.

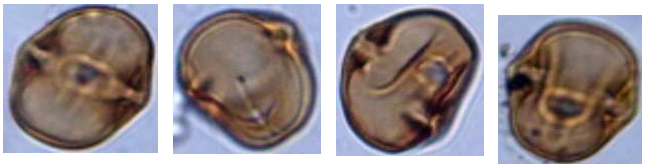


Figura 2. Imagens do tipo *Protium* obtidas através de acetólise.

Foram selecionados nove tipos polínicos (Figura 3) mais comuns em alguns méis do Estado de Mato Grosso do Sul e de cada tipo foram capturadas 30 imagens, totalizando um banco com 180 imagens, denominado de POLEN9E. As imagens contidas no banco de imagens foram utilizadas pela técnica BOW que identifica a variação de gradiente que ocorrem na imagem. O tamanho do dicionário utilizado pelo BOW representa o número de atributos e características que são extraídas do conjunto de imagens, para gerar o vetor de atributo Foram feitos testes exponenciais de 2 à 16384 no tamanho do dicionário, para saber o melhor tamanho de vocabulário. Experimentos foram realizados para verificar qual o melhor algoritmo de aprendizagem de máquina e o melhor tamanho de dicionário para melhor desempenho das informações extraídas do conjunto de imagens.

Foram verificados quatro algoritmos de aprendizagem, com o objetivo de obter um algoritmo ideal para este conjunto de imagens, foram testados: Máquina de Vetores de Suporte (SVM), C4.5, K Vizinhos Mais Próximos (KNN) e AdaBoost (Adaptive Boosting) ou estímulo adaptativo. Foram realizados alguns experimentos para testar o algoritmo KNN para escolher o melhor parâmetro para *k* a ser utilizado.

Os algoritmos foram comparados através das métricas: área sob a curva ROC (Area Under Roc), tempo de uso da CPU para teste (User CPU Time Testing), porcentagem de acerto (Percent Correct) e média ponderada da medida F (Weighted Average F Measure). Todos os experimentos foram feitos utilizando a validação cruzada com 10 dobras, presente no software WEKA.

Para teste de hipótese, utilizou-se o teste T-Student. A matriz de confusão foi utilizada para uma melhor visualização dos resultados dos experimentos, apresentando o resultado da classificação realizada pelo algoritmo. Cada linha representa uma espécie de pólen e cada coluna informa a quantidade de imagens que foram classificadas de acordo com a espécie. A diagonal da matriz demonstra a quantidade de imagens classificadas corretamente. Para a produção da matriz confusão, que apresenta os resultados obtidos pelo classificador com melhor desempenho, também foi utilizada a validação cruzada com 10 dobras presentes no programa WEKA.

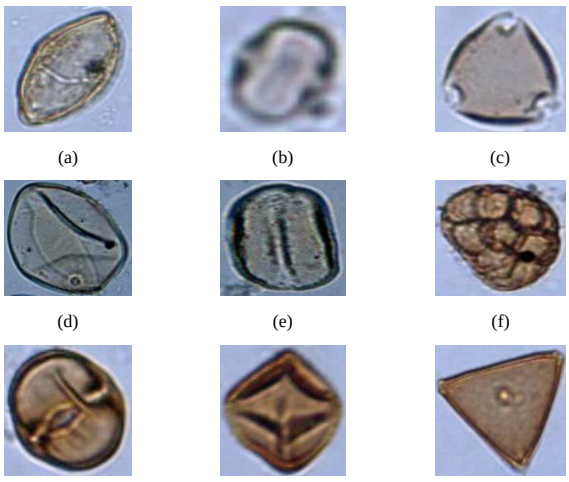



Figura 3. Imagens dos nove tipos polínicos presentes no banco de imagens. Arecaceae (a), *Cecropia* (b), *Myrcia* (c), Poaceae (d), *Syagrus* (e), *Anadenanthera* (f), *Protium* (g), *Schinus* (h), *Serjania* (i).

## V RESULTADOS E DISCUSSÕES

Com os experimentos realizados foi possível uma análise do desempenho dos algoritmos de aprendizagem para este tipo de problema. Nas tabelas 1 à 4 são demonstrados os resultados obtidos com o experimento. Na primeira coluna da esquerda estão os tamanhos de dicionário e na segunda coluna o primeiro algoritmo a ser comparado, e nas colunas seguintes da direita estão os demais algoritmos de aprendizagem. Os experimentos com o KNN resultaram em um $k = 15$, pois obteve um bom resultado em todas as comparações feitas.

A Tabela 1 apresenta os resultados da métrica Porcentagem de acerto com o algoritmo SVM sendo comparado com os demais algoritmos. A ordem dos algoritmos ordem interfere de forma direta na análise do desempenho dos algoritmos, pois o primeiro é comparado com os demais algoritmos que apresentam seus desempenhos melhores ou piores em relação ao primeiro. Foram obtidas outras tabelas como esta, porém com variação da ordem dos algoritmos, contudo mantendo esta métrica. Fazendo uma análise de forma geral de todos os resultados, é possível obter alguns dados que levam a visualizar para esta métrica um algoritmo ou conjunto de algoritmos mais eficiente.

A primeira análise foi em relação à métrica Porcentagem de acerto e feito esse estudo foi possível observar um índice maior de alguns algoritmos como os de melhores desempenhos. O classificador SVM apresentou melhor desempenho nos dicionários de tamanho 256 até 16384, em dicionários de tamanho inferior teve um resultado relativamente igual aos outros classificadores e com base nesses resultados é possível afirmar que o SVM tem melhor desempenho nesta métrica.

TABELA 1: Porcentagem de acerto

| Dic. | SVM | KNN | C4.5 | AdaBoost |
|---|---|---|---|---|
| Dic02 | 14.27066 | 14.27920 | 12.82051 | 12.76781 |
| Dic04 | 23.34900 | 23.10826 | 27.58262 | 15.82194 ○ |
| Dic08 | 23.65527 | 20.84046 | 29.99430 ○ | 20.30342 |
| Dic16 | 23.68234 | 23.51140 | 33.43305 ○ | 19.46439 |
| Dic32 | 29.12963 | 27.10399 | 40.73219 ○ | 24.20370 |
| Dic64 | 43.60541 | 37.81766 | 41.47436 | 23.60399 ○ |
| Dic128 | 50.79060 | 47.39744 | 45.28348 | 23.15527 ○ |
| Dic256 | 65.36182 | 50.13533 ○ | 44.77208 ○ | 23.30057 ○ |
| Dic512 | 64.71429 | 29.46746 ○ | 37.25952 ○ | 16.82540 ○ |
| Dic1024 | 68.93162 | 20.28632 ○ | 35.80199 ○ | 21.38319 ○ |
| Dic2048 | **74.03846** | 13.52564 ○ | 36.75926 ○ | 17.69231 ○ |
| Dic4096 | 71.24644 | 13.52564 ○ | 38.86325 ○ | 19.31197 ○ |
| Dic8192 | 64.82479 | 13.52564 ○ | 37.03989 ○ | 16.27635 ○ |
| Dic16384 | 59.35755 | 13.52564 ○ | 36.02991 ○ | 14.72365 ○ |

○, ○degradação ou melhoria estatisticamente significativa com nível de confiança de 95%.

Nesta seção será verificada a tabela com o SVM como primeiro algoritmo dos quatro. Na Tabela 2 estão os resultados obtidos com a métrica Área sob a curva ROC, em que é possível observar os dados em relação ao desempenho de cada algoritmo, e analisar a frequência dos algoritmos que apresentaram melhores desempenhos que os outros.

TABELA 2: Área sob a curva roc.

| Dic. | SVM | KNN | C4.5 | AdaBoost |
|---|---|---|---|---|
| Dic02 | 0.51232 | 0.54460 | 0.50884 | 0.57671 |
| Dic04 | 0.88855 | 0.81285 | 0.78349 | 0.52604 ○ |
| Dic08 | 0.72112 | 0.72522 | 0.91464 | 0.80386 |
| Dic16 | 0.81743 | 0.79911 | 0.90138 | 0.60014 ○ |
| Dic32 | 0.58646 | 0.53383 | 0.88178 ○ | 0.60597 |
| Dic64 | 0.91388 | 0.87948 | 0.89099 | 0.59530 ○ |
| Dic128 | 0.93886 | 0.91418 | 0.83312 ○ | 0.62248 ○ |
| Dic256 | 0.99512 | 0.99667 | 0.82828 ○ | 0.64099 ○ |
| Dic512 | 0.99132 | 0.95389 | 0.70877 ○ | 0.53602 ○ |
| Dic1024 | **0.99775** | 0.81863 ○ | 0.78275 ○ | 0.60642 ○ |
| Dic2048 | 0.99594 | 0.54729 ○ | 0.74811 ○ | 0.58431 ○ |
| Dic4096 | 0.99707 | 0.50000 ○ | 0.75321 ○ | 0.56020 ○ |
| Dic8192 | 0.98238 | 0.50000 ○ | 0.76420 ○ | 0.51347 ○ |
| Dic16384 | 0.98325 | 0.50000 ○ | 0.65010 ○ | 0.52882 ○ |

○, ○degradação ou melhoria estatisticamente significativa.

Os tamanhos de dicionário de 1024 a 16384 apresentam melhores desempenho para o classificador SVM, enquanto que para os outros tamanhos de dicionário este classificador obteve resultados semelhantes ao dos outros classificadores. A métrica Área sob a curva ROC teve o algoritmo SVM com uma maior incidência de vezes como melhor algoritmo.

O terceiro experimento (Tabela 3) foi feito utilizando a métrica Média Ponderada da medida F, sendo que para esta métrica o melhor resultado foi obtido com o algoritmo SVM. O algoritmo apresentou melhor desempenho nos dicionários de tamanho 256 á 16384, nos demais dicionários teve um desempenho relativamente parecido com os outros classificadores.

TABELA 3: Média Ponderada da medida F.

| Dic. | SVM | KNN | C4.5 | AdaBoost |
|---|---|---|---|---|
| Dic02 | 14.27066 | 14.27920 | 12.82051 | 12.76781 |
| Dic04 | 23.34900 | 23.10826 | 27.58262 | 15.82194 ○ |
| Dic08 | 23.65527 | 20.84046 | 29.99430 ○ | 20.30342 |
| Dic16 | 23.68234 | 23.51140 | 33.43305 ○ | 19.46439 |
| Dic32 | 29.12963 | 27.10399 | 40.73219 ○ | 24.20370 |
| Dic64 | 43.60541 | 37.81766 | 41.47436 | 23.60399 ○ |
| Dic128 | 50.79060 | 47.39744 | 45.28348 | 23.15527 ○ |
| Dic256 | 65.36182 | 50.13533 ○ | 44.77208 ○ | 23.30057 ○ |
| Dic512 | 64.71429 | 29.46746 ○ | 37.25952 ○ | 16.82540 ○ |
| Dic1024 | 68.93162 | 20.28632 ○ | 35.80199 ○ | 21.38319 ○ |
| Dic2048 | **74.03846** | 13.52564 ○ | 36.75926 ○ | 17.69231 ○ |
| Dic4096 | 71.24644 | 13.52564 ○ | 38.86325 ○ | 19.31197 ○ |
| Dic8192 | 64.82479 | 13.52564 ○ | 37.03989 ○ | 16.27635 ○ |
| Dic16384 | 59.35755 | 13.52564 ○ | 36.02991 ○ | 14.72365 ○ |

○, ○degradação ou melhoria estatisticamente significativa.

A Tabela 4 apresenta como primeiro algoritmo o C4.5, pois este apresentou melhor resultado com esta métrica. Com esta métrica obtivemos resultados parecidos entre alguns classificadores.

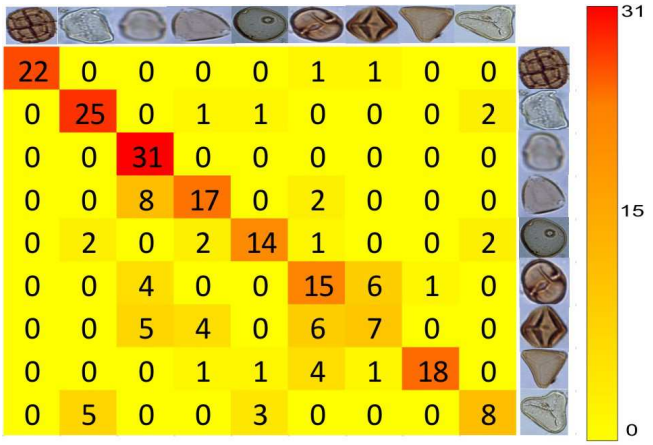
TABELA 4: Teste do tempo de uso da CPU.

| Dic. | C4.5 | SVM | KNN | AdaBoost |
|---|---|---|---|---|
| Dic02 | 0.00030 | 0.00030 | 0.00130 | 0.00010 |
| Dic04 | **0.00000** | 0.00010 | 0.00160 | 0.00000 |
| Dic08 | 0.00020 | 0.00020 | 0.00290 | 0.00000 |
| Dic16 | **0.00000** | 0.00030 | 0.00530 ○ | 0.00000 |
| Dic32 | 0.00020 | 0.00030 | 0.01040 ○ | 0.00020 |
| Dic64 | 0.00010 | 0.00140 | 0.02190 ○ | 0.00000 |
| Dic128 | 0.00030 | 0.00260 | 0.04300 ○ | 0.00000 |
| Dic256 | 0.00010 | 0.00440 ○ | 0.08810 ○ | 0.00010 |
| Dic512 | 0.00010 | 0.02750 ○ | 0.31040 ○ | 0.00030 |
| Dic1024 | 0.00100 | 0.01920 ○ | 0.37080 ○ | 0.00040 |
| Dic2048 | 0.00090 | 0.04190 ○ | 0.71740 ○ | 0.00060 |
| Dic4096 | 0.00240 | 0.08050 ○ | 1.42130 ○ | 0.00100 |
| Dic8192 | 0.00200 | 0.15860 ○ | 2.91610 ○ | 0.00140 |
| Dic16384 | 0.00420 | 0.28960 ○ | 6.39040 ○ | 0.00390 |

○, ○degradação ou melhoria estatisticamente significativa.

Através da observação das quatro tabelas apresentadas nesta seção, é possível determinar que o algoritmo C4.5 tem uma desempenho mais rápido do que os demais algoritmos. No entanto, apresenta um resultado muito parecido com o algoritmo AdaBoost. Assim apresenta-se o melhor resultado dos quatro algoritmos testados com esta métrica. A métrica Tempo de uso da CPU para teste faz uma análise do tempo que leva para o algoritmo testar o conjunto de dados gerado a partir das imagens. Como estes dois algoritmos obtiveram resultados parecidos é preciso analisar outros aspectos para verificar um melhor desempenho.

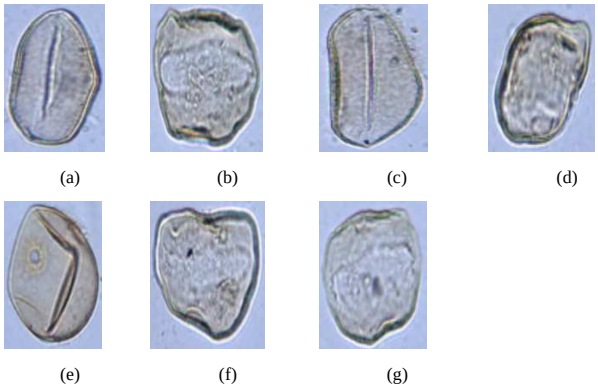
Com esses experimentos foi possível verificar que o algoritmo SVM teve uma boa classificação em relação aos outros classificadores. Por isso foi gerada uma matriz de confusão com o resultado de classificação deste algoritmo, para que este seja melhor avaliado. A matriz de confusão (Figura 4) a seguir apresenta os resultados da classificação do algoritmo de aprendizagem SVM.




a- *Anadenanthera*. b- Arecaceae; c- *Cecropia*; d- *Myrcia*; e- Poaceae; f- *Protium*; g- *Schinus*. h- *Serjania*; i- *Syagrus*.

Figura 4: Matriz de confusão gerada com o classificador SVM.

Para identificar visualmente quais imagens estavam sendo classificadas, outra classificação foi realizada a partir de um conjunto de 45 imagens de grãos de pólen. A Figura 5 mostra os resultados da classificação para o tipo Arecaceae. Foram classificadas corretamente as imagens (b), (d), (f) e (g). A imagem (a) e (c) são do tipo *Syagrus*, e a imagem (e) é do tipo Poaceae. A classificação errônea pode ter ocorrido devido às características parecidas, como por exemplo, a coloração e a textura.

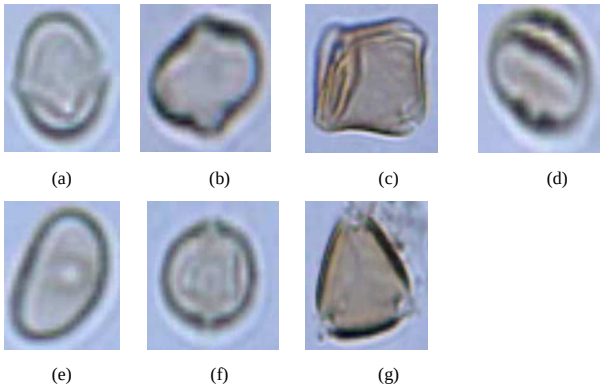


(a) (b) (c) (d)

(e) (f) (g)

Figura 5. Imagens classificadas como Arecaceae.

A Figura 6 mostra os resultados do tipo classificado como *Cecropia*. Nesses resultados as imagens (a), (b), (d), (e) e (f) foram classificadas corretamente. As imagens (c) e (g) são do tipo *Myrcia*. O erro na classificação nesse caso pode ter sido decorrente do tratamento da imagem, pois a resolução de ambas as espécies não estão boas, o que pode causar certa confusão na hora da classificação.

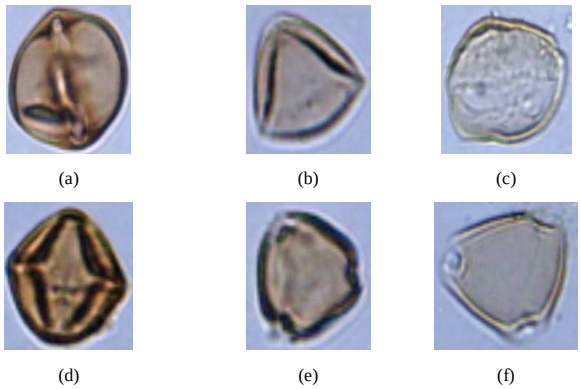


(a) (b) (c) (d)

(e) (f) (g)

Figura 6. Imagens classificadas como *Cecropia*.

A Figura 7 apresenta as imagens resultantes da classificação para o tipo *Myrcia*. As imagens (b), (e) e (f) foram classificadas corretamente. A imagem (a) é de *Protium*, a imagem (c) é de Arecaceae e a imagem (d) é do tipo *Schinus*. Um atributo que podemos notar certa semelhança é em relação às formas de cada grão, também é possível observar que a cor se assemelha com a do tipo Arecaceae.

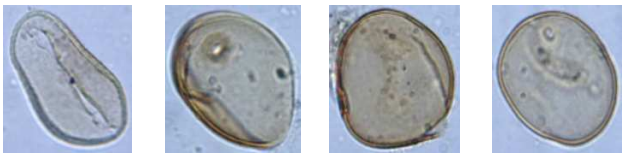


(a) (b) (c)

(d) (e) (f)

Figura 7. Imagens classificadas como *Myrcia.*

A Figura 8 mostra as imagens classificadas como Poaceae. As imagens (b), (c) e (d) estão classificadas corretamente. A Imagem (a) é do tipo *Syagrus*. A textura de *Syagrus* é semelhante ao tipo Poaceae, como também sua forma o que fez com que o classificador classificasse essa imagem de *Syagrus* como *Poaceae*.



(a) (b) (c) (d)

Figura 8. Imagens classificadas como *Poaceae*.

As imagens da Figura 9 apresentam os resultados da classificação para o tipo *Protium*, sendo que as imagens (b) e (d) foram classificadas corretamente. A imagem (a) é do tipo *Serjania* e a imagem (c) é do tipo *Schinus*. As imagens (b), (c) e (d) possuem formas muito semelhantes. A imagem (a) apresenta os atributos de cor e textura semelhante com a imagem (d). Analisando as semelhanças das imagens (d) e (a) podemos ver que a imagem (b), que é do mesmo tipo da

imagem (d), e não possuem semelhanças de cor e de morfologia com a imagem (a). Desta maneira podemos dizer que as imagens do mesmo tipo têm aspectos diferentes, o que é um problema na classificação.

(a) (b) (c) (d)

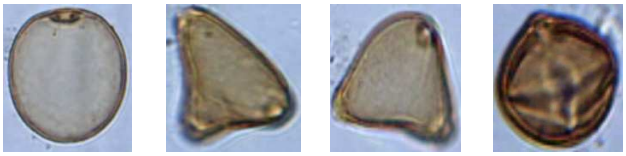
Figura 9. Imagens classificadas como *Protium*.

A Figura 10 mostra imagens que foram classificadas como *Schinus*. As imagens (b) e (c) estão classificadas corretamente. As imagens (a) e (d) são do tipo *Protium*. Forma, textura e cor são características muito semelhantes entre esses dois tipos, fazendo com que a classificação seja mais difícil.

(a) (b) (c) (d)

Figura 10. Imagens classificadas como *Schinus.*

A Figura 11 mostra os resultados da classificação para o tipo *Serjania*. As imagens (b) e (c) estão classificadas corretamente. A imagem (a) é do tipo *Poaceae* e a imagem (d) é do tipo *Schinus*. O erro na classificação pode-se observar que é em relação à forma como foi feita a captura das imagens. Por exemplo, as imagens (b) e (c) são do mesmo tipo, contudo existe uma variação na coloração, permitindo que o tipo *Serjania* apresente cores semelhantes as duas espécies *Poaceae* e *Schinus*.



(a) (b) (c) (d)

Figura 11. Imagens classificadas como *Serjania*.

Os resultados do tipo *Anadenanthera* foram todos classificados corretamente, em contrapartida, todas as imagens do tipo Syagrus foram classificadas erroneamente, sendo confundidas com imagens de outras espécies.

Com o algoritmo SMO 75% das imagens utilizadas no experimento foram classificadas corretamente. Este percentual já é um grande avanço na classificação polínica, pois [7] utilizando apenas dois gêneros conseguiu 97%, e como eles afirmam, quando aumenta o número de classe não é esperado que está porcentagem de acerto se mantenha.

## VI CONCLUSÃO

Este trabalho foi desenvolvido com o intuito de automatizar a classificação dos tipos polínicos. Foram realizados quatro experimentos utilizando métricas diferentes e algoritmos de aprendizagem diferentes. Com a análise do desempenho dos algoritmos nas métricas impostas foi possível determinar os algoritmos que mais se destacaram positivamente. Em relação ao desempenho em porcentagem de acerto, o SVM obteve melhor resultado. Em relação à métrica Área sob a curva ROC e Média Ponderada da média F o algoritmo SVM obteve melhor resultado. Para o problema da classificação de grãos de pólen os algoritmos SVM e KNN são os mais indicados para o problema. Em relação ao tamanho de dicionário mais adequado a esse conjunto de dados, foi possível observar que o dicionário 2048 se mostrou mais adequado, pois apresentou bons resultados na maioria das métricas para diferentes classificadores. Para trabalhos futuros espera-se que sejam aplicadas novas formas de extração de atributos para as mais variadas classes de grãos de pólen e que sejam testados novos algoritmos de aprendizagem utilizando outras métricas.

## REFERÊNCIAS


[1] Aha, David W., Dennis Kibler, and Marc K. Albert. "Instance-based learning algorithms." *Machine learning* 6.1 (1991): 37-66.

[2] Bay, Herbert, Tinne Tuytelaars, and Luc Van Gool. "Surf: Speeded up robust features." *Computer Vision–ECCV 2006*. Springer Berlin Heidelberg, 2006. 404-417.

[3] Bradley, Andrew P. "The use of the area under the ROC curve in the evaluation of machine learning algorithms." *Pattern recognition* 30.7 (1997): 1145-1159.

[4] Costa, Clayton M., and Suann Yang. "Counting pollen grains using readily available, free image processing and analysis software." *Annals of botany*104.5 (2009): 1005-1010.

[5] Cruz-Roa, Angel, Juan C. Caicedo, and Fabio A. González. "Visual pattern mining in histology image collections using bag of features." *Artificial Intelligence in Medicine* 52.2 (2011): 91-106.

[6] Csurka, Gabriella, et al. "Visual categorization with bags of keypoints."*Workshop on statistical learning in computer vision, ECCV.* Vol. 1. 2004.n

[7] Damian, M., et al. "Pollen classification of three types of plants of the family Urticaceae." *Proc. of the 12th Portuguese Conference on Pattern Recognition, Aveiro, Portugal*. 2002.

[8] France, I., et al. "A comparative study of approaches to automatic pollen identification." *Proceedings of British Machine Vision Conference*. 1997.

[9] Hripcsak, George, and Adam S. Rothschild. "Agreement, the f-measure, and reliability in information retrieval." *Journal of the American Medical Informatics Association* 12.3 (2005): 296-298.

[10] Kim, Kyungnam, et al. "Real-time foreground–background segmentation using codebook model." *Real-time imaging* 11.3 (2005): 172-185.

[11] Louveaux, J., Maurizio. A., Vorrwohl, G. (1970) Methods of melissopalynology, Bee World 51, 125-131.

[12] Paes, A. T. "Por dentro da estatística." *Einstein: Educ. Contin. Saúde* 6 (2008): 107-108.

[13] Quinlan, John Ross. *C4. 5: programs for machine learning*. Vol. 1. Morgan kaufmann, 1993.

[14] Schapire, Robert E. "The boosting approach to machine learning: An overview." *LECTURE NOTES IN STATISTICS-NEW YORK-SPRINGER VERLAG-* (2003): 149-172.

[15] Situ, Ning, et al. "Malignant melanoma detection by bag-of-features classification." *Engineering in Medicine and Biology Society, 2008. EMBS 2008. 30th Annual International Conference of the IEEE*. IEEE, 2008.

[16] Vapnik, Vladimir. *The nature of statistical learning theory*. Springer, 1999.

[17] Tamaki, Toru, et al. "A system for colorectal tumor classification in magnifying endoscopic NBI images." Computer Vision–ACCV 2010. Springer Berlin Heidelberg, 2011. 452-463.

[18] Wolff, L.F.; Reis, V.D.A.; Santos, R.S.S. Abelhas melíferas: bioindicadores de qualidade ambiental e de sustentabilidade da agricultura familiar de base ecológica. Pelotas: Embrapa Clima Temperado, 38p., 2008.

# Video-based Iris Recognition by *Quasi-Dynamic* Texture Analysis

Raissa Tavares Vieira, Virgílio de Melo Langoni, Adilson Gonzaga
Department of Electrical Engineering - EESC/USP
University of São Paulo
Av. Trabalhador São Carlense, 400
13560-590, São Carlos, SP, Brasil
raissa@ieee.org, virgilio.langoni@gmail.com, agonzaga@sc.usp.br

***Abstract*—In this paper, we evaluated two approaches for human iris recognition in video sequences. While applying pulses of a white led in the left eye, we have captured the iris texture in the right eye under near infrared illumination. Thus, the iris texture changes significantly during light pulse intervals. We established five time intervals according to the pupil movement. We applied a well-known method for dynamic texture analysis called LBP-TOP and we proposed to evaluate the iris texture as a sequence of frames from the same class. We have called this approach a *quasi-dynamic* method. The dynamic texture analysis captures texture information in three histograms concatenating the extracted features. The *quasi-dynamic* analysis uses the original LBP applied in video sequence as a class of static textures. The results demonstrate that our proposed methodology is better than LBP-TOP. We reached 88.6% of sensitivity and the LBP-TOP reached only 55.26%.**

**Keywords—*Dynamic texture; Image texture analysis; Iris recognition; Local Binary Pattern***

## I. INTRODUCTION

The human iris is a highly distinctive feature of an individual to establish its identity with very high accuracy. Among all the characteristics for biometric recognition, the pattern of iris texture is one believed to be the most distinguishable among different people. The iris possesses a textural pattern due to its anatomical structure consisting of fibrous tissues, blood vessels, crypts, furrows, freckles and pigmentation. The iris texture is believed to be unique to each eye, and is thought to be stable throughout the lifespan of an individual. Thus, human recognition using the iris texture is considered to be highly reliable.

Iris recognition through still-image is covered by the most of published research studies. However, the iris recognition in video is a relatively new research subject which needs to overcome a number of challenges [1]. Challenges include recognizing a person in infrared image sequences, coping with high and low resolution, processing video sequences of people walking through a portal, matching to still face images, etc.

Iris still-images usually have noise. By using an iris video, highlights and occlusions that occur in one frame may not be present in the next. Furthermore, it is possible to obtain a better image by using multiple frames to create a better iris image [2].

Broadly speaking, still-images from iris are analyzed by ordinary static texture methods. But, considering the iris movement in video images, the texture in the frame sequences shows temporal features that could be better recognized by dynamical techniques.

The term, dynamic texture, was firstly used by Saisan et al. [3] as sequences of images that exhibit temporal stationarity. They proposed to recognizing and to classifying dynamic textures in the space of dynamical systems where each dynamic texture is uniquely represented.

More recently, Zhao and Pietikainen [4] proposed two methods for dynamic texture recognition based on the Local Binary Pattern (LBP) approach. They called one of the methods as Volume Local Binary Patterns (VLBP), which are an extension of the LBP operator, combining motion and appearance. The second proposed method was called Local Binary Patterns from Three Orthogonal Planes (LBP-TOP). In this version, they reduced the computational cost by considering only the co-occurrences of the local binary patterns on three orthogonal planes (XY-XT-YT).

The aim of this paper is to propose a method that evaluates one iris video sequence as a class of static textures varying in time. Thus, each frame is processed by the local texture analysis based on LBP. For comparison, we applied the dynamic texture model called LBP-TOP, in video sequences of iris images. We generated our own video database, by capturing the iris videos under near-infrared illumination.

The remaining of this paper is divided into five sections. In section II we show the related works about iris recognition in video images. In section III we present our proposed methodology, applying the LBP-TOP and the *quasi-dynamic* approach. The results and discussions are shown in section IV. In section V we present our conclusions.

## II. IRIS IN VIDEO SEQUENCES

In 1993, Daugman [5] had already patented his method, which produces accuracy rates close to 99%. Despite the high accuracy rates, most iris recognition methods are applied to still-images.

More recently, some studies using iris video images were reported. The Video-based Automatic System for Iris Recognition (VASIR) is an iris recognition algorithm designed to work on both conventional iris images and iris images collected at a distance [1] [6]. All videos were captured while a person walked through a portal at a distance. However, video-based iris recognition at a distance is a relatively new research which still needs to overcome a number of issues.

Hollingsworth et al. [2] take advantage of the temporal continuity in an iris video to improve matching performance using signal-level fusion. From multiple frames of a frontal iris video, the authors create a single average image. Their experiments show that using average images created from ten frames of an iris video, performs better than experiments with single still-images.

Based on the eye consensual reaction, Costa and Gonzaga [7] proposed an innovative methodology to the extraction of information about the way the human eye reacts to light, and used such information for biometric recognition purposes. The results demonstrated that the proposed Dynamic Features, are discriminating, and may be employed for personal identification.

Furthermore, multiple still-images from iris have been used to improve iris recognition performance. Du [8] demonstrated higher rank on recognition rates by using three gallery images instead of one. Ma et al. [9] also enrolled three iris images and averaged the three Hamming distances to obtain the final score. Schmid et al. [10] demonstrated that fusing the scores using log likelihood ratio gave superior performance when compared to average Hamming distance. Liu and Xie [11], Roy and Bhattacharya [12] used multiple iris images for training classifiers.

Some papers were published, aiming to solve problems that occur in iris video image sequences, mainly in the "non-cooperative" or "iris-on-the move" environment. Du et al. [13] proposed a multi-level iris video image thresholding method that takes advantage of the correlations between consecutive images for video based thresholding. The thresholded images show clear pupil and iris areas, which can help the video iris segmentation and processing. Mahadeo et al. [14] propose a technique for selecting the best frames in an iris video. Taking advantage of the temporal correspondence in iris frames, they classified iris videos in three categories, namely Adequate, Motion Constrained and Time Constrained. Zuo and Schmid [15] introduced global and local factors that can be used to evaluate iris video and image quality. The proposed measures were evaluated by analyzing the relationship between the quality of the video iris images and the system performance. These relationships indicated that the proposed quality measures influenced the recognition performance.

## III. METHODOLOGY

### A. *Video Acquisition*

For our purpose, we developed a device for the acquisition of iris image sequences, as shown in Fig. 1. This device is similar to that shown in [7], but is composed of high-performance cameras with better resolution. The left eye receives visible light stimuli (white light) at computer-controlled specific time intervals, while the right eye image is captured and digitized into a video sequence under NIR illumination. The two sides of the device related to the left and right eye are optically isolated from each other.

Fig. 1. The proposed device to capture iris image sequences.

The video sequence obtained under NIR illumination is synchronized with the "visible light" pulses (a white LED was employed) applied to the other eye. Thus, it is possible to extract the features of the frames during pupil contraction or dilation without the interference of light reflections from the iris, pupil and sclera. Their movements are induced by the light stimulus applied to the other eye and, due to the consensual reflex, repeated by the eye whose image is being digitized without interference from visible light reflections.

The prototype was built with two multi-spectral cameras, digital and progressive scan model AD-080GE, GigE Vision compliant, belonging to the JAI C3 Advanced family [16]. The AD-080GE employs 2 CCDs, one for BAYER color and the other for NIR monochrome utilizing prism optics. The AD-080GE provides a frame rate of 30 frames/second at full resolution of 1024 (h) x 768 (v) active pixels.

In this work, we used only the right camera to capture the iris image sequence, while pulsing a white light on the left eye. We generated a database with 53 different persons and four sequences for each one taken in a random way. Some persons had their video sequences taken sequentially and others were digitized in different days. Each video sequence was established in the following way. During the video capture, we applied visible light pulses to the eye not being digitized using the white LED, whereas the camera digitizes the image sequence of the other eye that is illuminated in the near infrared band. Figure 2 shows the time intervals determined by our methodology. These intervals were defined empirically, and can be altered by the control software. Thus, each video has several frames of the iris with dilated pupil and other frames with contracted pupil.

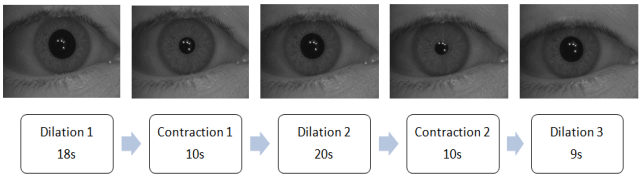



Fig. 2. Time interval during the pupil dilation and contraction.

*B. Video pre-processing*

In order to segment the iris, we have to exclude the regions that will interfere with the recognition, such as the eyelashes and the eyelids. Due to eye movements, it is necessary to delete those frames that cannot be used for the recognition method and select only the adequate ones. The algorithm excludes the inadequate frames and uses the remaining ones as did in [7].

We pre-processed each usable (or good) frame in order to calculate the pupil center (xc, yc). This is done with a low-pass filter, highlight reflections elimination, image thresholding, and morphological erosion and dilation.

We calculated the pupil radius for each frame by comparing the gray-level variations from the pupil center, through the horizontal direction. This is an estimated value, and we multiplied the radius by a factor of 1.1 to guarantee a texture sample without occlusions or highlight reflections. This value was empirically defined.

From each usable frame we extracted two texture samples of 120 x 45 pixels as shown in Fig. 3. The coordinates of this two texture samples are given by:

Left texture sample:

$$x_{1L} = x_c - 60 \quad and \quad y_{1L} = y_c - 1.1 * radius$$
$$x_{2L} = x_c - 60 \quad and \quad y_{2L} = y_c - 1.1 * radius - 45$$
$$x_{3L} = x_c + 60 \quad and \quad y_{3L} = y_c - 1.1 * radius - 45$$
$$x_{4L} = x_c + 60 \quad and \quad y_{4L} = y_c - 1.1 * radius$$

Right texture sample:

$$x_{1R} = x_c - 60 \quad and \quad y_{1R} = y_c + 1.1 * radius$$
$$x_{2R} = x_c - 60 \quad and \quad y_{2R} = y_c + 1.1 * radius + 45$$
$$x_{3R} = x_c + 60 \quad and \quad y_{3R} = y_c + 1.1 * radius + 45$$
$$x_{4R} = x_c + 60 \quad and \quad y_{4R} = y_c + 1.1 * radius$$

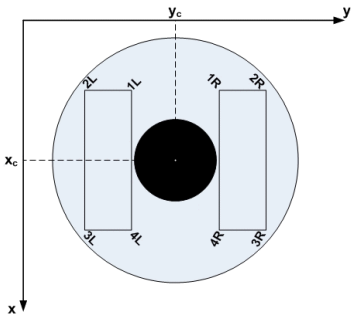



Fig. 3. Coordinates of the two texture samples.

The segmented texture samples were disposed in 10 different sets, as shown in Table I:

TABLE I. SEGMENTED TEXTURE SAMPLE SETS

| | |
|---|---|
| LD1 | Left Dilation 1 |
| RD1 | Right Dilation 1 |
| LC1 | Left Contraction 1 |
| RC1 | Rigth Contraction 1 |
| LD2 | Left Dilation 2 |
| RD2 | Rigth Dilation 2 |
| LC2 | Left Contraction 2 |
| RC2 | Rigth Contraction 2 |
| LD3 | Left Dilation 3 |
| RD3 | Rigth Dilation 3 |

*C. Applying dynamic texture analysis based on LBP*

The key problem of dynamic texture recognition is how to combine motion features with appearance features.

Zhao and Pietikainen [4] proposed to concatenating LBP on three orthogonal planes (XY, XT, and YT), considering only the co-occurrence statistics in these three directions. The video sequence is thought as a stack of XY planes in axis T, a stack of XT planes in axis Y and a stack of YT planes in axis X, respectively.

The XT and YT planes provide information about the space-time transitions. This approach was called LBP-TOP. The LBP-TOP uses three orthogonal planes that intersect in the center pixel. It considers the feature distributions from each separate plane and then concatenates them together.

The LBP code is extracted from the XY, XT, and YT planes, and denoted as XY-LBP, XT-LBP, and YT-LBP and then concatenated into a single histogram.

*D. Applying local texture analysis to video frames*

Our proposal for results comparison, is applying the static LBP methodology, frame by frame, in the iris video sequences. We have called this method as "*quasi-dynamic*", because the involved images are sequences of video frames. Each segmented texture sample has generated a histogram with 256 bins. We used the leave-one-out cross validation comparing each histogram with the other ones.

Among several classifiers used to compute goodness of fit between two histograms, such as log-likelihood ratio and histogram intersection, we choose the Chi-square distance [17], the same metric used by [18], as the classifier (1).

$$\chi^2(P_i, Q_i) = \frac{1}{2}\sum_{i=0}^{255}\frac{(P_i - Q_i)^2}{(P_i + Q_i)} \qquad (1)$$

Where, *Qi* are the gray-level's frequencies of the query sample and *Pi* are the gray-level's frequencies of the compared sample from the set.

We generated confusion matrices with True Positives (*TP*), or the number of correctly classified samples, and False Negatives (*FN*), or the wrongly classified samples, for all of the query samples. The Hit-rate or Sensitivity was calculated by (2).

$$S = \frac{TP}{TP + FN} \qquad (2)$$

Based on the texture samples extracted from each iris frame, we have established four experiments. Firstly, we selected 40 different people and two video sequences from each one. The choice was based on the best video samples with a minimal number of occlusions and reduced defocusing, avoiding twinkling. The texture samples were extracted as explained in the last section, and Table II shows the number of images used in the experiment number 01.

TABLE II. NUMBER OF IMAGES USED IN EXPERIMENT 01

| | Samples from each video | Number of images |
|---|---|---|
| LD1 | 140 | 140*40*2 = 11,200 |
| RD1 | 140 | 140*40*2 = 11,200 |
| LD2 | 121 | 121*40*2 = 9,680 |
| RD2 | 121 | 121*40*2 = 9,680 |
| LD3 | 77 | 77*40*2 = 6,160 |
| RD3 | 77 | 77*40*2 = 6,160 |
| LC1 | 12 | 12*40*2 = 960 |
| RC1 | 12 | 12*40*2 = 960 |
| LC2 | 12 | 12*40*2 = 960 |
| RC2 | 12 | 12*40*2 = 960 |

By joining the right and left textures within each interval of contraction and dilation we established the sets for the experiment number 02. By combining two regions of the segmented samples within the same side of the iris (left or right) we established the experiment number 03. In the experiment number 04, we calculated the hit-rate of one texture sample from one side (left or right) of the iris to be correctly classified, taking into account all the intervals of contraction and dilation. The sets for these experiments are shown in Table III.

TABLE III. NUMBER OF IMAGES USED IN EXPERIMENTS 02, 03, 04

| Experiment 02 | Number of images |
|---|---|
| D1 (LD1 + RD1) | 140*40*2*2 = 22,400 |
| D2 (LD2 + RD2) | 121*40*2*2 = 19,360 |
| D3 (LD3 + RD3) | 77*40*2*2 = 12,320 |
| C1 (LC1 + RC1) | 12*40*2*2 = 1,920 |
| C2 (LC2 + RC2) | 12*40*2*2 = 1,920 |
| **Experiment 03** | **Number of images** |
| LD12 (LD1 + LD2) | (140+121)*40*2 = 20,880 |
| RD12 (RD1 + RD2) | (140+121)*40*2 = 20,880 |
| LD13 (LD1 + LD3) | (140+77)*40*2 = 17,360 |
| RD13 (RD1 + RD3) | (140+77)*40*2 = 17,360 |
| LD23 (LD2 + LD3) | (121+77)*40*2 = 15,840 |
| RD23 (RD2 + RD3) | (121+77)*40*2 = 15,840 |
| LC12 (LC1 + LC2) | (12+12)*40*2 = 1,920 |
| RC12 (RC1 + RC2) | (12+12)*40*2 = 1,920 |
| **Experiment 04** | **Number of images** |
| LS (LD1 + LD2 + LD3 + LC1 + LC2) | (140+121+77+12+12)*40*2 = 28,960 |
| RS (RD1 + RD2 + RD3 + RC1 + RC2) | (140+121+77+12+12)*40*2 = 28,960 |

## IV. RESULTS

In this section, we show the results for each experiment explained in the last section. For each one, we calculated the correct classifications (hit-rate) and we used the chi-square distance between the query sample and the samples in each set by leave-one-out cross validation.

### *A. Results for the static LBP applied to sequences of video frames (quasi-dynamic)*

As explained in Section III.D, we applied the texture analysis method based on the static LBP in the video frames of the acquired iris' sequences. We divided our tests in four experiments, and the achieved results for the experiment number 01 are shown in Table IV.

TABLE IV. *QUASI-DYNAMIC* HIT-RATE OF THE EXPERIMENT 01

| | Number of Samples | Hit-rate |
|---|---|---|
| LD1 | 11,200 | 88.58% |
| RD1 | 11,200 | 82.61% |
| LC1 | 960 | 76.45% |
| RC1 | 960 | 72.91% |
| LD2 | 9,680 | 88.13% |
| RD2 | 9,680 | 83.11% |
| LC2 | 960 | 77.91% |
| RC2 | 960 | 73.75% |
| LD3 | 6,160 | 81.62% |
| RD3 | 6,160 | 82.01% |

We can see that during the contraction intervals the classification performance is worse than during the dilation intervals. The best result was reached in the first interval from the left side (LD1) when the eye was not submitted to light pulses yet. During the pupil dilation intervals, at the end of the process, even with a half of samples (LD3), the hit-rate for the left textures, was reduced by 6.96%. It is interesting to observe that, in general, the left side of the iris shows a better recognition rate than the right side. When we consider the texture samples within each interval (dilation and contraction) joined together, the method performance was worse than when the texture samples were analyzed by their individual sets. The Table V shows the hit-rate calculated for the experiments 02, 03 and 04.

Considering the results obtained by the experiment number 02, we can observe that the method performance is better in the first interval, that is, when the light pulse was not applied to the person's eye. In the experiment number 03 the hit-rate shows that the best performance occurs in the first time interval by analyzing samples from the left dilation associated with samples from the same place during the second dilation.

TABLE V. *QUASI-DYNAMIC* HIT-RATE OF THE EXPERIMENTS 02,03,04

| **Experiment 02** | **Number of Samples** | **Hit-rate** |
|---|---|---|
| D1 | 22,400 | 80.96% |
| C1 | 1,920 | 68.07% |
| D2 | 19,360 | 80.60% |
| C2 | 920 | 68.75% |
| D3 | 12,320 | 75.97% |
| **Experiment 03** | **Number of Samples** | **Hit-rate** |
| LD12 | 20,880 | 84.38% |
| RD12 | 20,880 | 78.98% |
| LD13 | 17,360 | 82.10% |
| RD13 | 17,360 | 78.01% |
| LD23 | 15,840 | 83.08% |
| RD23 | 15,840 | 80.65% |
| LC12 | 1,920 | 77.39% |
| RC12 | 1,920 | 72.55% |
| **Experiment 04** | **Number of Samples** | **Hit-rate** |
| LS | 28,960 | 80.76% |
| RS | 28,960 | 76.80% |

Taking into account all the texture samples extracted from one side of the video sequence for all the 40 individuals, and comparing the correct classifications for the left side and for the right side, we saw in the results obtained in the experiment number 04 that the left side is more discriminant than the right side. However, the best result was achieved when we have analyzed the correct classifications given by the texture's samples extracted from the left side during the first interval of acquisition. In this time interval we have not applied a light pulse and the pupil was dilated.

Based on our tests, when we applied the LBP methodology for iris texture analysis the best results were achieved when the eye's pupil was dilated. We think that due to the iris muscles (trabeculas) more contracted, the iris texture is more homogeneous, and thus, more discriminant. Furthermore, the samples extracted from the left side have performed better than the samples from the right side of the iris. We don't have a scientific conclusion about this fact. Maybe it is related to the acquisition equipment and the eye position. It requires further investigations.

The texture samples extracted from the left side during the first time interval (LD1) compared to the set of samples from the same place and same time interval, gave the best correct classifications (88,58%). All the combinations of samples from other places and different time intervals show inferior performances.

### *B. Results for the LBP-TOP applied to video frames*

We have chosen only the first experiment of the "*quasi-dynamic*" method for testing the dynamic texture based on LBP-TOP, due to achieving the best correct classifications among all. The texture samples have a concatenated histogram of three planes (XY-XT-YT) whereas the static LBP generates histograms of one single plane (XY) in the *quasi-dynamic* approach. We used the same sets of texture frames shown in the *quasi-dynamic* tests. The results are shown in Table VI.

TABLE VI. LBP-TOP HIT-RATE FOR EACH SET OF VIDEO SEQUENCES

| | **Number of Samples** | **Hit-rate** |
|---|---|---|
| LD1 | 11,200 | 55.26% |
| RD1 | 11,200 | 44.74% |
| LC1 | 960 | 53.95% |
| RC1 | 960 | 43.42% |
| LD2 | 9,680 | 53.95% |
| RD2 | 9,680 | 40.79% |
| LC2 | 960 | 48.68% |
| RC2 | 960 | 32.89% |
| LD3 | 6,160 | 42.11% |
| RD3 | 6,160 | 40.79% |

The graphics in the Fig. 4 shows the difference in performance, taking into account the time intervals of the pupil dilations and contractions.

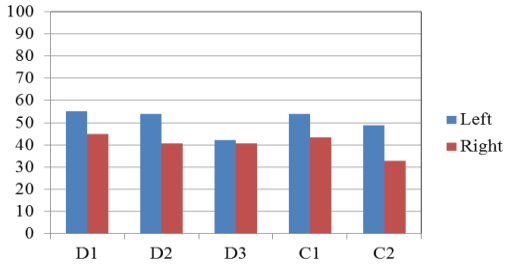



Fig. 4. Dynamic texture performance comparison to the left and right side of the iris

The proposed "*quasi-dynamic*" method for iris recognition by local texture analysis, by far surpassed the performance of the well-known method based on LBP-TOP. Fig. 5 shows the performance comparison between the two approaches, when the frames are considered as a video sequence within each interval of contraction and dilation.

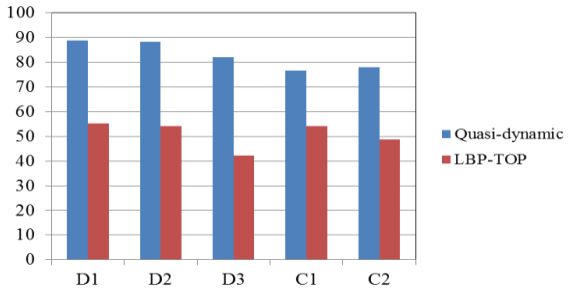



Fig. 5. Performance comparison between our proposed method and the LBP-TOP

We believe that the saccadic eye movements are the main responsible by the weak performance of the analysis based on dynamic texture. That is, the person's eye changes its position related to the camera axis, during the video acquisition. Thus, the texture variation in time, responsible to the XT and YT histogram generation, is not correlated frame by frame. Notwithstanding, the texture variations in time is not well behaved, causing the bad performance in iris recognition.

## V. Conclusions

In this paper, we proposed a way of recognizing human iris, in video images with controlled pupil dilation and contraction. As we used sequence of videos, and we evaluated the local texture in each video frame during defined time intervals, we called our approach as "*quasi-dynamic*" methodology.

For results comparison, we applied a well known method for dynamic texture analysis. The LBP-TOP was chosen due to its similarity with our proposed approach. The difference between the two methods is that the dynamic texture analysis captures texture information in three histograms (XY-XT-YT) concatenating the extracted features. The *quasi-dynamic* analysis evaluates the frames in the video sequence as a class of static textures. The results demonstrate that the *quasi-dynamic* iris recognition, proposed by our methodology, are discriminating, and may be employed for iris identification in video sequences. The performance of our method surpasses the LBP-TOP for this application. The best hit-rate reached by our method was about 88.6% and the by the LBP-TOP was only 55.26%.

Furthermore, another interesting observation was that the samples extracted from the left side of the iris have performed better than the samples extracted from the right side. And more, we observed that the samples extracted during the first time interval, when the eye was not submitted to light pulse, have performed better than that samples extracted in time interval after light pulse application.

In addition, the possibility to extract features from living irises could increase the resistance of our approach to fraud attempts in personal identification. For example, the proposed method can check if the input image being analyzed is actually from a "living iris" or not by determining if the subject to be validated responds to the illumination stimuli applied, or if the subject is using artificial irises in an attempt to cheat the recognition method. In addition to personal recognition, the methodology proposed also allows for the evaluation of the iris behavior at different moments under different illumination stimuli.

## Acknowledgment

The authors would like to thank the S˜ao Paulo Research Foundation (FAPESP), grant #2011/18645-2, and National Council for Scientific and Technological Development (CNPQ) for their financial support of this research.

## References

[1] Y. Lee, P. J. Phillips, and R. J. Micheals, "An Automated Video-Based System for Iris Recognition", in LNCS 5558, M. Tistarelli and M. S. Nixon (Eds.), ICB2009, Springer-Verlag Berlin Heidelberg 2009, pp. 1160–1169, 2009.

[2] K. Hollingsworth, T. Peters, K. W. Bowyer, and P. J. Flynn, "Iris Recognition Using Signal-Level Fusion of Frames From Video", IEEE Transactions on Information Forensics and Security, vol. 4, no. 4, pp. 837-848, Dec. 2009.

[3] P. Saisan, G. Doretto, Y. N. Wu, and S. Soatto, Dynamic Texture Recognition, in *Proc. of Conference on Computer Vision and Pattern Recognition* - CVPR '01, vol. 2, December 2001, pp. 58-63.

[4] G. Zhao and M. Pietikainen, "Dynamic Texture Recognition Using Local Binary Patterns with an Application to Facial Expressions", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 6, June 2007, pp. 915-928.

[5] J. Daugman, "How iris recognition works," in Proc. *Int. Conf. Image Process.*, Dec. 2002, vol. 1, pp. I-33–I-36.

[6] Y. Lee, R. J. Micheals, J. J. Filliben, and P. J. Phillips, "Robust Iris Recognition Baseline for the Grand Challenge", National Institute of Standards and Technology, Gaithersburg, MD 20899, NISTIR 7777, March 2011.

[7] R. M. da Costa, A. Gonzaga, "Dynamic Features for Iris Recognition, IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, vol. 42 , no. 4, pp. 1072 – 1082, Aug. 2012

[8] Y. Du, "Using 2D log-Gabor spatial filters for iris recognition," *SPIE Biometric Technology for Human Identification III*, vol. 62020, 2006.

[9] L. Ma, T. Tan, Y. Wang, and D. Zhang, "Efficient iris recognition by characterizing key local variations," *IEEE Transactions on Image Processing*, vol. 13, no. 6, pp. 739-750, 2004.

[10] N. A. Schmid, M. V. Ketkar, H. Singh, and B. Cukic, "Performance analysis of iris based identification system at the matching score level," *IEEE Transactions on Information Forensics and Security*, vol. 1, no. 2, pp. 154-168, 2006.

[11] C. Liu and M. Xie, "Iris recognition based on DLDA," in *Proc. International Conference on Pattern Recognition*, pp. 489-492, 2006.

[12] K. Roy and P. Bhattacharya, "Iris recognition with support vector machines," in *Proc. International Conference on Biometrics*, pp. 486-492, 2006.

[13] Y. Du, N. L. Thomas, and E. Arslanturk, "Multi-level Iris Video Image Thresholding", *IEEE Workshop on Computational Intelligence in Biometrics: Theory, Algorithms, and Applications, CIB 2009*, pp. 38 – 45, 2009.

[14] N. K. Mahadeo, A. P. Paplinski, S. Ray, "Automated Selection of Optimal Frames in NIR Iris Videos", in *Proc. of 2013 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, Hobart, TAS, 26-28 Nov., pp.1 – 8, 2013.

[15] J. Zuo and N. A. Schmid, "Global and Local Quality Measures for NIR Iris Video", in *Proc. of IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops*, Miami, FL, 20-25 June, pp.120-125, 2009.

[16] JAI – AD-080GE camera: http://www.jai.com/en/products/ad-080ge

[17] M. W. O. Pele, "The quadratic-chi histogram distance family". in *Proc. of the 11th European conference on Computer vision Part II (ECCV'10)*, Berlin, Heidelberg, 2010. Springer-Verlag, pp. 749–762.
Z. Guo, L. Zhang, D. Zhang, "A completed modeling of local binary pattern operator for texture classification", *IEEE Transactions on Image Processing*, vol. 19, Issue 6, June 2010.

# Detection, Extraction and Text Translation in Digital Images using Android Platform

Luiz P. Rabachini, Marco A. Piteri, Almir O. Artero
Universidade Estadual Paulista – Unesp
Faculdade de Ciências e Tecnologia – FCT
Presidente Prudente, SP, 19060-900
{luiz.rabachini, piteri, almir}@fct.unesp.br

Francisco A. da Silva, Danillo R. Pereira
Faculdade de Informática de
Presidente Prudente, FIPP, Unoeste
Presidente Prudente, SP, 19050-920
chico@unoeste.br, dpereira@ic.unicamp.br

***Abstract*—The purpose of this paper is present a software for mobile devices using the Android operating system, able to detect, extract and translate texts present in digital images, initially with support for english and portuguese languages. The ultimate goal is modify the captured image so that the original text seems to have been created in the target language of the translation. In the experiments were submitted twenty images for the system to analyze your operation in different situations in which texts are commonly found. Their use in everyday environments is feasible and produces an easy mechanism for translation of texts, since it eliminates the need for manual entry of characters.**

## I. Introdução

Em decorrência da globalização, é cada vez mais comum encontrar indivíduos provenientes de outras localidades que desconhecem o idioma do país visitado e enfrentam dificuldades de comunicação. Compreender textos presentes em locais que fazem parte do cotidiano de pessoas, como estradas, restaurantes e aeroportos, é uma tarefa que deve ser executada com certa rapidez, pois o dinamismo com o que o cenário é modificado pode fazer com que a informação seja alterada ou perdida rapidamente.

Sob esta perspectiva, a proposta deste trabalho é elaborar um sistema para dispositivos móveis que utilizam a plataforma Android [1], [2], capaz de analisar imagens digitais arbitrárias e realizar a tradução dos textos presentes nestas. Inicialmente, os idiomas suportados são o português e inglês, mas outros podem ser facilmente adicionados. Um sistema com essas características é composto fundamentalmente por três etapas de processamento: detecção de regiões de texto, reconhecimento óptico de caracteres e uma ferramenta de tradução [3]. Além dessas, neste trabalho foi incluída uma quarta etapa, que compreende a edição automática da imagem original, com o propósito de fazer com que o resultado da operação seja apresentado ao usuário através da substituição das regiões de texto originais, de forma que o texto da imagem pareça ter sido escrito na língua de destino da tradução.

## II. Trabalhos Relacionados

A etapa de detecção de regiões de textos em imagens digitais é a mais crítica do processo de tradução. Distinguir caracteres e agrupá-los em palavras e linhas de texto é uma tarefa complexa, pois geralmente estão misturados a elementos como vegetação, textura e objetos diversos, dentre outros. Assim, inúmeras pesquisas foram realizadas nas últimas décadas a fim de identificar regiões de texto em imagens naturais. O trabalho de Jung et al. [5] apresenta uma síntese das principais técnicas presentes na literatura e serviu de base para a pesquisa bibliográfica do presente trabalho. Segundo ele, essas técnicas podem ser classificadas em dois tipos: as que utilizam características de textura e as baseadas em regiões. A primeira realiza a análise de propriedades da textura dos textos para distingui-los do fundo através de técnicas como Wavelet [6], Filtros de Gabor [7] e Transformada Rápida de Fourier [8]. A segunda consiste em identificar subestruturas na imagem original, como elementos conectados e bordas, e uni-las através de regras até que sejam definidas as potenciais regiões de texto.

Como as técnicas de detecção baseadas em textura são suscetíveis à variações de fonte e estilo dos caracteres, são gerados diversos descritores de características com base em representações de regiões de texto válidas, que são posteriormente utilizadas na fase de classificação dos elementos da imagem. O método de obtenção desses descritores possui alta complexidade computacional e apresenta um tempo de processamento elevado, fazendo com que seja executado em uma etapa prévia de treinamento do sistema. A busca por regiões de texto é realizada por meio da fragmentação da imagem de entrada em diversas regiões que são exaustivamente comparadas com os descritores obtidos para que seja possível definir as prováveis áreas que contenham caracteres de texto [5].

A pesquisa de Minetto et al. [9] apresenta o descritor T-HOG, que utiliza o conceito de histograma de gradientes orientados (HOG) baseado em texturas para detectar linhas únicas de caracteres textuais. Esse descritor serve de base para a criação de um conjunto de treinamento que posteriormente é empregado na busca de prováveis regiões de texto através do classificador *Support Vector Machines* (SVM) [10].

A técnica proposta por Ohya et al. [11] utiliza binarização baseada em um limiar local e detecção de prováveis regiões de caracteres através da análise de diferenças nos níveis de cinza, que posteriormente são filtrados através de comparação aproximada com uma base de características previamente estabelecida. Esse método gerou bons resultados segundo a

pesquisa publicada, porém a segmentação por binarização é inadequada para objetos que apresentam alterações bruscas nos níveis de cinza, grande quantidade de ruído e diferenças de iluminação [5]. O método apresentado por Lee e Kankanhalli [12] não é tão suscetível a esse tipo de problema e utiliza o conceito de componentes conectados para detectar e reconhecer regiões de texto. Porém, os resultados apresentados foram realizados apenas com imagens de textos em *conteiners* de carga. Zhong et al. [13] também emprega componentes conectados em conjunto com o método de redução de cores, pois pressupõe que estas sejam uniformes ao longo do texto e que este ocupe um espaço significativo da imagem. Para isso, realiza a quantização do espaço de cores pela análise dos picos presentes no histograma RGB. Posteriormente as regiões de texto são identificadas por heurística com base na análise do diâmetro, área e alinhamento dos elementos.

Por outro lado, o trabalho de Epshtein et al. [4] apresenta um método de detecção de texto utilizando o operador local *Stroke Width Transform* (SWT), que se baseia na análise da largura dos contornos dos componentes da imagem para definir possíveis regiões de texto, que na sequência são filtradas e agrupadas de acordo com similaridades geométricas. Isso se deve ao fato de textos apresentarem uma largura de fonte muito similar, ao contrário de folhagem e outros elementos que possuem dimensões diversas. Dessa forma, este método foi escolhido para o presente trabalho por não utilizar treinamento prévio sobre os caracteres a serem identificados e empregar apenas características locais dos pixels que compõem a imagem.

## III. Desenvolvimento do Sistema

A aplicação desenvolvida consiste em um sistema que realiza a tradução automática de textos presentes em imagens digitais através de dispositivos móveis com sistema operacional Android [1], [2]. Para organização e caracterização das etapas presentes ao longo desse processo, a aplicação foi dividida nos seguintes módulos:

**Aquisição de Imagem:** Captura a imagem de uma cena a partir da câmera do dispositivo móvel e controla funções como luminosidade e foco;

**Detecção de Regiões de Texto:** Analisa a imagem obtida na etapa anterior a fim de identificar potenciais regiões que contenham caracteres textuais;

**Extração de Texto:** Realiza o reconhecimento óptico dos possíveis caracteres presentes nas regiões de texto detectadas;

**Tradução de Texto:** Efetua a correção ortográfica das palavras obtidas na fase anterior e executa as operações necessárias para obter a versão traduzida do texto;

**Apresentação do Resultado:** Modifica automaticamente a imagem da cena capturada a fim de sobrescrever o texto original com o resultado da tradução.

Esses módulos possuem um relacionamento direto e sequencial, como pode ser visualizado na Figura 1. Assim, eventuais erros provenientes de qualquer etapa são diretamente refletidos nas subsequentes, o que pode afetar a qualidade do resultado final. A seguir é realizada uma descrição detalhada do funcionamento de cada uma dessas diferentes etapas e a forma como o resultado do processamento obtido em cada uma é propagado para as demais, desde a detecção dos textos na imagem digital até a modificação desta para sobrescrever o texto original com o resultado da tradução.

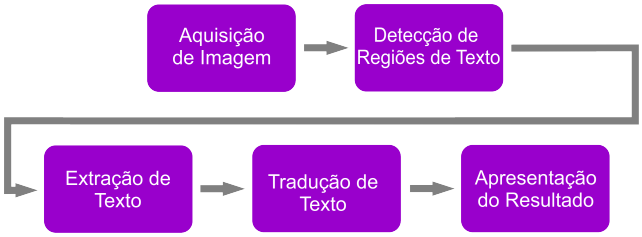



Figura 1. Fluxo do processo de tradução de textos em imagens digitais.

### A. Aquisição de Imagem

As imagens da cena são adquiridas pela aplicação por meio do *Charge Coupled Device* (CCD) do dispositivo móvel em conjunto com o SDK do Android [1], que controla os recursos de câmera integrados ao sistema. Com isso, o foco e o ajuste de luminosidade são manipulados de forma automática através de mecanismos de aceleração presentes no *hardware* da câmera do dispositivo. Isso melhora o desempenho da aplicação, pois não há necessidade de cálculos para correção via *software*. As imagens capturadas pelo sistema possuem dimensão fixa de 800x480 pixels que, embora não seja a resolução máxima obtida por dispositivos mais modernos, satisfaz as necessidades de detecção e reduz o tempo de busca por possíveis regiões de texto. Essa resolução pode ser modificada automaticamente para adequação aos recursos de *hardware* e *software* presentes no dispositivo móvel em que o sistema é executado.

Vale realçar que o tempo de exposição da câmera fotográfica de dispositivos móveis é geralmente alto, o que pode ocasionar borrões nas imagens devido a movimentos bruscos do usuário no momento da captura. Esse tipo de interferência afeta diretamente as etapas seguintes do processamento e podem inviabilizar a tradução dos textos presentes na imagem. O tratamento desses casos não é realizado na fase de aquisição de imagem, mas sim na de detecção de textos, descrita a seguir.

### B. Detecção de Regiões de Texto

O método de detecção de textos de Epshtein et al. [4] possui implementação disponível no trabalho de Zhu [14], em que o foco é a localização de textos em ambientes internos para a biblioteca de robótica Literate PR2, parte do projeto *Robot Operating System* (ROS) [15]. Por possuir código fonte aberto e ser desenvolvida totalmente em linguagem C/C++, essa biblioteca foi utilizada no presente trabalho na etapa de detecção de textos. Também foram utilizados alguns algoritmos de processamento de imagens, como detecção de bordas e *blur*, disponíveis na biblioteca de visão computacional OpenCV [16].

O primeiro passo na busca por regiões de texto é a conversão da imagem RGB proveniente da etapa de aquisição em tons

de cinza, em que cada pixel desta recebe o valor obtido pela expressão $0,299 \cdot R + 0,587 \cdot G + 0,114 \cdot B$ de seu correspondente na imagem colorida. Em seguida, é realizada a detecção de bordas dos elementos presentes na cena através do método de Canny [17]. Com base nos testes realizados e em consonância com a referência [14], os limiares inferior e superior utilizados neste algoritmo foram $50$ e $120$, respectivamente. A imagem em tons de cinza ainda serve de base para o cálculo dos gradientes nas direções $x$ e $y$ para cada pixel de borda definido anteriormente, obtidos através do método de Sobel [17] com máscara de ordem $3$ e que são utilizados para orientar a detecção da largura interna dos objetos presentes na imagem. Esse cálculo é realizado uma única vez e utilizado nos dois ciclos do método de detecção, um para identificar caracteres mais claros e outro para caracteres mais escuros do que o fundo da imagem, em que o valor de cada um dos gradientes é invertido entre esses dois ciclos [14].

A largura interna é obtida através do cálculo da distância euclidiana entre o pixel inicial $p$ e seu oposto $q$, ambos pertencentes a bordas dos componentes da imagem, conforme pode ser observado na Figura 2. Esses pixels são considerados opostos caso os dois vetores gradiente nas direções $x$ e $y$, aqui definidos como $d_p$ e $d_q$, satisfazem a condição $d_q = -d_p \pm \pi/6$. Caso essa correspondência não exista ou as dimensões da imagem sejam excedidas sem que o pixel $q$ seja encontrado, a largura é descartada. A sequência de pixels percorridos entre $p$ e $q$ é dada pela expressão $r = p + n \cdot d_p$, para $n \in \mathbb{R}^*$, que representa o refinamento do caminho [4]. Outra condição utilizada nos testes foi o descarte de caminhos cuja largura obtida foi superior a $4\%$ do maior valor entre a altura e a largura da imagem analisada [14].

Após todos os caminhos e larguras internas dos elementos que compõem a imagem serem definidas, é gerada uma nova imagem com as mesmas dimensões da original e inicialmente com todos as suas posições contendo o valor $0$. A largura obtida em cada caminho é então representada nesta imagem exatamente nos mesmos pontos $(x, y)$ dos pixels que os compõem. Durante esse procedimento, é possível que haja o cruzamento entre esses caminhos e um pixel possua dois ou mais valores de largura associados. Nessa situação é atribuído o menor valor possível para posterior normalização, processo em que cada pixel recebe o valor mediano das larguras presentes ao longo do caminho que o contém [4].

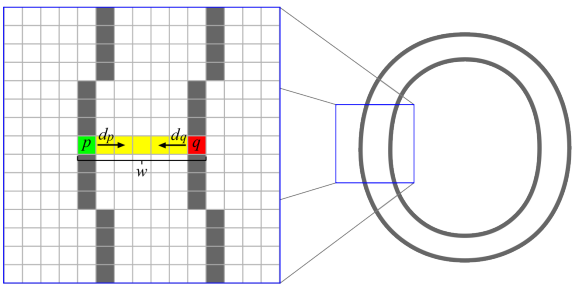



Figura 2. Busca pela largura do caractere. (Adaptado de Epshtein et al. [4])

O próximo passo do processo de detecção de regiões de texto consiste em agrupar os pixels semelhantes da imagem resultante do SWT através da análise da vizinhança de cada um. Dois pixels são considerados vizinhos caso o valor, ou largura, associado a cada um não exceda uma relação de proporção igual a $3$, o que torna a associação suscetível a diferentes fontes presentes em imagens naturais. Os pixels são então agrupados em regiões que possivelmente representam caracteres de texto. Em seguida, é necessário validar essas regiões em busca de reais candidatos a elementos textuais. Para isso, são obtidos os valores de variância e mediana para as larguras dos componentes conectados e regiões com variância maior do que a metade da mediana são descartadas. A mediana das larguras também é utilizada para verificar se a relação dela com o diâmetro da região não ultrapassa o valor $10$, o que também causa seu descarte. Por fim, são aceitas regiões cuja altura esteja entre $10$ e $300$ pixels e a relação de aspecto entre largura e altura esteja entre $0.1$ e $10$. A conjunção desses passos elimina boa parte de elementos não textuais, como vegetação presente em imagens naturais de zonas rurais e urbanas. Os elementos remanescentes são então adotados como potenciais caracteres de texto [4].

O último passo do método é o agrupamento dos caracteres para definição das palavras e linhas de texto, que geralmente aparecem com orientação e cores semelhantes. Para isso, dois caracteres são unidos caso a largura mediana dos pixels pertencentes a cada um não exceda uma relação igual a $2$. A razão entre a altura desses caracteres também não deve ultrapassar esse mesmo valor, o que torna possível a união entre letras maiúsculas e minúsculas. Para não conectar elementos dispersos, o espaçamento máximo permitido entre eles é o de três vezes a largura do mais largo. Os caracteres são assim agrupados dois a dois até que sejam obtidas as palavras e linhas de texto [4].

No caso da imagem capturada na fase de aquisição apresentar baixa qualidade, a detecção de textos pode não produzir resultados satisfatórios. Para não direcionar recursos computacionais excessivos na tentativa de correção desse tipo de situação, o sistema exibe ao usuário a imagem da cena com as regiões de texto identificadas em destaque e solicita confirmação para prosseguir com o processo de tradução. Assim, é possível cancelar a operação caso as regiões de texto não tenham sido corretamente identificadas.

### *C. Extração de Texto*

Com as potenciais regiões de texto definidas, é necessário converter os caracteres detectados em caracteres de texto que possam ser manipulados pelo sistema. Para isso é utilizado o Tesseract OCR [18], que é desenvolvido em linguagem C/C++ e possui uma adaptação para o sistema Android disponível em [19]. Essa ferramenta utiliza bases de treinamento previamente estabelecidas com referência nos caracteres e palavras encontradas na língua de origem do texto presente na imagem. Para este propósito, foram utilizados arquivos de treinamento das línguas inglesa e portuguesa disponível em [20].

As regiões de texto detectadas passam por uma fase de pré-processamento antes de serem submetidas à ferramenta de OCR. Primeiramente, a imagem é convertida em tons de cinza recebe um filtro de mediana, de máscara com tamanho 3, para remoção de ruídos e melhor definição dos elementos textuais. Posteriormente, as áreas de texto são extraídas da imagem original através da eliminação dos elementos presentes ao redor das áreas retangulares definidas na etapa de detecção de texto.

Após as imagens das regiões de textos serem extraídas, são submetidas a ferramenta de OCR e o resultado de cada uma é armazenado como um rótulo da região que representam na imagem original. Caso esse resultado seja nulo, a região é descartada.

### *D. Tradução de Texto*

Com base nos rótulos definidos para as regiões de texto, a etapa de tradução é realizada através da API MyMemory [21], que possui acesso público e suporte a diversos idiomas.

Antes da operação de tradução, é necessário verificar se os textos resultantes da fase anterior estão corretos, pois erros podem ter sido gerados na fase de OCR. Esta fase é muito suscetível à variações nos caracteres de texto presentes nas imagens capturadas do ambiente, que podem apresentar distorções e elementos estranhos nas proximidades mesmo após a aplicação do filtro na fase anterior.

Para a correção dos textos foi utilizado o corretor ortográfico Enchant [22], que possui suporte à diversas línguas e código fonte aberto. Essa ferramenta fornece uma lista de sugestões baseado na palavra fornecida através de consultas a dicionários locais previamente definidos para a busca. A melhor opção para correção é obtida através do cálculo da similaridade entre cada alternativa sugerida e a palavra originalmente pesquisada. Para obter esse valor foi utilizada a métrica de Levenshtein [23], que calcula a quantidade de edições necessárias entre duas palavras para que uma seja convertida na outra. Por fim, a alternativa que apresentar o menor valor para esse método é admitida como correta. Esse processo é realizado para cada uma das palavras dos textos a fim de obter suas versões corrigidas, que posteriormente são submetidas ao serviço de tradução e o resultado do processo é armazenado em conjunto com os rótulos das regiões definidas anteriormente.

A correção e tradução dos textos são efetuadas por servidores remotos acessíveis via internet. A comunicação entre os elementos envolvidos nesse processo é descrita na Seção IV.

### *E. Apresentação do Resultado*

O resultado obtido pela tradução da imagem original é apresentado na mesma posição do texto extraído, a fim de que seja estabelecida uma relação entre ambos. Para isso, inicialmente é necessário remover os caracteres de texto da imagem original afetando o menor número de objetos não relacionados ao texto. As regiões definidas na etapa de detecção de textos são borradas através do Filtro Gaussiano [17] com máscara de ordem 60 a fim de que os detalhes dos caracteres sejam removidos ao ponto de serem vinculados ao fundo da imagem. Os caracteres traduzidos são então sobrepostos nessa mesma região, respeitando a largura máxima e altura que esta possui. Caso não seja suficiente para representar o texto por completo, este é dividido em partes que são representadas uma após a outra na direção vertical.

O tamanho de fonte utilizado para apresentação do resultado é alterado até que seja possível encaixar todo o texto obtido na região de destino. Porém, em algumas situações a altura desta não é suficiente para representar todo o texto obtido com um tamanho de fonte mínimo. Nesses casos, é inserido um retângulo de cor cinza com transparência para destacar o resultado e ofuscar os elementos presentes no fundo da imagem.

## IV. Arquitetura do Sistema

O sistema desenvolvido foi dividido em uma aplicação local, executada no dispositivo móvel do usuário, e em um serviço remoto, hospedado por um servidor acessível via internet. Essa arquitetura foi desenvolvida a fim de diminuir a quantidade de dados manipulada pela aplicação local e obter acesso a recursos que são disponibilizados de forma *on-line*, como o de tradução [21]. Além disso, torna mais simples as alterações nos mecanismos de correção e tradução de textos, porém pode elevar o tempo de resposta do sistema de forma substancial, visto que a comunicação entre os dispositivos está sujeita a variações de velocidade na conexão com a internet. O serviço em execução no servidor remoto foi desenvolvido em linguagem de programação Python e é acessado através do protocolo *Hypertext Transfer Protocol* (HTTP).

Todas as operações envolvendo o processamento de imagens são realizadas no dispositivo móvel do usuário, que é responsável pelas fases de aquisição, detecção e extração de textos, além da apresentação do resultado. Isso atribui à aplicação local toda carga de processamento gerada pelas imagens envolvidas na metodologia apresentada, porém diminui o tráfego de rede na comunicação com os serviços remotos, pois apenas caracteres de texto precisam ser enviados e recebidos para correção e tradução dos textos obtidos a partir das imagens.

Os elementos envolvidos na arquitetura do sistema são definidos da seguinte forma:

**Dispositivo Móvel:** Executa a aplicação local, responsável por capturar as imagens do ambiente a partir da câmera acoplada ao sistema, detectar as regiões de texto, extrair o conteúdo destas e iniciar a comunicação com o servidor da aplicação para a tradução dos textos. Além disso, processa a imagem capturada para sobrescrever automaticamente o texto original com o resultado obtido;

**Servidor da Aplicação:** Responsável por receber os dados da aplicação em execução no dispositivo móvel, efetuar as correções dos textos submetidos e consultar o servidor de tradução. Além disso, é também responsável por processar estes dados e enviar o resultado da operação de volta para a aplicação local de forma estruturada;

**Servidor de Tradução:** *Application Programming Interface* (API) disponibilizada por MyMemory [21] e utilizada

pelo servidor da aplicação para tradução dos textos corrigidos.

O servidor da aplicação recebe o texto extraído pelo método POST do HTTP, além dos parâmetros de língua de origem e destino através da URL requisitada (método GET). O servidor da aplicação executa as correções necessárias nos textos recebidos e obtém a tradução destes através de consultas à API de tradução. As respostas obtidas desta são padronizadas em formato *JavaScript Object Notation* (JSON), que são recebidas e processadas pelo servidor da aplicação e reenviadas para a aplicação local utilizando esse mesmo formato.

## V. Experimentos

O experimento realizado consistiu de vinte testes com imagens de diferentes locais que pessoas cotidianamente frequentam, como parques e ruas, onde textos são encontrados em avisos e sinalização, sendo algumas apresentadas na Figura 3. Essas imagens são provenientes do trabalho de Shahab et al. [24] e algumas foram obtidas no ambiente de trabalho dos autores por meio da câmera do dispositivo móvel utilizado nos testes. O objetivo é verificar a eficácia da metodologia proposta ao utilizar imagens que apresentam características variadas, como diferenças de iluminação, resolução, fundo e texturas. Além disso, a maior parte delas são largamente empregadas em competições internacionais de identificação e reconhecimento de caracteres, o que fornece um bom parâmetro para verificar a qualidade do sistema.

Para todas as imagens do experimento, a aplicação local foi executada em um *smartphone* Nexus 4, que possui processador de quatro núcleos de 1.5Ghz, 2Gb de memória RAM, câmera de 8MP e Android 4.3. Esse dispositivo foi utilizado por seu *hardware* robusto e principalmente por não sofrer alterações de software por parte da fabricante, o que mantém o sistema operacional em sua versão original.

Com o propósito de avaliar o desempenho do sistema e verificar eventuais gargalos nas diferentes fases do processo de tradução, cada execução deste produz um relatório que contém o tempo decorrido em cada fase de processamento e as respectivas imagens geradas. Dos vinte testes realizados, optou-se por exibir algumas destas imagens a fim de embasar a discussão a seguir. Os tempos obtidos em cada etapa, além do total gasto pelo sistema para produzir o resultado final, podem ser vistos na Tabela I, que segue o fluxo descrito na Figura 1 a partir da segunda fase. A primeira, que corresponde à aquisição da imagem, por apresentar poucos procedimentos envolvidos e baixo tempo de processamento, não foi incluída no relatório de execução do sistema. As etapas descritas na Tabela I são definidas da seguinte forma:

**Etapa 1:** Detecção de Regiões de Texto;
**Etapa 2:** Extração de Texto;
**Etapa 3:** Tradução de Texto;
**Etapa 4:** Apresentação do Resultado.

Tabela I
Tempo (em ms) decorrido em cada etapa do processo de tradução durante os experimentos executados.

| Teste | Etapa 1 | Etapa 2 | Etapa 3 | Etapa 4 | Total |
|---|---|---|---|---|---|
| 1 | 3404 | 11313 | 3154 | 3642 | 21513 |
| 2 | 1852 | 13248 | 6893 | 2833 | 24826 |
| 3 | 393 | 6242 | 3305 | 509 | 10449 |
| 4 | 2190 | 9762 | 8393 | 3811 | 24156 |
| 5 | 938 | 5414 | 2107 | 979 | 9438 |
| 6 | 773 | 21347 | 20213 | 1486 | 43819 |
| 7 | 2571 | 54710 | 7841 | 3093 | 68215 |
| 8 | 2054 | 3863 | 5530 | 2689 | 14136 |
| 9 | 619 | 5174 | 1788 | 1079 | 8660 |
| 10 | 586 | 7261 | 4258 | 1108 | 13213 |
| 11 | 1643 | 3129 | 4607 | 2906 | 12285 |
| 12 | 2830 | 20946 | 18375 | 2682 | 44833 |
| 13 | 844 | 9076 | 13453 | 1188 | 24561 |
| 14 | 571 | 2994 | 1875 | 930 | 6370 |
| 15 | 451 | 3968 | 6400 | 1114 | 11933 |
| 16 | 2581 | 16613 | 6131 | 2721 | 28046 |
| 17 | 3179 | 22763 | 3310 | 2792 | 32044 |
| 18 | 743 | 3259 | 3934 | 900 | 8836 |
| 19 | 533 | 10594 | 6079 | 869 | 18075 |
| 20 | 491 | 3020 | 4800 | 648 | 8959 |
| **Média** | 1462,3 | 11734,8 | 6622,3 | 1898,95 | **21718,35** |

## VI. Conclusões

A partir da análise dos testes realizados, é possível constatar que os resultados obtidos são compatíveis com as necessidades de usuários inseridos em ambientes cotidianos, como ruas, aeroportos e parques.

O tempo médio gasto pelo sistema para produzir o resultado final, embora elevado se comparado a outros sistemas similares, está dentro de um valor aceitável para a fase de desenvolvimento atual do projeto, ficando próximo à 22 segundos. Melhorias podem ser realizadas para que esse valor seja otimizado, porém no atual estágio em que o sistema se encontra já é possível aferir um ganho de produtividade quando analisada sua utilização em situações corriqueiras, como a tradução de placas de sinalização e avisos, dentre outros. Em situações mais simples, em que os textos são apresentados sobre superfícies regulares com fundo homogêneo, o sistema pôde obter o resultado final em aproximadamente 9s.

Outro fator que influenciou no tempo elevado de processamento foi a Etapa 3, cujos processos são todos executados por meio de serviços hospedados em servidores externos. Nos experimentos, o servidor da aplicação foi executado em um *Virtual Private Server* (VPS) localizado nos Estados Unidos da América, o que em geral elevou o tempo de resposta do sistema devido à distância geográfica.

Os próximos passos no desenvolvimento do sistema envolvem a utilização de um sistema de correção e tradução local, executados no próprio dispositivo móvel, e de um analisador semântico. Este último tem como objetivo corrigir eventuais erros provenientes do OCR que são validados pela análise sintática, caso em que determinada palavra é substituída por outra muito parecida, mas que não existia no texto original. Um caso desse tipo pode ser visto no terceiro experimento apresentado nas imagens da Figura 3, em que a palavra “for” foi substituída por “or”, também válida, mas que produz um erro semântico. Outro ponto a ser corrigido é o agrupamento entre linhas de texto, que em alguns casos gerou palavras ou linhas isoladas e causou perda de contexto, o que afetou a

Figura 3. Utilizando imagens provenientes de [24], áreas de texto identificadas à esquerda e o resultado final da tradução à direita.

qualidade da tradução. Um exemplo dessa situação apresentado no segundo experimento da Figura 3, em que as palavras do texto da placa de aviso foram detectadas separadamente, porém sem grande comprometimento do resultado final.

De forma geral, os objetivos do projeto, que eram a identificação e tradução de textos em imagens digitais, foram alcançados e validados através dos testes realizados. Vale ressaltar que a maior parte das imagens utilizadas nos experimentos contém cenas complexas que dificultam a detecção e reconhecimento dos textos. Ao utilizar o sistema em ambientes cotidianos, foi possível averiguar sua utilidade em diversos momentos, principalmente nos quais não era possível dedicar atenção e tempo para fornecer manualmente os textos para tradução, o que cria uma forma de interação muito mais fluida com as informações presentes no ambiente.

## Referências


[1] Google. *Android Web Site*. Disponível em: http://www.android.com/. Acessado em 22 de maio de 2014.

[2] Open Handset Alliance. *Android Overview*. Disponível em: http://www.openhandsetalliance.com/android_overview.html. Acessado em 7 de novembro de 2013.

[3] Yang, J.; Chen, X.; Zhang, J.; Zhang, Y.; Waibel, A., *Automatic detection and translation of text from natural scenes*, Vol. 2. Acoustics, Speech, and Signal Processing (ICASSP), IEEE International Conference, pg. II-2101 - II-2104, Maio de 2002.

[4] Epshtein, B.; Ofek, E.; Wexler, Y. *Detecting Text in Natural Scenes with Stroke Width Transform*. Computer Vision and Pattern Recognition (CVPR), IEEE Conference, pg. 2963 - 2970, Junho de 2010.

[5] Jung, K; Kim, K. I.; Jain, A. K. *Text information extraction in images and video: a survey*. Pattern Recognition, Vol. 37, pg. 977 – 997. Maio de 2004.

[6] Chui, C. *An introduction to wavelets*. Academic Press Professional, Inc. San Diego, CA, USA. Janeiro de 1992.

[7] Gabor, D. *Theory of communication*. Journal of the Institute of Electrical Engineers, Vol. 93, pg. 429 – 441. Novembro de 1946.

[8] Cooley, J. W.; Tukey, J. W. *An algorithm for the machine calculation of complex Fourier series*. Mathematics of Computation, Vol. 19, pg. 297 - 301. 1965.

[9] Minetto, R.; Thome, N.; Cord, M; Leite, N. J.; Stolfi, J. *T-HOG: An effective gradient-based descriptor for single line text regions*. Pattern Recognition, Vol. 46, pg. 1078 – 1090. Janeiro de 2013.

[10] Vapnik, V. *The nature of statistical learning theory*. Springer-Verlag New York, Inc. New York, NY, USA. Janeiro de 1995.

[11] Ohya, J.; Shio, A.; Akamatsu, S. *Recognizing characters in scene images*. Pattern Analysis and Machine Intelligence, IEEE Transactions, Vol. 16, pg. 214 – 220. Fevereiro de 1994.

[12] Lee, C. M.; Kankanhalli, A. *Automatic extraction of characters in complex images*. International Journal of Pattern Recognition and Artificial Intelligence, Vol. 9, pg. 67 – 82. Fevereiro de 1995.

[13] Zhong, Y.; Karu, K.; Jain, A. K. *Locating Text in Complex Color Images*. Document Analysis and Recognition. Proceedings of the Third International Conference, Vol. 1, pg. 146 - 149. Agosto de 1995.

[14] Zhu, M. *Text Detection on Android Devices*. Disponível em: https://github.com/dreamdragon/text-detection. Acessado em 18 de agosto de 2014.

[15] Open Source Robotics Foundation. *Robot Operating System (ROS)*. Disponível em: http://www.ros.org/. Acessado em 18 de agosto de 2014.

[16] Bradski, G.; Kaehler, A. *Learning OpenCV: Computer vision with the OpenCV Library*. O'Reilly. 2008.

[17] Gonzalez, R. C.; Woods, R. E. *Digital image processing*. Prentice Hall. 2008.

[18] Smith, R. *History of the Tesseract OCR engine: what worked and what didn't*, Vol. 8658. Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, Janeiro de 2013.

[19] Theis, R. *Tess-two*. Disponível em: https://github.com/rmtheis/tess-two. Acessado em 17 de novembro de 2013.

[20] Smith, R. *Tesseract Language Data*. Disponível em: https://code.google.com/p/tesseract-ocr/downloads/list. Acessado em 7 de novembro de 2013.

[21] Translated S.r.l. *MyMemory Translation API*. Disponível em: http://mymemory.translated.net/doc/spec.php. Acessado em 7 de novembro de 2013.

[22] Lachowicz, D. *Enchant: Generic Spell Checking Library*. Disponível em: http://www.abisource.com/projects/enchant/. Acessado em 14 de novembro de 2013.

[23] Levenshtein, V. I. *Binary codes capable of correcting deletions, insertions and reversals*. Soviet Physics, Vol. 10, pg. 707 - 710. Fevereiro de 1966.

[24] Shahab, A.; Shafait, F.; Dengel, A. *ICDAR 2011 Robust Reading Competition Challenge 2: Reading Text in Scene Images*. International Conference on Document Analysis and Recognition (ICDAR), pg. 1491 - 1496, Setembro de 2011.

# Effect of Nonrigid Alignment Using Free-Form Deformation in Frontal Face Images

Igor R. R. Xavier
Centro Universitário da FEI
São Bernardo do Campo, Brasil.
iradames@fei.edu.br

Carlos E. Thomaz
Centro Universitário da FEI
São Bernardo do Campo, Brasil.
cet@fei.edu.br

***Abstract*—Face alignment is an important step in any task related to face recognition. Basically, this step is performed applying a transformation that maps points from a face image to the same points in the reference (or target) one. These transformations can be rigid or affine, but due to some specific and local deformations caused by facial expressions, for instance, the application of different alignment techniques must be necessary. In this paper we have used Free-Form Deformation based on B-Splines to align frontal face images with support of a number of basic facial landmarks extended here automatically. Our results have shown that with such approach is possible to create very realistic human-like mean images using only few samples of frontal faces.**

***Keywords*—Nonrigid alignment, Free-Form Deformation, microexpressions, expressions, face recognition.**

## I. Introdução

No contexto de reconhecimento de faces e expressões faciais, uma importante etapa é o alinhamento das imagens com relação a uma face de referência para posterior extração de características e comparações com uma base de conhecimento pré-existente [1], [2]. Em outras palavras, antes de comparar a informação existente em duas faces distintas, deve-se alinhar espacialmente a localização das características de uma imagem em relação às mesmas características da outra.

Na verdade, mesmo duas imagens de face da mesma pessoa podem apresentar variações de pose, escala e expressão, fazendo com que a tarefa de identificação se torne muito mais problemática sem o alinhamento correto dessas amostras. No caso de imagens da mesma pessoa, um alinhamento rígido pode corrigir a variação de pose e um alinhamento afim pode corrigir a variação de escala. Porém, quando trabalhamos com imagens de pessoas distintas ou quando lidamos com expressões faciais, pode haver variações de natureza não-rígida.

Em algumas áreas de pesquisa, a etapa de alinhamento pode ser ainda mais determinante. Por exemplo, na tentativa de detecção e reconhecimento de micro-expressões faciais, o alinhamento precisa ser feito entre muitas imagens, pois uma micro-expressão é uma variação espacial e temporal da mesma face [3] usualmente manifestada sutilmente. Em [4], cada face em uma sequencia de quadros de um vídeo foi alinhada rigidamente às demais utilizando as coordenadas dos centros dos olhos. Em [3], [5] e [6], nos quais experimentos foram conduzidos no intuito de detectar e reconhecer micro-expressões faciais de origem espontânea, as imagens de face foram marcadas com 68 pontos usando o método *Active Shape Model* [7], amplamente conhecido, para depois alinhá-las a uma face de referência aplicando uma função de transformação não-rígida descrita em [8]. Esta última abordagem reforça a necessidade de alinharmos as imagens de forma não-rígida para levar em consideração deformações globais e locais simultaneamente.

Neste trabalho, utilizou-se além de transformações rígidas e afins, uma deformação livre de forma baseada em *B-Splines* e comumente denominada em ingles de *Free-Form Deformation* (FFD) [9]. Esta técnica vem sendo amplamente aplicada com sucesso no contexto de imagens médicas. Rueckert *et al.* [10] utilizaram FFD para alinhar imagens de ressonância magnética de mamas e consequentemente melhorar o contraste e detecção de tumores. Neste caso, a descoberta de tumores pode ser prejudicada pela movimentação do paciente no momento do exame, entre outros fatores. Chandrashekara *et al.* [11] utilizaram *Multilevel* FFD em imagens de ressonância magnética marcadas para construção de modelos 4D de funcionamento do coração. Papatheodorou [12] usou FFD para aprimorar técnicas de reconhecimento facial através do alinhamento e representação de faces em 3D. Mais recentemente, Sandbach *et al.* [13] utilizaram FFD em um sistema de reconhecimento de expressões faciais em 3 dimensões.

Neste artigo, estudou-se o efeito do alinhamento de faces baseado em pontos fiduciais usando FFD e o impacto do aumento do número de pontos na qualidade do alinhamento para poucas amostras disponíveis. A principal contribuição deste estudo está na melhoria da qualidade do registro de imagens frontais de face, sem o uso intensivo de segmentação de imagens, necessária em abordagens holísticas de reconhecimento de padrões [14], [2], [15] e [16]. Portanto, acredita-se que qualquer abordagem que trate uma imagem de face como um vetor de atributos e aplique métodos estatísticos para redução de dimensionalidade e extração de características possa tirar proveito de um alinhamento espacial melhor entre as imagens devido à redução de variabilidade das amostras obtida pelo algoritmo descrito aqui.

O restante do trabalho está dividido da seguinte forma. A seção II descreve sucintamente os métodos de alinhamentos existentes e apresenta o algoritmo proposto. A seção III descreve as imagens utilizadas neste trabalho. A seção IV discute os resultados obtidos e, por fim, a seção V finaliza o artigo com conclusões e possibilidades de trabalhos futuros.

## II. Métodos de Alinhamento

### A. Conceitos Fundamentais

Geralmente, o alinhamento de imagens é realizado aplicando-se uma transformação $\boldsymbol{T}$ que mapeia o ponto $\boldsymbol{a}$ de uma face para o correspondente ponto $\boldsymbol{b}$ de outra face, tal que:

$$\boldsymbol{b} = \boldsymbol{T}(\boldsymbol{a}), \tag{1}$$

onde $\boldsymbol{a} = (a_x, a_y)$ e $\boldsymbol{b} = (b_x, b_y)$ [12]. As transformações podem ser classificadas em dois tipos: aquelas que preservam as proporções geométricas das coordenadas iniciais, portanto, não deformam o objeto transformado, e aquelas que não preservam essas proporções e, por sua vez, deformam o objeto. Transformações rígidas e afins são do tipo que preservam as proporções geométricas. Uma transformação rígida é uma transformação geométrica que quando aplicada a um objeto, por exemplo, uma imagem de face, mantém todas as distâncias entre os pontos do objeto e também seus ângulos internos, sendo uma combinação linear de rotação e translação, tal que $\boldsymbol{b} = \boldsymbol{R}_{AB}\boldsymbol{a} + \boldsymbol{t}_{AB}$, cuja forma matricial é descrita por:

$$\boldsymbol{T}_{rigida}(a_x, a_y) = \begin{pmatrix} b_x \\ b_y \end{pmatrix} = \begin{pmatrix} r_{11} & r_{12} \\ r_{21} & r_{22} \end{pmatrix} \begin{pmatrix} a_x \\ a_y \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix}, \tag{2}$$

onde $\mathbf{R}_{AB} = \{r_{ij}\}, i, j \in \{1, 2\}$, é a matriz de rotação da transformação. Um exemplo de transformação rígida aplicada em uma imagem de face pode ser observado na Figura 1.

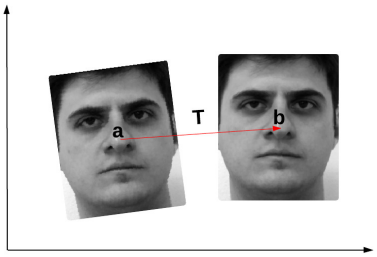



Figura 1: Transformação rígida aplicada a uma imagem do banco de dados de faces da FEI. Percebe-se que houve rotação e translação da face exemplificada pelo deslocamento do ponto a para o ponto b.

Uma classe mais geral de transformações lineares é a transformação afim. Esta além de combinar rotação com translação também promove mudança de escala e recortes no objeto transformado. Sua forma matricial é:

$$\boldsymbol{T}_{afim}(a_x, a_y) = \begin{pmatrix} b_x \\ b_y \end{pmatrix} = \begin{pmatrix} \alpha_{11} & \alpha_{12} \\ \alpha_{21} & \alpha_{22} \end{pmatrix} \begin{pmatrix} a_x \\ a_y \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix}, \tag{3}$$

onde $\mathbf{A}_{AB} = \{\alpha_{ij}\}, i, j \in \{1, 2\}$, é a matriz que descreve os componentes de escala, recorte e rotação da transformação. Um exemplo de transformação afim pode ser visto na Figura 2.

Porém, muitas vezes, o mapeamento entre dois objetos pode não ser linear para todas as coordenadas, ou seja, uma transformação global não é suficiente para capturar a deformação necessária para alinhar um objeto ao outro, como no caso de imagens de face onde as deformações podem

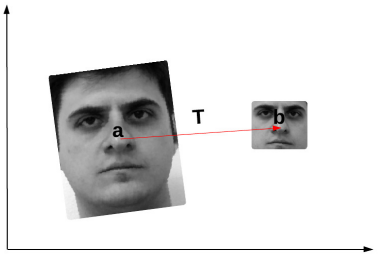



Figura 2: Transformação afim aplicada a uma imagem do banco de dados de faces da FEI. Percebe-se que além de rotação e translação da face, houve também mudança de escala e recorte da região central da face.

ocorrer global e localmente. Tais transformações são ditas não-afins, não-lineares ou não-rígidas. Na prática, esse tipo de transformação é definida por pontos de controle ou marcações no objeto e a deformação é suavemente interpolada por pontos intermediários [17]. Transformações baseadas em *Splines* atuam nos níveis local e global. Estas foram originalmente concebidas para modelagem computacional, projeto de aviões, carros ou em atividades onde os projetistas usavam tiras de metal ou madeira com ajuda de pesos para deformação [18]. Um esquema visual de transformação não-rígida pode ser observado na Figura 3.

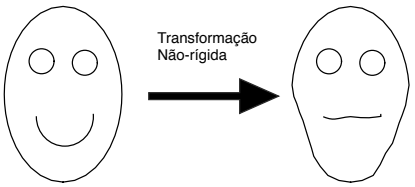



Figura 3: Transformação não-rígida aplicada a uma face pode deformar uma face global e localmente.

### B. Algoritmo Proposto

O diagrama de blocos do algoritmo proposto neste trabalho pode ser visto na Figura 4. O algoritmo consiste dos seguintes passos:

1) Estimar a transformação rígida necessária para alinhar os pontos da imagem de face k aos pontos da imagem padrão com base apenas em um conjunto de pontos fiduciais previamente computados;
2) Estimar a transformação afim necessária para alinhar os pontos da imagem de face k aos pontos da imagem padrão com base apenas em um conjunto de pontos fiduciais previamente computados e o resultado do alinhamento rígido;
3) Estimar a transformação não-rígida necessária para alinhar os pontos de face k aos pontos da imagem padrão com base apenas em um conjunto de pontos fiduciais e o resultado do alinhamento afim;

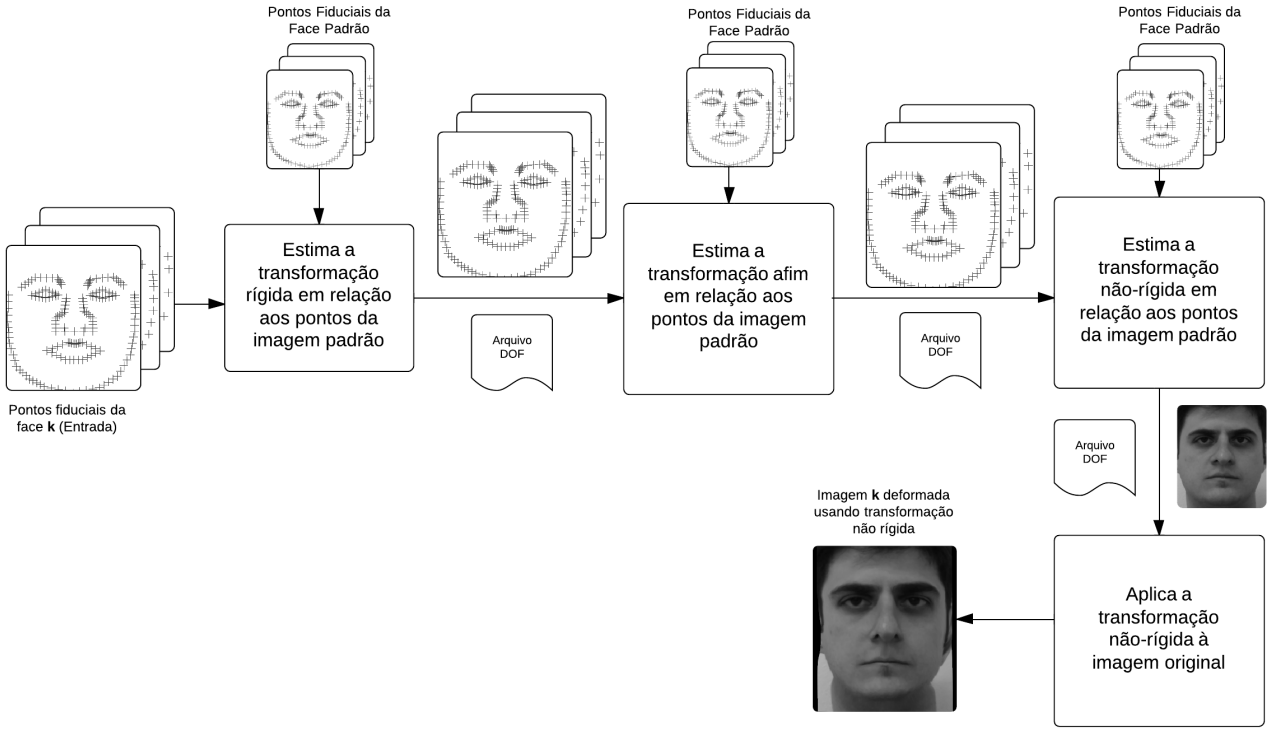



Figura 4: Diagrama de blocos do algoritmo proposto.

4) Computar a imagem de face k aplicando a transformação não-rígida com base na face k original e o resultado do alinhamento não-rígido.

O arcabouço computacional descrito na Figura 4 é aplicado para diversas combinações de pontos fiduciais. A partir de um conjunto inicial de $m$ pontos, utilizou-se o algoritmo 1 para computar um ponto médio entre cada par de pontos sequenciais, gerando como saída um novo conjunto de $n$ pontos, onde $n \geq m$ e $\varepsilon$ é a máxima distância que dois pontos podem ter para serem interpolados, determinada empiricamente neste trabalho.

**Algorithm 1** InterporlarPontos(m)

$k \leftarrow 1$
**for** $i = 1$ **to** $comprimento[m] - 1$ **do**
  **if** $0 < dist(m(i), m(i+1)) < \varepsilon$ **then**
    $n(k) \leftarrow m(i)$
    $n(k+1) \leftarrow [m(i) + m(i+1)]/2$
    $n(k+2) \leftarrow m(i+1)$
    $k \leftarrow k + 2$
  **end if**
**end for**
**return** $n$

## III. Material

Para este estudo, escolheu-se aleatoriamente 15 imagens de face de 250x300 pixels, neutras e frontais, do banco de dados de faces da FEI [19], conforme Figura 5. Além disso, foram utilizados como pontos fiduciais iniciais as $m = 46$ marcações feitas no trabalho [20] no qual foi utilizado o mesmo banco de dados. A face média das imagens neutras do banco de dados da FEI também foi anotada com os mesmos 46 pontos fiduciais com intuito de ser utilizada como face padrão (referência), conforme Figura 6.

## IV. Experimento e Resultados

Para efeito de comparação, calculou-se a média das 15 imagens selecionadas, as quais já estavam alinhadas no banco de dados em relação somente às coordenadas dos centros dos olhos de cada face. O resultado pode ser visto na Figura 7.

O experimento foi dividido em três etapas, cada uma com um determinado número de pontos fiduciais anotados. Aproveitou-se a marcação feita previamente no banco de dados [20] e a partir dela aumentou-se o número de pontos para as etapas seguintes. Em cada etapa, aplicou-se o algoritmo da Figura 4 nas 15 imagens selecionadas e calculou-se a média das faces para comparação.

Utilizando as imagens anotadas, aplicou-se a transformação não-rígida. Essa transformação deforma as imagens originais na tentativa de alinhar os pontos fiduciais de cada imagem aos pontos da imagem de referência. Um resultado desta transformação pode ser visto na Figura 8 usando 46 pontos fiduciais. A fim de melhorar o resultado do alinhamento, interpolaram-se os 46 pontos iniciais utilizando o algoritmo

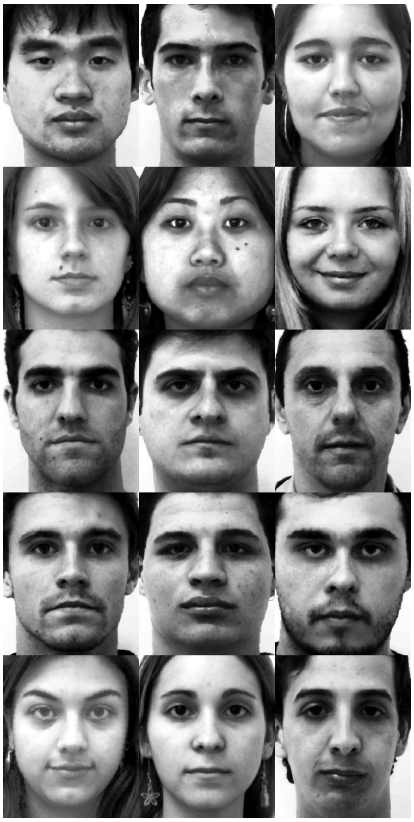


Figura 5: As 15 imagens de faces selecionadas aleatoriamente do banco de dados da FEI.

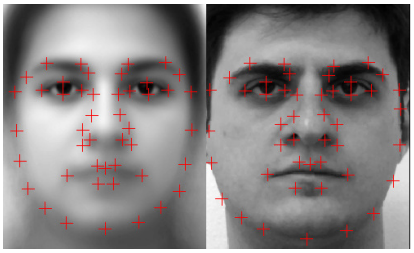


Figura 6: Faces originais do banco de dados da FEI marcadas com 46 pontos fiduciais. A imagem da esquerda mostra a face padrão (face média de todos os individuos da base) marcada com os 46 pontos e a imagem da direita mostra uma das faces do banco de dados marcada com os 46 pontos correspondentes aos da imagem padrão.

1 proposto. Os resultados da primeira e segunda rodadas de inclusão de novos pontos podem ser vistos na Figura 9.

O processo automático de inclusão de novos pontos criou duas novas máscaras com 86 e 166 pontos. Aplicou-se a estas novas configurações o algoritmo de alinhamento não-rígido e o resultado pode ser observado na Figura 10.

Por fim, calculou-se a imagem média dos resultados do alinhamento não-rígido das 3 configurações de pontos: 46, 86 e 166 pontos, conforme a Figura 11.

Figura 7: Imagem média resultante do alinhamento rígido das faces selecionadas aleatoriamente do banco de dados da FEI.

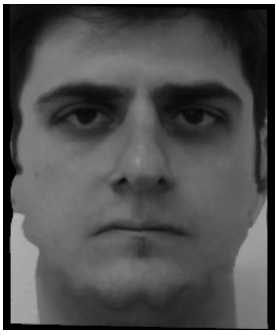


Figura 8: Resultado de uma transformação não-rígida usando 46 pontos fiduciais como entrada para o algoritmo proposto.

Pode-se observar comparando, especificamente, as Figuras 8 e 10 que o aumento na quantidade de pontos fiduciais de entrada para o algoritmo de alinhamento não-rígido fez com que a rugosidade do resultado final diminuísse consideravelmente. Isto pode ser observado tanto nas imagens de exemplo quanto na média da amostra, onde a rugosidade das amostras permanecia clara nas faces médias correspondentes. Percebe-se também, pela Figura 11, que o uso do alinhamento não-rígido proporcionou uma melhora significativa na visualização das características médias das amostras. Além dos olhos, a boca, o nariz e as sobrancelhas puderam ser alinhados de forma bastante realista em comparação com a face média do alinhamento rígido (Figura 7).

Como medida da qualidade do alinhamento, calculou-se o desvio padrão médio de cada pixel em cada conjunto de imagens, ou seja, para o conjunto original sem a aplicação do algoritmo proposto neste trabalho e para os conjuntos de imagens transformados não-rigidamente com os acréscimos automáticos das quantidades de pontos fiduciais. Conforme pode-se observar na Figura 12, o alinhamento não-rígido diminuiu a variância em regiões da face onde os estudos, principalmente sobre expressões e micro-expressões faciais, mais se concentram, por exemplo.

Como medida quantitativa, calculou-se a decomposição espectral da matriz de covariância de cada conjunto de imagens alinhadas. A Figura 13 mostra a magnitude desses autovalores. Esta análise de componentes principais foi aplicada à região de interesse definida pela envoltória convexa dos pontos fiduciais

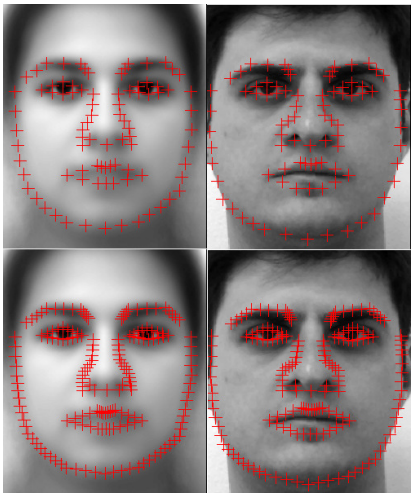


Figura 9: Resultado da inclusão automática de novos pontos na marcação inicial de 46 pontos. Acima e à esquerda: a face média do banco de dados da FEI marcada com 86 pontos fiduciais, acima e à direita: uma das faces selecionadas marcada com 86 pontos, abaixo e à esquerda: a face média do banco de dados da FEI marcada com 166 pontos fiduciais e abaixo e à direita: uma das faces selecionadas marcada com 166 pontos.

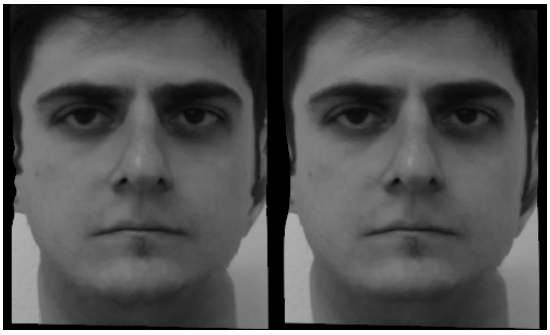


Figura 10: Resultado do alinhamento não-rígido utilizando as máscaras com 86 pontos (esquerda) e 166 pontos (direita) calculados automaticamente pelo algoritmo proposto.

anotados em cada grupo de imagens.

Calculou-se também, o coeficiente de correlação médio de cada grupo de imagens definido pela região de interesse, conforme tabela I.

| Tipo de Alinhamento | CC |
|---|---|
| Rígido | 0.83 |
| Não-rígido 46 pontos | 0.96 |
| Não-rígido 86 pontos | 0.97 |
| Não-rígido 166 pontos | 0.96 |

Tabela I: Coeficiente de correlação médio obtido entre todas as imagens de cada grupo.

Figura 11: Médias de imagens de face pós alinhamento não-rígido com 46 (esquerda), 86 (centro) e 166 (direita) pontos respectivamente.

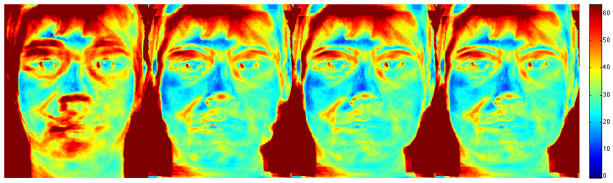


Figura 12: Desvio padrão médio de cada pixel em cada grupo de imagens gerado. Da esquerda para direita: faces alinhadas pela transformação afim usando os centros dos olhos, faces alinhadas pelo método não-rígido usando 46, 86 e 166 pontos fiduciais, respectivamente.

## V. Conclusão

Neste trabalho, explorou-se uma técnica de alinhamento de faces utilizando um algoritmo para alinhamento não-rígido com o objetivo de diminuir os efeitos de problemas de escala e deformação, e até mesmo de morfologia, entre as amostras de um banco de dados de face disponível publicamente.

Os resultados observados no cálculo das faces médias indicam que o aumento automático proposto de pontos fiduciais melhora o alinhamento das características da face e que este aumento pode ser conseguido partindo-se de uma marcação básica e incrementando o número de pontos por meio de uma simples interpolação de pares sequenciais com seus respectivos pontos médios. Adicionalmente, nota-se que o alinhamento espacial foi feito utilizando somente 15 imagens de face, ou seja, mesmo com poucas amostras a média das faces transformadas apresenta características de uma face humana extremamente realista. Este resultado pode ser comparado ao de outras técnicas de síntese e transformação de faces como em [21], porém com a vantagem de utilizar, por meio do algoritmo proposto, poucos pontos base fiduciais.

Como trabalhos futuros, acredita-se que seria interessante verificar a combinação desta técnica com métodos automáticos de marcação de pontos fiduciais como em [22] e extrapolar essa marcação inicial utilizando, possivelmente como informação *a priori*, medidas antropométricas da face.

## Agradecimentos

Os autores gostariam de agradecer o suporte financeiro provido pelo programa PROSUP/CAPES e também ao Dr. Daniel Rueckert por disponibilizar as ferramentas utilizadas neste trabalho para o alinhamento das faces. Estas ferramentas se encontram disponíveis em http://www.doc.ic.ac.uk/~dr/software/.

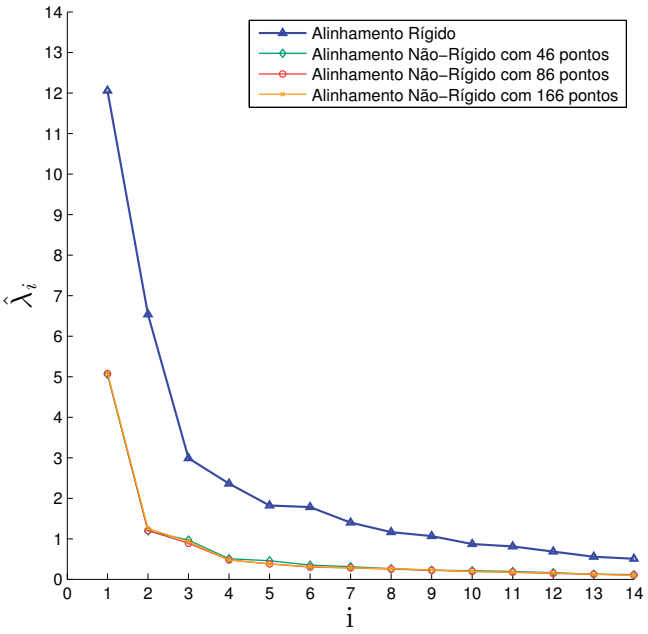



Figura 13: Os $n$ autovalores não nulos das matrizes de covariância, obtidos através da decomposiçao espetral de cada conjunto de imagens.

## REFERÊNCIAS


[1] T. Ahonen, A. Hadid, and M. Pietikäinen, "Face recognition with local binary patterns," in *Computer vision-eccv 2004*. Springer, 2004, pp. 469–481.

[2] M. Kirby and L. Sirovich, "Application of the Karhunen-Loeve procedure for the characterization of human faces," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 12, no. 1, pp. 103–108, 1990.

[3] T. Pfister, X. Li, G. Zhao, and M. Pietikainen, "Recognising spontaneous facial micro-expressions," in *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE, 2011, pp. 1449–1456.

[4] M. Shreve, S. Godavarthy, D. Goldgof, and S. Sarkar, "Macro-and micro-expression spotting in long videos using spatio-temporal strain," in *Automatic Face & Gesture Recognition and Workshops (FG 2011), 2011 IEEE International Conference on*. IEEE, 2011, pp. 51–56.

[5] J. A. Ruiz-Hernandez and M. Pietikainen, "Encoding local binary patterns using the re-parametrization of the second order gaussian jet," in *Automatic Face and Gesture Recognition (FG), 2013 10th IEEE International Conference and Workshops on*. IEEE, 2013, pp. 1–6.

[6] W.-J. Yan, X. Li, S.-J. Wang, G. Zhao, Y.-J. Liu, Y.-H. Chen, and X. Fu, "Casme ii: An improved spontaneous micro-expression database and the baseline evaluation," *PloS one*, vol. 9, no. 1, p. e86041, 2014.

[7] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham, "Active shape models-their training and application," *Computer vision and image understanding*, vol. 61, no. 1, pp. 38–59, 1995.

[8] A. Goshtasby, "Image registration by local approximation methods," *Image and Vision Computing*, vol. 6, no. 4, pp. 255–261, 1988.

[9] T. W. Sederberg and S. R. Parry, "Free-form deformation of solid geometric models," in *ACM Siggraph Computer Graphics*, vol. 20, no. 4. ACM, 1986, pp. 151–160.

[10] D. Rueckert, L. I. Sonoda, C. Hayes, D. L. Hill, M. O. Leach, and D. J. Hawkes, "Nonrigid registration using free-form deformations: application to breast MR images," *Medical Imaging, IEEE Transactions on*, vol. 18, no. 8, pp. 712–721, 1999.

[11] R. Chandrashekara, R. H. Mohiaddin, and D. Rueckert, "Analysis of 3-D myocardial motion in tagged MR images using nonrigid image registration," *Medical Imaging, IEEE Transactions on*, vol. 23, no. 10, pp. 1245–1250, 2004.

[12] T. Papatheodorou, "3D Face Recognition Using Rigid and Non-Rigid Surface Registration," Ph.D. dissertation, University of London, 2006.

[13] G. Sandbach, S. Zafeiriou, M. Pantic, and D. Rueckert, "Recognition of 3D facial expression dynamics," *Image and Vision Computing*, vol. 30, no. 10, pp. 762–773, 2012.

[14] M. Turk and A. Pentland, "Eigenfaces for recognition," *Journal of cognitive neuroscience*, vol. 3, no. 1, pp. 71–86, 1991.

[15] B. Moghaddam and A. Pentland, "Probabilistic visual learning for object representation," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 19, no. 7, pp. 696–710, 1997.

[16] W. Zhao, A. Krishnaswamy, R. Chellappa, D. L. Swets, and J. Weng, "Discriminant analysis of principal components for face recognition," in *Face Recognition*. Springer, 1998, pp. 73–85.

[17] S. Lee, G. Wolberg, and S. Y. Shin, "Scattered data interpolation with multilevel B-splines," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 3, no. 3, pp. 228–244, 1997.

[18] R. H. Bartels, J. C. Beatty, and B. A. Barsky, *An introduction to splines for use in computer graphics and geometric modeling*. Morgan Kaufmann, 1987.

[19] C. E. Thomaz and G. A. Giraldi, "A new ranking method for principal components analysis and its application to face image analysis," *Image and Vision Computing*, vol. 28, no. 6, pp. 902–913, 2010.

[20] E. Z. Tenorio and C. E. Thomaz, "Análise Multilinear Discriminante de Formas Frontais de Imagens 2D de Face," *Proceedings of the X Simposio Brasileiro de Automacao Inteligente SBAI 2011*, pp. 266–271, September 2011.

[21] B. Tiddeman, M. Stirrat, and D. I. Perrett, "Towards realism in facial image transformation: Results of a wavelet mrf method," in *Computer Graphics Forum*, vol. 24, no. 3. Wiley Online Library, 2005, pp. 449–456.

[22] M. Valstar, B. Martinez, X. Binefa, and M. Pantic, "Facial point detection using boosted regression and graph models," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010, pp. 2729–2736.

# Augmented Tattoo:
# *a proposal of tattoo visualization in augmented reality*

Jairo Calmon
Mestrado em Ciência da Computação
Universidade Estadual de Feira de Santana – Bahia - Brazil
jairocalmon@ecomp.uefs.br

João Queiroz
Instituto de Artes e Design
Universidade Federal de Juiz de Fora
Juiz de Fora – Minas Gerais - Brazil
queirozj@pq.cnpq.br

Claudio Goes, Angelo Loula
Departamento de Ciências Exatas
Universidade Estadual de Feira de Santana – Bahia – Brazil
cegoes@gmail.com
angelocl@ecomp.uefs.br

***Abstract*—Technologies that aim user immersion in a synthetic environment have been used for various purposes. One such technology is Augmented Reality (AR) that allows the user to view the surrounding environment with overlap and composition of virtual objects, creating new ways of computer interaction. One of the objects for such virtual interactions is the human body. The human body is also an element that undergoes real physical changes, a common cultural practice for centuries, and one of the ways of expression through body modification is tattoo. Here we describe an AR system for visualization of tattoos on skin surface, following the body surface deformations. The system uses a few small markers to obtain a 2D model of skin surface. Results show that the system is able to overlay an augmented tattoo on the skin even with distortions and occlusions of markers.**

***Keywords—augmented reality; tattoo;***

## I. INTRODUÇÃO

Tecnologias gráficas desenvolvidas para imergir o usuário em ambientes sintéticos, de diferentes formas e em diferentes níveis, têm sido utilizadas para as mais diversas finalidades, como, medicina, educação, terapia, entretenimento, robótica, entre outros [1]. A Realidade Aumentada (RA) permite ao usuário a visualização do mundo real a sua volta, através da sobreposição e composição de objetos virtuais modelados por computador. Aplicações de RA complementam a realidade, ao invés de tentar substituí-la [1], a partir da visualização em tempo real de um mundo real físico que foi "melhorado" através da adição de informação gerada por computador [2].

As interações entre o mundo real e o virtual, em aplicações de Realidade Aumentada são praticadas, na maioria das vezes, de forma não intrusiva. Existem, por outro lado, antigas práticas culturais, exercidas há muitos séculos, de intervenção em camadas epiteliais do corpo humano [3]. São muitos os tipos de intervenções praticadas no corpo, incluindo as mais populares e recentes, como a tatuagem e o *piercing*, e também são diversas as motivações para tais práticas. Tal popularidade é confirmada através de dados que estimam que 23% dos americanos e 21% dos canadenses possuem pelo menos uma tatuagem. Estima-se ainda que estas pessoas consomem em tatuagens, somente nos Estados Unidos, a quantia de 1.65 bilhão de dólares por ano [4].

Este artigo descreve um sistema de Realidade Aumentada para visualização de tatuagens virtuais na pele, que acompanha a superfície do corpo humano. É, portanto, uma forma de exploração de novas interfaces aliada à motivação dos indivíduos para modificarem seus corpos. Esta nova forma de tatuagem – virtual – permite abordagens dinâmicas, e mesmo iterativas e multimodais, através de animações e vídeos, além da possibilidade do usuário trocá-la quando desejar.

A próxima seção apresenta uma discussão sobre tatuagem, tecnologias envolvidas e trabalhos relacionados. A seção III descreve a metodologia utilizada, descrevendo a especificação e restrições do projeto do sistema e também as etapas da aplicação proposta. A seção IV apresenta os resultados obtidos, e, por fim, as considerações finais são apresentadas.

## II. TECNOLOGIA E TATUAGEM

Intervenções físicas na superfície do corpo constituem práticas observadas em diversas culturas e são motivadas por várias razões. Na última década, a popularidade das tatuagens e *piercings* aumentou dramaticamente, tanto em termos numéricos quanto em relação ao espectro de classes sociais de seus usuários [3]. As motivações para realizar intervenções corporais por meio de tatuagem são diversas. Um estudo descrito em [3] definiu algumas categorias motivacionais que descrevem os impulsos dos indivíduos para modificar o corpo: beleza, arte e moda; narrativa pessoal; tolerância física; afiliação a grupos e comprometimento; resistência; espiritualidade e tradição cultural; vício; e motivação sexual.

De forma simplificada, uma tatuagem é uma inscrição invasiva na segunda camada da pele (derme), através de repetidas micro-incisões. Portanto, abordagens no desenvolvimento de tatuagens devem ter como base o seu suporte, a pele. Após um longo período sem mudanças significativas de concepção e aplicação tecnológicas, novos aparatos tecno-científicos começaram a emergir, permitindo, por exemplo, o uso de dispositivos epiteliais dinâmicos, isto é, artefatos intencionalmente inscritos ou anexados à pele nos

quais a aparência é potencialmente modificada no tempo [5]. Estes projetos estão associados com a pesquisa de novos dispositivos biotecnológicos, bem como a síntese de novos materiais e aplicações tecno-científicas. Algumas dessas novas tecnologias permitem, por exemplo, a leitura de variações de estados orgânicos (por exemplo, endócrino) criando novas formas de relações interpessoais e novos padrões de comunicação *(embodied communication)*.

### A. Realidade Aumentada e Tatuagem

Realidade Aumentada (RA) é definida como a visualização em tempo real direta ou indireta de um mundo físico que foi "ampliado" através da adição de informação gerada por computador [2]. As aplicações práticas decorrentes da tecnologia são diversas, uma vez que os objetos virtuais adicionam ao mundo real informações que o usuário não pode perceber diretamente. Assim, as informações transmitidas pelos objetos virtuais podem ajudar o usuário a executar tarefas muito diversas, entre as quais pode-se citar: visualização médica, entretenimento, publicidade, manutenção e reparo, anotação, planejamento de rota, entre outros [2].

Trabalhos acadêmicos na área de RA envolvendo a visualização de tatuagens através da detecção da superfície da pele não foram encontrados nas buscas do tema. Mas alguns projetos comerciais e aplicativos para dispositivos móveis foram encontrados. Nestes casos é tatuado um marcador, ou a própria tatuagem do usuário é utilizada; então a aplicação detecta e sobrepõe um objeto animado virtual, como o projeto ThinkAnApp[1]. Tais projetos são similares aos de detecção realizados em cartões, a diferença sendo apenas relativa a natureza da superfície em uso. Assim, o objeto virtual é apenas posicionado de modo a flutuar sobre a pele, sem qualquer deformação ou integração com a superfície.

O uso de projeções em realidade aumentada sobre a pele pode ser observado em muitos campos de pesquisa como, por exemplo, em medicina. Nicolau et al. [6], por exemplo, apresentam um sistema de RA para auxílio em procedimentos médicos no fígado. É possível identificar similaridades importantes entre esta aplicação e nosso trabalho. Em contraste, uma considerável diferença encontra-se no fato da aplicação médica basear-se em um ambiente bastante controlado e fazer uso de dispositivos de captura sofisticados, caros e de uso restrito, como a tomografia computadorizada.

## III. METODOLOGIA

Enquanto muitos trabalhos utilizam a pele apenas como a superfície que porta um cartão a ser detectado para a aplicação de RA, propomos uma abordagem capaz de identificar as deformações da pele, de modo que uma tatuagem virtual se ajusta para acompanhar tais deformações.

No projeto do sistema, diversas restrições e especificações foram consideradas. A aplicação resultante pode ser utilizada por diversos usuários em ambientes não controlados ou preparados. Deve-se permitir visualização em tempo real, utilizando imagens capturadas por uma câmera monocular, executada em dispositivos móveis e posteriormente em dispositivos vestíveis. Para reconhecer as deformações da pele, é necessário analisar o conteúdo da imagem em busca de informações que permitam extrair a pose de determinadas superfícies bem como possíveis deformações. A detecção da superfície da pele neste trabalho foi obtida, então, através do registro 2D proveniente da detecção e identificação de marcadores *adhoc* adicionados à pele. Apesar desta estratégia exigir a modificação da pele (através da inscrição do marcador), esta foi considerada a mais adequada pelo baixo custo computacional de análise da imagem, preservação da textura da pele, tolerância em relação a oclusões e facilidade de remoção da cena.

O atual desenvolvimento da aplicação é realizado em um ambiente *desktop*, facilitando a realização de testes e experimentos. Foram utilizadas as bibliotecas de visão computacional OpenCV, juntamente com as bibliotecas gráficas OpenGL.

As várias etapas de processamento do sistema proposto, estão representadas no diagrama da Fig. 1. O processo tem inicio com a aquisição da imagem, conversão de espaço de cores e obtenção da máscara de segmentação de pele. Em seguida o processo de detecção de marcadores é realizado através dos processos de detecção de bordas e de componentes conexos, a partir dos quais é possível identificar (ou presumir) o posicionamento de cada um dos marcadores. Com a informação da posição de cada marcador, é construída uma malha com a textura da tatuagem escolhida em uma imagem, na qual será aplicada a máscara de segmentação de pele de modo a tornar invisível as partes que não estão sobre a pele. Os marcadores são removidos da imagem original e então a tatuagem pode ser sobreposta. Por fim, a imagem composta é exibida. As seções a seguir descrevem em maiores detalhes cada uma das etapas.

### A. Segmentação de Pele

Nessa etapa, a intenção é separar a pele de todo o ambiente circundante, para que a identificação dos marcadores presentes na pele seja facilitada, eliminando possíveis *outliers* presentes na imagem original (Fig. 3-a) que dificultariam o processo. O resultado da segmentação também é utilizado na sobreposição da tatuagem como máscara que impede a tatuagem de ser exibida em partes que não correspondam à pele.

O resultado desejado dessa etapa é uma máscara binária que informa se determinado pixel é considerado pele ou não (Fig. 3-c). Para realizar tal segmentação de pele foi utilizado o espaço de cor HSV, sendo dada a opção ao usuário de selecionar uma pequena área circular na imagem na tela, como amostra representativa de sua pele, como forma de calibração da segmentação (Fig. 3-b). A partir dessa informação de amostra de pele, é calculada a média (M) e desvios padrões ($\sigma$) dos canais de matiz (H) e saturação (S), suficientes para realização satisfatória da segmentação de pele.

A segmentação é obtida automaticamente pela seleção dos pixels que possuem valores entre a média e uma margem de tolerância de 4 desvios-padrão para os canais H e S. Para os casos nos quais o usuário seleciona uma área extremamente

[1] Veja notícia Dybwad [7], B. Augmented Reality Tattoo Makes Your Skin Come Alive, FEB 17, 2010. <http://mashable.com/2010/02/17/augmented-reality-tattoo/

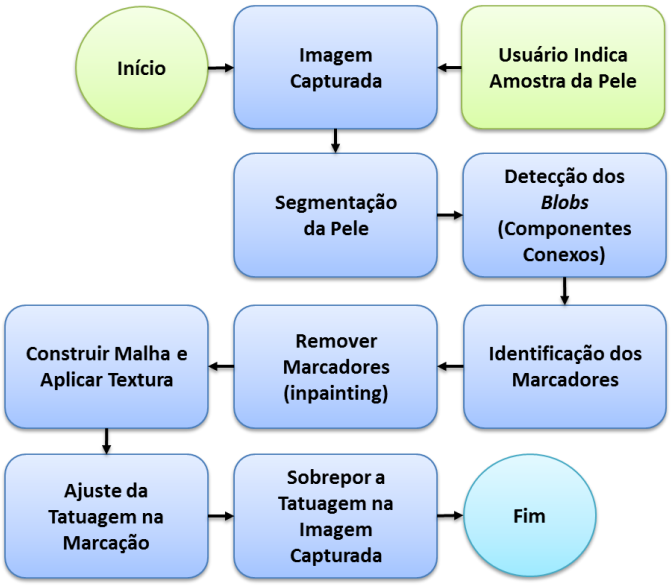



Fig. 1. Diagrama com fluxo de ações executadas neste sistema.

uniforme, com pouca variabilidade nos canais H e S, o desvio padrão terá pouca variação, e ao invés de serem utilizados quatro desvios-padrão como margem de tolerância, um valor mínimo predeterminado de margem por canal é utilizado neste caso.

Nessa etapa também são aplicados os operadores de dilatação e erosão [8] para eliminar pequenos ruídos. Estes procedimentos são realizados através da convolução de um elemento estruturante na imagem e foram úteis para refinar os resultados obtidos. Filtros de média e desfoque também foram úteis para eliminar pequenos ruídos e tornar os contornos da imagem de máscara mais suave (Fig. 3-c).

### B. *Marcadores e Detecção*

Os marcadores são utilizados para obter informações sobre a forma da superfície na qual foram aplicados. Para o sistema proposto, foi utilizada uma grade regular de marcadores de pequeno tamanho (Fig. 2), uma vez que são marcadores simples de serem inseridos na pele (utilizando uma caneta, por exemplo) e ocupam pouca área em relação à superfície na qual é sobreposta a tatuagem, facilitando o processo de eliminação deste marcador na composição final. Além disso, a detecção é relativamente simples e permite tratar oclusões ou falhas na detecção. Além do conjunto regular de marcadores circulares são usados marcadores especiais, de forma a se obter a orientação dos marcadores e possibilitar a obtenção do modelo da superfície mesmo com oclusões parciais de alguns marcadores.

O processo de detecção destes marcadores tem início através da detecção de bordas da imagem, utilizando uma versão monocromática da imagem capturada e máscara de segmentação de pele já aplicada, para eliminar possíveis *ouliers* (Fig. 3-d). Este processo é realizado utilizando o detector de bordas de Canny [9] que consegue identificar variações bruscas entre os pixels vizinhos (Fig. 3-e).

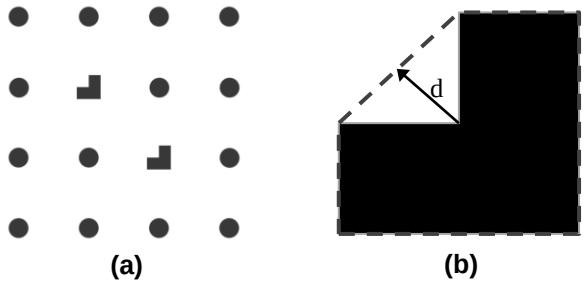



Fig. 2. (a) Conjunto de marcadores utilizados. Note a presença de dois marcadores especiais no centro. (b) Representação do marcador especial em formato de "L" (sólido) juntamente com seu *convex hull* (tracejado) e a representação em forma de vetor "d" do chamado "defeito de convexidade".

Em seguida são aplicados alguns operadores morfológicos na (Fig. 3-f) e então um algoritmo de detecção de *blobs* [10] é executado.

### C. *Identificação e Registro de Marcadores*

A etapa anterior obtém como resultado um conjunto de *blobs* com possíveis *outliers*. No entanto, para identificar o conjunto de marcadores desejados são verificadas as relações de distância e área dos *blobs*, de modo a achar o agrupamento que tem mais possibilidade de ser um conjunto de marcadores de interesse, eliminando eventuais outros *blobs*. Como resultado temos o conjunto de *blobs* que representam o conjunto de marcadores vísiveis.

A partir dos marcadores detectados, são identificados os dois marcadores especiais. Para isso, verificamos os marcadores que possuem a menor solidez (*solidity*), correspondente à área do *blob* sobre a área do seu *convex hull* [11]. Como os marcadores normais são círculos, a solidez tende a ser unitária, uma vez que as áreas são coincidentes. Os marcadores especiais possuem um formato de "L", deixando seu *convex hull* com área significativamente maior, e consequentemente seu valor de solidez menor e facilmente identificado.

Uma vez identificados os dois marcadores especiais em formato de L, podemos obter sua orientação através da análise dos defeitos do *convex hull*. Porém, ainda não é possível distinguir um marcador especial do outro. Para isso, é realizada a comparação entre o vetor dos defeitos e o vetor entre os marcadores obtidos.

Com a informação dos dois marcadores especiais e a orientação do conjunto de marcadores, é possível associar cada *blob*-marcador com sua respectiva posição relativa na grade (Fig. 3-h). Para isso, é presumida uma possível localização na imagem do marcador, baseada nas posições dos marcadores já identificados (utilizando interpolação/ extrapolação linear), do centro da grade para as bordas. A partir desta localização presumida, procura-se qual marcador encontra-se mais próximo deste local, com distância não maior que a metade da menor distância possível entre os *blobs*.

Nos casos onde não há oclusão dos marcadores, espera-se que a busca do marcador no raio informado seja bem sucedida, de forma que podemos atribuir a correspondência entre o

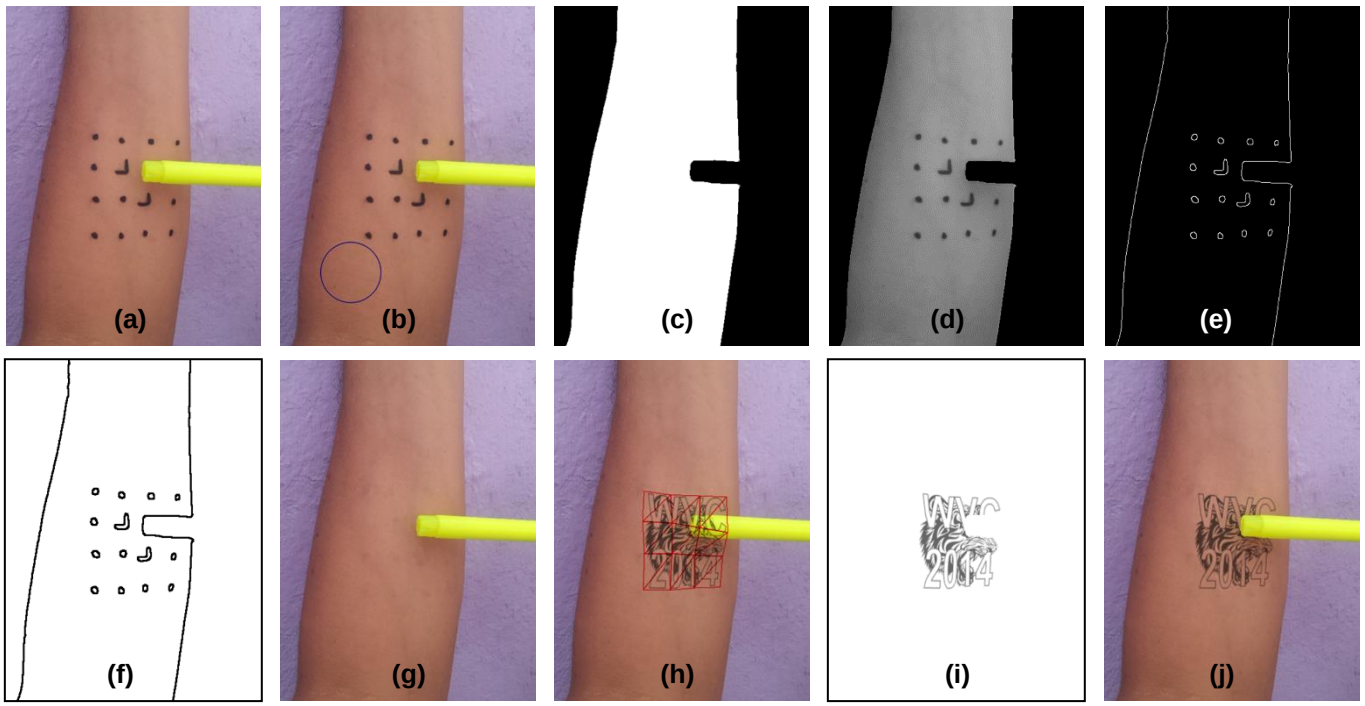



Fig. 3. Diagrama do fluxo de ações executadas pelo programa. (a) Imagem original de um braço com marcadores capturada pela câmera. (b) Usuário calibra o sistema através da seleção de uma amostra de pele. (c) Segmentação de pele utilizando o valores de média e desvios-padrão dos canais matiz e saturação obtidos no processo de calibração. (d) Máscara de segmentação aplicada à imagem monocromática. (e) Detecção de bordas através do filtro de Canny. (f) Aplicação de filtros morfológicos sobre a imagem de detecção de bordas. (g) Processo de remoção de marcadores através da técnica *inpainting*. (h) Deformação aplicada na tatuagem de acordo com a posição dos marcadores (o *wireframe* é mostrado para evidenciar a deformação). (i) Corte da tatuagem de acordo com a máscara de segmentação de pele. (j) Superposição da imagem da tatuagem.

marcador identificado e a sua posição relativa na grade. Porém, nos casos de oclusão, a busca em tal raio pode não encontrar o marcador procurado, de forma que a informação de posição presumida é utilizada para representar tal *blob*. Isso permite a construção da malha que modela a superfície da pele mesmo sem a identificação de todos marcadores. Dessa forma, obtêm-se o registro da grade completa, ainda que extrapolando a área demarcada como pele pelo processo de segmentação. Ressalta-se que a aplicação requer que pelo menos os dois marcadores especiais estejam visíveis.

### *D. Construção da malha*

Para obter um modelo 2D da superfície da pele, construímos uma malha de triângulos utilizando OpenGL utilizando a posição dos marcadores (identificados e previstos) como vértices da malha. Este processo consiste em relacionar, a cada frame, a posição de cada vértice às coordenadas do centro de massa de seu respectivo marcador.

No processo de construção da malha, o mapeamento UV (representação 2D do mapeamento da superfície 3D) é realizado e a textura da tatuagem, escolhida pelo usuário, é atribuída à malha. Dessa forma, temos uma malha em OpenGL que será renderizada para obtenção da tatuagem com as mesmas deformações da superfície da pele.

A renderização deste processo é feita separadamente, não sobrepondo diretamente à imagem original, gerando uma imagem a parte, uma vez que precisamos tratar o caso de possíveis oclusões, além de realizar o processo de mesclagem, que pode envolver mais que uma simples sobreposição.

### *E. Oclusões*

Como resultado da etapa anterior, temos a imagem de uma malha renderizada com os vértices nas posições correspondentes aos seus respectivos marcadores. Partes da região de pele delimitada pela grade podem não estar visíveis na imagem original devido a oclusões. Assim, a imagem da tatuagem resultante deve também considerar esta restrição, não exibindo a tatuagem em áreas que não correspondam à pele.

Uma vez que se deseja aplicar a tatuagem apenas sobre a superfície da pele, passamos a utilizar a máscara obtida no processo de segmentação de pele, que já indica as regiões com pele visível na imagem original, para cortar a parte desejada da imagem da tatuagem. O resultado é mostrado na Fig. 3-i.

### *F. Remoção de Marcadores*

Uma das etapas que precede o processo de sobreposição da imagem da tatuagem, porém, é o procedimento de exclusão dos marcadores da imagem original. Isto é necessário para aprimorar o resultado final, uma vez que a tatuagem a ser inserida pode possuir áreas vazadas e tais marcadores podem prejudicar a visualização. Para realizar a remoção de tais marcadores utilizamos a técnica *inpainting*, disponível no *OpenCV*, que tem como objetivo restauração de pequenas partes em imagens [12].

A técnica de *inpainting* utiliza informação de vizinhança das áreas a serem restauradas para determinar a aparência da área a ser restaurada. No nosso sistema, as áreas restauradas são as áreas correspondentes aos marcadores, os quais são excluídos da imagem original, nesta etapa, com substituição pela textura da pele, disponibilizando uma nova imagem para sobreposição da tatuagem (Fig. 3-g).

### *G. Sobreposição da Tatuagem*

Nesta etapa, a imagem da tatuagem é sobreposta à imagem original obtida, com as devidas deformações e limites da superfície da pele aplicadas, de modo a fornecer a percepção de que a tatuagem encontra-se na superfície da pele e acompanha seu contorno (Fig. 3-j). A descrição da tatuagem é feita através de uma imagem que é composta, na maioria dos casos, por uma parte totalmente transparente (vazada). Além dessa transparência, ao aplicar a tatuagem sobre a pele, uma operação de mesclagem é aplicada à parte opaca da imagem, de forma a não eliminar completamente a textura da pele nessas áreas. Esta mesclagem foi realizada utilizando uma multiplicação ponderada entre as imagens.

## IV. Resultados e Discussão

Os experimentos para avaliação do sistema foram executados conforme a metodologia descrita na Seção III. Nessa seção serão apresentados os resultados de alguns testes realizados em condições variadas de modo a avaliar alguns dos possíveis casos de uso da aplicação. A Fig. 4 mostra a imagem de tatuagem utilizada para realização dos testes.

Inicialmente realizamos alguns testes considerando os casos mais simples para nossa aplicação (Fig. 5-a). Nessa situação, temos o plano de fundo uniforme com suficiente contraste, facilitando a segmentação de pele. Também temos um clara visualização de todos os marcadores através de um ponto de vista frontal. Além disso, existe pouca deformação entre os marcadores, reduzindo a possibilidade de erros no processo de detecção. A ausência de oclusão também colabora para a obtenção de um resultado final sem erros.

Em seguida, foram testadas as oclusões com, basicamente, duas possibilidades para aplicação: a oclusão proveniente da própria superfície da pele relativa ao posicionamento da câmera; e a oclusão referente a algum corpo interposto entre a pele e a câmera. Realizamos alguns testes (Fig. 5-b) utilizando a superfície do braço como exemplo. Em seguida, foi colocada uma caneta sobre os marcadores para produzir uma oclusão (Fig. 5-c).

Alguns tipos de distorções possíveis também foram testadas. Um dos testes foi realizado através da modificação da posição da câmera em relação aos marcadores (Fig 5-d e Fig. 5-e). Com isso, pequenas distorções de perspectiva são introduzidas, podendo dificultar o processo de detecção de marcadores. Em um outro teste (Fig. 5-f), forçamos a deformação na superfície de maneira não linear para verificar a robustez do algoritmo em detectar as mais diversas superfícies.

Condições adversas de iluminação também podem produzir erros em diversas etapas do processo. Ambientes com pouca iluminação, por exemplo, tendem a causar ruídos e

Fig. 4. Imagem de tatuagem a ser sobreposta na pele

borramentos *(motion blur)* na imagem como resultado da tentativa de compensação automática realizada pela câmera. Alguns testes foram realizados variando a condição de iluminação, como mostrado nas diversas situações da Fig. 5. A Fig. 5-g mostra também a aplicação em um ambiente externo.

Outro fator importante levado em consideração foi a adequação da aplicação para funcionar com qualquer tonalidade de pele. O processo de calibração através de amostragem trata dessa questão na etapa da segmentação de pele. De maneira semelhante, a detecção dos *blobs* a partir da detecção de borda apenas exige que os marcadores tenham alto contraste, e não uma cor específica. Estes dois fatores tornam o sistema mais robusto e não faz limitações baseados nas características da pessoa. Porém, outras características podem influenciar a detecção do sistema, como a presença de pelos e marcas, por exemplo. As Figuras 5-h, 5-i e 5-j mostram alguns testes realizados em peles diferentes.

## V. Considerações finais

O desenvolvimento do sistema está sendo inicialmente realizado para ambiente *desktop* para facilitar a pesquisa e investigação de soluções. Porém, um dos propósitos é gerar um produto que funcione em ambientes móveis e posteriormente em dispositivos vestíveis. Finalizada a implementação dessa primeira etapa, o sistema será portado e avaliado em dispositivos móveis.

De outra perspectiva, este projeto antecipa o que pode ser considerado uma modificação radical do papel semiótico da pele como nicho de processamento de informação e linguagem. Seu desenvolvimento, e proliferação, podem, rápida e facilmente, alterar o que conhecemos como "comunicação incorporada" (*embodied communication*). A pele, e seu "novo" *layout* e interface, permitirá a exploração de novos padrões de comunicação. Mais especulativamente, novas modalidades de relações interagentes devem ser estabelecidas devido à criação de novos padrões de comunicação, associados ao uso de dispositivos de RA epiteliais (*augmented tattoos*).

## Agradecimentos

Os autores gostariam de agradecer à FAPESB pelo apoio a esta pesquisa.

Fig. 5. Testes realizados sob diversas condições. (a) Imagem capturada frontalmente e sem oclusões. (b) Oclusão identificada no braço com alguns marcadores não visíveis. (c) Oclusão de um objeto sobreposto. (d, e) Imagem com distorções de perspectiva. (f) Deformação provocada na superfície da pele. (g) Teste realizado em ambiente externo com mais iluminação. (h, i, j) Testes realizados em peles de tonalidades diferentes e características diferentes.

## Referências


[1] Azuma, R. T. (1997). A survey of augmented reality. Presence-Teleoperators and Virtual Environments, 6(4).

[2] Furht, B. (2011). Handbook of augmented reality.

[3] Wohlrab, S., Stahl, J., and Kappeler, P. M. (2007). Modifying the body: Motivations for getting tattooed and pierced. Body image, 4(1):87–95

[4] Faille, M. and Edmiston, J. (2013). Graphic: The tattoo industry. http://news.nationalpost.com/2013/08/16/graphic-the-tattoo-industry/.

[5] Bitarello, B., Fuks, H., and Queiroz, J. (2011). New technologies for dynamic tattoo art. In Proceedings of the fifth international conference on Tangible, embedded, and embodied interaction, TEI '11, pages 313–316, New York, NY, USA. ACM.

[6] Nicolau, S., Pennec, X., Soler, L., and Ayache, N. (2005). A complete augmented reality guidance system for liver punctures: First clinical evaluation. Medical Image Computing, pages 539–547.

[7] Dybwad, B., (2010). Augmented Reality Tattoo Makes Your Skin Come Alive. http://mashable.com/2010/02/17/augmented-reality-tattoo/

[8] Gonzalez, R. C. and Woods, R. E. (2006). Digital Image Processing (3rd Edition). Prentice-Hall, Inc., Upper Saddle River, NJ, USA.

[9] Canny, J., *A Computational Approach To Edge Detection*, IEEE Trans. Pattern Analysis and Machine Intelligence, 8(6):679–698, 1986.

[10] Suzuki, S. and Abe, K., Topological Structural Analysis of Digitized Binary Images by Border Following. CVGIP 30 1, pp 32-46 (1985)

[11] Russ, J. C. (2010). The image processing handbook. CRC press.

[12] Telea, A. (2004). An Image Inpainting Technique Based on the Fast Marching Method. Journal of Graphics, GPU, and Game Tools 9 1, pp 23-34.

# Salient Region Detection Through Random Pixel Dissimilarities

Nelson Garcia de Paula, Mikhail A. Koslowski, Humberto R. Gamba
Programa de Pós-Graduação em Engenharia Elétrica e Informática Industrial – CPGEI
Universidade Tecnológica Federal do Paraná – UTFPR
Curitiba, Brasil
nelson@utfpr.edu.br

Gustavo B. Borba
Programa de Pós-Graduação em Engenharia Biomédica – PPGEB
Universidade Tecnológica Federal do Paraná – UTFPR
Curitiba, Brasil
gustavobborba@utfpr.edu.br

***Abstract*—This paper presents an algorithm that creates a map of the salient pixels of an image, by means of comparing each pixel of the image with a few randomly selected ones. The comparisons consist of dissimilarity tests considering three low-level dimensions: color, intensity and orientation. The saliency map is further thresholded and the resulting binary region is inserted into a bounding-box, in order to measure against a publicly available dataset and its associated ground-truth. For the quantitative analysis, classical figures of merit were adopted: precision, recall, F1 and area of overlap. The obtained results show that the proposed method is efficient in capturing salient regions in images, with good precision.**

***Keywords— Salient region detection; salient region evaluation, image dataset, visual attention.***

## I. INTRODUÇÃO

Um objeto saliente em uma imagem é um objeto que se destaca do resto da cena ou, em outras palavras, que captura a atenção do observador. A habilidade do sistema visual humano para a detecção de saliências é amplamente considerada como facilitador da aprendizagem e da própria sobrevivência dos organismos vivos, pois permite a concentração dos recursos cognitivos limitados nas regiões mais importantes da cena. A atenção nos seres humanos é capturada de dois modos que estão geralmente combinados, denominados *bottom-up* e *top-down* [1][2].

A atenção visual *bottom-up* é automática e depende exclusivamente das propriedades de baixo nível (não semânticas) da imagem, isto é, não depende de um conhecimento prévio. A atenção visual *top-down* é voluntária e está presente, por exemplo, quando o observador procura algo em uma imagem. A atenção *top-down* é geralmente guiada pela memória ou antecipação. Como no ato de prestar atenção aos carros antes de atravessar a rua, por exemplo, é necessário ter um evidente conhecimento prévio de que atravessar ruas é um processo perigoso por causa dos carros [2].

Embora o estudo da atenção visual humana seja do interesse constante de áreas como a Neurologia e Psicologia, o interesse pela atenção visual alcança também as áreas da computação [2]. Especificamente nas áreas de processamento de imagens e visão computacional, pode-se desenvolver técnicas para a detecção automática das regiões importantes ou salientes de uma imagem, utilizando-se como inspiração o mecanismo de atenção visual *bottom-up* [2][3]. Pode-se dizer que as regiões salientes de uma imagem correspondem a contrastes, ou diferenças significativas destas regiões em relação ao resto da imagem, seja, por exemplo, nas dimensões cor, forma, ou orientação [4].

Neste artigo, apresenta-se um método para a detecção de saliências em imagens que explora o conceito da busca aleatória por dissimilaridades em características de baixo nível entre os pixels. Dentre os pontos importantes do método proposto, destacam-se a simplicidade e a flexibilidade, no sentido de que diferentes variações do conceito central - busca aleatória por dissimilaridades - podem ser concebidas e testadas. Para a avaliação quantitativa foram utilizadas 1000 imagens de uma base de dados disponível publicamente, de cenas reais com as regiões salientes previamente selecionadas e disponibilizadas na forma de máscaras binárias, consideradas como *ground-truth.*

A próxima seção do artigo descreve em detalhes o método proposto. Em seguida, apresenta-se a metodologia empregada na avaliação quantitativa do método, incluindo a interpretação das figuras de mérito utilizadas. Posteriormente, são apresentados os resultados e, finalmente, as considerações finais e sugestões para trabalhos futuros.

## II. MÉTODO PROPOSTO

O algoritmo desenvolvido gera um mapa de saliências [5][2] a partir de comparações de características de baixo nível entre pixels aleatórios da imagem. As características consideradas são cor, intensidade e orientação. Segundo o modelo computacional de atenção visual *bottom-up* proposto por [6], cuja saída são coordenadas, ou pontos de atenção na imagem, estas três dimensões compreendem uma combinação adequada para a captura de saliências em imagens. O mapa de saliências é então limiarizado e, na última etapa, obtém-se um retângulo (*bounding-box*) correspondente à região considerada saliente pelo algoritmo. Esta saída na forma de uma *bounding-box* permite a comparação com um banco de imagens público,

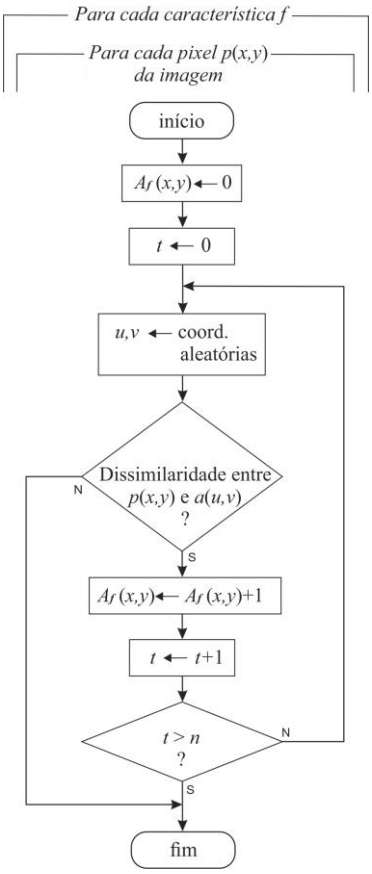



Fig. 1. Fluxograma do método proposto. Esta operação é realizada para ca pixel da imagem, para cada uma das três características: cor, intensidade e orientação.

para o qual o *ground-truth* disponível está na forma de retângulos.

O objetivo das comparações entre um pixel $p(x,y)$ e pixels aleatórios, considerando cada uma das três características independentemente, é detectar dissimilaridades entre os pixels. Quanto maior for a dissimilaridade, mais saliente é o pixel $p(x,y)$. O fluxograma na Fig. 1 detalha o procedimento de comparação, realizado independentemente para cada uma das três características. Para cada pixel $p(x,y)$ da imagem, realiza-se um teste de dissimilaridade com até $n$ pixels aleatórios da imagem. Caso o resultado do teste de dissimilaridade seja positivo, uma imagem acumuladora $A_f$ é incrementada em $A_f(x,y)$. Conforme mencionado anteriormente, o fluxograma da Fig. 1 é executado para cada uma das três características, originando três imagens acumuladoras: $f \in \{c, i, o\}$, sendo $c$ a característica cor, $i$ a característica intensidade e $o$ a característica orientação.

O teste de dissimilaridade para a característica cor entre o pixel $p$ e o pixel aleatório $a$ é verdadeiro quando a diferença absoluta entre qualquer um dos canais de cor $R$, $G$ e $B$ for maior que $\delta$ (Eq. (1)). As Figs. 2(a) e (b) mostram uma imagem sintética e a imagem acumuladora $A_c$ obtida. Pode-se observar que a região de cor verde, saliente na imagem de entrada, foi capturada corretamente em $A_c$.

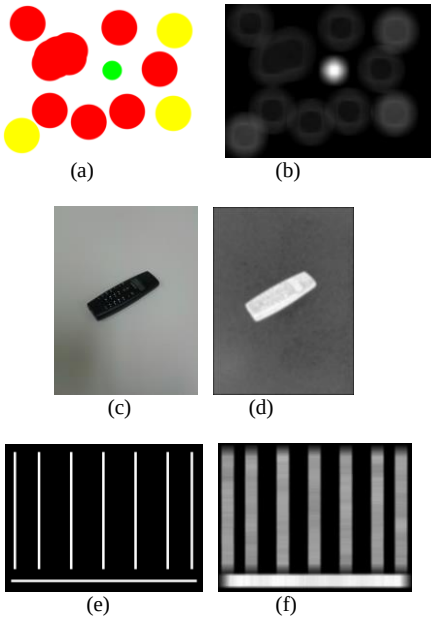



Fig. 2. (a) Imagem sintética e (b) imagem acumuladora $A_c$. (c) Imagem exemplo e (d) image acumuladora $A_i$. (e) Imagem sintética e (f) imagem acumuladora $A_o$.

$$\exists\, |p^h - a^h| > \delta \;\mid\; h \in \{R,G,B\} \qquad (1)$$

O teste de dissimilaridade para a característica intensidade entre o pixel $p$ e o pixel aleatório $a$ é verdadeiro quando a diferença absoluta entre os níveis de cinza for maior que $\gamma$ (Eq. (2)). A intensidade $g$ é obtida a partir de (3), onde $k$ é um pixel qualquer. As Figs. 2(c) e (d) mostram uma imagem exemplo e a imagem acumuladora $A_i$ obtida. Pode-se observar que a região de intensidade escura, saliente na imagem de entrada, foi capturada corretamente em $A_i$.

$$\exists\, |g(p) - g(a)| > \gamma \qquad (2)$$

$$g(k) = 0{,}30R + 0{,}59G + 0{,}11B \qquad (3)$$

O teste de dissimilaridade para a característica orientação entre o pixel $p$ e o pixel aleatório $a$ é verdadeiro quando a diferença absoluta entre as orientações dos vetores gradiente for maior que $\beta$ (Eq. (4)). A orientação do gradiente é obtida a partir de (5), onde $D_H$ e $D_V$ são as imagens gradiente nas direções horizontal e vertical, respectivamente. As Figs. 2(e) e (f) mostram uma imagem sintética e a imagem acumuladora $A_o$ obtida. Pode-se observar que a região de orientação vertical, saliente na imagem de entrada, foi capturada corretamente em $A_o$.

$$\exists\, |\theta(p) - \theta(a)| > \beta \qquad (4)$$

$$\theta(k) = tan^{-1}(D_H(k) / D_V(k)) \qquad (5)$$

Para a obtenção do mapa de saliências $S$, as imagens acumuladores $A_c$, $A_i$ e $A_o$ são combinadas através de (6), onde $N$ é a função normalização. Posteriormente, como etapa final do algoritmo, é feita a limiarização do mapa de saliências em um limiar *th* e a criação de um retângulo que engloba o objeto considerado saliente, obtendo-se uma máscara binária $M$. O limiar *th* é obtido conforme (7), onde *autoTh* é o método de

(a) (b) (c)

Fig. 3. (a) Imagem de entrada, (b) mapa de saliências *S* e (c) máscara binária *M*.

limiarização [7] e *L* é 255 para imagens de 8 bits por pixel. A Fig. 3 apresenta o mapa de saliências S e a máscara binária *M* para uma imagem exemplo, extraída de [4].

$$S = N(A_c + A_i + A_o) \quad (6)$$

$$th = ( autoTh(S) + L ) / 2 \quad (7)$$

Este algoritmo baseia-se assim na idéia de que no modo *bottom-up* o sistema visual humano detecta pontos interessantes que são diferentes do resto da imagem. Nosso algoritmo pode ser considerado uma versão simplificada do algoritmo proposto por [8].

## III. Metodologia Para a Avaliação Quantitativa

### A. Banco de Imagens e Ground-truth

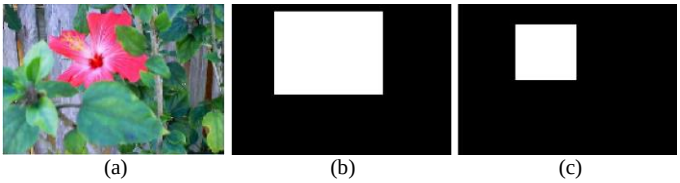
O banco de imagens utilizado para a avaliação quantitativa do método foi o proposto por [9], disponível publicamente, e contém 1000 imagens. Este banco é derivado do *MSRA Salient Object Database* [4].

No contexto de detecção de regiões salientes, um *ground-truth* pode ser obtido requisitando-se que observadores humanos determinem que regiões da imagem consideram salientes. O *MSRA* disponibiliza, para cada imagem, uma imagem binária *ground-truth* contendo retângulos delimitando as regiões salientes, estabelecidos manualmente por voluntários.

### B. Figuras de Mérito

Para medir o desempenho do método de detecção saliências proposto, foram utilizadas as figuras de mérito abrangência (*recall*), precisão, *F*1 e área de sobreposição [10]. Todas elas variam entre 0 e 1, sendo 1 o melhor valor. Cada uma destas figuras de mérito será explicada e interpretada em detalhes adiante. Elas podem ser descritas em função das operações de interseção e união de conjuntos entre as regiões binárias determinadas pelo algoritmo, denominadas de regiões salientes (RS), e as regiões *ground-truth* (GT). Ainda, podem ser descritas em função de:

- *verdadeiros positivos* (VP): a área onde a RS obtida concorda com a GT;
- *falsos positivos* (FP): a área onde RS diz ser saliente mas a GT não;
- *falsos negativos* (FN): a área onde RS não considera saliente mas a GT sim.



(a) (b) (c)

Fig. 4. (a) Imagem de entrada, (b) máscara binária *ground-truth* e (c) máscara binária *M* de saída do algoritmo proposto. Neste exemplo, os valores obtidos para precisão e abrangência são *P* = 1 e *R* = 0,38.

A abrangência, *R*, mede a fração da região de GT coberta pela nossa região saliente RS. Em termos de VP e FN, ela é definida pela Eq. (5). Em termos de interseção entre regiões binárias, é definida pela Eq. (6), onde $A(\cdot)$ retorna uma área em pixels. A abrangência penaliza a falha em reconhecer uma região do *ground-truth*. Em outras palavras, penaliza os falsos negativos. Entretanto, esta métrica não penaliza os falsos positivos – uma abrangência perfeita igual a 1 pode ser obtida simplesmente escolhendo toda a área da imagem sob teste como saliente.

$$R = \frac{\mathrm{VP}}{\mathrm{VP} + \mathrm{FN}} \quad (5)$$

$$R = \frac{A(\mathrm{RS} \cap \mathrm{GT})}{A(\mathrm{GT})} \quad (6)$$

A precisão, *P*, mede a fração da região obtida que fica dentro da região de *ground-truth*. É definida pela Eq. (7) em termos de VP e FP. Em termos de interseção entre regiões binárias, é definida pela Eq. (8). A precisão penaliza pixels detectados fora da *ground-truth*, ou falsos positivos. Esta métrica falha em penalizar falsos negativos, pois uma pequena fração de acerto exclusivamente dentro da *ground-truth* já fornece uma precisão de igual a 1.

$$P = \frac{\mathrm{VP}}{\mathrm{VP} + \mathrm{FP}} \quad (7)$$

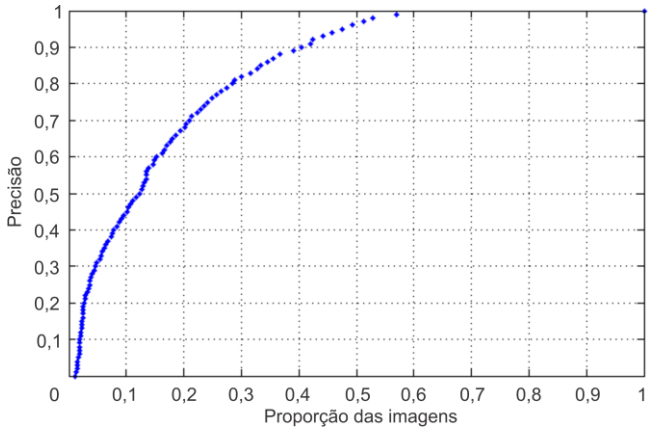
$$P = \frac{A(\mathrm{RS} \cap \mathrm{GT})}{A(\mathrm{RS})} \quad (8)$$

Portanto, as métricas abrangência e precisão devem sempre ser avaliadas em conjunto. A métrica *F*1, definida na Eq. (9), combina as métricas abrangência e precisão através de uma média harmônica. Desta forma, penaliza tanto falsos negativos quanto falsos positivos.

$$F1 = \frac{2 \times P \times R}{P + R} \quad (9)$$

A métrica área de sobreposição (*AS*) é uma outra maneira de combinar abrangência e precisão neste contexto. A *AS* também penaliza tanto falsos negativos quanto falsos positivos. É definida como a divisão entre a área da interseção da região saliente obtida com a área da *ground truth* e a união das mesmas áreas, Eq. (10).

$$AS = \frac{A(\mathrm{SR} \cap \mathrm{GT})}{A(\mathrm{SR} \cup \mathrm{GT})} \quad (10)$$



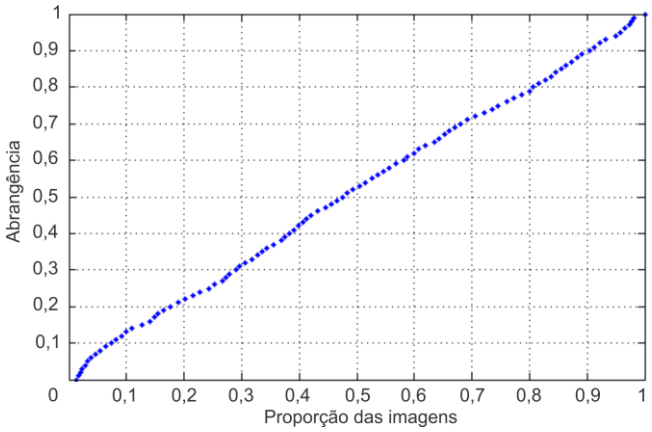




Fig. 5. Histogramas cumulativos da precisão e da abrangência do algoritmo proposto para as 1000 imagens.

Neste contexto, a *AS* apresenta duas vantagens sobre a métrica *F*1:

*1)* a interseção e a união das regiões obtidas e da *ground-truth* podem ser facilmente visualizadas em termos de pixels na imagem.

*2)* a relação entre a interseção e união expressa mais claramente que *F*1 o quão bem a região obtida coincide com a região de *ground-truth.*

As figuras de mérito abrangência, precisão e área de sobreposição podem ser computadas para uma única imagem, ou no caso de um conjunto de imagens, globalmente. Para isso, acumula-se individualmente o numerador e o denominador da figura de mérito em questão e ao final realiza-se a divisão, obtendo-se assim o valor global.

## IV. Resultados

Para todas as avaliações, os parâmetros utilizados para o algoritmo (seção II) foram $n = 10$ e $\delta = \gamma = \beta = 50$. Estes valores foram obtidos empiricamente.

A Fig. 4(a) apresenta uma imagem exemplo e 4(b) o seu *ground-truth* correspondente. O algoritmo proposto deve então encontrar a mesma região saliente que foi determinada manualmente no *ground-truth*. A máscara binária *M* de saída

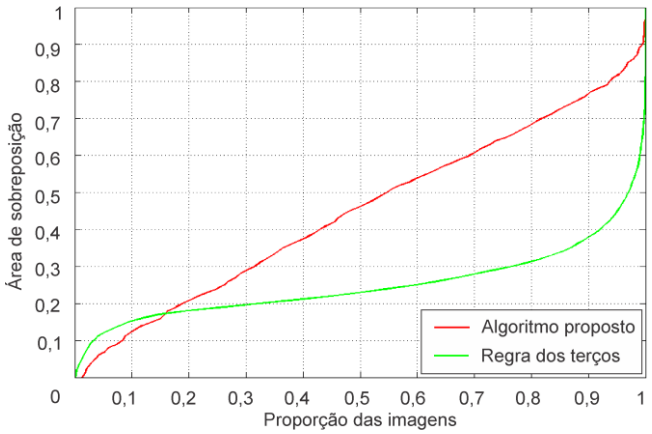



Fig. 6. Histograma cumulativo da área de sobreposição do algoritmo proposto para as 1000 imagens.

do algoritmo é mostrada na figura 4(c). Note que, neste exemplo, o algoritmo encontrou parte da região saliente, mas em um retângulo consideravelmente menor que o da *ground-truth*. Neste caso então temos uma precisão igual a 1 porque a região saliente está completamente contida no *ground-truth*. A região saliente encontrada é porém somente parte da região desejada (*ground-truth*). Assim, embora o algoritmo tenha encontrado corretamente o objeto (flor), o retângulo em *M* corresponde somente a uma parte do que foi determinado como *ground-truth*, resultando em uma abrangência baixa, igual a 0,38. Isto indica que, neste cenário, a figura de mérito abrangência é bastante exigente, já que não é suficiente que o algoritmo encontre corretamente a parte principal do objeto. Esta observação aplica-se também para a figura de mérito área de sobreposição.

A precisão e a abrangência obtidas para as 1000 imagens são mostradas nas curvas das Fig. 5, na forma de histogramas cumulativos. Assim, o eixo das abcissas corresponde à proporção do número total de imagens. Nestes histogramas cumulativos, quanto antes a curva tender para o canto superior esquerdo do sistema de coordenadas, melhor é o desempenho do algoritmo para a figura de mérito em questão. A curva de precisão demonstra que o algoritmo apresentou bons resultados na detecção de parte da região *ground-truth*, e em aproximadamente 60% das imagens obteve uma precisão maior que 90%. Já a curva de abrangência demonstra que o desempenho do algoritmo para esta figura de mérito é baixo se comparado à precisão. No entanto, é interessante ressaltar que, conforme comentado anteriormente, a abrangência é uma figura de mérito bastante exigente neste cenário.

A Fig. 6 apresenta a curva da área de sobreposição para as 1000 imagens, também na forma de histograma cumulativo. É importante mencionar que a área de sobreposição também é uma figura de mérito bastante exigente neste cenário. A curva *regra dos terços* corresponde a uma região central imagem, obtida segundo a tradicional regra prática dos terços, utilizada em fotografia. Ela é apresentada para efeito de comparação, simulando um algoritmo ingênuo (*naive*) para detecção de regiões salientes.

Os valores globais de cada uma das figuras de mérito obtidos para o algoritmo, considerando as 1000 imagens,

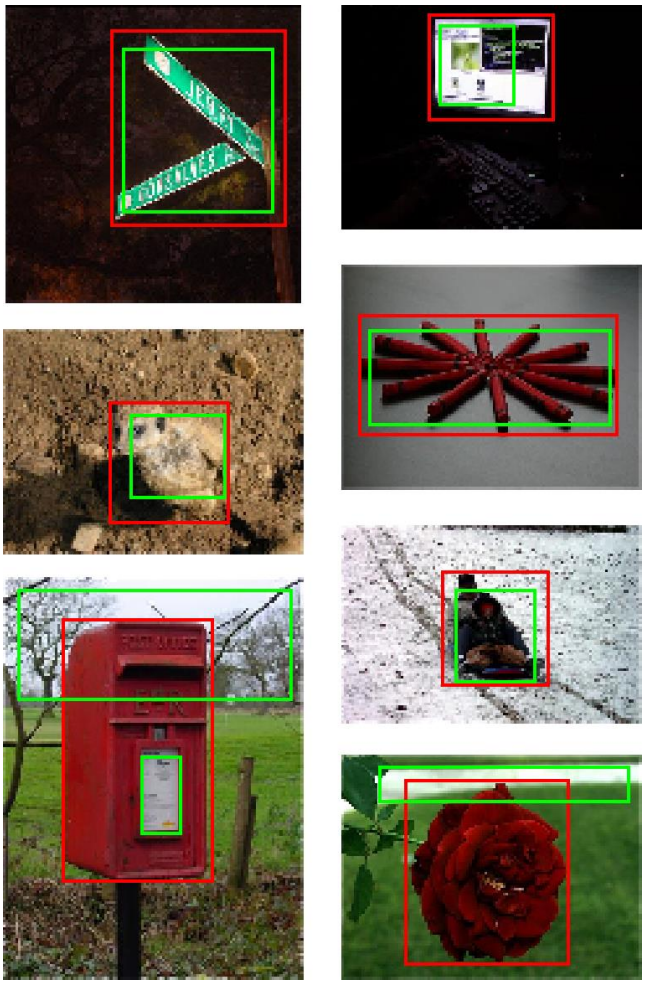


Fig. 7. Exemplos da saída do algotimo (retângulos verdes) para diferentes imagens. Os retângulos vermelhos correspondem ao *ground-truth*. Para as duas últimas imagens observa-se que o algoritmo não detectou as regiões esperadas.

foram: $R = 0{,}50$; $P = 0{,}79$; $AS = 0{,}44$ e $F1 = 0{,}61$.

Finalmente, a Fig. 7 apresenta alguns exemplos das regiões salientes detectadas pelo algoritmo para diferentes imagens de entrada. Os retângulos em verde correspondem às regiões detectadas e, os retângulos em vermelho, às regiões *ground-truth*. As imagens da última linha ilustram situações nas quais o algoritmo não detectou as regiões esperadas.

## V. Considerações Finais

Conclui-se que este algoritmo é viável como detector de regiões salientes pois detecta as regiões com precisão maior que 0,9 em 60% do banco de imagens testado, como pode ser observado no gráfico da Fig. 5. Na Fig. 7 estão ilustrados cinco casos em que a região saliente foi corretamente detectada e dois casos onde o algoritmo falhou. A partir destes exemplos é possível constatar a natureza *bottom-up* do algoritmo - não há conhecimento semântico das imagens e simplesmente são detectadas as regiões mais salientes segundo características de baixo nível. Em outras palavras, este algoritmo não utiliza qualquer mecanismo *top-down*, o que consistiria na interpretação semântica do conteúdo das imagens. Por exemplo, no caso da imagem da rosa na Fig. 7, o algoritmo detectou a trilha no fundo como mais saliente (mais dissimilar considerando os critérios adotados).

Outro ponto importante a ser tratado é a abrangência baixa em relação à precisão. Os exemplos da Fig. 7 ilustram que o algoritmo tende a detectar regiões menores que as estabelecidas no *ground-truth*.

Para trabalhos futuros, pretende-se avaliar o impacto de cada um dos parâmetros ($n$, $\delta$, $\gamma$, $\beta$) no resultado final. Ainda, é possível atribuir pesos diferente para cada imagem acumuladora $A_f$, avaliando assim o impacto de cada característica de baixo nível (cor, intensidade, orientação) nas figuras de mérito. O objetivo é melhorar a abrangência sem prejudicar a precisão para o maior número de imagens possíveis do banco de teste.

## Referências


[1] Z. Wang and B. Li, "A two-stage approach to saliency detection in images," IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2008), Las Vegas, pp. 965-968, March 2008.

[2] S. Frintrop, E. Rome, and H. I. Christensen, "Computational Visual Attention Systems and Their Cognitive Foundations: A Survey," ACM Trans. Appl. Percept., vol. 7, no. 6, pp. 1-39, 2010.

[3] E. Erdem and A. Erdem, "Visual saliency estimation by nonlinearly integrating features using region covariances," Journal of Vision, vol. 13, no. 4, 2013.

[4] T. Liu, Z. Yuan, J. Sun, J. Wang, N. Zheng, X. Tang, and H.-Y Shum, "Learning to Detect a Salient Object," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 33, pp. 353-367, 2011.

[5] K. Koehler, F. Guo, S. Zhang, and M. P. Eckstein, "What do saliency models predict?," Journal of Vision, vol. 14, no. 3, 2014.

[6] L. Itti and C. Koch, "Feature combination strategies for saliency-based visual attention systems," Journal of Electronic Imaging, vol. 10, pp. 161-169, 2001.

[7] N. Otsu, "A Threshold Selection Method from Gray-Level Histograms," IEEE Transactions on Systems, Man and Cybernetics, vol. 9, pp. 62-66, 1979.

[8] F. Stentiford, "Attention-based similarity", Pattern Recognition, vol. 40, pp. 771-783, 2007.

[9] R. Achanta, S.Hemami, F. Estrada, and S. Süsstrunk, "Frequency-tuned Salient Region Detection," IEEE International Conference on Computer Vision and Pattern Recognition (CVPR 2009), Miami, pp. 1597-1604, June 2009.

[10] G. B. Borba, H. R. Gamba, O. Marques., A. Colic, and V. Adzic, "Comparing Figures of Merit and Images Datasets for Evaluation of Salient Region Detection Algorithms," The IASTED International Conference on Signal Processing, Pattern Recognition and Applications (SPPRA 2010), Innsbruck, 2010.

# Image Segmentation applied on Registration Process for Remote Sensing

Juliano Marques Nunes
Universidade Federal do ABC
Santo André, Brasil
juliano.nunes@ufabc.edu.br

Francisco de Assis Zampirolli
Centro de Matemática, Computação e Cognição
Universidade Federal do ABC
Santo André, Brasil
fzampirolli@ufabc.edu.br

Helena França
Centro de Engenharia, Modelagem e Ciências Sociais Aplicadas
Universidade Federal do ABC
Santo André, Brasil
helena.franca@ufabc.edu.br

***Abstract*—In several studies of remote sensing in which satellite images are used and there's a need to adjust the coordinates of two or more images so that the same point in these images correspond to the same part of the earth's surface in the other, this step is usually done manually. This paper presents segmentation techniques that were applied to images for extracting homologous points in regions bounded by bodies of water.**

***Resumo*—Em vários trabalhos da área de sensoriamento remoto em que se utilizam imagens orbitais e existe a necessidade de ajustar as coordenadas de duas ou mais imagens de modo que um mesmo ponto nessas imagens corresponda ao mesmo local da superfície terrestre nas outras, essa etapa é geralmente feita manualmente. Este trabalho apresenta técnicas de segmentação de imagens e extração de pontos homólogos em regiões delimitadas por corpos d'água.**

***Keywords—image processing; mathematical morphology; image registration; remote sensing.***

## I. INTRODUÇÃO

Em sensoriamento remoto é importante inter-relacionar diversas imagens espaciais de um mesmo ponto da superfície terrestre, ou seja, cada pixel em várias imagens adquiridas em tempos distintos tem que representar aproximadamente o mesmo ponto da superfície terrestre. Como aplicações, é possível verificar o crescimento do desmatamento e da queimada, verificar níveis de água em reservatórios, verificar o crescimento urbano, estimar a produção de grãos etc. [1].

Segundo Federov [2], o processo de registro tem como objetivo referenciar determinada imagem (imagem de ajuste) dentro de um sistema de coordenadas geográficas com base em outra imagem com coordenadas conhecidas (imagem de referência). Para isso, é necessária a aquisição de pontos de controle (ou homólogos), geralmente feita de maneira manual, que consiste em localizar visualmente o mesmo ponto nas duas imagens: a de referência e a de ajuste. É comum selecionar pontos de fácil identificação em ambas as imagens e que sofrem pouca variação de forma ao longo do tempo. Bons exemplos são: cruzamento de estradas, pontes, encontro de rios, barragens em represas etc. Com a aquisição de vários pontos homólogos nas imagens, é possível modelar uma função de distorção (ou função de mapeamento) entre elas que será utilizada para correção da imagem de ajuste.

Ainda de acordo com Federov, a aquisição dos pontos homólogos pode ser realizada basicamente de três maneiras: (1) manualmente, quando há grande intervenção do especialista de sensoriamento remoto e este define os pontos homólogos visualmente nas duas imagens. Esta tarefa é lenta, repetitiva e pode ser difícil encontrar estes pontos de maneira uniforme em regiões com textura bem diversificada, sem feições (estruturas salientes ou contornos) relevantes ou visíveis; (2) de maneira semiautomática, quando ainda existe a intervenção de um especialista, geralmente na definição de regiões visualmente interessantes para seleção automática dos pontos homólogos, mas a definição dos pontos selecionados é feita automaticamente; e (3) automaticamente, quando não existe intervenção ou esta é muito baixa [3]. Este artigo trata deste terceiro caso, ou seja, uma região de interesse é escolhida e os pontos homólogos são encontrados de forma automática.

Segundo Zitová e Flusser [4], a maioria dos processos de registro de imagens consiste em 4 etapas: Detecção de feições; casamento de feições; estimação do modelo de transformação; e, finalmente, reamostragem e transformação da imagem. Na etapa de detecção de feições, objetos distintos e mais salientes (regiões de contorno fechado, bordas, contornos, intersecções de linhas, cantos etc.), em ambas as imagens, são detectados. A correspondência entre as feições detectadas na imagem de referência e de ajuste é estabelecida na etapa de casamento de feições. Posteriormente, são determinados os tipos e parâmetros para as funções de mapeamento, que são aplicadas à imagem de ajuste na última etapa, de reamostragem e de transformação.

Os métodos para detecção de feições semiautomáticos ou automáticos podem se basear na segmentação de contornos de objetos nas imagens e são recomendados que as imagens possuam objetos distinguíveis e facilmente detectáveis. Este é, usualmente, o caso das aplicações de sensoriamento remoto e visão computacional. Imagens típicas contêm bastante detalhes (edifícios, rios, estradas, florestas etc.) [4, 5]. Porém, neste trabalho serão abordadas imagens com contorno em água, que são facilmente segmentadas, para a obtenção das feições e dos pontos homólogos nas imagens de ajuste e de referência.

## II. Materiais e Métodos

Para encontrar os pontos homólogos em duas imagens, a de ajuste e a de referência, foi utilizado o algoritmo apresentado no fluxograma na Fig. 1, que será descrito nas próximas seções.

### A. *Leitura das imagens*

Para este artigo, foram escolhidas duas imagens de épocas diferentes, uma de 14/08/2003 e a outra de 02/09/2010. As imagens, obtidas pelo sensor de segunda geração denominado "*Thematic Mapper*" (TM) a bordo da plataforma Landsat 5, cobrem a região do litoral dos estados de São Paulo e Rio de Janeiro na órbita ponto 218/76 (sistema WRS - Landsat).

As imagens foram obtidas do site da Divisão de Geração de Imagens do Instituto Nacional de Pesquisas Espaciais (DGI-INPE) com resolução espacial de 30 x 30m, resolução radiométrica de 8 bits/pixel e escala nominal de 1:100.000, com correção geométrica.

Fig. 1. Fluxograma do algoritmo para encontrar pontos homólogos.

Usando o *software* IDRISI Andes [6] as imagens foram recortadas em setores de tamanho 1071 x 625 pixels. Esse recorte recobriu apenas a área teste, no caso, a Ilha Grande, ilha do litoral do Rio de Janeiro (Fig. 2). Para o estudo, foi utilizada a banda 2 de ambas as cenas. Estes recortes foram salvos em formato **.tif* e processados no *software* Matlab, versão R2013a.

### B. *Limiarização*

Inicialmente, foi encontrado um limiar que possibilitasse distinguir as porções terrestres das porções de água e, assim, as imagens foram transformadas em imagens binárias que tivessem como conteúdo apenas as porções de terra.

Após identificar as porções terrestres das imagens, foram eliminadas as porções de terra menores e aplicada uma função de fechamento morfológico por área para cobrir os "buracos" e "ruídos" existentes nas porções encontradas, restando como resultado apenas a Ilha Grande.

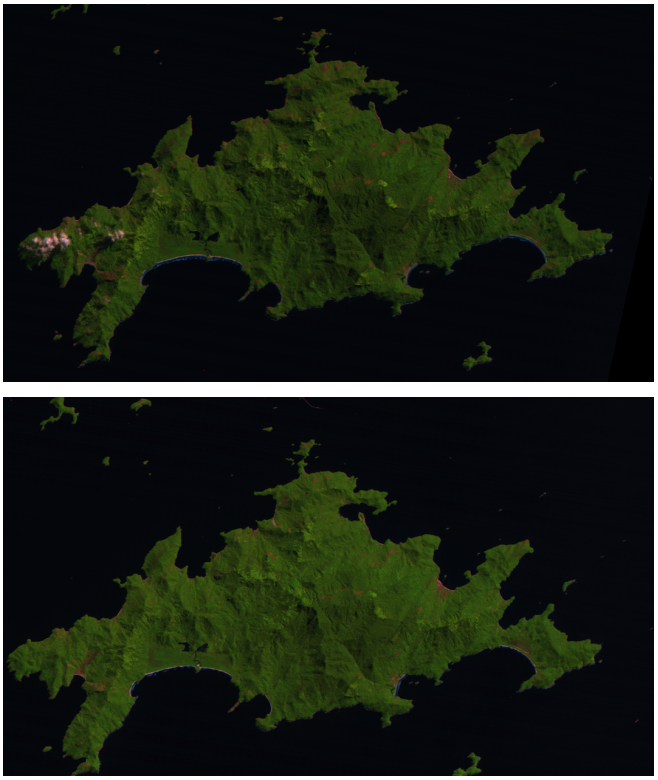


Fig. 2. Imagens da Ilha Grande recortadas, cenas de 2003 (acima) e 2010.

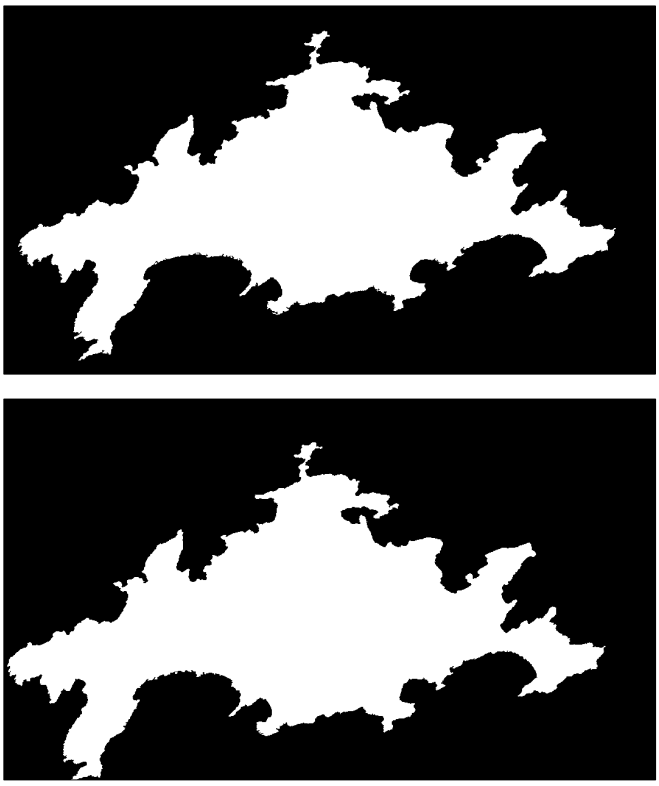


Fig. 3. Imagens binárias da Ilha Grande, cenas de 2003 (acima) e 2010.

### C. *Extrações de Feições (extremidades)*

Feito o tratamento inicial da imagem, são definidas as regiões de onde serão extraídos os pontos homólogos. Tipicamente, ao fazer um registro manual, seriam escolhidas as extremidades da ilha. Para reproduzir este procedimento computacionalmente, foram utilizadas funções de abertura e fechamento morfológicos considerando uma vizinhança (função estruturante) em forma de disco de raio crescente e, posteriormente, feita a diferença entre essas imagens. Como resultado, foram obtidas pequenas regiões que demarcam as regiões localizadas nas extremidades do contorno da ilha. O número de regiões foi restringido através da exclusão de áreas com poucos pixels. Pelos teste empíricos, para descartar regiões escolhidas erroneamente, que deviriam ser equivalentes nas duas imagens, foi considerado 50% a mais de regiões sobre o número de feições a serem consideradas, outro parâmetro de entrada. O resultado final dessa etapa é apresentado na Fig. 4 (as imagens foram invertidas para uma melhor visualização).

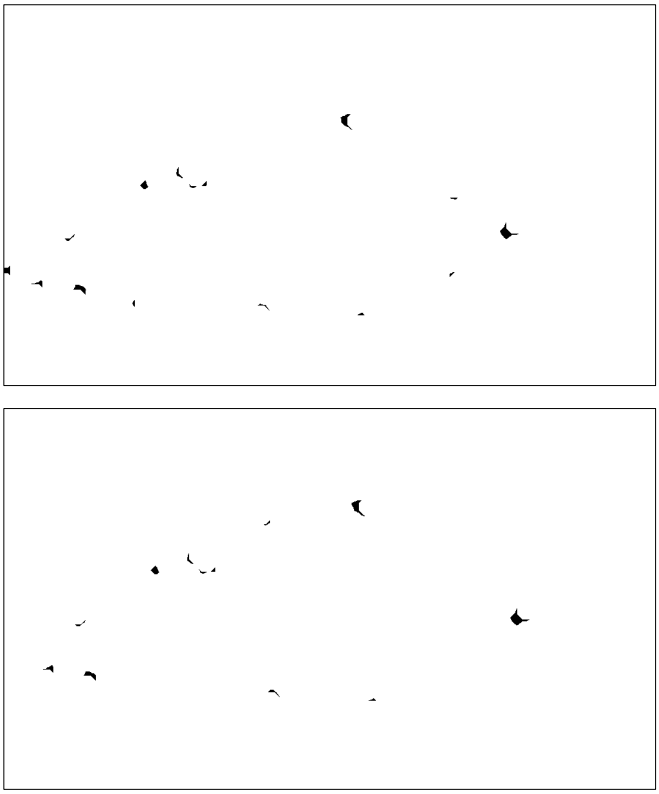


Fig. 4. Feições extraídas das cenas de 2003 (acima) e 2010.

### D. *Definições dos Centroides*

Dos agrupamentos de pixels extraídos das imagens da Fig. 4 foram calculados os centros geométricos, também conhecidos como centroides. Veja na Fig. 5 as soma das imagens em níveis de cinza das cenas de 2003 e 2010, destacando os centroides calculados.

### E. *Definições dos Pontos Homólogos*

A definição dos centroides em ambas as imagens (a de ajuste e a de referência) não é suficiente para encontrar os pontos homólogos, já que ambas possuem também pontos sem correspondência na outra imagem. Para isto, foi feita a comparação entre os centroides em ambas imagens. A identificação dos pares de pontos foi feita através da comparação das distâncias (distância horizontal somada à distância vertical) entre todos os pontos de ambas as imagens. Inicialmente, foram geradas matrizes para os centroides de cada imagem com a distância relativa de um ponto para todos os outros. Essas matrizes foram "cruzadas" e, de acordo com o número de correspondências encontradas, foram suprimidos os pontos para os quais não foram encontrados os pontos correspondentes na outra imagem (veja Fig. 6). Além disso, foi usada a função estatística moda para descartar alguns pontos encontrados erroneamente, devido à proximidade de alguns pixels.

Para um melhor entendimento do algoritmo, nota-se que a segmentação identificou 11 pontos na imagem 1 e, na imagem 2, 15 pontos, sendo que apenas 10 pontos correspondentes existem em ambas. O primeiro passo consiste em gerar uma matriz que contém a soma das distâncias horizontais e verticais entre todos os pontos, para ambas as imagens.

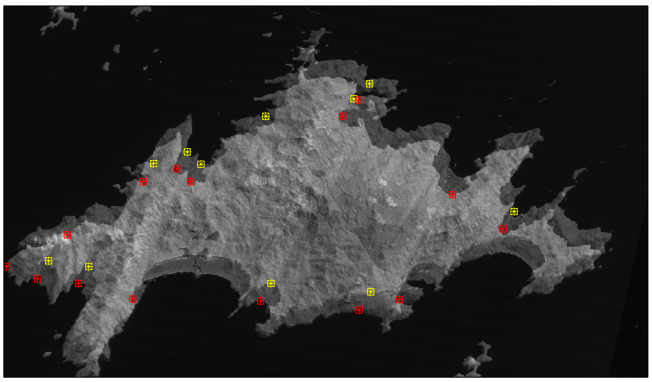


Fig. 5. Soma das imagens em níveis de cinza da Fig. 2, destacando os centroides, em amarelo da cena de 2003 e vermelho da cena de 2010.

A Tab. 1 permite verificar que o ponto 1 da imagem 1, por exemplo, está a 76 pixels de distância do ponto 2 (também da imagem 1), considerando a soma da distância em X (horizontal) com a distância em Y (vertical). Da mesma forma, podemos afirmar através da Tab. 2 que o ponto 2 da imagem 2 está a 75 pixels de distância do ponto 4 da imagem 2. Assim, podemos concluir que os pontos 1 e 2 da imagem 1 possuem distância próxima com relação os pontos 2 e 4 da imagem 2.

Nas Tabelas 1 e 2 o destaque em cinza mostra as correspondências entre as distâncias, evidenciando que os pontos 1, 2, 4, 5, 7, 8, 9 e 10 da imagem 1 correspondem aos pontos 2, 4, 7, 8, 9, 10, 11, 12 e 15 na outra imagem, já que as distâncias entre esses pontos e todos os outros são semelhantes, ou seja, com diferença máxima de 2 pixels.

Para reproduzir o procedimento em um algoritmo computacional, foi gerada uma tabela de comparação entre as Tabelas 1 e 2 (tabelas de distância, veja Tab. 3), que mostra a semelhança entre os pontos.

A Tab. 3 mostra que o ponto 2 da imagem 2 tem 10 semelhanças com o ponto 1 da imagem 1, assim como alguns outros casos. Sendo assim, o algoritmo seleciona e exclui os

pontos com menor número de semelhanças, restando apenas os pontos correspondentes. Na Tab. 3 os valores que demonstram esta relação de correspondência entre os pontos estão destacados em cinza escuro.

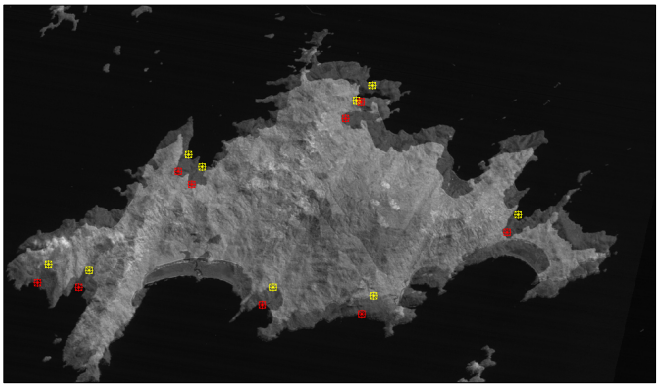


Fig. 6. Pontos correspondentes em destaque. Os pontos amarelos foram extraídos da cena de 2003 e os vermelhos da cena de 2010.

Tab. 1: Distâncias entre os 11 pontos da imagem 1.

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 76 | 333 | 407 | 409 | 595 | 403 | 769 | 819 | 581 | 847 |
| 2 | 76 | 0 | 277 | 351 | 353 | 539 | 327 | 713 | 763 | 505 | 791 |
| 3 | 333 | 277 | 0 | 74 | 78 | 262 | 390 | 436 | 486 | 568 | 672 |
| 4 | 407 | 351 | 74 | 0 | 42 | 188 | 354 | 362 | 412 | 532 | 636 |
| 5 | 409 | 353 | 78 | 42 | 0 | 186 | 312 | 360 | 410 | 490 | 594 |
| 6 | 595 | 539 | 262 | 188 | 186 | 0 | 284 | 174 | 224 | 462 | 566 |
| 7 | 403 | 327 | 390 | 354 | 312 | 284 | 0 | 440 | 490 | 178 | 518 |
| 8 | 769 | 713 | 436 | 362 | 360 | 174 | 440 | 0 | 50 | 346 | 450 |
| 9 | 819 | 763 | 486 | 412 | 410 | 224 | 490 | 50 | 0 | 344 | 448 |
| 10 | 581 | 505 | 568 | 532 | 490 | 462 | 178 | 346 | 344 | 0 | 368 |
| 11 | 847 | 791 | 672 | 636 | 594 | 566 | 518 | 450 | 448 | 368 | 0 |

Tab. 2: Distâncias entre os 15 pontos da imagem 2.

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 72 | 154 | 147 | 261 | 365 | 442 | 443 | 475 | 802 | 854 | 652 | 701 | 853 | 880 |
| 2 | 72 | 0 | 122 | 75 | 189 | 333 | 410 | 411 | 403 | 770 | 822 | 580 | 629 | 821 | 848 |
| 3 | 154 | 122 | 0 | 99 | 213 | 211 | 288 | 289 | 427 | 648 | 700 | 604 | 653 | 699 | 726 |
| 4 | 147 | 75 | 99 | 0 | 114 | 274 | 351 | 352 | 328 | 711 | 763 | 505 | 554 | 762 | 789 |
| 5 | 261 | 189 | 213 | 114 | 0 | 210 | 287 | 288 | 214 | 647 | 699 | 391 | 440 | 698 | 725 |
| 6 | 365 | 333 | 211 | 274 | 210 | 0 | 77 | 78 | 390 | 437 | 489 | 567 | 616 | 530 | 671 |
| 7 | 442 | 410 | 288 | 351 | 287 | 77 | 0 | 43 | 355 | 360 | 412 | 532 | 581 | 495 | 636 |
| 8 | 443 | 411 | 289 | 352 | 288 | 78 | 43 | 0 | 312 | 359 | 411 | 489 | 538 | 452 | 593 |
| 9 | 475 | 403 | 427 | 328 | 214 | 390 | 355 | 312 | 0 | 439 | 491 | 177 | 230 | 490 | 517 |
| 10 | 802 | 770 | 648 | 711 | 647 | 437 | 360 | 359 | 439 | 0 | 52 | 346 | 395 | 309 | 450 |
| 11 | 854 | 822 | 700 | 763 | 699 | 489 | 412 | 411 | 491 | 52 | 0 | 346 | 395 | 309 | 450 |
| 12 | 652 | 580 | 604 | 505 | 391 | 567 | 532 | 489 | 177 | 346 | 346 | 0 | 83 | 343 | 370 |
| 13 | 701 | 629 | 653 | 554 | 440 | 616 | 581 | 538 | 230 | 395 | 395 | 83 | 0 | 260 | 287 |
| 14 | 853 | 821 | 699 | 762 | 698 | 530 | 495 | 452 | 490 | 309 | 309 | 343 | 260 | 0 | 141 |
| 15 | 880 | 848 | 726 | 789 | 725 | 671 | 636 | 593 | 517 | 450 | 450 | 370 | 287 | 141 | 0 |

Sendo assim, seriam excluídos os pontos 3 e 6 da imagem 1 e os pontes 1, 3, 5, 6, 13 e 14 da imagem 2 (veja destaques em laranja nas colunas C1 e C2 da Tab. 4, respectivamente). Nesta tabela, são apresentadas também as colunas Rel1 e Rel2 com os pontos selecionados das imagens 1 e 2, assim como as suas diferenças, veja coluna Diff. Observe que nesta coluna o valor mais frequente é (4,5), ou seja, existe um deslocamento que mais ocorre entre os pontos encontrados anteriormente das imagens 1 e 2, encontrados através da função moda. Desta forma, também são excluídos o ponto 5 da imagem 1 e o ponto 7 da imagem 2, veja destaque em amarelo na Tab. 4.

Tab. 3: Comparações entre as tabelas de distâncias das imagens 1 e 2.

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 10 | 1 | 2 | 1 | 4 | 4 | 5 | 2 | 2 | 2 | 2 | 2 | 2 | 3 |
| 2 | 1 | 2 | 1 | 12 | 1 | 3 | 5 | 5 | 3 | 2 | 2 | 2 | 2 | 2 | 2 |
| 3 | 3 | 4 | 1 | 2 | 3 | 8 | 2 | 2 | 2 | 2 | 1 | 3 | 2 | 2 | 2 |
| 4 | 2 | 6 | 1 | 5 | 2 | 2 | 9 | 6 | 2 | 2 | 3 | 2 | 1 | 2 | 2 |
| 5 | 1 | 5 | 1 | 3 | 1 | 4 | 9 | 12 | 5 | 3 | 6 | 2 | 1 | 2 | 2 |
| 6 | 2 | 2 | 1 | 1 | 3 | 2 | 1 | 3 | 1 | 1 | 1 | 2 | 3 | 2 | 2 |
| 7 | 2 | 2 | 1 | 3 | 3 | 3 | 3 | 4 | 11 | 2 | 3 | 4 | 2 | 2 | 2 |
| 8 | 2 | 2 | 1 | 2 | 2 | 2 | 4 | 3 | 2 | 11 | 4 | 3 | 2 | 2 | 3 |
| 9 | 1 | 6 | 1 | 3 | 1 | 2 | 5 | 6 | 3 | 4 | 11 | 5 | 1 | 5 | 3 |
| 10 | 1 | 2 | 1 | 2 | 1 | 4 | 3 | 2 | 4 | 3 | 5 | 13 | 2 | 4 | 2 |
| 11 | 1 | 2 | 1 | 2 | 1 | 3 | 2 | 3 | 2 | 3 | 3 | 3 | 1 | 2 | 12 |

Tab. 4: C1 e C2 são os centroides de Im1 e Im2, visualizados na Fig.5. Ref1 e Ref 2 são os pontos homólogos visualizados na Fig. 6 (excluindo os pontos destacados em amarelo). Diff são as diferenças entres os pontos Rel1 e Rel2.

| | Rel1 | | Rel2 | | Diff | |
|---|---|---|---|---|---|---|
| 1 | 76 | 427 | 58 | 457 | 4 | 5 |
| 2 | 142 | 437 | 125 | 465 | 4 | 5 |
| 3 | 304 | 248 | 287 | 276 | 4 | 5 |
| 4 | 326 | 268 | 287 | 276 | 6 | 3 |
| 5 | 326 | 268 | 309 | 297 | 4 | 5 |
| 6 | 442 | 464 | 425 | 493 | 4 | 5 |
| 7 | 578 | 160 | 560 | 189 | 4 | 5 |
| 8 | 604 | 136 | 586 | 163 | 4 | 5 |
| 9 | 606 | 478 | 587 | 508 | 4 | 5 |
| 10 | 842 | 346 | 824 | 375 | 4 | 5 |

| | C1 | | C2 | |
|---|---|---|---|---|
| 1 | 76 | 427 | 6 | 437 |
| 2 | 142 | 437 | 58 | 457 |
| 3 | 249 | 267 | 107 | 384 |
| 4 | 304 | 248 | 125 | 465 |
| 5 | 326 | 268 | 214 | 490 |
| 6 | 433 | 189 | 231 | 297 |
| 7 | 442 | 464 | 287 | 276 |
| 8 | 578 | 160 | 309 | 297 |
| 9 | 604 | 136 | 425 | 493 |
| 10 | 606 | 478 | 560 | 189 |
| 11 | 842 | 346 | 586 | 163 |
| 12 | | | 587 | 508 |
| 13 | | | 653 | 491 |
| 14 | | | 740 | 318 |
| 15 | | | 824 | 375 |

### *F. Geração do arquivo de Pontos Homólogos*

Uma vez selecionados os pontos homólogos, foram calculadas suas coordenadas geográficas UTM (*Universal Transversa de Mercator*), levando em conta a resolução espacial e as coordenadas das bordas das imagens, que foram encontrados nos metadados dos arquivos de formato *tif*. O algoritmo então gera um arquivo de extensão **.cor*, que é compatível com o arquivo de correspondência utilizada no aplicativo de correção geométrica/registro do software IDRISI. Este arquivo **.cor* é armazenado em código ASCII e tem a seguinte estrutura: em sua primeira linha é informado o número de pontos de correspondência e nas demais são apresentadas as coordenadas da imagem de ajuste e da imagem de referência. O arquivo de correspondência foi importado no software IDRISI para a realização do registro da imagem, através da ferramenta RESAMPLE. O modelo de transformação utilizado foi o linear com reamostragem pelo vizinho mais próximo, já implementado no IDRISI. A qualidade do arquivo de correspondência foi avaliada através do resíduo médio quadrático (*RMS: Root Mean Square*) total.

## III. Resultados e Discussões

Foram obtidos 9 pontos homólogos para o experimento da Ilha Grande, sendo que inicialmente foram encontrados 11 pontos na imagem de 2003 e 15 na imagem de 2010 e, visualmente, foi possível identificar os pontos homólogos. Assim, o algoritmo utilizado para encontrar a correspondência entre os pontos foi satisfatório. É importante notar que os pontos homólogos estão bem distribuídos pela imagem, o que é essencial para o registro.

A Fig. 7 apresenta a sobreposição entre a banda 4 de ambas as imagens antes e após o registro. Em verde e magenta aparecem destacadas as regiões em que as cenas de 2003 e 2010 não se sobrepõem, deixando evidente o deslocamento que há entre as cenas antes do registro e sua posterior correção (observações análogas podem ser vistas nas Figuras 5 e 6).

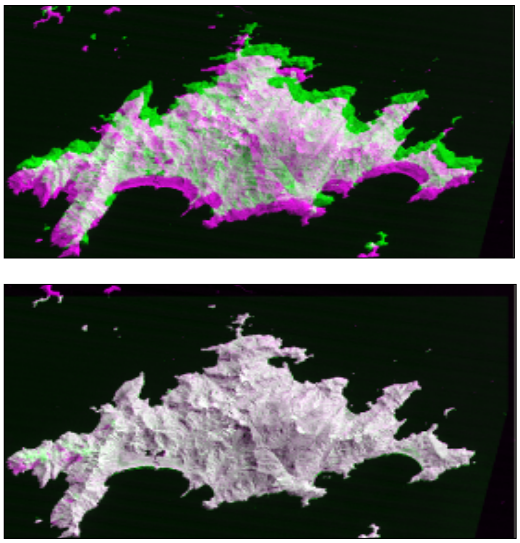


Fig. 7. Sobreposição das imagens antes do registro (acima) e após o registro com a utilização dos pontos homólogos para a Ilha Grande, RJ.

O mesmo algoritmo apresentado na Fig. 2 foi testado em outro par de imagens com características semelhantes (regiões de terra delimitadas por corpos d'água). A área escolhida foi a Represa de Peixoto, localizada nos limites dos municípios de Delfinópolis, Cássia e Passos, situados no estado de Minas Gerais. Foram obtidas imagens de 04/06/1984 e 22/04/2009 capturadas pelo sensor TM a bordo da plataforma Landsat 5 através do site da DGI/INPE. As cenas da órbita ponto 220/74 (sistema WRS - Landsat) foram recortadas através do software IDRISI visando delimitar somente a região da represa. De 15 pontos identificados em cada imagem, foram encontrados 7 pontos homólogos, que foram utilizados para registrar a imagem, cujo resultado final é apresentado na Fig. 8.

Estes experimentos podem ser reproduzidos no Matlab através do código disponível no seguinte endereço na internet: http://vision.ufabc.edu.br/RemoteSensing.

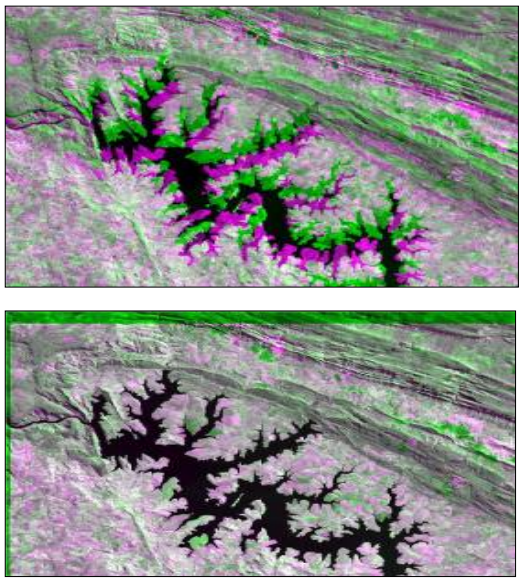


Fig. 8. Sobreposição das imagens antes do registro (acima) e após o registro com a utilização dos pontos homólogos para a Represa de Peixoto, MG.

## IV. Conclusão

Nesse artigo foi possível desenvolver um algoritmo para encontrar pontos homólogos entre duas imagens para a realização de um registro satisfatório. O método desenvolvido inicialmente foi aplicado em um par de imagens de uma ilha e, posteriormente, testado em imagens de uma represa; em ambos os casos se mostrou eficiente na identificação de pontos homólogos bem distribuídos. Nos testes realizados com as imagens da represa notou-se a necessidade de alterar o tamanho da vizinhança utilizada nos filtros de abertura e fechamento morfológicos. Ajustes de limiares também se fazem necessários para outros pares de imagens para que o algoritmo seja aplicável a uma gama maior de situações. O bom resultado obtido deverá se repetir caso o algoritmo seja aplicado a outros conjuntos de imagens com características semelhantes.

## Referências


[1] Salvador A.C., Oliveira J.C., Mello M.P. "Registro automático de imagens utilizando pixels de baixa freqüência de ocorrência". In: Anais do XIV Simpósio Brasileiro de Sensoriamento Remoto (SBSR 2009), Natal, RN: INPE. p. 7127-7134, 2009.

[2] Fedorov D. "Sistema semi-automático de registro e mosaico de imagens". 155 p. (INPE-9582-TDI/838). Dissertação (Mestrado em Computação Aplicada) - Instituto Nacional de Pesquisas Espaciais, São José dos Campos. 2002

[3] Bunting P., Labrosse F., Lucas R.M. "A multi-resolution area-based technique for automatic multi-modal image registration". Image Vision Computing, 28 (8), p. 1203-1219, 2010.

[4] Zitova B., Flusser J. "Image registration methods: a survey," Image Vision Computing, vol. 21(11), p. 977-1000, 2003.

[5] Jensen, J. R. Introductory digital image processing: a remote sensing perspective. 3 ed. Upper Saddle River, NJ, USA: Pearson Prentice Hall, 2005, 526 p.

[6] IDRISI. Disponível em www.idrisi.com.br. Acesso em Maio de 2014.

# Comparação de Métodos de Interpolação de Imagens Digitais.

Wesley Dourado
Universidade Estadual Paulista - UNESP
Programa de Pós-graduação em Matemática Aplicada e Computacional - FCT
Presidente Prudente, SP - Brasil

Aylton Pagamisse
Universidade Estadual Paulista
UNESP - Departamento de Matemática e Computação - FCT
Presidente Prudente, SP - Brasil

Marco A. Piteri
Universidade Estadual Paulista
UNESP - Departamento de Matemática e Computação - FCT
Presidente Prudente, SP - Brasil

***Resumo*—In this paper a comparative study of some types of interpolation algorithms for digital images is performed. This paper presents the methods that have improvements in comparison with the classical methods bilinear and bicubic interpolation. All methods evaluated are compared visually and quantitatively using statistical metrics such as MSE, SNR, and the IQI.**

## I. Introdução

Inúmeras são as aplicações dos métodos de interpolação em processamento de imagens digitais. Entre elas estão o uso no processo de redimensionamento de escala (*zoom*) e em métodos de rotação de imagens. O processo de aumento de escala (*zoom in*) tanto quanto a rotação de imagens são muito utilizados em dispositivos móveis e em programas de edição de fotos, em sensoriamento remoto, computação gráfica entre outros.

Os métodos de interpolação de imagens mais clássicos encontrados na literatura de processamento de imagens são: método da replicação, bilinear e bicúbico [1], [2]. Em geral o método mais utilizado por *softwares* comerciais é o bicúbico.

Os métodos clássicos são técnicas lineares de interpolação de imagens. Neste artigo são apresentados técnicas não-lineares. Os algoritmos desta técnica melhoram a qualidade da imagem utilizando restrições para cada característica da própria imagem. Os métodos destes tipos de técnicas comparados são: o método NEDI, iNEDI, ICBI, o método de interpolação utilizando o filtro bilateral e um método de interpolação baseado na transformada wavelet.

Todos os métodos apresentados ampliam a imagem por um fator de $2^k$, com $k \in \mathbb{N}$ e foram implementados no ambiente Matlab.

### A. Problemas da Interpolação de Imagens.

A interpolação é, basicamente, o processo que utiliza dados conhecidos para estimar valores em pontos desconhecidos. Dependendo do tipo de interpolação que é utilizada, a imagem resultante pode apresentar alguns artefatos. É o caso de *jaggies*, que causa um serrilhamento nas bordas da imagem e o *blurring*, que resulta em uma imagem de aspecto borrado ou desfocado (Figure 1).

Com o intuito de ampliar a imagem sem introduzir estes artefatos, foram propostos vários algoritmos de interpolação, que serão apresentados a seguir.

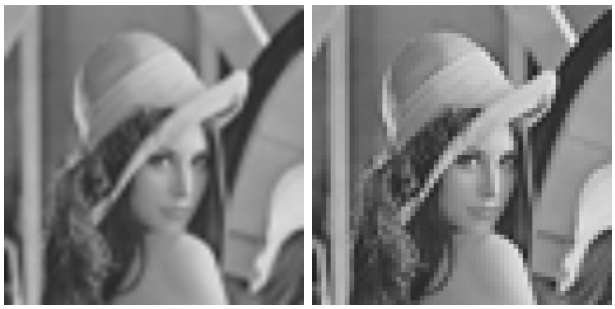


Figura 1. (a) Imagem com *blurring*. (b) Imagem com *jaggies*.

## II. Métodos Avaliados neste Trabalho.

Nesta seção serão apresentados os métodos de interpolação estudados e avaliados. O primeiro método é o New Edge-Direction Interpolation (NEDI) proposto por Li [6]. Este método utiliza um cálculo de estimativa da covariância local entre os pixels da imagem de baixa e alta resolução como peso da média ponderada dos pixels vizinhos. Melhorias deste algoritmo é feito por Giachetti [7], [8] que também apresenta o método de interpolação iterativa com base em curvatura [9]. Na subseção II-D é apresentado o método adaptativo de super-resolução de frame único com wavelets redundantes proposto por Pagamisse [4]. Logo após, na subseção II-E, o último algoritmo avaliado é método de interpolação utilizando o filtro bilateral.

### A. New Edge-Direction Interpolation.

O método NEDI, tem como metodologia interpolar cada ponto da imagem de alta resolução ($Y_{2i,2j}$) utilizando uma estimativa da covariância da imagem de baixa resolução ($X_{i,j}$). Para obter os valores dos novos pixels, o algoritmo realiza o seguinte cálculo (Equação 1):

$$\hat{Y}_{2i+1,2j+1} = \sum_{k=0}^{1}\sum_{l=0}^{1} \alpha_{2k+l} Y_{2(i+k),2(j+l)} \qquad (1)$$

em que $\hat{Y}_{2i+1,2j+1}$ são aqueles que possuem pixels originais nas diagonais. Já o coeficiente $\alpha_{2k+l}$ é o peso estimado conforme a Equação 2.

$$\vec{\alpha} = R_{yy}^{-1}\vec{r}_y \qquad (2)$$

Na Equação 2, tem-se $\alpha = [\alpha_0, \alpha_1, \alpha_2, \alpha_3]$, $R_{yy} = [R_{kl}]$ com $0 \leq k, l \leq 3$ e $r_y = [r_0, r_1, r_2, r_3]$. Os termos $[\alpha_0, \alpha_1, \alpha_2, \alpha_3]$, são os coeficientes a serem calculados utilizando os quatro pixels originais que estão nas diagonais. O termo $R_{kl}$ é a covariância-local entre dois pontos originais vizinhos do ponto interpolado. Já o termo $r_y$, por sua vez, é o cálculo da covariância-local entre o ponto interpolado e os vizinhos mais próximos da imagem de baixa resolução. Por exemplo, $R_{03} = E[Y_{2i,2j}Y_{2i,2j+2}]$ e $r_0 = E[Y_{2i,2j}Y_{2i+1,2j+1}]$, em que a função $E[XY]$ é a esperança entre duas variáveis $X$ e $Y$. Observe a Figura 2.

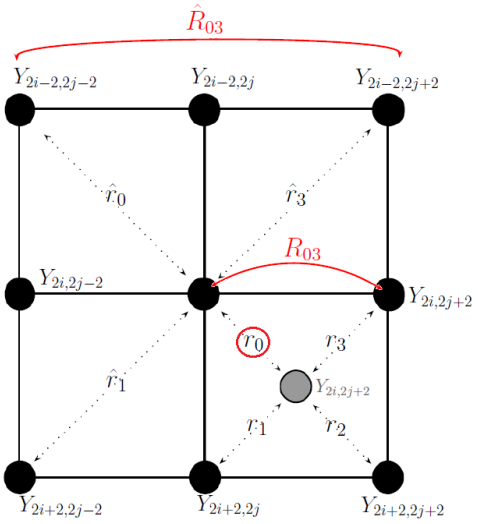



Figura 2. Representação do cálculo dos termos $R_{kl}$, $\hat{R}_{kl}$, $r_k$ e $\hat{r}_k$. Os pontos pretos são os pixels originais enquanto o ponto cinza é o pixel que será interpolado pelo método. Fonte: Adaptado a partir de Li [6].

A covariância $r_0$ circulado na Figura 2 precisa do valor $Y_{2i+1,2j+1}$ para ser calculada. Contudo, não é conhecido o valor da intensidade deste pixel e portanto é necessário estimar o valor de $r_k$. Para isto, ao invés de utilizar $r_k$, é calculado o valor de $\hat{r}_k$ com os valores dos pixels da imagem original. Desta maneira, o ponto $Y_{2i+1,2j+1}$ é interpolado utilizando-se $\hat{r}_k$ e $\hat{R}_{kl}$ no calculo do coeficiente $\alpha$. Segundo Li [6], $\hat{R}_{kl} = \frac{1}{M^2} C^T C$ e $\hat{r}_k = \frac{1}{M^2} C^T \vec{y}$, em que $\vec{y} = [y_1, \ldots, y_k, \ldots, y_{M^2}]$ é o vetor que contém todos os pixels da janela local. A matriz $C$ contém em cada linha o valor dos quatro vizinhos da diagonal de cada $y_k \in \vec{y}$.

Todos os coeficientes $\alpha_i$s podem ser resolvidos através do método de mínimos quadrados. Desta forma, o valor $Y_{2i+1,2j+1}$ é interpolado com cada valor de $\alpha$ substituída na Equação 1. O método NEDI aplica a interpolação adaptativa baseada na covariância apenas nas áreas ao redor de bordas. Em regiões homogêneas, o algoritmo aplica a interpolação bilinear.

Depois de calculado todos os pontos centrais $Y_{2i+1,2j+1}$, o algoritmo precisa interpolar os pontos do tipo $Y_{2i+1,2j}$ e os $Y_{2i,2j+1}$. Para isto, o algoritmo faz uma rotação de $45^o$ e aplica a Equação 1 nestes pixels. Observe que nesta posição os pontos originais (pontos pretos) ficam na posição vertical e os estimados no primeiro passo (pontos cinzas) estão na posição horizontal. A estimativa dos termos $R_{kl}$ e $r_k$ são calculados da mesma maneira.

### B. *Improved New Edge Directed Interpolation.*

O algoritmo NEDI tem como resultado a preservação da forma contínua das bordas, contudo, pode introduzir artefatos indesejáveis na imagem resultante. Com o intuito de melhorar este método e resolver este problema, Giachetti [7] propõe uma modificação no método e apresenta o *Improved New Edge Directed Interpolation* (iNEDI). Pode-se listar as seguintes limitações do método NEDI:

- A suposição da covariância estacionária local é violada em alguns casos;
- Introduz artefatos como *Jaggies* e distorções em regiões de altas frequências;
- O sistema é frequentemente mal condicionado. O uso de largas janelas melhora o condicionamento da matriz $C^T C$, porém, isto suaviza a imagem, dando a aparência de imagem borrada;
- Valores de pixels interpolados trocam com o brilho global;

Para resolver este problemas, o algoritmo iNEDI propõe as seguintes mudanças:

- A primeira modificação está no tamanho da janela utilizada. De acordo com Giachetti [7], utilizar a janela quadrada pode introduzir artefatos direcionais e de qualquer modo faz com que o algoritmo seja não isotrópico. Em análise de imagens um operador isotrópico é aquele que responde de forma igual a variações de intensidade sem qualquer preferência por uma orientação. Os artefatos podem ser reduzidos ao calcular os parâmetros em janelas aproximadamente circulares, ou seja, janelas que variam seu formato de acordo com a posição do pixel interpolado;
- O problema do mal condicionamento é consequência do fato que a matriz $C$ pode ter uma linha ou coluna como combinação linear das outras. Neste caso o determinante da matriz $C$ é igual a zero e sua solução dos mínimos quadrados não é única. Para resolver esta questão, considere o valor interpolado é dado por $I_{2i+1,2j+1} = \vec{\alpha}^* \cdot \vec{I}_4$, em que $\vec{I}_4 = \vec{I}_0 + \vec{I}_{err}$, com $\vec{I}_{err}$ o erro. Desta maneira, o método escolhe como única, a menor norma de solução para $\vec{\alpha}^*$;
- Os valores dos pixels interpolados no método NEDI sofrem alterações no brilho. Isto ocorre porque eles não dependem somente das diferenças entre os valores dos vizinhos mas também do seus valores absolutos. Para resolver este problema, é subtraído de cada valor da matriz $C$ a média da intensidade dos quatros vizinhos dos valores dos termos pertencente a $C$;

- Ao invés de utilizar a interpolação bilinear, que suaviza demais as regiões homogêneas, o método iNEDI utiliza a interpolação bicúbica, o que lhe dá regiões suaves mais nítidas;
- Um outro problema do método NEDI é garantir que os pontos da janela utilizada na estimação do pixel interpolado estejam totalmente inseridos na mesma borda. O fato de pixels de regiões distintas estarem na mesma janela, pode ocasionar uma interpolação equivocada e introduzir artefatos indesejáveis nas bordas. Para solucionar este problema, é colocado um limiar substituindo qualquer valor interpolado fora da faixa de intensidade dos quatro vizinhos com o mais próximo dos valores que delimitam esse intervalo.

### C. *Iterative Curvature-Based Interpolation.*

O método *iterative curvature-based interpolation* (ICBI) também é proposto por Giachetti [7], [9] e tem como base um preenchimento em duas etapas e uma correção iterativa dos pixels interpolados obtidos através da minimização de uma função objetiva, dependendo das derivadas direcionais de segunda ordem da intensidade da imagem. Este método utiliza restrições para derivar as funções relacionadas com o método NEDI, contudo é computacionalmente mais vantajosa por sua velocidade. O método ICBI amplia a imagem duas vezes o tamanho original. Para o preenchimento dos novos pixels, o primeiro passo é calcular as aproximações da segunda ordem de derivadas denotadas por $\tilde{I}_{11}(2i+1,2j+1)$ e $\tilde{I}_{22}(2i+1,2j+1)$, dada pela Equação 3, em que $I(i,j)$ representa o valor de intensidade da imagem na posição $(i,j)$.

$$\begin{aligned}\tilde{I}_{11}(2i+1,2j+1) &= I(2i-2,2j+2)+I(2i,2j)+\\ &I(2i+2,2j-1)-3I(2i,2j+2)-3I(2i+2,2j)+\\ &I(2i,2j+4)+I(2i+2,2j+2)+I(2i+4,2j)\\ \tilde{I}_{22}(2i+1,2j+1) &= I(2i,2j-2)+I(2i+2,2j)+\\ &I(2i+4,2j+2)-3I(2i+2,2j+2)-3I(2i,2j)+\\ &I(2i-2,2j)+I(2i,2j+2)+I(2i+2,2j+4)\end{aligned} \quad (3)$$

Depois de calculados os termos $\tilde{I}_{11}(2i+1,2j+1)$ e $\tilde{I}_{22}(2i+1,2j+1)$, é atribuído ao ponto $I(2i+1,2j+1)$ o valor da média dos dois vizinhos que possui a menor derivada direcional,ou seja média entre $I(2i,2j)$ e $I(2i+2,2j+2)$ se $\tilde{I}_{11}(2i+1,2j+1) < \tilde{I}_{22}(2i+1,2j+1)$ ou entre $I(2i+2,2j)$ e $I(2i,2j+2)$ caso contrário.

Depois deste processo, o método ainda calcula o valor de três componentes para amenizar artefatos como *blurring* ou *jaggies* que a interpolação possa ocasionar. Os componentes são:

- O primeiro componente é o termo que verifica a curvatura de continuidade ($U_c$). Este termo diminui qualquer suavidade que possa ocorrer na imagem;
- O segundo componente é o termo chamado de curvatura de realce ($U_e$). Este termo torna a imagem mais nítida;
- O último termo apresenta um método para melhorar a aparência da imagem em consideração a suavidade dos contornos ($U_i$).

Após ter calculado todos os termos, o algoritmo os reúne e, a este é somado ou subtraído um pequeno valor $\delta$ para cada pixel $I_{2i+1,2j+1}$. A menor intensidade destes é atribuído ao pixel interpolado. Depois de todo este processo, é repetido para os pontos do tipo $I_{2i+1,2j}$ e $I_{2i,2j+1}$ utilizando os pontos na vertical e horizontal.

### D. *Método Adaptativo de Super-resolução de Frame Único com Wavelets Redundantes.*

Assim como no processo de decimação, a transformada wavelet diádica também separa as informações em suave e em detalhes. A imagem de entrada é separada em três sub-bandas; a camada suave, a de informações nas bordas horizontais e nas bordas verticais. Observe a Expressão 4.

$$\begin{aligned}W^1_{2^{j+1}}f &= S_{2^j}f * (G_j \cdot I)\\ W^2_{2^{j+1}}f &= S_{2^j}f * (I \cdot G_j)\\ S^1_{2^{j+1}}f &= S_{2^j}f * (H_j \cdot H_j)\end{aligned} \quad (4)$$

o termo $I$ é o filtro de Dirac e a operação $*$ é a convolução. $H(n)$ é o filtro passa-baixa e $G(n)$ o filtro passa-alta ambos correspondentes a uma função suavizante $\varphi(t)$ e à wavelet mãe $\psi(t)$, respectivamente.

O método utiliza um frame único para sintetizar uma imagem em uma grade mais fina do que a original reduzindo a perda de informação de detalhes presentes. Para tal utiliza-se a expansão wavelet para realizar a interpolação juntamente com o cálculo de pixels, simulando um deslocamento sub-pixel na diagonal da imagem.

A transformada wavelets utilizada foi desenvolvida por Mallat [10] e é conhecida como transformada wavelet diádica discreta, em que, segue o processo de níveis sem a decimação. Ou seja, a cada nível a quantidade de informação da original é mantida em cada camada, obtendo assim uma redundância de informações.

Este algoritmo utiliza também o método adaptativo proposto por Battiato [3] para a interpolação dos pixels centrais da forma $I_{2i+1,2j+1}$.

Após este processo, os pontos $I_{2i+1,2j}$ e $I_{2i,2j+1}$ ainda precisam serem interpolados. Para isto, o algoritmo faz uma rotação de $45^o$ e aplica a transformada wavelet diádica na imagem obtendo três bandas: a versão suavizada, e duas de detalhes (horizontal e vertical). Depois deste processo de rotação, é aplicado ao redor dos pixels vazios o cálculo da média dos pontos de menor variância, proposto por Ward [11].

Este cálculo consiste em construir conjuntos $S_k$ definidos pela Expressão 5:

$$\begin{aligned}S_1 &= \{P_1,P_2,P_3\}, \quad S_2 = \{P_1,P_2,P_4\}\\ S_3 &= \{P_1,P_3,P_4\}, \quad S_4 = \{P_2,P_3,P_4\}\end{aligned} \quad (5)$$

desta maneira, verifica-se qual dentre os conjuntos $S_{k's}$ com $k \in \{1,2,3,4\}$ possui os pontos $P_k$ de tal maneira que $k = arg\ \ min_j \{var\,(S_j)\}$ e $j \in \{1,2,3,4\}$.

Por fim, a imagem é rotacionada novamente a sua posição inicial e é aplicado a transformada wavelet inversa, obtendo a imagem final.

### *E. Método de Interpolação utilizando o Filtro Bilateral.*

Uma nova abordagem que utiliza o filtro bilateral foi proposto por Han [5] e seu método interpola a imagem e ainda realça as bordas filtrando alguns ruídos nas regiões homogêneas. A ideia do algoritmo é utilizar o filtro bilateral para separar a imagem em duas camadas. Uma possuirá toda a informação dos detalhes da imagem original, enquanto a outra terá toda a informação de baixa frequência, ou seja, as regiões suaves. O método interpola ambas as camadas de maneira específica e no final as une com um realce nos detalhes.

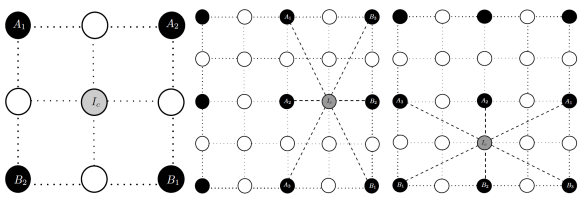
O primeiro passo do algoritmo é aplicar o filtro bilateral na imagem, obtendo a camada base. A imagem original subtraída desta camada base fornece a camada de detalhes. Para reduzir o realce dos ruídos e assim, melhorar a imagem interpolada, é feito uma filtragem na camada de detalhes. Para isto é utilizada a Expressão 6.

$$\hat{I}_d(x) = \frac{|E_b(x)|^2}{|E_b(x)|^2 + \gamma_1 |E_d(x)|^2} E_d(x) \tag{6}$$

em que, $E_b$ e $E_d$ são os gradientes das energias das camadas base e detalhe, respectivamente; e $\gamma_1$ é um fator de regularização de controle entre a redução de ruídos e a preservação da textura da imagem definido pela Equação 7:

$$\gamma_1 = \frac{\sum |E_b(x)|^2}{\sum |E_d(x)|^2} \tag{7}$$

Após a filtragem, ambas as camadas são expandidas individualmente. Para isto o método utilizado por Han [5] é a interpolação de preservação de bordas. Para isto observe a Figura 3.




(a) $1^o$ Configuração Pixels (b) $2^o$ Configuração Pixels (c) $2^o$ Configuração Pixels

Figura 3. Representação das configurações do algoritmo para o cálculo das intensidades de pixels na presença de borda. Fonte: Imagem retirada e adaptada a partir de Han [5].

Como a imagem é expandida pelo fator de dois, a primeira configuração está em interpolar os pontos centrais $I_{2i,2j}$. A segunda e terceira configuração são para os pontos $I_{2i+1,2j}$ e $I_{2i,2j+1}$. Para esta interpolação é feito o seguinte cálculo (Equação 8):

$$d_a^* = arg\ \ min_{d\in 1,2}(|A_d - B_d|) \tag{8}$$

em que $d_a^*$ representa a borda direcional desta primeira configuração. Desta maneira, é verificado qual diferença é menor, se $|A_1 - B_1|$ ou $|A_2 - B_2|$. O valor de $I_c$ é a média aritmética dos pontos pertencentes à menor intensidade (9).

$$I(\cdot) = \frac{A_{d_a^*} + B_{d_a^*}}{2} \tag{9}$$

Os pontos centrais da Figura 3 possuirá a média dos pontos mais próximos de intensidade de tons de cinza entre si. Se houver uma borda passando por $A_1$ e $B_1$ por exemplo, a diferença $|A_1 - B_1|$ será muito menor do que $|A_2 - B_2|$.

Depois de todo este processo, junta-se as duas camadas (Equação 10) com os detalhes realçados.

$$\hat{I}_{final}(x) = \hat{I}_b(x) + \lambda(x)\hat{I}_d(x) \tag{10}$$

em que $\hat{I}_b(x)$ e $\hat{I}_d(x)$ são, respectivamente, a camada base e a camada de detalhes interpolados. A função $\lambda(x)$ é um fator peso utilizado para realçar as informações na camada de detalhes (onde estão as bordas).

## III. Experimentos e Resultados.

Nesta seção é realizada a comparação dos métodos discutidos nas subseções anteriores. Esta comparação é feita de duas maneiras: a visual e a estatística.

Todas as imagens utilizadas foram imagens monocromáticas. Contudo, todos os métodos podem ser aplicados em imagens coloridas. Para isto, basta utilizar o modelo de cores YUV (Pedrini [2]). A imagem colorida RGB, ao ser convertida em YUV, é separada em três componentes, eles são: a luminância (Y), e a crominância (U e V). Toda a informação fundamental dos detalhes da imagem está na luminância, logo basta interpolar o componente Y utilizando o método desejado. Para os componentes U e V pode-se utilizar um método mais simples, como por exemplo a replicação de pixels e refazer a imagem no modelo RGB.

### *A. Análise Visual.*

Para a comparação visual, foram aplicados os métodos vistos em diversas imagens com distintas texturas. Observe as Figuras 4 e 5 dois exemplos dos resultados obtidos.

Na Figura 4(a) está a imagem da Lena de tamanho 512x512. A região retirada para a comparação é de tamanho 128x128 e inclui partes homogêneas como o ombro e a bochecha e partes ricas em detalhes, como as penas e bordas do chapéu.

A Figura 4(b) apresentam bordas bem preservadas e contínuas. O problema do NEDI nesta aplicação se encontra nas regiões que delimitam a imagem. Observe que as penas e ao redor do olho são introduzidas artefatos como se fossem contornos adicionais. Na Figura 4(c) estes problemas não aparecem mais, pois o iNEDI consegue conservar as informações sem adicionar estes tipos de ruídos.

O algoritmo ICBI é capaz de manter as informações contidas na imagem original e sem que o resultado perca o aspecto

(a) Lena (b) NEDI (c) iNEDI

(d) ICBI (e) Wavelets (f) Filtro bilateral

Figura 4. Aplicação dos métodos na imagem Lena.

(a) Babuíno (b) NEDI (c) iNEDI

(d) ICBI (e) Wavelets (f) Filtro bilateral

Figura 5. Aplicação dos métodos na imagem Babuíno.

de naturalidade, como visto na Figura 4(d). Em comparação com o iNEDI, o ICBI consegue conservar melhor os detalhes da pena no chapéu. Na Figura 4(e) está a aplicação do método de interpolação wavelet. Este mantém os detalhes das bordas mas introduz *jaggies*.

Na Figura 4(f) tem-se a imagem da Lena depois da aplicação da interpolação com o filtro bilateral. Os detalhes de bordas são bem preservadas assim como os ruídos nas regiões homogêneas, que são filtradas. Contudo, pela característica de suavizar, alguns detalhes são suprimidos como os riscos transversais do chapéu. Isto também ocorre nas penas e no cabelo.

De maneira geral, os método que exibiram melhores resultados visuais nesta imagem foram; o iNEDI, o ICBI e o bilateral.

A Figura 5(a) possui tamanho 512x512 e dela foi retirada uma região de 128x128 para a aplicação dos métodos.

A Figura 5(b) demonstra a aplicação pelo método NEDI. Este interfere nos detalhes presentes no pelo e introduz ruídos dentro da região do olho do babuíno.

O método iNEDI também não consegue reamostrar os detalhes dos pelos com perfeição (Figura 5(c)); contudo o contorno dos olhos são mais contínuos. Na Figura 5(d), o ICBI produz uma imagem menos artificial. Apesar de contornos e bordas melhores e mais suaves, também introduz ruídos dentro do olho do babuíno. A aplicação do método das wavelets apresenta resultados satisfatórios em imagens que possui muita textura e muitas curvas, e é possível ver este resultado na Figura 5(e)). A interpolação que utiliza o filtro bilateral reduz os defeitos nas regiões mais homogêneas, porém também interfere nas regiões dos pelos, aonde estão concentrados os detalhes.

De maneira geral os métodos que conseguiram interpolar a imagem e manter os aspectos e detalhes do babuíno foram os métodos ICBI e wavelets.

### *B. Análise Estatística.*

Para a comparação estatística, foram utilizadas as métricas MSE (*Mean Square Error*), SNR (*Signal Noise Ration*) e o IQI (*Universal Image Quality Index*). As métricas MSE e SNR são métodos clássicos utilizados na literatura de processamento de imagens (Gonzalez [1] e Pedrini [2]). A métrica IQI pode ser definida pela Equação 11:

$$IQI = \frac{4\mu_x\mu_y\sigma_{xy}}{(\mu_x^2+\mu_y^2)(\sigma_x^2+\sigma_y^2)} \tag{11}$$

em que $\mu_x$ representa a intensidade média estimada de $x$ e $\sigma_x$ é o seu desvio padrão. Segundo Wang [12], o método IQI é mais sensível à distorção da estrutura da imagem do que as métricas clássicas como o MSE.

Tais métricas foram aplicadas nas imagens da Figura 6, e os dados obtidos estão na Tabela I.

(a) Rosto (b) Olho (c) Babuíno (d) Nariz

Figura 6. Imagens utilizadas na comparação estatística.

Pela Tabela I, a Figura 6(a) possui os melhores resultados estatísticos com o método da wavelet com $MSE = 129,4$ seguido do bilateral ($MSE = 143,4$). Pela métrica IQI, as melhores interpolações ocorrem com os métodos ICBI, com $IQI = 0,851$ seguido da wavelet ($IQI = 0,843$). O bilateral ($IQI = 0,843$) e iNEDI ($IQI = 0,842$) possuem valores próximos.

Sobre a Figura 6(b), a interpolação ICBI e iNEDI apresentam os melhores resultado pela métrica MSE. Contudo as piores estão com a wavelet e a bilateral. Pelo IQI, a melhor é o método ICBI com $IQI = 0,865$. Os métodos que não conseguiram aproximar a imagem resultante com a original

foram o método da wavelet ($IQI = 0,780$) e da bilateral ($IQI = 0,783$).

Tabela I: Comparação de todos os métodos de interpolação.

| Imagem | Método | MSE | SNR | IQI |
|---|---|---|---|---|
| Rosto | NEDI | 157.015 | 20.03 | 0.839 |
| | iNEDI | 151.295 | 20.19 | 0.842 |
| | ICBI | 145.057 | 20.38 | 0.851 |
| | Wavelet | 129.431 | 20.87 | 0.847 |
| | Bilateral | 143.417 | 20.43 | 0.843 |
| Olho | NEDI | 200.790 | 20.28 | 0.845 |
| | iNEDI | 189.815 | 20.52 | 0.853 |
| | ICBI | 174.863 | 20.88 | 0.865 |
| | Wavelet | 628.596 | 15.32 | 0.780 |
| | Bilateral | 419.581 | 17.07 | 0.783 |
| Babuíno | NEDI | 757.078 | 13.67 | 0.603 |
| | iNEDI | 743.484 | 13.75 | 0.618 |
| | ICBI | 724.259 | 13.86 | 0.646 |
| | Wavelet | 709.961 | 13.95 | 0.637 |
| | Bilateral | 734.647 | 13.80 | 0.613 |
| Nariz | NEDI | 399.899 | 16.59 | 0.712 |
| | iNEDI | 393.051 | 16.66 | 0.723 |
| | ICBI | 393.358 | 16.66 | 0.736 |
| | Wavelet | 379.745 | 16.81 | 0.726 |
| | Bilateral | 392.507 | 16.67 | 0.710 |

Os métodos wavelet ($MSE = 709,9$) e ICBI ($MSE = 724,2$) apresentam os melhores resultados estatísticos pela métrica MSE da Figura 6(c). O mesmo ocorre pela métrica IQI com o ICBI apresentando $IQI = 0,646$ e wavelet $IQI = 0.637$.

Na Figura 6(d), em relação à métrica MSE os algoritmos mais eficientes é a wavelet ($MSE = 379,7$). Todos os outros apresentaram valores muito próximos. Em comparação com o IQI, os métodos mais eficientes é o método ICBI ($IQI = 0,736$).

De maneira geral, em comparação quantitativa, pode-se afirmar que os métodos ICBI e método de interpolação baseado em wavelets redundantes são os que apresentaram os melhores resultados.

## IV. Conclusões

Estre trabalho teve como objetivo apresentar alguns métodos recentes de interpolação de imagens e compará-los. Foi possível observar que alguns métodos como ICBI e o método das wavelets possuíram resultados mais satisfatórios na imagem do babuíno, cuja textura é cheia de detalhes e bordas em diversas direções, tanto na comparação visual quanto na quantitativa. Contudo, apesar das métricas estatísticas apresentarem estes métodos como os melhores, na imagem Lena, visualmente os métodos que se destacam são o iNEDI, ICBI e o bilateral. Esta imagem possui menos texturas do que o babuíno e bordas mais contínuas.

Por meio destes resultados é possível dizer que o método ICBI apresenta uma boa interpolação levando em consideração o aspecto visual, quantitativo e custo computacional, pois ele possui a vantagem de ser mais rápido do que os algoritmos NEDI e iNEDI. O método bilateral também é promissor, contudo a aplicabilidade em imagens com muitas texturas pode degradá-las um pouco com o efeito *blurring*, que é característica do próprio filtro utilizado. O método das Wavelets redundantes traz resultados satisfatórios visual e estatisticamente, porém ainda introduz *jaggies* em bordas contínuas em uma direção, como é o caso da borda do chapéu da imagem Lena.

É importante enfatizar que os métodos NEDI e iNEDI possuem melhores performance em imagens que apresentam muitas bordas contínuas e orientadas. Por exemplo, bordas que estejam na vertical e horizontal. Imagens que possuem bordas não direcionais, ou seja, curvas que não seguem uma direção contínua, podem apresentar melhores resultados quando aplicados os métodos que utilizam o filtro bilateral ou a transformada Wavelets. Logo, cada método possui eficiência de interpolação dependendo do tipo de textura presente. Somente o método ICBI apresentou bons aspectos visuais independente das características das imagens.

Para trabalhos futuros, o estudo de novas métricas estatísticas que melhor reflitam as diferenças entre duas imagens e que levem em consideração a percepção da visão humana é primordial. Para isso, o desenvolvimento de novas métricas, ou o aperfeiçoamento das existentes, requer pesquisas mais aprofundadas sem desconsiderar os fatores que envolvam a forma de interpretar a imagem visualizada pelo sistema visual humano. Desta maneira será possível obter métricas mais fiéis ao aspecto visual, e isso é essencial para a comparação e escolha dos métodos mais acurados.

## Referências


[1] Rafael C. Gonzalez. *Processamento Digital de Imagens. Tradução: Cristina Yamagami e Leonardo Piamonte.*, 3rd ed. São Paulo: Pearson Prentice Hall, 2010.

[2] Hélio Pedrini. *Análise de Imagens Digitais. Princípios, Algoritmos e Aplicações.*, 3rd ed. São Paulo: Thomson Learning, 2008.

[3] S. Battiato and G. Gallo and F. Stanco. *A Locally-Adaptive Zooming Algorithm for Digital Images.*, vol. 20, pages 805-812. September, Italy.Image and Vision Computing, 2002.

[4] A. Pagamisse and Leila M. G. Fonseca. *Método Adaptativo de Superresolução de Frame único com Wavelets Redundantes.*, pages 230 - 235, Julho, Anais do VI Workshop de Visão Computacional, 2010.

[5] J. Han, Woo and Jun-Hyung Kim. *A Novel Image Interpolation Method Using the Bilateral Filter.*, pages 175 - 181 , February. São Paulo: IEEE Consumer Electronics Society, 2010.

[6] X. Li. *New Edge-Directed Interpolation.*, October. São Paulo: IEEE Trans. on Image Processing, 2001.

[7] Andrea Giachetti and Nicola Asuni. *Fast artifact free image interpolation from a single image.*, In Proc. BMVC, 2008.

[8] Andrea Giachetti and Nicola Asuni. *Accuracy improvements and artifacts removal in edge based image interpolation.*, Proc. 3rd Intl. Conf. on Computer Vision Theory and Applications, 2008.

[9] Andrea Giachetti and Nicola Asuni. *Real time artifact-free image upscaling.*, pages 2760 - 2768. Image Processing, IEEE Transactions on, 2011.

[10] S. Mallat, and S. Zhong. *Characterization of Signals from Multiscale Edges.* IEEE Trans. on PAMI, 1992.

[11] Daniel L. Ward. *Redundant Discrete Wavelet Transform Based Super-Resolution Using Sub-Pixel Image Registration.* Ohio: Air Force Institute of Technology, 2003.

[12] Zhou Wang. *Modern Image Quality Assessment.*, 1rd ed. Texas: Morgan and Claypool, 2006.

# Off-line Signature Verification Using a Mixed Segmentation

William C. Farias, Edson J. R. Justino, Jacques Facon, Luiz E. S. Oliveira, Robert Sabourin
*Pontifícia Universidade Católica do Paraná*
*Universidade Federal do Paraná*
*École de Technologie Supérieure*
*jean.felsky@pucpr.br*
*{facon, justino}@ppgia.pucpr.br*
*lesoliveira@inf.ufpr.br*
*robert.sabourin@etsmtl.ca*

## Abstract

*This paper presents an approach for off-line signatures verification based on a mixed segmentation process. The features used were selected to meet two levels of visual perception of the human cognition and a simple computational implementation. The approach is based on a model of two classes, authorship and not authorship. It uses a dissimilarity concept. The results demonstrate the robustness of the method in determining the three basic types of forgeries, reaching an average accuracy of 93.5%*

## 1. Introdução

A verificação de assinaturas, assim como todas as demais biometrias, é um método baseado em características individuais, tendo como base o fato de que as pessoas possuem maneiras particulares de escrever ou assinar. Este método de análise biométrica possui boa aceitação por ser pouco invasivo e ter baixo custo de aquisição, além de ser aplicável em diversas situações onde as assinaturas já são utilizadas, tais como contratos, cheques bancários, entre outros.

Assinar não é um processo perfeitamente preciso e, por conseguinte, apresenta variabilidades intrapessoais. Essa variabilidade pode ser mais ou menos significativa, dependendo do autor. A única certeza é que, neste domínio, quando duas assinaturas são idênticas, uma delas é considerada cópia da outra [4]. Na figura 1 é possível observar as variabilidades intrapessoais de autores entre os vários exemplares.

Existem basicamente, duas abordagens computacionais em verificação de assinaturas, a chamada *on-line* e a *off-line*. Métodos de verificação *on-line* possuem acesso às informações grafocinéticas do autor durante o processo de escrita. Já na *off-line,* a aquisição das características é feita utilizando uma imagem da assinatura.

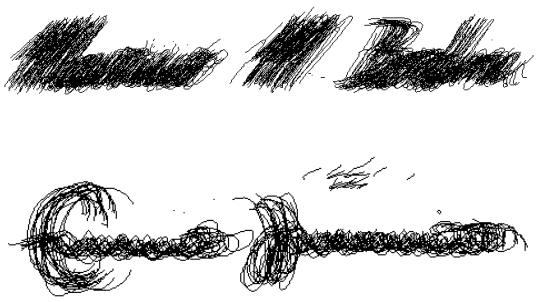


**Figura 1.** Variabilidades intrapessoais. [4]

O processo de verificação de assinaturas *off-line* é o que apresenta os maiores desafios. Usualmente envolvem processos de pré-tratamento complexos, pois exigem a retirada de informações desnecessárias do documento e que podem vir a interferir no desempenho do método de verificação [2]. É comum a presença de carimbos, selos e outros. Outro aspecto relevante está na incapacidade de mensurar, com relativa precisão, características grafocinéticas, tais com a pressão da caneta sobre o papel ou a velocidade imposta pelo autor durante o processo de escrita.

Assim como na etapa de classificação é possível identificar recursos para a melhoria no processo de verificação como um todo, tal como o uso da combinação de classificadores [3,8], na etapa de segmentação e extração de características também o é.

Este artigo apresenta uma abordagem para a verificação de assinaturas *off-line,* com base na combinação de distintos processos de segmentação. O processo utiliza características globais e locais de implementação computacional simples. O método proposto se baseia no uso de um classificador SVM e numa abordagem de geração de modelo independente do escritor [7].

## 2. Segmentação e extração de características

Em verificação de assinaturas a segmentação e extração de características se baseiam principalmente em duas abordagens principais, a contextual e a não contextual [4]. A contextual aborda a assinatura como um texto e busca, dessa forma, segmentar a assinatura em palavras e letras, com o objetivo de identificar padrões de semelhança [2]. Já a não contextual, a assinatura é considerada um desenho produzido pelo escritor, cujas propriedades estão mais vinculadas à forma [2,6]. Esta última, mais abrangente, atende aos requisitos de análise de dois tipos de assinaturas, as cursivas e as rubricas.

A segmentação e extração de características não contextuais podem ocorrer sobre duas formas [4]. A primeira se norteia pela análise da ocupação do espaço gráfico, isto é, na forma em que o escritor distribui a assinatura no espaço destinado a mesma. A segunda se dedica à análise do traçado da assinatura. Em ambos os casos podem-se usar técnicas de extração de características globais, isto é, sem o uso de segmentação [4], ou local, utilizado algum processo de segmentação, tais como uma grade ou *grid* [4, 10].

## 3. Método proposto

O método de verificação de assinatura proposto busca atender aos três tipos básicos de falsificação [4]:

- **A aleatória**: que seria a comparação do espécime original com a de outro escritor;
- **A simples**: que seria a comparação de um espécime original com uma falsificação baseada no nome do escritor;
- **A exercitada**: que seria a comparação de um espécime original com uma falsificação produzida com base na original.

O método é composto por uma etapa de pré-tratamento, onde a assinatura é preparada para a segmentação e extração de características. Essa etapa provê recursos de filtragem morfológicas a fim de atender à etapa de segmentação e extração de características. A etapa de segmentação e extração de características se baseia em dois dos três níveis de percepção visual humana. Essa etapa será descrita em detalhas no tópico seguinte. Por fim, encontra-se a etapa de treinamento e classificação. Essa última utiliza uma abordagem independente do escritor, isto é, o modelo gerado no processo de treinamento é capaz de identificar se uma assinatura é falsa ou não, não importando de que escritor ela tenha vindo. Essa etapa também será descrita em detalhes na seção 3.2.

### 3.1 Extração das características

O método proposto está sedimentado nas duas abordagens de segmentação e extração de características vista anteriormente, a abordagem local e a global. Para a obtenção das características locais adotou-se um processo de segmentação utilizado amplamente pela literatura [4, 2, 10]: o *grid*. O uso do *grid* favorece a obtenção de características oriundas do traçado e também do espaço gráfico em torno da assinatura. Confira na figura 2.

**Figura 2.** Exemplo de grid usado na segmentação.

As características utilizadas nesse trabalho foram selecionadas de forma a atender a dois dos três níveis de percepção visual cognitiva humana [11]. O primeiro trata-se do nível de entrada (*entry level*) ou global e o segundo do nível subordinado (*subordinate level*) ou local. A tabela 1 mostra o conjunto de características e suas propriedades.

**Tabela 1.** Conjunto de características segundo o nível de percepção visual cognitiva humana.

| Característica | Segmentação *pixels* | Nível de Percepção Visual |
|---|---|---|
| Densidade de Pontos | *grid (25x25) a 30%* | entrada |
| Pontos de Máximo, Mínimo e Centro de Gravidade | - | subordinado |
| Inclinação Estrutural | *grid (10x25)* | subordinado |
| Inclinação Global | - *k=4 e L=14* | entrada |
| Inclinação no *Grid* | *grid (4x5) k=4 e L=14* | subordinado |

De todas as características a densidade de pontos seria a mais simples e, por conseguinte, a menos rigorosa do conjunto [4,5] no aspecto de similaridades. A densidade de pontos consiste numa simples contagem dos

pontos dentro de uma célula do *grid*. A incorporação do *grid* nessa característica favorece a análise do espaço gráfico. Em teste preliminares observou-se que existe uma tendência de deformação acentuada do traçado da assinatura, no decorrer do processo. Isto é, quanto mais próximo do final da escrita, maior a deformidade imposta pelo escritor. Confira na figura 3. Com a contagem de pixels e, consequentemente, a análise gradativa do *grid*, em testes preliminares, com as assinaturas alinhadas à esquerda, observou-se que aproximadamente em 30% do *grid* foi estabelecido um ponto de equilíbrio entre as taxas de rejeição das assinaturas genuínas e as taxas de aceitação das falsificações, evidenciando a região da assinatura que melhor a representa. Essa propriedade não foi observada nas outras características.

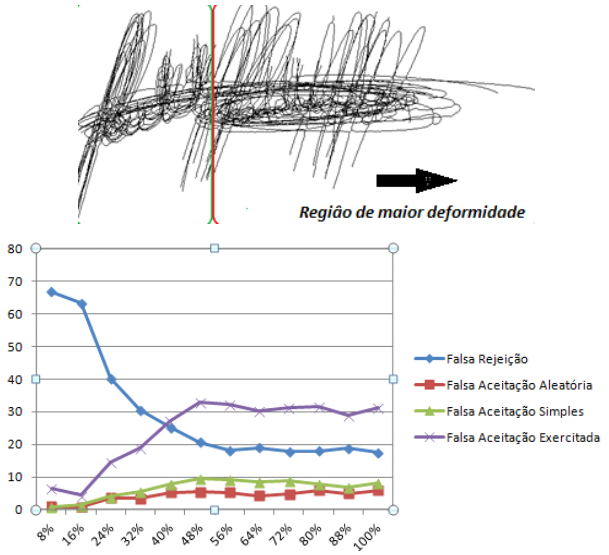



**Figura 3.** Exemplo de deformidade gradativa da assinatura e do melhor ponto de corte para a contagem de pixels (eixo x sendo o percentual da assinatura considerado e o eixo y a taxas de erros).

Para os elementos de ataque e remate foram extraídos os pontos extremos mais à esquerda, mais à direita, mais acima e abaixo da assinatura. Para cada um desses pontos foram calculadas os ângulos internos e a distâncias das arestas em relação aos pontos vizinhos. Também foram calculadas as distâncias e ângulos formados em relação ao centro de gravidade (*CG*) da assinatura e as distâncias entre o ponto de intersecção (*PI*) da reta que liga os pontos mais acima com o mais abaixo e com a reta que ligam os pontos mais a esquerda com o mais a direita. Na figura 4 é possível observar os pontos extremos da assinatura, o polígono formado por estes pontos, o ponto de intersecção das restas e o centro de gravidade da assinatura.

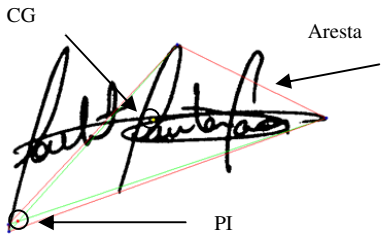



**Figura 4.** Características de Pontos de Máximo, Mínimo e Centro de Gravidade.

A extração da Inclinação Estrutural, já implementada com sucesso por [1,3,4,5], utiliza como método de segmentação o *grid*. A imagem da assinatura esqueletizada é, em cada célula, percorrida por um conjunto de elementos estruturantes que contabiliza a maior ocorrência. Na figura 5 são apresentados os elementos e um exemplo de assinatura esqueletizada.

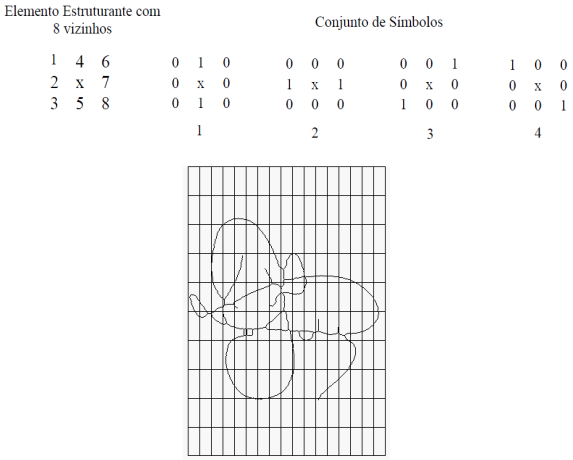



**Figura 5.** Conjunto de elementos estruturantes que representam os tipos de inclinação e uma imagem do esqueleto no *grid*.[4]

A Inclinação Global é uma característica grafocinética que descreve o aspecto dinâmico do traçado e o ângulo de inclinação da escrita [12]. Esta característica foi extraída da imagem do contorno do traçado. Confira na figura 6. A imagem do contorno e percorrida, considerando-se cada pixel preto do traçado no centro do elemento estruturante retangular, figura 7. Em seguida, verificam-se os fragmentos de borda em todas as direções, partindo deste pixel central e conferindo os pixels posteriores com um operador lógico E (*AND*), finalizando nas extremidades do elemento estruturante apenas se houver a presença de um fragmento de borda inteiro. Ou seja, se todos os pixels vizinhos forem pretos, considera-se então o fragmento da borda e calcula-se a posição do fragmento em um vetor de

posições para a construção do histograma que determina a inclinação que pode ser à esquerda, à direita ou nula.

(a) (b)

**Figura 6.** (a) Imagem binária da assinatura, (b) contorno dos traçados.

Foram selecionadas variações do elemento estruturante $k$ = 3, 4, 5 e 10 (distância a partir do central, incluindo-o) ao longo do fragmento no qual, para cada elemento estruturante, são quantificadas as direções de inclinação. Na figura 7, para $k$ = 5, *L possui* 17 direções (ângulos: 0°, 11°, 23°, 34°, 45°, 56°, 68°, 79°, 90°, 102°, 113°, 124°, 135°, 146°, 158°, 169°, 180°), as quais também representam a dimensionalidade do vetor para essa característica. O melhor resultado obtido em experimentos preliminares, com a característica isolada foi $k$=4 e $L$=14.

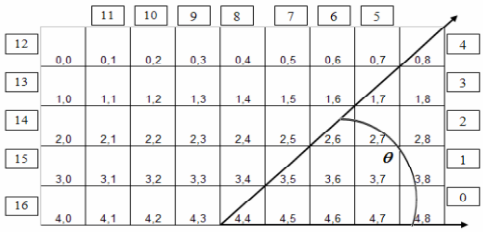



**Figura 7:** Exemplo de um elemento estruturante com comprimento $k$=5 e $L$=17 direções.

Essa característica também foi selecionada utilizando o *grid* como método de segmentação. O processo de extração de características não sofreu alterações. Primeiramente foi realizada uma análise para obtenção da melhor configuração de *grid*. Neste caso, a resolução de 4x5 pixels para o *grid* foi a que apresentou o melhor desempenho, junto com $k$=4 e $L$=14. Neste caso, com a característica isolada.

## 3.2 Cálculo da dissimilaridade

O conceito de dissimilaridade foi proposto por Cha [7] com o objetivo de transformar um problema de múltiplas classes em um problema de duas classes. A dissimilaridade é obtida através da diferença entre as características observáveis de dois objetos. Para o cálculo dessa diferença usa-se uma medida de distância. Entre outras, encontra-se a euclidiana, cityblock, chebychev, correlação e spearman [2,8]. Para esse trabalho foi selecionada a euclidiana, função 1, por ter gerados os melhores resultados em testes preliminares.

$$D(x, x_i) = \sqrt{\sum_{k=1}^{d} (x_k - x_{i,k})^2} \qquad 1$$

Sendo $d$ o número máximo de características e $x$ o vetor de características.

A dissimilaridade propicia o uso de um modelo independente do escritor. Esse modelo apresenta a vantagem de não necessitar da geração de novos modelos, quando da entrada de um novo escritor na base de escritores. O modelo de dissimilaridade, nesse contexto, revela a presença da autora, classe $w_1$, e não autoria, classe $w_2$. Para tanto, usa-se o conceito de Referência (Ref), isto é, amostras genuínas de um escritor que serão comparadas com a questionada (Q), usando um modelo de dissimilaridade independente do escritor como árbitro. Confira na figura 8.

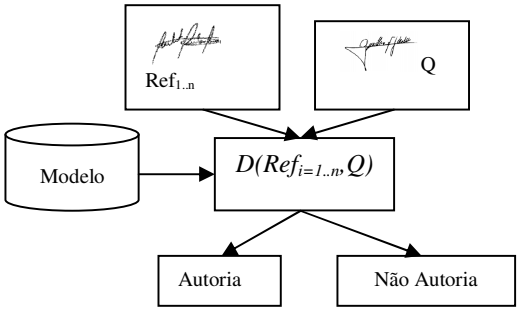



**Figura 8.** Diagrama em bloco do Modelo de classificação.

Como se pode utilizar um número de referências maior que 1, isto é, Ref ≥ 1, um método de fusão dos resultados deve ser aplicado. Esse conceito é o mesmo utilizado por um perito grafotécnico que usa um conjunto de amostras conhecidas do autor *Ref* na confrontação com a amostra questionada *Q*, a fim de determinar o padrão de variabilidade intrapessoal. Os métodos de fusão mais utilizados são o voto majoritário, a soma dos resultados e a média [4].

Neste trabalho foi usado um classificador SVM num modelo de duas classes. O valor escalar com o grau de certeza produzido pelo mesmo foi usado no método de fusão por soma, pois esse foi o que gerou o melhor resultado.

## 4. Análise dos resultados

A base de dados usada nesse trabalho é composta de 5.200 imagens de assinaturas. Sendo 4.000 assinaturas genuínas oriundas de 100 escritores diferentes (40 amostras de assinaturas por escritor) e 1.200 falsificações,

geradas a partir dos 60 primeiros escritores. Foram produzidas 10 falsificações simples (FS) e 10 falsificações exercitadas (FE) para cada escritor. As assinaturas foram coletadas numa resolução de 300 dpi e 256 níveis de cinza. Em seguida as imagens foram recortadas e gravadas em um arquivo no formato BMP, num retângulo de tamanho 3x10 cm ou 400x1.000 pixels, correspondendo à área destinada à assinatura [4].

A base de 100 escritores foi dividida em duas sub-bases. A primeira destinada ao processo de treinamento ou geração do modelo. A segunda, contendo as falsificações, foi utilizada somente para os testes. Cabe salientar que nenhum autor que participou do processo de treinamento foi utilizado nos testes. Outro aspecto importante é que não foram utilizadas falsificações simples e exercitadas na geração do modelo. Isto é, somente amostras de escritores distintos foram usados como falsificações aleatórias (FA). Portanto, a base de treinamento do modelo foi composta de vetores de dissimilaridades entre amostra do mesmo escritor (autoria) e vetores de dissimilaridades entre amostra de escritores diferentes (não autoria). Foram utilizadas 20 amostras de cada escritor, num total de 240 vetores de dissimilaridade das amostras genuínas e 288 das falsas aleatórias.

Nos testes foram utilizados os escritores da segunda sub-base. De cada um dos 60 autores da base de teste foram separadas 7 amostras como referência (*Ref*) e 10 para testes, mais as falsificações, totalizando 30 amostras questionadas por escritor. As 10 amostras genuínas de teste foram usadas como verdadeiras para esse escritor e como falsas aleatória para os demais escritores. Cada amostra de assinatura questionada foi confrontado com as de referência de um determinado escritor, tendo como árbitro o modelo gerado pelo SVM. Confira na figura 7. O *kernel* utilizado pelo SVM foi o linear, já utilizado por outros autores [1,3,4,5]. Os resultados obtidos podem ser vistos na tabela 2.

**Tabela 2.** Resultados da classificação.

| Número Ref | Erro I (%) | Erro II (%) | | | Acerto (%) |
|---|---|---|---|---|---|
| | | FA | FS | FE | |
| **3** | **5.17** | **0.83** | **0.33** | **19.67** | **93.50** |
| 5 | 6.17 | 0.00 | 1.17 | 20.50 | 93,04 |
| 7 | 8.17 | 1.67 | 1.50 | 23.67 | 91,25 |

É possível observar que o modelo responde adequadamente para as falsificação aleatória (FA) (Falsa aceitação ou Erro tipo II < 1%) e a falsa rejeição (Erro tipo I < 6%). Isto ocorre pois o modelo gerado foi treinado com falsificações aleatória. Mas mesmo com as demais falsificações, a simples e a exercitada, o modelo teve como resultado FS < 1% e FE < 20%. Os fatores que mais influenciaram no erro das falsificações exercitadas (FE) estão no uso de um modelo generalista, uma vez que não se usou um modelo por escritor e também pela falta de treinamento com outros tipos de falsificações. Mas mesmos nessas condições, os resultados se mostram promissores. Um fator importante está no número de referências utilizado, Não se verificou variações significativas no uso de um número superior a 3 referência. Isto indica que o método se mostra adequado a uma aplicação real, o uso de 3 a 4 referências seria aceitável para um processo de colheita de exemplares.

A tabela 3 mostra os resultados comparativos com outros métodos que utilizaram a mesma base de dados e protocolos de testes semelhantes. Os resultados obtidos se mostraram muito próximos aos do método 2, com exceção da FA. A superioridade apresentada pelo método 2 deve-se principalmente ao modelo dependente do autor e ao número de amostra de treinamento e validação cruzada usados. Apesar dos resultados obtidos pelo método proposto serem inferiores em alguns casos, ainda é o que apresenta um modelo independente do autor, com o menor número de amostras de referências.

**Tabela 3.** Resultados comparativos com outros trabalhoso.

| Número de características / Número de classificador / Modelo / Número de referências *Ref* | | Erro Tipo I (%) | Erro Tipo II (%) | | |
|---|---|---|---|---|---|
| | | | FA | FS | FE |
| **1** Bertolini [1] | 18 características / Algoritmos genéticos e combinação de 66 classificadores SVM / Independente do Autor / 15 referências | 11,3 | 3,0 | 4,3 | 6,4 |
| **2** Justino [4] | 3 características / 1 classificador HMM / Dependente do Autor / 30 referências (20 no treinamento e 10 na validação cruzada) | **3,3** | **1,33** | 1,83 | **5,51** |
| **3** Método Proposto | 5 características / 7 classificadores SVM / Independente do Autor / 7 referências *Ref* | 8,1 | 1.6 | **1.5** | 23.6 |

## 5. Conclusão e trabalhos futuros

Este artigo apresentou uma abordagem para a verificação de assinaturas *off-line* com base em um

processo misto de segmentação. As características utilizadas foram selecionadas de forma a atender a dois níveis de percepção visual cognitiva humana e apresentam simplicidade na implementação computacional. A abordagem se baseia em um modelo de duas classes, autoria e não autoria, usando o conceito de dissimilaridade. Os resultados demonstram a robustez do método na verificação dos três tipos básicos de falsificações, atingindo uma média de acerto de 93,5%.

Como trabalhos futuros existem dois pontos a serem explorados. O primeiro está em investigar a inclusão de um modelo por autor, o que deve melhorar significativamente o desempenho para as falsificações exercitadas. A segunda seria manter o mesmo modelo, incluído os três tipos de falsificações na geração do mesmo.

## 6. Referências


[1] Bertolini, D. ; Oliveira, L.S. ; Justino, E. ; Sabourin, R. . Reducing forgeries in writer-independent off-line signature verification through ensemble of classifiers. *Pattern Recognition*, v. 43, 2010,p. 387-396,,.

[2] D. Impedovo, G. Pirlo, Automatic signature verification: the state of the art*, IEEE Transactions on Systems, Man, and Cybernetics*—Part C 38 (5) ,2008, 609–635.

[3] Oliveira, L.S. ; Justino, E. ; Sabourin, R.; Bortolozzi, F. Combining Classifiers in the ROC-space for Off-line Signature Verification. *Journal of Universal Computer Science*, vol. 14, no. 2 ,2008, 237-251.

[4] E. Justino, F. Bortolozzi, R. Sabourin, Off-line signature verification using HMM for random, simple and skilled forgeries, in: *6th International Conference on Document Analysis and Recognition*, 2001, pp. 1031–1034.

[5] C. Santos, E. Justino, F. Bortolozzi, R. Sabourin, An off-line signature verification method based on document questioned experts approach and a neural network classifier, in: *9th International Workshop on Frontiers in Handwriting Recognition*, 2004, p. 498–502.

[6] Shomaker, L.; Advances in Writer Identification and Verification, *Proceedings of the International Conference on Document Analysis and Recognition (ICDAR)* Curitiba, Brazil, 2007.

[7] Cha, S. H., *Use of the Distance Measures in Handwriting Analysis, Doctor Theses. State University of New York at Buffalo*, EUA, 2001,p. 208.

[8] H. Baltzakis, N. Papamarkos, A new signature verification technique based on a twostage neural network classifier, *Engineering Applications of Artificial Intelligence* 14 (2001) 95–103.

[9] B. Fang, Y. Tang, Improved class statistics estimation for sparse data problems in offline signature verification, *IEEE Transactions on Systems, Man and Cybernetics* 35 (3) ,2005, 276–286.

[10] R. Sabourin, G. Genest, An extended-shadow-code based approach for off-line signature verification: part I evaluation of the bar mask definition, in: *12th International Conference on Pattern Recognition*, 1994, p. 450–453.

[11] Jolicoeur, P., Gluck, M., & Kosslyn, S. M.. Pictures and names: Making the connection. *Cognitive Psychology*, *16*, 1984, 243-275.

[12] Hanusiak, R. K., Oliveira, L. S., Justino, E., Sabourin, R., Writer verification using texture-based features. *International Journal on Document Analysis and Recognition* (Print). , v.15, 2012, p.213 - 226.

# Reconhecimento de pele em imagens digitais utilizando redes neurais artificiais

Igor Leonardo Oliveira Bastos
Instituto de Matemática
Universidade Federal da Bahia - UFBA
Feira de Santana, Brasil
igorcrexito@gmail.com

Michele Fúlvia Angelo
Departamento de Tecnologia
Universidade Estadual de Feira de Santana - UEFS
Feira de Santana, Brasil
mfangelo@ecomp.uefs.br

***Abstract* — This paper aims to present an approach to recognize skin pixels in images. To do this, a feature vector is assembled for every pixel of every image of the training dataset, combining components from different color spaces and utilizing a Perceptron Multilayer classifier. At the end of this paper the accuracy of the classifier is presented and compared to three of the most used algorithms to recognize skin in digital images, using for this 243 images from 3 different image datasets. It is important to say that the approach presented in this paper showed superior results in most cases than the algorithms, reaching, for some images, accuracy rates above 99%.**

***Keywords—colorspaces, skin detection, feature extraction, neural networks***

## I. INTRODUÇÃO

A detecção de pele em imagens digitais tem sido utilizada cada vez mais nos últimos anos, sendo que esta se mostra frequente em aplicações como as de vídeo-vigilância, reconhecimento de face e/ou gestos, além de aplicações que realizam interação homem-computador através do reconhecimento de partes do corpo [1].

Muitos são os métodos utilizados para o reconhecimento de pele em imagens digitais. Dentre estes, pode-se destacar os não-paramétricos, tais como o classificador bayeasiano [2]; os paramétricos, como o modelo por mistura de gaussianas [2]; e por fim, métodos onde regiões (limites) são estabelecidas para definir se um pixel é pele ou não [2, 3].

Devido à simplicidade dos métodos onde estes limites são definidos e dos bons resultados que eles apresentam, tomou-se como base para o presente trabalho 3 (três) dos algoritmos mais usados para a detecção de pele. Estes algoritmos são os propostos por Buhyian, Ampornaramveth e Ueno [4]; Gomez, Sanchez e Sucar [5] e Peer, Kovac e Solina [6] os quais utilizam os espaços de cores Vermelho, Verde e Azul (RGB); Matiz, Saturação e Intensidade (HSV) e Luminância, Crominância laranja-azul e Crominância roxo-verde (YIQ).

Assim, baseando-se na pesquisa realizada a respeito dos algoritmos para detecção de pele e nos espaços de cores RGB, HSV e YIQ, o objetivo deste trabalho é apresentar uma abordagem para o reconhecimento de pele em imagens digitais utilizando uma Rede Neural Perceptron Multicamada para classificar os pixels das imagens.

As seções seguintes visam descrever os espaços de cores utilizados, como o vetor de características do classificador foi criado, além de apresentar aspectos referentes à rede neural utilizada e os resultados obtidos, os quais foram comparados, em termos de eficácia, com os algoritmos de detecção de pele propostos por Buhyian, Ampornaramveth e Ueno [4], Gomez, Sanchez e Sucar[5 ] e Peer, Kovac e Solina [6].

## II. ESPAÇOS DE CORES.

### A. *Espaço de cor RGB*

Apesar de características indesejáveis, como a mistura de informação de crominância e luminância, a alta correlação entre os canais [2] e a não uniformidade com a percepção da visão humana (cores com valores próximos dos canais RGB não são necessariamente próximas em termos de percepção) [7], o espaço de cores RGB é o mais utilizado na formulação de algoritmos de reconhecimento de pele.

Fatores como a facilidade de se operar os canais RGB e o fato da maior parte dos dispositivos de aquisição de imagens operarem diretamente este espaço de cor [8] estimularam a sua utilização no desenvolvimento de alguns trabalhos que visam detectar pele, como os propostos por Yang e Ahuja [9] e Peer, Kovac e Solina [6].

### B. *Espaço de cor HSV*

O espaço de cores HSV resulta de uma transformação linear do espaço RGB [2]. Devido à separação, em seus canais, da informação referente à crominância e luminância, este espaço é extremamente utilizado em pesquisas relacionadas ao reconhecimento de pele pela cor [7]. O espaço HSV é utilizado em trabalhos como os propostos por Gomez, Sanchez e Sucar [5], Sigal, Sclaroff e Athitsos [10] e Liu e Peng [11].

### C. *Espaço de cor YIQ*

Assim como o espaço HSV, o espaço de cores YIQ (derivado do YCbCr), apresenta os componentes de luminância e crominância separados. Devido a esta característica e a simplicidade deste espaço, ele e os espaços ortogonais como o YUV, YES e YCbCr são muito utilizados em algoritmos para a detecção de pele, como o proposto por Buhyian, Ampornaramveth e Ueno [4].

## III. ALGORITMOS PARA A DETECÇÃO DE PELE BASEADOS EM LIMITES

Os algoritmos para a detecção de pele baseados em valores de limite representam métricas para determinar se os pixels da imagem apresentam coloração de pele ou não. Estes algoritmos foram, em sua grande maioria, definidos empiricamente e relacionam-se a um ou mais espaços de cores, para os quais as margenspara a separação de pixels como pele ou não são definidas.

Alguns dos algoritmos mais usados para a detecção de pele em imagens digitais são os propostos por Buhyian, Ampornaramveth e Ueno [4], Gomez, Sanchez e Sucar [5] e Peer, Kovac e Solina [6].

### A. *Algoritmo de Buhyian, Ampornaramveth e Ueno[4]*

O algoritmo para a detecção de pele proposto por Buhiyan, Ampornaramveth e Ueno [4] utiliza duas das componentes do espaço de cores YIQ para estabelecer limites para a classificação de pixels como pele ou não. É importante salientar que estes valores de limite foram obtidos empiricamente e podem ser calculados através das inequações 1 e 2, as quais, quando satisfeitas, determinam que um pixel corresponde a pele.

$$60 < Y < 200 \quad (1)$$

$$20 < I < 50 \quad (2)$$

### B. *Algoritmo de Gomez, Sanchez e Sucar [5]*

O algoritmo para detecção de pele proposto por Gomez, Sanchez e Sucar [5] combina componentes dos espaços de cores RGB e HSV, além de um coeficiente intitulado de Wr, proposto pelos autores para a determinação das margens para a detecção de pixels com cor de pele. As inequações 3, 4 e 5, a seguir, regem o funcionamento deste algoritmo e, quando satisfeitas, indicam que um pixel corresponde a pele.

$$-17.4545 < H < 26.666 \quad (3)$$

$$GY < -5.9216 \quad (4)$$

$$Wr < 0.0271 \quad (5)$$

O coeficiente Wr é calculado pela equação 6, apresentada a seguir. Este coeficiente também foi calculado empiricamente e os termos r, g e b correspondem aos canais do espaço RGB normalizados.

$$Wr = \left(\frac{r}{r+g+b} - 1/3\right)^2 + \left(\frac{2}{r+g+b} - 1/3\right)^2 \quad (6)$$

### C. *Algoritmo de Peer, Kovac e Solina [6]*

O algoritmo de pele proposto por Peer, Kovac e Solina [6] utiliza as componentes do espaço RGB para determinar se um pixel é pele ou não. Para tal, margens relacionadas aos valores das componentes deste espaço foram empiricamente determinadas. As inequações 7, 8 e 9 regem o funcionamento deste algoritmo.

$$(R > 95), (G < 40), (B > 20) \quad (7)$$

$$(max(R,G,B)-min(R,G,B)) > 15, (|R-G|) > 15 \quad (8)$$

$$(R > G), (R > B) \quad (9)$$

## IV. IMAGENS PARA TESTE E TREINAMENTO DA REDE NEURAL

Para o presente trabalho, foi criado um conjunto de testes com 243 imagens de pessoas. Este conjunto foi agrupado em 3 grupos menores. O primeiro deles é composto por 50 imagens adquiridas aleatoriamente do *dataset LFW Face Database* [12]. O segundo grupo é composto por 90 imagens pertencentes ao *dataset Annotated Skin Database* [13]. Este segundo *dataset* fornece, além das imagens de pessoas, imagens com máscaras binárias com os pixels de pele realçados. É válido ressaltar que ambos os *datasets* apresentam imagens com diferentes resoluções. Já o terceiro grupo é composto por 103 imagens de pessoas, as quais foram tiradas pelos pesquisadores responsáveis por este trabalho. Estas imagens apresentam resolução de 320x240 e foram adquiridas de forma a apresentar diferentes condições de iluminação, além de envolverem sujeitos com diferentes tons de pele.

Fig.1 – Imagens e suas correspondentes máscaras binárias

Para avaliação dos resultados de imagens do primeiro e terceiro grupo, fez-se necessária uma etapa de geração de máscaras binárias que correspondem às zonas de pele das mesmas. Estas foram obtidas a partir de marcações manuais e algumas podem ser vistas na Figura 1, acima.

O conjunto de treinamento, assim como o terceiro grupo mencionado anteriormente, foi composto com imagens tiradas pelos pesquisadores. Este conjunto foi criado com 43 imagens de pessoas e suas respectivas máscaras binárias. Apesar de constituído por imagens diferentes, este conjunto de treinamento foi tirado no mesmo ambiente e envolve os mesmos sujeitos das imagens do terceiro grupo e assim como ele, apresenta a resolução de 320x240. Como para o realce de pele utiliza-se a informação de cada pixel da imagem isoladamente (cada pixel é utilizado separadamente em cada um dos algoritmos citados ou, no caso do presente trabalho, para compor um vetor de característica), é importante salientar que foi utilizado um total de 3.302.400 pixels para o treinamento do classificador, o qual foi realizado tomando-se como base, os pixels realçados como pele nas máscaras

binárias para informar quais entradas correspondiam à pele ou não.

## V. Metodologia Empregada

Com base nas pesquisas realizadas, elaborou-se para o presente trabalho uma abordagem baseada na utilização dos canais dos espaços de cores citados anteriormente, os quais são utilizados pelos algoritmos de pele mencionados na seção III [4, 5, 6]. Estes canais, pertencentes aos espaços de cores RGB, HSV e YIQ, foram utilizados para compor um vetor de características, como apresentado na Figura 2.

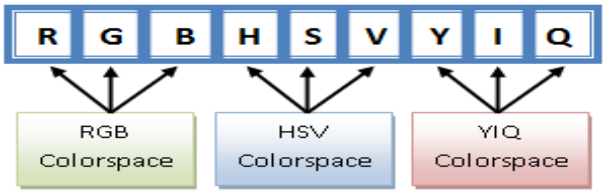



Fig. 2 – Vetor de características combinando os elementos de diferentes espaços de cores.

Para cada pixel de cada imagem utilizada, um vetor de característica, como o mostrado na Figura 2, foicriado. Na etapa de treinamento, este vetor possui uma posição extra, a qual indica se o vetor relaciona-se a um pixel de pele ou não.

O classificador utilizado no presente trabalho foi uma rede neural Perceptron Multicamada. A configuração de suas camadas foi feita de forma a se adequar ao vetor de características apresentado. Como o vetor possui nove posições e a saída da rede apenas indica se um pixel é pele ou não (saída binária), a camada de entrada da rede possui nove neurônios e a camada de saída apenas um neurônio. Quanto à camada intermediária, testes foram realizados para a obtenção dos melhores resultados e definição da quantidade de neurônios. Variou-se a quantidade de neurônios de 1 a 70, como mostrado na Figura 3. É válido ressaltar que a função de ativação usada foi a sigmóide e a regra de aprendizado foi a *Backpropagation*.

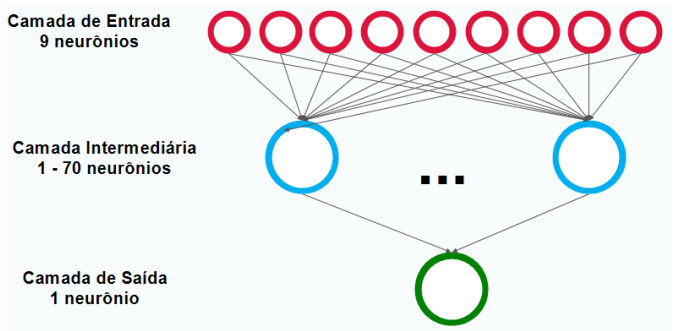



Fig. 3 – Arquitetura da rede neural utilizada.

Após a definição da arquitetura da rede neural utilizada e a composição do vetor de características, foi realizada a implementação dos algoritmos de detecção de pele apresentados na seção III. Estes, assim como o classificador implementado, foram testados utilizando imagens dos 3 grupos citados anteriormente e tiveram sua taxa de acerto (acurácia) comparada à da presente abordagem.

## VI. Resultados e Discussões

A avaliação do desempenho da presente abordagem, assim como dos algoritmos de detecção de pele implementados [4, 5, 6], foi realizada em termos de taxa de acerto (acurácia). Esta taxa de acerto foi calculada segundo a equação 10, onde os termos FP, FN, VP e VN correspondem aos falsos positivos, falsos negativos, verdadeiros positivos e verdadeiros negativos, respectivamente. Vale lembrar que para a obtenção destes valores, foi necessária a comparação entre as imagens resultantes da classificação e dos algoritmos de pele com as máscaras binárias geradas anteriormente.

$$\text{Taxa de acerto} = (VP+VN)/(VP+VN+FP+FN) \qquad (10)$$

Utilizando a equação 10 e, primeiramente, o terceiro grupo de imagens de teste, calculou-se a taxa de acerto do classificador, realizando uma variação da quantidade de neurônios na sua camada escondida. Esta quantidade variou de 1 a 70. Pôde-se observar que os melhores resultados para as 103 imagens do grupo 3 foram obtidos com as configurações de 35 e 45 neurônios na camada escondida. Para estes valores, obtiveram-se resultados (para 92 das 103 imagens) de 15% a 60% superiores às configurações com outros valores. É importante salientar que, para algumas imagens, obtiveram-se taxas de acerto superiores a 99%, indicando uma classificação muito fiel às máscaras binárias geradas.

Na Figura 4, a seguir, é apresentado o resultado da classificação com diferentes quantidades de neurônios na camada escondida. Percebe-se que, para 35 e 45 neurônios, realçou-se, quase que exclusivamente, zonas que correspondem à pele na imagem original, apresentando baixa presença de ruídos.

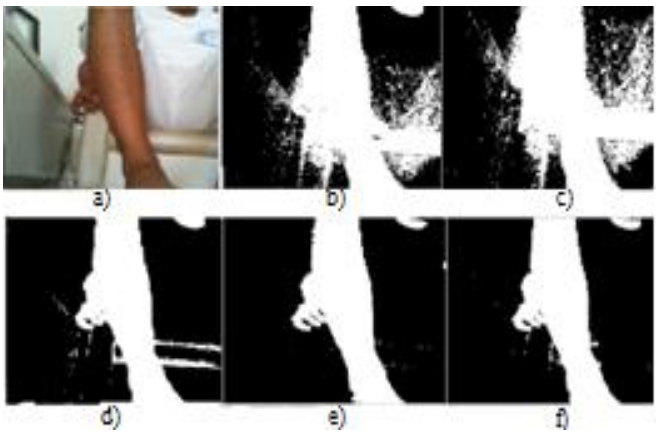



Fig. 4 – Resultado da classificação para diferentes números de neurônios na camada escondida. (a) Imagem original. (b) Rede com 5 neurônios. (c) Rede com 15 neurônios. (d) Rede com 25 neurônios. (e) Rede com 35 neurônios. (f) Rede com 45 neurônios.

Devido o bom desempenho do classificador com as configurações de 35 e 45 neurônios na camada escondida,

estas configurações foram utilizadas para a realização da comparação da acurácia desta abordagem com os algoritmos de detecção de pele implementados [4, 5, 6].

Utilizando mais uma vez o terceiro grupo, o qual possui imagens com os mesmos sujeitos do conjunto de treinamento, obtiveram-se taxas de acerto para o classificador, em média, 25% superiores aos algoritmos implementados. É importante salientar que os resultados encontrados para o classificador com 35 e 45 neurônios foram muito similares e estes foram superiores aos dos algoritmos para 98 das 103 imagens do terceiro grupo. Na Figura 5 é apresentado o resultado da classificação utilizando a rede neural com as configurações citadas e os resultados dos algoritmos de detecção de pele implementados [4, 5, 6]. É perceptível que a classificação realizada pela abordagem proposta, para esta imagem, apresentou uma taxa de acerto muito superior à dos algoritmos.

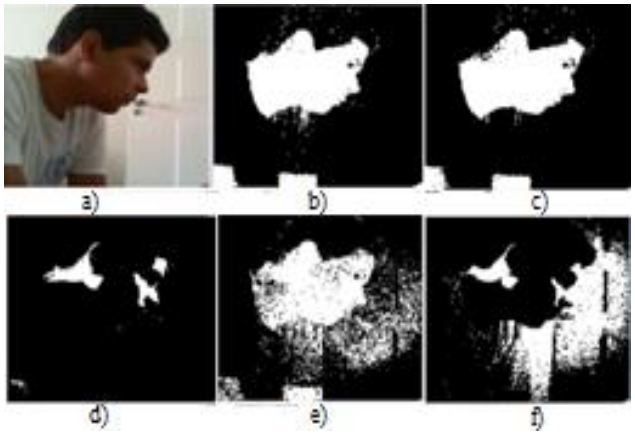



Fig. 5 – Classificação com diferentes configurações e algoritmos de pele implementados. (a) Imagem original. (b) Rede com 35 neurônios. (c) Rede com 45 neurônios. (d) algoritmo de Peer, Kovac e Solina. (e) algoritmo de Gomez, Sanchez e Sucar. (f) algoritmo de Buhyian, Ampornaramveth e Ueno.

Assim como para o terceiro grupo de imagens (103 imagens), testes foram realizados para a obtenção da taxa de acerto do classificador e dos algoritmos de detecção de pele implementados [4, 5, 6] para o primeiro (50 imagens) e para o segundo (90 imagens) grupo. Para o primeiro grupo, assim como para o terceiro, as configurações de 35 e 45 neurônios apresentaram resultados muito similares. Estas promoveram uma detecção com taxa de acerto, em média, 5% superior ao algoritmo de melhor resultado. É válido ressaltar que, para ambas as configurações, o classificador apresentou resultados superiores aos algoritmos para 34 das 50 imagens. Na Figura 6, uma das imagens pertencentes ao primeiro grupo pode ser vista. Esta foi submetida à classificação pela rede neural e aos algoritmos de detecção de pele. Neste caso, a rede neural, com ambas as configurações e o algoritmo de Buhyian, Ampornaramveth e Ueno [4] apresentaram taxas de acerto superiores a 90%.

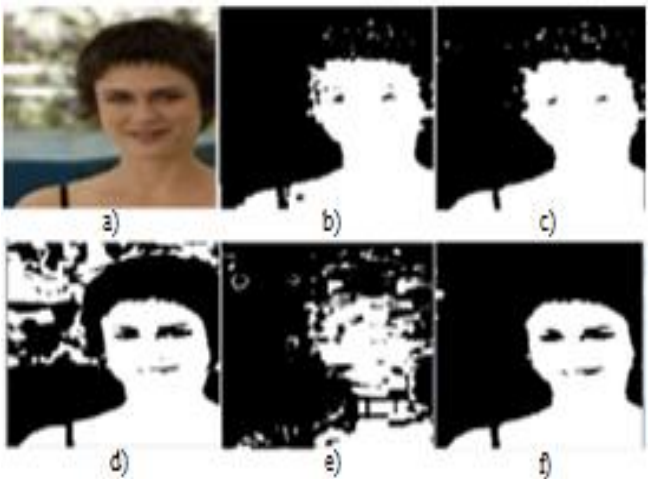



Fig. 6 – Classificação com diferentes configurações e algoritmos de pele implementados. (a) Imagem original. (b) Rede com 35 neurônios. (c) Rede com 45 neurônios. (d) algoritmo de Peer, Kovac e Solina. (e) algoritmo de Gomez, Sanchez e Sucar. (f) algoritmo de Buhyian, Ampornaramveth e Ueno.

Finalmente, para o segundo grupo de imagens, notou-se que, para 59 das 90 imagens, uma das duas configurações da rede apresentou resultados superiores aos algoritmos, com destaque para a de 35 neurônios, que apresentou resultado superior à de 45 neurônios para 56 das 90 imagens. Para este grupo, o classificador apresentou, em média, considerando a configuração com melhor resultado, uma taxa de acerto 2% superior em relação ao algoritmo de melhor resultado. Na Figura 7, a seguir, uma imagem pertencente ao segundo grupo pode ser vista juntamente com as detecções provenientes dos algoritmos de pele e das redes neurais com 35 e 45 neurônios.

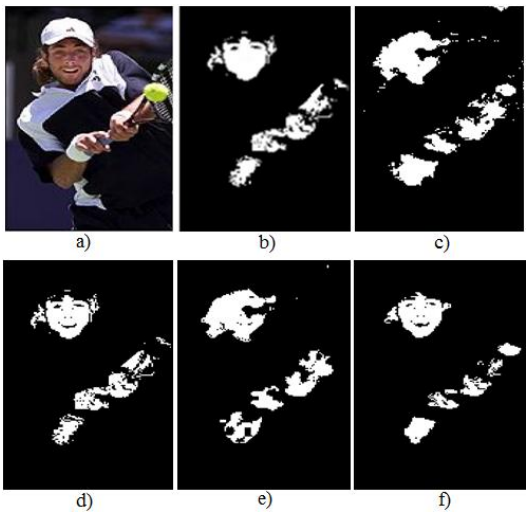



Fig. 7 – Classificação com diferentes configurações e algoritmos de pele implementados. (a) Imagem original. (b) Rede com 35 neurônios. (c) Rede com 45 neurônios. (d) algoritmo de Peer, Kovac e Solina. (e) algoritmo de Gomez, Sanchez e Sucar. (f) algoritmo de Buhyian, Ampornaramveth e Ueno.

A partir destes resultados, observou-se que os algoritmos de detecção de pele e a abordagem apresentada neste trabalho atingiram taxas de acerto similares, com uma sutil vantagem

para a abordagem proposta. Porém, para o terceiro grupo de imagens, o qual é composto por imagens de sujeitos também presentes no conjunto de treinamento, notou-se um ganho considerável quanto à taxa de acerto da abordagem em relação aos algoritmos.

Estes resultados indicam que a abordagem proposta apresenta resultados bastante precisos quando aplicada no contexto em que o classificador foi treinado, isto é, envolvendo o mesmo ambiente de aquisição e os mesmos indivíduos nas imagens. A Tabela 1 apresenta a acurácia média, em termos percentuais, para cada configuração da rede neural e para os algoritmos de detecção de pele implementados, encontrada em relação a cada grupo de imagens.

TABELA I – ACURÁCIA (%) DOS ALGORITMOS IMPLEMENTADOS E DO CLASSIFICADOR PARA OS DIFERENTES GRUPOS DE IMAGENS

| **Classificador/Algoritmos** | **Grupos de Imagens** | | |
|---|---|---|---|
| | **Primeiro** | **Segundo** | **Terceiro** |
| Rede 35N | 66,8% | 61,9% | 86,6% |
| Rede 45N | 67,2% | 58,2% | 86,8% |
| Peer, Kovac e Solina | 61,5% | 59,5% | 53,2% |
| Gomez, Sanchez e Sucar | 54,2% | 60,0% | 61,9% |
| Buhyian, Ampornaramveth e Ueno | 58,7% | 55,3% | 60,6% |

Interessantemente, mesmo em condições adversas para o reconhecimento de pele baseado em cor, como baixa iluminação, a abordagem proposta neste trabalho apresentou resultados superiores aos algoritmos, desde que as imagens com estas condições fizessem parte do conjunto de treinamento, como foi o caso. Na Figura 8, uma imagem com uma baixa iluminação pode ser vista, juntamente com a detecção realizada pelo classificador para ambas as configurações (35 e 45 neurônios), e a detecção realizada pelos algoritmos implementados. Nota-se que, apesar da incidência de ruídos, os resultados do classificador se mostram precisos.

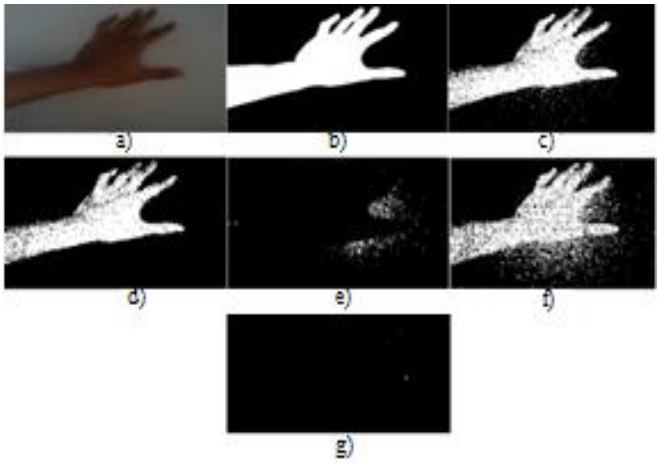



Fig. 8 – Imagem com baixa iluminação. (a) Imagem original. (b) Máscara binária com pixels de pele realçados. (c) Rede com 35 neurônios. (d) Rede com 45 neurônios. (e) algoritmo de Peer, Kovac e Solina. (f) algoritmo de Gomez, Sanchez e Sucar. (g) algoritmo de Buhyian, Ampornaramveth e Ueno.

A imagem mostrada na Figura 8 pertence ao terceiro grupo de imagens e foi usada para apresentar a boa eficácia do classificador em condições adversas de iluminação. Além desta, utilizou-se as 5 imagens de cada grupo com pior condição de iluminação e foram obtidos, em média, os resultados apresentados na Tabela 2.

TABELA II – ACURÁCIA (%) DOS ALGORITMOS IMPLEMENTADOS E DO CLASSIFICADOR PARA OS DIFERENTES GRUPOS DE IMAGENS PARA IMAGENS COM CONDIÇÕES ADVERSAS

| **Classificador/Algoritmos** | **Grupos de Imagens** | | |
|---|---|---|---|
| | **Primeiro** | **Segundo** | **Terceiro** |
| Rede 35N | 73,0% | 73,6% | 79,3% |
| Rede 45N | 72,5% | 72,4% | 84,5% |
| Peer, Kovac e Solina | 16,4% | 66,3% | 1,1% |
| Gomez, Sanchez e Sucar | 68,3% | 6,8% | 67,9% |
| Buhyian, Ampornaramveth e Ueno | 54,8% | 17,8% | 1,0% |

## VII. CONCLUSÕES

A partir dos resultados obtidos, pôde-se concluir que a abordagem proposta neste trabalho apresenta resultados satisfatórios, sendo mais precisos do que os resultados dos algoritmos de detecção de pele implementados para a maior parte das imagens. Esta abordagem pode se apresentar bastante útil em contextos onde um treinamento prévio pode ser feito, como para o terceiro grupo de imagens do *dataset* de

testes, para o qual, foi obtida, para mais de 80% das imagens, uma acurácia entre 80% e 100%.

Como trabalho futuro, pretende-se realizar testes para conjuntos maiores de imagens, além de comparações da presente abordagem com um conjunto maior de técnicas de detecção de pele.

## REFERÊNCIAS


[1] A. Cheddad, J. Condell, K. Curran, and P.M. Kevitt, "A new colour space for skin tone detection". IEEE – International Conference on Image Processing, Cairo, 2009.

[2] C. Prema, and D. Manimegalai, "Survey on Skin Tone Detection using Color Spaces".*International Journal of Applied Information Systems*, Foundation of Computer Science, New York, vol.2 n.2, 2012.

[3] V. Vezhnevets, V. Sazonov, and A. Andreeva, "A Survey on Pixel-Based Skin Color Detection Techniques". International ConferenceGraphicon, Moscow, 2003.

[4] M.A.A., Buhyian, V. Ampornaramveth, and S.Y.H. Ueno, "Face detection and Facial Feature Localization for Human-machine interfaces". *NII Journal*, n.5, 2003.

[5] G. Gomez, M. Sanchez, and L. Sucar."On selecting an appropriate colour space for skin detection".*Mexican International Conference on Artificial Intelligence*, Yucatan, pp. 70-79, 2002.

[6] P. Peer, J. Kovac, and F. Solina."Human skin colour clustering for face detection".EUROCON - International Conference on Computer as a Tool, Ljubljana, 2003.

[7] H.L. Ribeiro. "Reconhecimento de Gestos Usando Segmentação de Imagens Dinâmicas de Mãos Baseada no Modelo de Mistura de Gaussianas e Cor de Pele", master´sthesis, Dept. ofElectricalEngeneering, 2006.

[8] G. Yang, H. Li, L. Zhang and Y. Cao, "Research on a skin color detection algorithm based on self-adaptive skin color model", *Communications and Intelligence Information Security*, Nanning, pp. 266-270, 2010.

[9] M.-H. Yang, and N. Ahuja. "Gaussian Mixture Modelling of Human Skin Color and It´s Applications in Image and Video Databases".*Storage and Retrieval for Image and Video Databases*, San Jose, pp. 458-466, 1999.

[10] L. Sigal, S. Sclaroff, and V. Athitsos."Skin color-based video segmentation under time-varying illumination".*Pattern Analysis and Machine Intelligence*, v. 26, pp. 862-877, 2004.

[11] Q. Liu, and G. Peng. "A Robust Skin Color Based Face Detection Algorithm". *2nd International Asia Conference on Informatics in Control, Automation and Robotics*,Wuhan, 2010.

[12] "Labeled Faces in the Wild (LFW) Database". Jan. 2014; http://vis-www.cs.umass.edu/lfw/.

[13] "Annotated Skin Database". Jan. 2014; http://agami.die.uchile.cl/skindiff/

# Self-Organizing Traffic Lights: A Pedestrian Oriented Approach

Jessica S. Souza[1], Cesar A. M. Ferreira[2], Cassio E. dos Santos Jr[3], Victor H. C. Melo[4], William Robson Schwartz[4]
Computer Science Department, Federal University of Minas Gerais, Belo Horizonte, Brazil
{jessicasena,camf,victorhcmelo,cass,william}@dcc.ufmg.br

***Abstract*—The traffic light is one of the valuable devices to control the vehicular and pedestrian traffic. One of its main issues is that several traffic lights might be improperly calibrated once they do not consider the differences in pedestrian mobility from region to region. As each region presents different pedestrians with different characteristics, there is a need for automatic approaches. In this work, we propose a new approach to automatically adjust the pedestrian traffic light program to provide greater safety for pedestrians and to allow a better flow of the traffic. We deal with two challenging cases in transport engineering literature. The first case happens when pedestrians with reduced speed cannot cross the street within the available time. The second case happens when the traffic light for pedestrians remains open for a long time even when there are no pedestrians waiting to cross. Our proposed approach consists of a new segmentation step based on the pedestrians' distribution map; a pedestrian tracking step based on a Kalman Filter model and a new association metric; and, finally, a module that uses the information provided by these two previous steps to control the self-manageable traffic light. The experimental evaluation shows that the proposed approach allows to minimize cases in which pedestrians are crossing the street and the traffic light turns red.**

***Keywords*-Pedestrian tracking; self-manageable traffic light.**

## I. Introduction

Brazilian cities have been modified by the influence of automobile traffic. The growth of the private car fleet affects urban centers in a negative way, creating disharmony between pedestrians and vehicles. Traffic light is one of the valuable devices for the control of vehicular and pedestrian traffic, used to regulate, warn, or guide traffic [1]. However, several traffic lights might be improperly calibrated because they do not consider the differences in pedestrian mobility from region to region within the city.

The 70th article of the Brazilian Traffic Code states that, in case there is a crosswalk available, the crossing priority will be in accordance with the emitted traffic light for pedestrians and vehicles, where pedestrians always have priority on completing the crossing, even if the traffic light changes from pedestrian traffic to car traffic. On these occasions, more than the discomfort and embarrassment suffered by the pedestrian, the increase of the risk of accidents is a major concern, especially for those with reduced mobility.

Many studies seek to analyze the optimum pedestrian speed. FHWA [1] shows that the ideal pedestrian speed is 1.22 m/s, a value accepted by different laws around the world. However, there is not a consensus in the research community. For instance, ITE [2] suggests that the most appropriate value to be used is between 0.91 and 0.99 m/s. Silva *et al.* [3] establish that 1.22 m/s is the average speed to cross the street, with a minimum of 0.63 m/s to a maximum of 1.83 m/s, indicating the existence of several factors which affect the speed of pedestrians, such as age, crossing being performed by pedestrians alone or in groups, and the vehicles average speed. A worrying factor is the high speed of the vehicles, which can affect mainly elderly crossing the road. Therefore, we can notice that the value of 1.22 m/s is not the best pedestrian speed estimation due to different pedestrian types.

Each region in a city presents different characteristics, such as hospital regions, which have a higher number of pedestrians with mobility problems, characteristic of the elderly and infirm population. Displacement speeds might change in different regions due to certain characteristics of the region, such as shops, parks, and others points of distraction for pedestrians. Furthermore, factors such as time and seasonal periods affect the characteristics of crossing. Due to the large number of cases, it becomes hard to measure the pedestrian speed manually to adjust each traffic light program because each traffic light would have countless setups in its controller and would not be able to adapt to new situations. Thus, there is need for automatic approaches.

Current automated approaches are divided into two groups. The first group, called *green-waves*, works coordinating a series of traffic lights to allow continuous traffic flow over several intersections in one main direction. The second, called *self-organizing traffic lights*, works to improve traffic flow. In this approach, the traffic lights are able to self-organize and adapt to changing traffic conditions. Both approaches have their limitations, as described in Section II-A, and mainly, do not give preference to pedestrians.

In this work, we propose the Traffic Light Manager (TLM), a framework to automatically adjust the traffic light program, changing the light signal to provide greater safety for pedestrians and to allow better flow traffic. We deal with two challenging cases in transport engineering literature, which will the traffic light to be self-manageable. The first case happens when pedestrians with reduced speed cannot cross the street within the available time, i.e., their estimated crossing time exceeds the remaining green time. Thus, the time of flashing red – upraised hand, in English language countries – will be increased according to the need of the pedestrian who cannot achieve a crossing speed of 1.22m/s. The second case happens when the traffic light for pedestrians remains open

for a long time even when there are no pedestrians waiting to cross, in which by watching the sidewalk the green light time might be increased for cars.

This work proposes an approach where the traffic light is self-organizable and makes decisions based upon tools of the computation vision. As pedestrian detection is important for several steps of our methodology, we employ the HOG detector [4] for such purpose. As pedestrian tracker we employ the Kalman filter due to its simplicity, optimality, tractability and robustness [5]. Since TLM require information of whether detected pedestrians lies on the crosswalk, we propose a segmentation step to determine the sidewalk and crosswalk regions of a monitored street. To perform such segmentation with robustness and better localization, we propose to use the pedestrian's distribution over a locality map.

The remaining of this work is organized as follows. In Section II, we reviewed the literature on the methods used. In Section III, we describe our model and its individual components. Finally, the proposed method is experimentally evaluated in Section IV.

## II. Related Work

In the following sections we discuss methods to coordinate traffic lights (Section II-A) and object tracking (Section II-B).

### A. Traffic Light Coordination

There are two major types of approaches in literature to coordinate traffic lights: *green-waves* and *self-organizing* [6]. Green-waves consist in determining the green light time in traffic lights between intersections according to the expected time travel between those intersections. The green light time is chosen considering the minimization of the time travel, often performed offline, for periods of the day such as morning, afternoon, rush hours, sportive events, weekends. This way, vehicles that follow the direction of successively green lights usually do not require stopping. Green light times, however, are usually optimized considering an average situation, which may not always result in optimal time travel. If the vehicles are moving slower than expected, for instance, the time determined to the green lights may, in fact, increase the average time travel.

Self-organizing traffic lights [6]–[12] consider the actual state of the traffic to determine the time of the green lights. Since optimizing the time travel for several intersections in a large urban area is computational expensive and unfeasible to calculate in real time, the optimization is performed in groups of few intersections. The advantage of approaches based on self-organizing traffic lights is that green lights can adapt when the traffic condition changes. However, self-organizing traffic lights usually require more sensors to proper measure variables in intersections, such as number of vehicles in the intersection, their average speed, and state of the traffic lights in the neighborhood.

Several approaches have been proposed to measure parameters for the self-organizing traffic lights, including car-to-car communication [13], magnetic loops [14], and computer vision techniques [15]. Although works in literature focus on coordinating traffic lights oriented to vehicles, in this work, we focus on developing a self-organizing traffic light oriented to pedestrians.

### B. Pedestrian Tracking

The pedestrian tracking presents several challenges related to partial occlusion and change in the appearance of the pedestrian due to, for instance, changes in the pose or in the lightning conditions. Some approaches consider online models to adaptively learn the appearance of the tracked subject [16], [17]. Such approaches, however, should carefully select samples of the subject being tracked to update the online model. Otherwise, accumulated errors from samples acquired using the online model can drift the model towards learning the appearance of the background in the scene. The drifting problem can be solved considering *tracking-by-detection* [18], [19], where only samples filtered by the pedestrian detection are presented to update the online model. The best results in tracking-by-detection are acquired when samples of the background are considered as counter-examples when training the detector [17].

In general, pedestrian tracking approaches consist in a motion model, an appearance model, and a tracking algorithm responsible to determine the subject location based on the models of motion and appearance [20], [21]. Regarding the appearance model, there are several examples in literature, including methods based on superpixel [16], pixel intensity templates [22], HOG descriptor [17], Haar-like features [23], among others. However, the use of appearance model is not mandatory, some approaches [18], [19] rely only in a motion model and in the proper assignment of successively detections to track subjects. Regarding the motion model, there are approaches in literature that consider Gaussian distributions [16], particle filters [23], Kalman filters [24], among others.

## III. Proposed Method

In this work, we propose a new approach able to automatically adjust the traffic light program by changing the light signal to provide greater safety for pedestrians and to allow better traffic flow (more details in Section III-C). The proposed approach requires information about the pedestrians to perform adjustments in the traffic light program, such as the need for increasing the time for pedestrian crossing. To provide such information, we propose a pedestrian tracking approach, based on Kalman filter, from which we obtain information such as the pedestrian's position and velocity (described in Section III-A). As our approach also needs to know the remaining distance of a pedestrian to the sidewalks, we propose a segmentation step based on the pedestrian's distribution map (described in Section III-B). These steps occur simultaneously and provide the required information to the manager, as illustrated in Figure 1.

### A. Tracking

We use pedestrian tracking to improve the accuracy of detection and to estimate the pedestrian speed. The velocity is

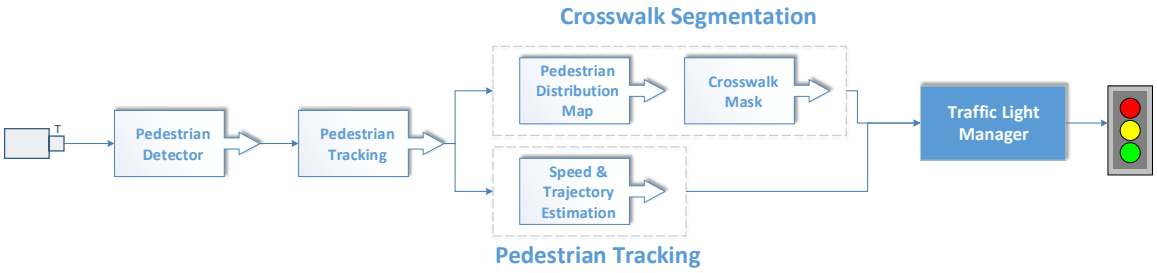



Fig. 1. Workflow of the proposed approach. The pedestrian position in each frame is determined in the pedestrian detector and pedestrian tracking modules. Then, the pedestrian position is used to estimate a distribution map where pedestrians walk when crossing the street and to estimate the speed and trajectory of these pedestrians to set the red flashing time in the pedestrian traffic light.

used to predict how long pedestrians that are in the crosswalk will take to reach the sidewalk. Therefore, if any pedestrian has low speed, e.g. pedestrians with reduced mobility, the time of the pedestrian traffic light will be increased proportionally. This prevents accidents that occur when the traffic light opens to the cars, and they move forward towards the pedestrians that are still crossing the road. We divided pedestrian tracking in three steps. First, we detect pedestrians using the HOG detector as described in [4]. Second, a motion model implemented using the Kalman filter is associated to each individual detected. Finally, we make associations between the detections of the new frame with individuals already been tracked.

*1) Pedestrian Detection:* Pedestrian detection is an important step for our methodology, since the remaining steps depends upon it. As the HOG detector presents lower computational cost and comparable false positive rate in our application, we employed it in this work. Nevertheless, one may consider any other pedestrian detector [25], more robust and/or faster ones, such as the PLS detector [26], the VeryFast detector [27], and the FPDW [28].

The HOG detector is a sliding-window based method consisting of the extraction of Histograms of Oriented Gradients (HOG). The HOG descriptor is extracted from a dense grid, which is obtained by dividing a detection window into overlapping blocks of size $16 \times 16$. Each block is divided into non-overlapping cells of size $8 \times 8$. For each cell, a 9-bin histogram of oriented gradients is computed, followed by a L2-norm block normalization with an optional clipping. The resulting histograms are then concatenated yielding a feature vector of 3,780 dimensions. Such feature vector is then presented to a linear SVM classifier.

*2) Pedestrian Tracking:* To model the motion of pedestrians using the Kalman filter, we consider the position $(p_i, p_j)$ and velocity $(v_i, v_j)$ of the center of the bounding box returned by the pedestrian detector. The $(v_i, v_j)$ parameters are estimated considering successively detections presented to the Kalman filter. Then, the prediction of the next bounding box of a pedestrian is calculated according to Equation 1, where $k$ is the current state, $k-1$ is the previous state, $A$ is the state transition matrix, and $w$ is the process noise.

$$X_k = Ax_{k-1} + w_{k-1} \quad (1)$$

The detections of the new frame may belong to one of two categories: a detection of a new pedestrian coming into the crosswalk or a new detection of the pre-existing pedestrian. In the first case, a Kalman filter model is created for each new subject. This new subject is tracked across the crosswalk and his/her velocity is taken into account when the traffic signal for pedestrians is about to close. In the second case, the new detection is associated with a pre-existing subject. To discover whether the new detection belongs to some pre-existing subject, we use a metric based on average and variance that will be explained later.

In addition to the aforementioned cases, the pedestrian might be occluded in the scene, which creates two cases to be processed. In the first case, the crosswalk may have a point of occlusion, for instance, as under the shade of a tree, or the pedestrian can stay occluded by another pedestrian. In this case, the subject motion model continues predicting, and the pedestrian can still be tracked. In the second scenario, the pedestrian may have left the crosswalk, and went into the sidewalk, being safe and no longer need to be tracked. Since we do not aim at identifying the individuals and only provide a safe crossing, we do not need to keep tracking the subjects after they reach the sidewalk. Thus, we fix $N$ as the number of frames that the pedestrian is then regarded outside the crosswalk and not only occluded. This $N$ is set according to the size of the crosswalk.

*3) Association Metric:* Given $n$ Kalman motion models learned considering detections from the past frames and $d$ detections of the current frame, the objective of the association metric is to link each one of the $d$ detections to one, and only one, subject relative to one of the $n$ motion models.

The association is performed as follow: the minimum distance $m$ between a detection from the current frame and the predicted state of each Kalman motion model is compared to the expected distance of the Kalman motion model being considered. The predicted state of the Kalman motion model is given in Equation 1. The expected distance between the prediction of a Kalman motion model and a detection is

calculated considering the exponentially weighted moving average and variance, presented in Equation 2.

$$\begin{aligned} \epsilon &= m - \mu_t \\ \mu_{t+1} &= \mu_t + \alpha \times \epsilon \\ \sigma^2_{t+1} &= (1-\alpha) \times (\sigma^2_t + \alpha \times \epsilon^2) \end{aligned} \tag{2}$$

The association is performed only if the probability of $m$ being draw from the same distribution $D$ of distances between predictions and detections of the Kalman motion model is above $p$. Considering $D$ normally distributed, the aforementioned comparison is calculated according to Equation 3.

$$m < \mu_t + k * \sigma \tag{3}$$

When creating new distribution models, we set $\mu$ to a maximum distance in pixels by which a pedestrian cannot move between two frames. Higher initial values of $\mu$, however, may incorrectly associate the detection of a subject being tracked in initial frames, when $\mu$ and $\sigma^2$ did not converged yet. The initial value of $\sigma^2$ is set to zero, although, a fast growth is observed while $\mu$ is converging.

### *B. Segmentation Based Upon Pedestrian Distribution Map*

We propose a segmentation step based on a pedestrian distribution map to estimate the crosswalk's borders with the sidewalk and the regions presenting pedestrian traffic. Such segmentation is required to determine whether a pedestrian is in the street or in the sidewalk. The segmentation is performed using the speed and position information generated by the tracking step. The speed vector contains the pedestrian speed on the $x$-axis and on the $y$-axis, we compute the norm of this vector, to determine whether the pedestrian is moving or not. If the norm of this vector is less than a threshold, then the pedestrian is stationary and is more likely to be on the sidewalk waiting for crossing. Otherwise, if the pedestrian exceeds the threshold, he/she is likely to be moving and most likely to be at the crosswalk.

The pedestrian's distribution map consists of a matrix accumulated over time, in which a position is incremented according to the detected pedestrian. We use the pedestrian's distribution map to segment the regions belonging to the crosswalk and the regions that belongs to the sidewalks. During a crossing, it is expected that the largest concentration of moving pedestrians will be on the crosswalk, while stationary pedestrians will be at the sidewalk.

### *C. Traffic Light Manager (TLM)*

Given the crosswalk location, the TLM verifies whether a pedestrian $p$ is in the crosswalk, and whether the crossing time duration of $p$ is higher than the remaining green time of the Pedestrian Traffic Light.

The crosswalk location is given by a binary mask from semaphore surveillance video. Let M be a matrix that representing this mask the locations where $M_{ij} = true$ determine the crosswalk area. The crossing time of $p$ is given by the equation of uniform rectilinear motion $t = \frac{d}{s}$, where $d$ is the distance between $p$ and the end of the crosswalk, and $s$ is the crossing speed of $p$. For calculated de distance $d$, we need determine the pedestrian direction, since end of the crosswalk is relative to the direction to which the pedestrian is moving. To determine the pedestrian direction, we calculate the angle $\theta$ between the vector $u$ that represents crosswalk direction , and the vector $v$ that represents the pedestrian position. The angle $\theta$ is given by Equation 4. If $\theta$ is negative we conclude that the pedestrian is moving to the left of the crosswalk, and if $\theta$ is positive we conclude that he is moving to the right.

$$\theta = \frac{u \cdot v}{||u|| \cdot ||v||} \tag{4}$$

The distance $d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$ where $(x_1, y_1)$ is the position of the $p$, and $(x_2, y_2)$ is the pixel of the crosswalk end border that is more distant from the pedestrian. This comparison between pedestrian and the more distant pixel, ensures the safety of the pedestrian crossing even if he makes the longest trajectory to exit the crosswalk. The speed of $p$ is given by the length of the velocity vector $v$. The vector $v$ is given by the Kalman model associated with it, and the computation of the length of $v$ is given by $||v||$.

Given the equation and definitions above, if the crossing time $t$ of the pedestrian $p$ is greater than the remaining time $r$ of the open semaphore for pedestrian crossing, the state red-flashing light for pedestrians will increment in $n$ frames and the traffic light for cars will remain closed, where $n = t - r$ ensuring a safety crossing for pedestrians. Since pedestrians are tracked along time by pedestrian tracking. Therefore, if the speed of the $p$ change and its crossing time become shorter than the remaining green time, the pedestrian traffic light will not be incremented. The TLM will increase the traffic light according to the greatest need for increment. Thus, if there is another pedestrian who has crossing time greater than the time of $p$, the traffic light will be increased with the time required for this slower pedestrian.

## IV. Experiments

In this section, we present experiments to evaluate the steps of our proposed approach. First, we demonstrate the applicability of the proposed tracking method for tracking pedestrians in the crosswalk. Second, we evaluate the TLM compared to the real traffic signal, which does not dynamically adapt to pedestrian and vehicle flow.

### *A. Experimental Setup*

The parameters of the proposed approach include the parameters from the HOG detector, Kalman filter, and association metric. Regarding the implementation, we use the Smart Surveillance Framework (SSF) [29], since it contains a public implementation of the HOG detector.

**Pedestrian Detector.** We tested several combinations of HOG detector input parameters to obtain a more robust method for each specific situation. To evaluate the pedestrian detection, we set the horizontal and vertical stride of the sliding window

Fig. 2. In the left image, we show a common situation in the region where the dataset [30] was recorded, elderly pedestrians and mothers with babies on their lap crossing in the crosswalk. In the right image, we show one of the three sequences of the PETS dataset, the challenges of this dataset are occlusion and pedestrians crowd. PETS was used only to evaluate the tracking proposed method and the dataset [30] was used to evaluate the TLM.

to 6 pixels and scale step to 0.5. These parameters control how much displacement is given between two detection windows and between different scales. To evaluate the TLM, we set the horizontal and vertical stride to 8 pixels and the scale to 1.05.

**Association Metric.** In the tracking and traffic lights manager evaluation, we fix the parameter $\mu$ as 0.1, and we evaluate two different values for the $\alpha$ parameter, 15 and 25. The $\alpha$ parameter set as 15 becomes the association more restrictive, while as 25, the association is more flexible.

**Kalman Filter.** We divide the evaluation into two groups, one that the positions are given by the Kalman predicted state and other that the positions are given by the estimated state. Within each group, we present the best result by testing several different parameters from the HOG detector and from the association metric.

**Datasets.** The experiments were conducted on two datasets, show in the Figure 2. PETS 2009 Dataset S2 and in a dataset composed of urban crosswalk images acquired from a hospital area [30]. The former is used to evaluate the proposed tracking method, while the latter is used to evaluate the TLM. PETS 2009 Dataset S2 is a standard dataset widely used for tracking evaluate. This dataset have three scenarios with three difficult levels, L1, L2 and L3, respectively. Scenario L1 consists of a sequence of 784 frames and contains a sparse crowd sequence where some pedestrians walk through a courtyard. The scenarios L2 and L3 consist of a sequence of 435 and 239 frames, respectively. In these scenarios, the difficult level increases as they present pedestrian crowd and many occlusion cases. In the evaluation of the TLM, we use a video recorded near a hospital area in Belo Horizonte/Brazil [30]. For this purpose, we labeled every sixth frame, resulting in a total of 3775 labelled frames. The information annotated is the traffic light signal (green, red, flashing red). For the frames where the signal is red or flashing red, we also annotate whether or not the pedestrian is present in a crosswalk. This was done to evaluate our method, the information about the light signal state is considered to increase the time of the red flashing lights whenever our approach detect pedestrians in the crosswalk.

### B. Tracking Evaluation

We evaluate the proposed tracking considering four parameter setups, which consist in the combination of two values for $\mu_0$, 15 and 25, and predicted or estimated state of the Kalman filter. The setups are presented in Table I.

TABLE I
SETUPS CONSIDERED IN THE COMBINATION OF KALMAN FILTER AND HOG DETECTOR.

| | Tracking | Association Metric |
|---|---|---|
| SetupA | Prediction | $\mu_0 = 25$ |
| SetupB | Prediction | $\mu_0 = 15$ |
| SetupC | Estimated | $\mu_0 = 25$ |
| SetupD | Estimated | $\mu_0 = 15$ |

Results considering miss rate and false positives per-image (FPPI), for each setup and in the PETS 2009 dataset, are presented in Table II. We shows in Table II the results of comparison of the our proposed methods against the HOG detector. Although our proposed tracking method achieved a smaller recall rate than the HOG detector in the L1 sequence, it achieved higher recall rates in the L2 and L3 sequences (*i.e.*, more difficult sequences) because it is better to handle occlusion than the detector by itself.

TABLE II
COMPARISON OF HOG DETECTOR AND OUR PROPOSED TRACKING METHOD. RECALL IS COMPUTE AT 1 FPPI

| | Scenario L1 | Scenario L2 | Scenario L3 |
|---|---|---|---|
| HOG Detector [4] | 0.679 | 0.227 | 0.305 |
| Tracking(A) | 0.497 | 0.274 | 0.135 |
| Tracking(B) | 0.515 | 0.274 | 0.336 |
| Tracking(C) | 0.462 | 0.222 | 0.324 |
| Tracking(D) | 0.483 | 0.274 | 0.317 |

### C. Traffic Light Manager Evaluation

We evaluate the self-organizing traffic lights proposed method. For this purpose, we use six sequences of the traffic signal states (green, flashing red, red) of the dataset in [30]. Our major concern is to improve the safety of the pedestrian crossing allowing the traffic light to self-organize according to pedestrian speed at that region of the city. In Figure 3 we present results indicating the number of pedestrians in the crosswalk when the pedestrian traffic light turns red and the time increased to allow the pedestrians to safely cross the street.

According to Figure 3, best results are achieved when increasing around 40 seconds (6.66 seconds on average in each red flashing light cycle), when all pedestrians make to cross the street in the red flashing light. Table III show time increased in each sequence. The results of the TLM show the feasibility of the proposed approach. However, we do not consider the impact of the increase in time for the red flashing light on the vehicle flow. To balance between safety and vehicle flow, we suggest varying the additional time in Figure 3.

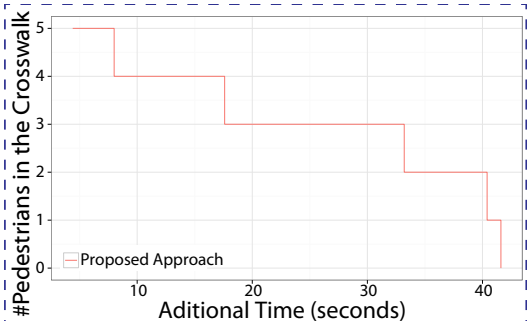



Fig. 3. Number of pedestrians in the crosswalk when the pedestrian traffic light turns red compared to the time given to pedestrians to cross the street. Better approaches present few pedestrians in the crosswalk and few additional time.

TABLE III
TIME (IN SECONDS) ADDED TO EACH RED FLASHING LIGHT SEQUENCE TO ALLOW ALL PEDESTRIANS TO SAFELY CROSS THE STREET.

| Seq. 1 | Seq. 2 | Seq. 3 | Seq. 4 | Seq. 5 | Seq. 6 |
|---|---|---|---|---|---|
| 6.4 | 6.6 | 7.0 | 6.6 | 7.4 | 11.0 |

## V. CONCLUSION

We proposed and evaluated an approach of self-organizing traffic light oriented to pedestrians based on computer vision techniques. To enhance the results of the pedestrian detection, we model temporal information considering the Kalman filter. Results show the feasibility in consider the proposed approach of self-organizing traffic light, which adaptively increase the time of different sequences of red flashing light based on the velocity of the pedestrians crossing the street. In future works, one may consider appearance models in our tracking approach since it has been shown to increase the accuracy of the tracking. A comparison of the proposed self-organizing traffic light with real pedestrian traffic lights may also be presented along with a study of impact of the the proposed traffic light in the vehicle flow.

## ACKNOWLEDGMENTS

The authors would like to thank the Brazilian National Research Council – CNPq (Grant #477457/2013-4) and the Minas Gerais Research Foundation - FAPEMIG (Grant APQ-01806-13).

## REFERENCES

[1] Federal Highway Administration (FHWA), *Manual on Uniform Traffic Control Devices for Streets and Highways*, 2009.
[2] J. L. Pline, *Traffic Engineering Handbook*, 5th ed. Institute of Transportation Engineers, 1999.
[3] A. M. C. B. Silva, J. R. R. da Cunha, and M. R. d. C. A. Ferreira, "Velocidade Pedonal em Atravessamentos - Avaliação da Relevância dos Fatores Explicativos," in *Congresso de Ensino e Pesquisa em Transportes (ANPET)*, 2012.
[4] N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection," in *Intl. Conference on Computer Vision and Pattern Recognition*, 2005.
[5] S. J. Julier and J. K. Uhlmann, "A new extension of the kalman filter to nonlinear systems," in *Aerosense*, 1997.
[6] C. Gershenson and D. A. Rosenblueth, "Modeling self-organizing traffic lights with elementary cellular automata," *Complexity*, 2009.
[7] N. Geroliminis and A. Skabardonis, "Identification and analysis of queue spillovers in city street networks," *Intelligent Transportation Systems*, 2011.
[8] D. Helbing and A. Mazloumian, "Operation regimes and slower-is-faster effect in the controlof traffic intersections," *The European Physical Journal B*, 2009.
[9] D. A. R. Carlos Gershenson.
[10] D. Mckenney and T. White, "Distributed and adaptive traffic signal control within a realistic traffic simulation," *Eng. Appl. Artif. Intell.*, pp. 574–583, 2013.
[11] C. Gershenson and D. A. Rosenblueth, "Self-organizing traffic lights at multiple-street intersections," *Complex.*, pp. 23–39, 2012.
[12] D. Helbing, P. Molnr, I. J. Farkas, and K. Bolay, "Self-organizing pedestrian movement," *Environment and Planning B: Planning and Design*, pp. 361–383, 2001.
[13] N. Maslekar, M. Boussedjra, J. Mouzna, and H. Labiod, "Vanet based adaptive traffic signal control," in *Vehicular Technology Conference*, 2011.
[14] M. Tubaishat, Y. Shang, and H. Shi, "Adaptive traffic light control with wireless sensor networks," in *Consumer Communications and Networking Conference*, 2007.
[15] A. Fernández-Caballero, F. J. Gómez, and J. López-López, "Road-traffic monitoring by knowledge-driven static and dynamic image analysis," *Expert Systems with Applications*, 2008.
[16] S. Wang, H. Lu, F. Yang, and M.-H. Yang, "Superpixel tracking," in *Intl. Conference on Computer Vision*, 2011.
[17] H. Pirsiavash, D. Ramanan, and C. C. Fowlkes, "Globally-optimal greedy algorithms for tracking a variable number of objects," in *Computer Vision and Pattern Recognition*, 2011.
[18] B. Benfold and I. Reid, "Stable multi-target tracking in real-time surveillance video," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, 2011.
[19] J. Berclaz, F. Fleuret, E. Turetken, and P. Fua, "Multiple object tracking using k-shortest paths optimization," *Pattern Analysis and Machine Intelligence*, 2011.
[20] G. P. F. Gisele M. Simas, "Using computer vision for 3d propabilistic reconstruction an motion," in *Computational Modeling (MCSUL)*, 2007.
[21] N. T. Siebel, "Fusion of multiple tracking algorithms for robust people tracking," in *European Conference on Computer Vision (ECCV)*, 2011.
[22] W. Zhong, H. Lu, and M.-H. Yang, "Robust object tracking via sparsity-based collaborative model," in *Computer Vision and Pattern Recognition*, 2012.
[23] B. Babenko, M.-H. Yang, and S. Belongie, "Robust object tracking with online multiple instance learning," *Pattern Analysis and Machine Intelligence*, 2011.
[24] K. Van Beeck, T. Goedemé, and T. Tuytelaars, "Towards an automatic blind spot camera: robust real-time pedestrian tracking from a moving camera," in *Conference on Machine Vision Applications*, 2011.
[25] P. Dollar, C. Wojek, B. Schiele, and P. Perona, "Pedestrian Detection: An Evaluation of the State of the Art," *Pattern Analysis and Machine Intelligence*, 2012.
[26] W. Schwartz, A. Kembhavi, D. Harwood, and L. Davis, "Human Detection Using Partial Least Squares Analysis," in *Intl. Conference on Computer Vision and Pattern Recognition*, 2009.
[27] R. Benenson, M. Mathias, R. Timofte, and L. Van Gool, "Pedestrian Detection at 100 Frames per Second," in *Intl. Conference on Computer Vision and Pattern Recognition*, 2012.
[28] P. Dollár, R. Appel, S. Belongie, and P. Perona, "Fast Feature Pyramids for Object Detection," *Pattern Analysis and Machine Intelligence*, 2014.
[29] A. C. N. Jr., C. E. dos Santos, R. Ferreira, and W. R. Schwartz, "Smart Surveillance Framework: A Versatile Tool for Video Analysis," in *IEEE Winter Conference on Applications of Computer Vision*, 2014.
[30] F. Gualberto, J. A. Dias, M. M. Thebit, and H. M. Barbosa, "Velocidade de pedestres em área hospitalar – um estudo exploratório," in *Congresso Nacional de Pesquisa e Ensino em Transporte (ANPET)*, 2013.

# Development of an application for security based in face recognition on Android platform.

Thaisa A. Correia, Marco A. Piteri, Almir O. Artero,
Unesp - Universidade Estadual Paulista
FCT - Faculdade de Ciências e Tecnologia
Presidente Prudente – SP, Brasil
thaisa.correia3@gmail.com, {piteri, almir}@fct.unesp.br

Francisco Assis da Silva, Danillo Roberto Pereira
Unoeste - Universidade do Oeste Paulista
FIPP – Faculdade de Informática de Presidente Prudente
Presidente Prudente – SP, Brasil
chico@unoeste.br, dpereira@ic.unicamp.br

***Abstract*—The primary objective of this work is to develop an application based on the Android platform that is capable of capturing a digital image, detect and recognize a human face, in order to verify if it belongs to a database of previously defined data for the following authenticate the user, allowing the locking and unlocking applications and access to personal data, promoting security of them. Among various existing methods of face recognition, we have chosen two that are available in the OpenCV computer vision library, Eigenfaces and Local Binary Patterns (LBP), which were developed and used in conjunction with three preprocessing techniques in order to improve the final results. The analysis of the experiments, which were conducted into uncontrolled conditions, indicate a greater than 86% accuracy, allowing to consider the approach acceptable, taking into account the stage of development of work and comparing it to an application available on the market.**

***Keywords—Android; security; computer vision; face recognition;***

## I. INTRODUÇÃO

Os dispositivos móveis estão cada vez mais ganhando espaço no cenário atual, possuem alto desempenho, agregam diferentes tipos de tecnologias (telefonia, internet, processamento e consumo de dados) antes separadas, e, um fator diferencial, a mobilidade. Com o desenvolvimento e expansão dessa plataforma, houve a necessidade da criação de sistemas operacionais direcionados a essa nova família de dispositivos e entre eles se destaca o Android, que tem obtido o crescimento mais acentuado no mercado.

Indivíduos em diferentes lugares do mundo usam dispositivos móveis para realizar diversas tarefas, como armazenar e acessar dados, pagamento de contas, navegação precisa por diferentes ambientes baseados em georreferenciamento, entre outras. Como consequência da importância que estes dispositivos estão assumindo nas vidas das pessoas é natural que muitos dados de natureza sigilosa sejam utilizados em variadas aplicações. Logo, uma questão emergente e de extrema importância está relacionada à segurança dessas informações, que em geral, são mais valiosas que o próprio hardware e/ou software [12].

Para agravar ainda mais este cenário, vários problemas podem ocorrer, como roubo, perda ou até mesmo tentativas de violação. Logo, assegurar por meio de algum esquema de certificação quem usa as aplicações e garantir que somente o proprietário dos dados confidenciais armazenados no dispositivo tenha acesso a eles, deve ser uma preocupação permanente. Há muitas maneiras de certificação e autenticação de usuários em dispositivos móveis, entretanto, esse trabalho irá fazer uso do reconhecimento facial, que é uma das características biométricas explorada em diferentes contextos, não é invasiva ao usuário e não exige equipamentos de captura custosos e de alto desempenho, sendo possível elaborar soluções em plataformas com algumas limitações, como dispositivos móveis.

Reconhecer faces humanas em imagens digitais é uma área de pesquisa atual, está em constante desenvolvimento e o maior problema enfrentado é intrínseco à qualidade da imagem capturada, como o excesso ou falta de iluminação, baixa resolução, ruídos, orientação e pose, ocultação parcial da face, diferentes expressões ou imagens com grande volume de informações [1]. Tais situações dificultam a detecção e extração de características, que são passos cruciais no processo de reconhecimento facial, daí a importância da etapa de pré-processamento para o sucesso dessa tarefa, principalmente ao se utilizar imagens capturadas em ambientes comuns dos usuários de dispositivos móveis, em que os fatores externos normalmente não podem ser controlados.

A Seção II faz uma breve revisão das principais abordagens de reconhecimento facial presentes na literatura. Posteriormente, na Seção III é realizada a descrição da metodologia desenvolvida, e na seção subsequente os experimentos e resultados são discutidos. Por fim, a Seção V apresenta a conclusão do trabalho e pontos a serem explorados na continuidade do mesmo.

## II. RECONHECIMENTO FACIAL

O processo de identificar diferentes tipos de padrões é uma tarefa bastante simples para os seres humanos e, entre elas, está o reconhecimento de pessoas baseado em suas características faciais. Entretanto, automatizar esse processo e outorgá-lo a sistemas computacionais é uma tarefa extremamente complexa.

Em geral, os métodos de reconhecimento facial são divididos em duas etapas, a saber: o treinamento e o reconhecimento. Na etapa de treinamento, são extraídas as características das faces por meio da metodologia aplicada e estas são salvas (base de dados, arquivo de texto), para que num momento posterior possam ser recuperadas pelo sistema

computacional. Por sua vez, na etapa de reconhecimento são extraídas características de uma face arbitrária que é então comparada com as características previamente armazenadas na etapa anterior. Com base nos resultados dessa comparação uma resposta é inferida e disponibilizada. A imagem capturada em ambas as etapas é normalmente submetida a um conjunto de operações que procuram melhorar a qualidade da imagem original, como a normalização e identificação da região que contém a face. Na sequência são extraídas, selecionadas e classificadas as características definidas previamente pela metodologia [2].

Atualmente, muitos métodos buscam solucionar o problema de reconhecimento facial, porém nenhum deles apresenta resultados com exatidão da resposta alcançada. As abordagens de reconhecimento facial podem ser classificadas em três categorias: métodos holísticos, estruturais e híbridos. Uma análise e taxonomia completa das técnicas podem ser obtidas a partir dos trabalhos de Zhao [3] e Jafri [4]. No contexto desse trabalho a discussão será restrita as abordagens *Eigenfaces* [5, 6, 7, 8] e LBP (*Local Binary Patterns*) [9, 10, 11], ambas pertencentes à classe dos métodos holísticos.

### *A. Métodos Holísticos*

A imagem bidimensional é representada por uma matriz de *pixels*, onde cada elemento é definido por um valor de intensidade. Os métodos holísticos se baseiam nessa representação total da imagem, utilizam a informação global da face para o reconhecimento. A ideia básica é realizar a correlação entre as características da imagem de referência e imagens reunidas anteriormente para o reconhecimento com base nos seus valores de intensidade, para efetuar a classificação. Essa abordagem é bastante flexível, permite aplicações de técnicas de extração e classificação que se ajustem ao problema e apresenta melhores resultados que os métodos estruturais, pois variações muito pequenas entre a referência e a consulta não afetam tanto o resultado final. A principal desvantagem deve-se pelo fato de serem métodos muito sensíveis à variação da iluminação, fator determinante para a definição dos valores de intensidade numa imagem. Também é preciso trabalhar em um espaço de alta dimensão de informações, o que torna o processo custoso, principalmente em um grande volume de dados. Nesse contexto, são utilizados métodos de redução de dimensionalidade, que em geral são estatísticos ou empregam redes neurais. Em seguida, é realizada uma breve descrição das duas metodologias holísticas empregadas no trabalho: *Eigenfaces* e LBP.

- *Eigenfaces*

A imagem que representa a face possui grande dimensão de informações, o objetivo é identificar características que são determinantes para diferenciar uma face da outra, utilizando autovetores associados com os autovalores dominantes do padrão da matriz de covariância, baseia-se na projeção linear do espaço de imagens das faces em um espaço de características com dimensões reduzidas fazendo uso da técnica de Análise de Componentes Principais (PCA), com algumas modificações otimizadas. O fluxo básico desse processo é ilustrado na Fig. 1. A primeira etapa é o treinamento, que é realizado a partir de conjunto de $N$ faces. Seja $\mathbf{I}_i$ ($\forall i \in [1,N]$) uma imagem, onde $p$ e $q$ são, respectivamente, o número de linhas e de colunas da matriz que as representa. Cada uma delas é considerada um padrão de $p \times q$ características estruturadas pelo vetor $\mathbf{\Gamma}_i = (x_1,x_2,...,x_n)$ definido pelas colunas da matriz, onde $x_i$ é o valor de intensidade e $n = p \times q$. A Equação 1 representa a média do conjunto dos vetores de características, enquanto a Equação 2 indica o cômputo entre cada vetor $\mathbf{\Gamma}_i$ e a média calculada.

$$\psi = \frac{1}{N}\sum_{i=1}^{N}\Gamma_i \quad (1)$$

$$\phi_i = \Gamma_i - \psi \quad (2)$$

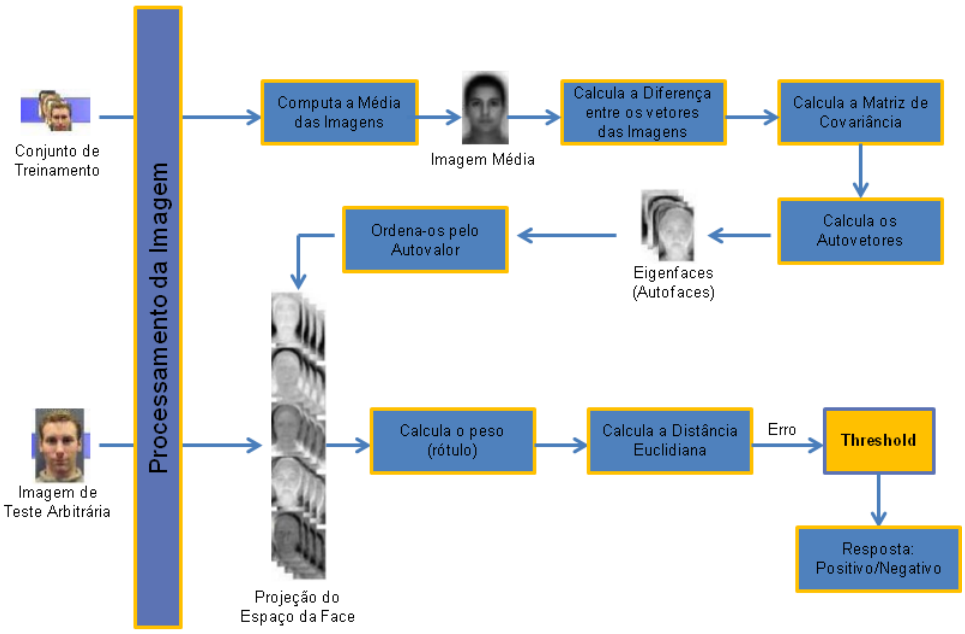



Fig. 1. Fluxo básico do reconhecimento facial através do método *Eigenfaces*. Adaptado de [8].

Com o propósito de eliminar a redundância das informações fornecidas pelas características é obtida a matriz de covariância dada pela Equação 3, com suas colunas definidas pela aproximação do valor esperado de cada vetor padrão. Em seguida, esta é diagonalizada, calculando outra matriz de mudança de base do espaço de características através dos autovalores e autovetores, representado pela Equação 4.

$$C = \frac{1}{N}\sum_{i=1}^{N} \phi_i \phi_i^t \quad (3)$$

$$Cv_i = \lambda_i v_i \quad (4)$$

Os autovetores calculados são ordenados do maior para o menor conforme o autovalor correspondente e os principais componentes *k* são selecionados para formar o novo espaço, constituído pelos autovetores de maiores autovalores associados, também denominados *eigenfaces*. O último passo é efetuar a classificação do padrão da face, atribuindo um peso para cada, sendo $\Omega_\Lambda = [\omega_1, \omega_2, ..., \omega_k]$ para cada.

Na etapa de reconhecimento, dada uma imagem arbitrária, esta é projetada no espaço formulado na etapa de treinamento, e um peso $\omega$ é associado a ela. E então é aplicado um classificador, que em geral é a distância euclidiana, entre o peso calculado e todos os *k* pesos, dado pela Equação 5. Geralmente um limiar é definido para indicar a tolerância da diferença na escolha do padrão.

$$\varepsilon_i = \left\| \omega - \omega_i \right\| \quad (5)$$

*Eigenfaces* é a metodologia mais estudada e utilizada atualmente, pois apresenta bons resultados. Entretanto, possui algumas limitações como a alta sensibilidade a condições de iluminação do ambiente, expressão e orientação da face.

- *LBP (Local Binary Patterns)*

Essa técnica foi originalmente desenvolvida para a análise e descrição de texturas em imagens, mas tem sido amplamente utilizada para o reconhecimento da face e suas expressões. Em síntese, extrai características de uma imagem baseando-se na comparação de cada *pixel* com sua vizinhança, fornecendo informações detalhadas da imagem analisada.

No processo, ilustrado na Fig. 3, a imagem é dividida em *m* regiões, sendo que em cada região é aplicado um operador definido que compara o *pixel* central com os seus vizinhos, aplicando-se um limiar. Se o valor do vizinho for menor que o elemento central, então é atribuído a ele o valor 0 (zero), caso contrário, o valor 1 (um) , conforme pode ser observador na Fig. 2. O resultado é um valor binário para cada *pixel*, dos quais realizando a concatenação forma um número de $2^V$ combinações, sendo *V* a quantidade de vizinhos definida pelo operador, como é ilustrado no exemplo da Fig. 2.

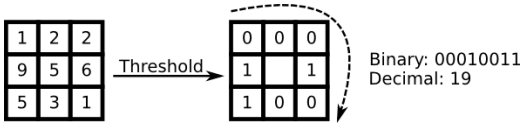



Fig. 2. Exemplo operador LBP 3X3. Retirado de [16].

A partir da aplicação do operador LBP é composto um histograma para cada região, do qual descreve o padrão da mesma. Por sua vez, a descrição da face é definida através da concatenação dos histogramas resultantes.

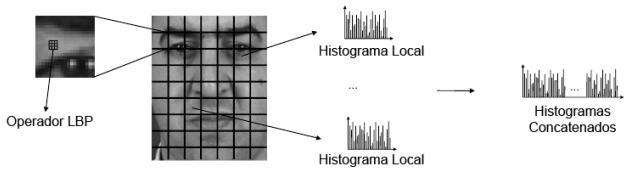



Fig. 3. Autenticação da face utilizando LBP. Adaptador de [11].

A etapa do reconhecimento facial é realizada calculando-se a correlação dos histogramas das imagens obtidos na fase de treinamento e o histograma de uma imagem arbitrária a ser verificada.

Devida limitações do operador diante de imagens com diferentes escalas, essa técnica foi estendida para ser possível utilizar o tamanho da vizinhança de acordo com a necessidade do problema, por meio de uma vizinhança circular e interpolação dos valores de intensidade definindo um raio de abrangência.

O método apresenta bom desempenho diante de variações de iluminação, escala ou orientação, e também não é necessário utilizar mais de uma imagem para o reconhecimento, ou seja, não é necessário consumir grande quantidade de memória, em contraste com a metodologia *Eigenfaces*.

### *B. Métodos Estruturais*

Os métodos estruturais, também conhecidos como baseados em características, não analisam a imagem como um todo, mas utilizam componentes locais da face para o reconhecimento como os olhos, nariz, boca, entre outros. É realizada a extração de características discriminantes e mensuráveis da imagem da face e então é computada a relação entre essas características.

A grande vantagem dessas metodologias é que fazem uso de pouca quantidade de memória e o processamento é realizado diretamente na imagem da face e de forma muito rápida. Entretanto, alguns dos métodos pertencentes a esta classe possuem limitações determinantes quanto ao reconhecimento. Se os limites de tolerância forem muito grandes o reconhecimento será comprometido, e em contrapartida se forem muito pequenos será impossível encontrar a correlação existente, ou seja, é necessário ter precisão exata quanto às medidas utilizadas.

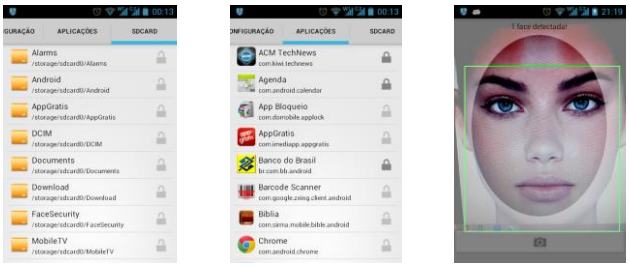
### *C. Métodos Híbridos*

Estudos indicam que o sistema de percepção do ser humano utiliza a metodologia holística e estrutural para reconhecer a face de um indivíduo, analisando os componentes locais e mensuráveis da face, assim como características globais, atingindo grande sucesso de desempenho. Por sua vez, os sistemas computacionais buscam emular esse comportamento, de modo que, cada abordagem pode suprir a deficiência da outra, possibilitando extrair as suas respectivas vantagens e

reforçar os resultados, oferecendo uma maior confiabilidade na resposta alcançada. Em contrapartida, a complexidade estrutural da solução, tempo de resposta e custo computacional são consideravelmente elevados.

## III. METODOLOGIAS ADOTADAS

O conjunto de funcionalidades associadas à aplicação desenvolvida se concentra essencialmente em dois módulos – o de segurança e o de reconhecimento facial. Na Fig. 4 podem ser observadas algumas telas capturadas do aplicativo.



(a) (b) (c)

Fig. 4. Capturas de tela do aplicativo em desenvolvimento: (a) seleção de bloqueio de arquivos (b) e de aplicações; (c) Autenticação da face e o consequente desbloqueio.

### A. *Metodologia de Segurança*

Dois pontos importantes foram levantados, a segurança dos dados e das aplicações, sendo que para cada um o tratamento foi realizado de maneira diferente, levando em consideração os três pilares da segurança da informação, confidencialidade, integridade e disponibilidade [13, 14].

Para a segurança das aplicações foi implementado um mecanismo de bloqueio onde o usuário define previamente quais aplicações deseja submeter. Quando ele ativa a segurança um serviço em *background* é disparado, o qual detecta se alguma aplicação foi iniciada e analisa se está faz parte da lista de bloqueio ou não. Se fizer parte, o bloqueio é disparado pelo serviço, que é a tela de reconhecimento facial, caso contrário nenhuma ação é lançada. O usuário pode tentar efetuar o desbloqueio apenas três vezes, se a aplicação não reconhecer a face nessas tentativas, então a tela de desbloqueio mediante a senha alfanumérica é disparada, para que a disponibilidade não seja afetada, se porventura o usuário estiver em um ambiente onde não é possível assegurar a qualidade da imagem capturada.

O bloqueio de dados é restringido a apenas diretórios raízes e seus respectivos subdiretórios do cartão de memória externa do dispositivo. Dentro da aplicação o usuário define qual diretório deseja bloquear. Ao efetuar o bloqueio este é compactado em um arquivo e submetido à criptografia de chave simétrica, garantindo a integridade dos dados, e então esse arquivo é oculto ao sistema, o qual só é possível ser acessado através da aplicação que efetuou o bloqueio. No desbloqueio o mesmo processo é realizado inversamente.

### B. *Metodologia de Reconhecimento Facial*

O processo de reconhecimento facial em imagens bidimensionais abordado no trabalho foi dividido em etapas básicas: aquisição da imagem, detecção da face na imagem, pré-processamento ou normalização da imagem, extração de características definidas e, por fim, a classificação.

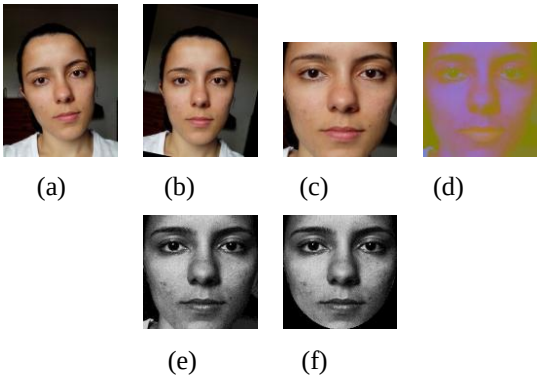
A partir da captura da imagem através da API (*Application Programming Interface*) disponibilizada pelo Android o próximo passo é efetuar a detecção da face na mesma.

Inicialmente foi utilizada para a detecção da face, a técnica Viola-Jones [15] disponível na OpenCV [16], mas pôde-se analisar que ela é muito sensível a variações da condição de iluminação do ambiente, não sendo aplicável no contexto dos dispositivos móveis. Então se optou pela abordagem de detecção de face disponibilizada pelas ferramentas do Android [17], que emprega dentre outras, técnicas baseadas na segmentação da pele para efetuar a detecção.

Salienta-se que etapa de pré-processamento é crucial para o sucesso da abordagem de reconhecimento facial, pois nesse procedimento o objetivo é conservar o padrão do objeto eliminando os fatores que dificultam a extração de suas características. Com a finalidade de melhorar o desempenho das metodologias utilizadas devido ao fato de se utilizar imagens capturadas em ambientes comuns dos usuários de dispositivos móveis em que os fatores externos não são controlados, o que aumenta consideravelmente o nível de dificuldade do processo de reconhecimento [18], neste trabalho foram implementadas três técnicas que são discorridas na sequência.

O módulo de reconhecimento facial realiza a extração e seleção de características da face e efetua a sua classificação. No desenvolvimento realizado foram empregadas duas metodologias holísticas, *Eigenfaces* e LBP (*Local Binary Patterns*), das quais foram descritas anteriormente e que são disponibilizadas pela biblioteca de visão computacional *open source* OpenCV [16]. Diferentes técnicas de pré-processamento foram implementadas em conjunto com cada método, para ser analisado os resultados através de testes, em relação à melhora do desempenho e taxa de acerto.

- *Equalização do Canal de Luz (Espaço cor YCbCr)*



(a) (b) (c) (d)

(e) (f)

Fig. 5. Etapas da equalização do canal de luz (Espaço cor YCbCr): (a) Imagem capturada; (b) Rotação da imagem; (c) Recorte da área da face; (d) Conversão YCbCr e equalização do canal Y; (e) Conversão em níveis cinza; (f) Aplicação da máscara e normalização do histograma.

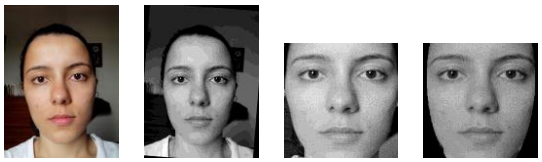
A partir de uma imagem capturada pelo dispositivo, a correção da distância e orientação é realizada a partir da detecção dos olhos na face. Ao detectar os olhos, a posição do olho esquerdo é fixada, obtendo-se o ângulo formado com o olho direito, sendo utilizada para efetuar a rotação da imagem. Posteriormente a face é recortada proporcionalmente à distância dos olhos obtida.

A imagem capturada pelos dispositivos móveis normalmente são representadas no espaço de cor RGB (*Red, Green and Blue*), que utiliza três canais de cores para cada *pixel*, sendo que cada um deles sofre influência da luz ambiente, dificultando o processo de correção do excesso ou falta de iluminação na imagem capturada. Devido a isso, é realizada a conversão da imagem para o espaço de cor YCbCr que utiliza apenas um canal para a informação de luz, e logo em seguida é feita a equalização do canal Y, a fim de corrigir a distribuição de luz na imagem.

O próximo passo é a conversão da imagem em níveis de cinza, e aplicação de uma máscara, para serem removidos detalhes que interfiram no processo de reconhecimento e, então, o histograma da imagem resultante é normalizado. Os efeitos de todos os passos descritos são ilustrados na Fig. 6.

- *Correção da iluminação aplicando um fator γ*

A partir da imagem capturada é realizada a sua rotação para o ajuste da orientação, e realizada a conversão do espaço de cor em níveis de cinza [19]. A área da face é recortada e a correção através do fator $\gamma$ é aplicada, definindo um valor de intensidade médio para que todos os valores de intensidade da imagem sejam ajustados proporcionalmente, nivelando os níveis de cinza da imagem e corrigindo as possíveis regiões escuras ou claras demais. Os efeitos de todos os passos descritos são ilustrados na Fig. 7.



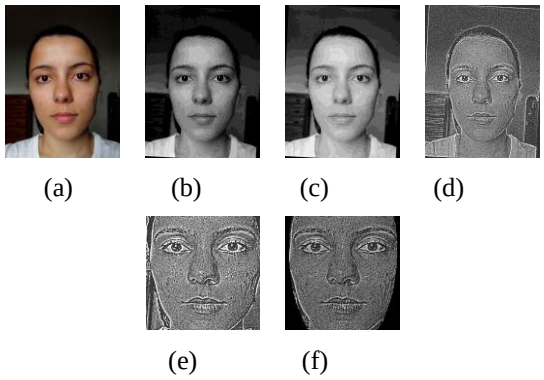
(a) (b) (c) (d)

Fig. 6. Etapas da correção da iluminação aplicando um fator $\gamma$: (a) Imagem capturada; (b) Rotação e conversão em níveis de cinza; (c) Aplicação do fator $\gamma$ e recorte da área da face; (d) Aplicação da máscara e normalização do histograma.

- *Abordagem de Xiaoyang Tan and Bill Trigg*

A abordagem descrita na referência [19] tem como objetivo eliminar os efeitos indesejados da iluminação e conservar os detalhes do objeto. Os passos iniciais são os mesmos que a abordagem de pré-processamento anterior, a partir de uma imagem, é realizado o ajuste de orientação, conversão em níveis de cinza e a aplicação do fator $\gamma$.

Em seguida, é feita duas cópias da imagem do processo e em cada uma é aplicado uma filtragem gaussiana diferente, uma com valor *sigma* 1 e outra valor *sigma* 2, e então é realizada a diferença entre elas, resultando em uma imagem com detalhes realçados. O recorte da área da face é realizado e finalmente a máscara e normalização do histograma são aplicadas.



(a) (b) (c) (d)

(e) (f)

Fig. 7. Abordagem de Xiaoyang Tan and Bill Trigg: (a) Imagem capturada; (b) Rotação e conversão em níveis de cinza; (c) Aplicação do fator $\gamma$; (d) Diferença da filtragem gaussiana; (e) Recorte da área da face; (f) Filtragem, aplicação da máscara e normalização do histograma.

## IV. EXPERIMENTOS E RESULTADOS PRELIMINARES

A aplicação desenvolvida na plataforma Android, foi escrita na linguagem de programação Java. Entretanto as funções de pré-processamento da imagem e métodos de reconhecimento facial foram escritos na linguagem de baixo nível C/C++, para ter um ganho no poder de processamento e aproveitamento da memória do dispositivo, que executa todo o fluxo da aplicação, sendo que a interface de ambos módulos é realizada através das ferramentas disponibilizadas pelo NDK (*Native Development Kit*) [20].

Dois métodos de reconhecimento facial foram empregados em conjunto com as técnicas de pré-processamento previamente descritas. Na sequência, uma análise foi feita a partir dos testes realizados, comparando-se os resultados de cada abordagem.

O objetivo dos experimentos foi avaliar por meio dos resultados obtidos, se a abordagem é aplicável no contexto da realidade do usuário de dispositivos móveis. Nessa linha, todas as imagens utilizadas nos experimentos realizados são frontais, porém são capturadas em ambientes comuns dos usuários, onde os fatores externos não são controlados, como orientação, distância e condições de iluminação variáveis, em diferentes locais e períodos do dia.

As abordagens do trabalho foram comparadas entre si, e também entre um aplicativo que realiza o reconhecimento facial, disponível no mercado, pago e bem conceituado pelos usuários Android.

Os resultados são quantificados de acordo com o erro total médio *E* definido pela soma de *FP* (Falso Positivo), que é a porcentagem das faces reconhecidas indevidamente, e *FN* (Falso Negativo) as faces que não foram reconhecidas erroneamente, sendo finalmente dividido por dois. Quanto menor o erro médio, maior é a confiabilidade da resposta alcançada.

Foram elaborados quatro conjuntos de testes para cada abordagem no experimento, ou seja, cada junção da metodologia de reconhecimento facial e técnica de pré-

processamento, sendo que em cada conjunto de teste eram capturadas trinta imagens submetidas ao reconhecimento, para serem avaliadas as respectivas taxas de falso positivo e falso negativo.

A Tabela I e a Tabela II apresentam os resultados das metodologias *Eigenfaces* e LBP respectivamente, em conjunto com cada técnica de pré-processamento utilizadas no trabalho.

TABELA I. EXPERIMENTOS EIGENFACES.

| ***Técnica Pré-processamento*** | ***Erro Total Médio (E) - %*** |
|---|---|
| *Equalização do Canal de Luz (Espaço cor YCbCr)* | *16,65* |
| *Correção da iluminação aplicando um fator γ* | *16,65* |
| *Abordagem de Xiaoyang Tan and Bill Trigg* | *20,0* |

TABELA II. EXPERIMENTOS LBP (LOCAL BINARY PATTERNS).

| ***Técnica Pré-processamento*** | ***Erro Total Médio (E) - %*** |
|---|---|
| *Equalização do Canal de Luz (Espaço cor YCbCr)* | *23,3* |
| *Correção da iluminação aplicando um fator γ* | *13,3* |
| *Abordagem de Xiaoyang Tan and Bill Trigg* | *16,65* |

Na Tabela III são comparados os resultados da melhor abordagem alcançada no trabalho, utilizando o método LBP (*Local Binary Patterns*) junto com a técnica de pré-processamento que corrige a iluminação da imagem aplicando um fator *γ*, e do aplicativo existente no mercado.

TABELA III. COMPARAÇÃO DOS RESULTADOS.

| ***Técnica Pré-processamento*** | ***Erro Total Médio (E) - %*** |
|---|---|
| *LBP + Correção da iluminação aplicando um fator γ* | *13,3* |
| *Aplicativo do Mercado* | *25,0* |

## V. CONCLUSÃO E TRABALHOS FUTUROS

É importante destacar que atualmente não há uma solução de reconhecimento facial que apresente certeza exata da resposta. Os resultados possuem apenas possibilidade de acerto ou erro, e a taxa desse acerto ou erro é o que define se uma abordagem é aplicável ou não. Isso é particularmente grave no cenário dos dispositivos móveis, onde os fatores externos da captura da imagem não são controlados e a qualidade da captura da imagem é imprescindível para o sucesso da abordagem, afetando diretamente o seu resultado. Por isso, nos testes realizados houve um grande empenho em emular os diferentes ambientes que o usuário pode estar.

O método de reconhecimento facial LBP (*Local Binary Patterns*) em conjunção com correção de iluminação aplicando um fator *γ*, além de apresentar melhores resultados em comparação ao *Eigenface*, também utiliza menos quantidade de memória e o tempo de processamento é menor, possibilitando concluir que é a melhor abordagem a ser empregada.

É possível observar na Tabela III que os resultados alcançados foram superiores comparativamente ao aplicativo do mercado que foi usado nos testes.

Na continuidade do trabalho, pretende-se aprimorar a aplicação, explorando novas soluções em busca de maior confiabilidade, usabilidade e medidas *anti-spoofing* para posteriormente publicá-la na loja virtual de aplicativos Android – Google Play.

## REFERÊNCIAS BIBLIOGRÁFICAS


[1] D.A. Forsyth and J. Ponce, “Computer Vision”, 2ed., Prentice Hall, 2011.

[2] R.O. Duda, P.E. Hart and D.G. Stork, “Pattern Classification”, 2ed., John Wiley & Sons, 2001.

[3] W. Zhao, R. Chellappa and P.J. Phillips, “Face recognition: A literature survey”, ACM Computing Surveys, vol. 35, pp. 399-458, December 2003.

[4] R. Jafri and H.R.A. Arabnia, “A survey of face recognition techniques”, Journal of Information Processing Systems, vol. 5, pp. 41-68, June 2009.

[5] M. Turk and A. Pentland, “Eigenfaces for recognition”, Journal of Cognitive Neuroscience, vol. 3, pp. 71-86, 1991.

[6] P.N. Belhumeur, J.P. Hespanha and D. Kriegman, “Eigenfaces vs. Fisherfaces: recognition using class specific linear projection”, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 19, pp. 711-720, July 1997.

[7] F. Chichizola, L.D. Giusti and M. Naiouf, “Face recognition: reduced image Eigenfaces method”, 47th International Symposium, pp. 159-162, June 2005.

[8] V. Hiremath and A. Mayakar, “Face Recognition Using Eigenface Approach”, IDT Workshop on Interesting Results in Computer Science and Engineering, 2009.

[9] T. Ojala and M. Pietikäinen, “A comparative study measures with classification based on featured distributions”, Pattern Recognition Elsevier, vol. 29, pp. 51-59, January 1996.

[10] T. Ahonen, A. Hadid and M. Pietikäinen, “Face recognition with local binary patterns”, European Conference on Computer Vision, vol. 3021, pp. 469-481, 2004.

[11] Y. Rodriguez and S. Marcel, “Authentication Using Adapted Local Binary Pattern Histograms”, 9th European Conference on Computer Vision, pp. 1-12, 2006.

[12] Centro de Estudos, Resposta e Tratamento de Incidentes de Segurança no Brasil, Cartilha de Segurança para internet. 2014. Disponível em < http://cartilha.cert.br/fasciculos/>.

[13] A. Ramos, “Security Oficer - 1: Guia Oficial para Formação de Gestores em Segurança”, 2ed., Editora Zouk, 2008.

[14] J.M.S. Pinheiro, Biometria nos sistemas computacionais você é a senha, Ciência Moderna, 2008.

[15] P. Viola and M.J. Jones, “Robust real-time face detection”, International Journal of Computer Vision, vol. 52, pp. 137-154, May 2004.

[16] OPENCV. Documentação da biblioteca OpenCV. 2014. Disponível em < http://docs.opencv.org/>.

[17] Google Inc. Documentação plataforma Android. 2014. Disponível em < http://developer.android.com/index.html>.

[18] R.C. Gonzalez and R.E.Woods, “Processamento digital de imagens”, 3ed., Pearson Education do Brasil, 2011.

[19] T. Xiaoyang, B. Triggs, “Enhanced Local Texture Feature Sets for Face Recognition Under Difficult Lighting Conditions”, IEEE Transactions on Image Processing, vol. 19, pp. 1635-1650, February 2010.

[20] O. Cinar, “Pro Android C++ with the NDK”, Apress, 2012.

# A MULTI-CHANNEL AND MULTI-SCALE ARCHITECTURE FOR IMAGE SEGMENTATION BASED ON MORPHOLOGICAL GRADIENT

KARIN S. KOMATI[1], EVANDRO O. T. SALLES[2], MARIO SARCINELLI-FILHO[2]
[1]INSTITUTO FEDERAL DO ESPÍRITO SANTO (IFES) CAMPUS SERRA,
RODOVIA ES-010, KM 6.5 – 29.164-231 – SERRA – ES – BRASIL
[2]LABCISNE, UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO (UFES)
AV. FERNANDO FERRARI, 514, GOIABEIRAS, VITÓRIA, ES, BRAZIL, 29075-910
E-MAILS: KKOMATI@IFES.EDU.BR, {EVANDRO, MARIO.SARCINELLI}@ELE.UFES.BR

**Abstract—** This work proposes a multi-channel and multi-scale architecture for grayscale image segmentation. Each channel is specialized in recognizing a region type and finally all channels are combined through a fusion step. By their turn, the different scales identify different granularities of image regions. Our approach is based on an edge detection method and an edge preserving smoothing filter. We choose the morphological gradient as the edge detection method, because tests with the database image "*The Berkeley Segmentation Dataset and Benchmark*"(BSDS) show that the results for morphological gradient are closer to human perception than the traditional methods (Sobel, Lapalace, Prewitt and LoG). Tests of our architecture with the BSDS image database resulted in metrics that are quite close to those obtained for works available in the BSDS site.

**Keywords—** segmentation, morphological gradient, *The Berkeley Segmentation Dataset and Benchmark, edge preserving smoothing filter*

**Resumo—** Este trabalho propõe uma arquitetura multi-canal e multi-escala para segmentação de imagens em escala de cinza. Cada canal da arquitetura é especializado em reconhecer um tipo de região, e, ao final, há a fusão de todos os canais. Por sua vez, as diferentes escalas identificam diferentes granularidades das regiões das imagens. Nossa abordagem é baseada em método de detecção de bordas e filtro de suavização com preservação de bordas. Escolheu-se o gradiente morfológico como método de detecção de bordas, pois testes realizados usando o banco de imagens "*The Berkeley Segmentation Dataset and Benchmark*"(BSDS) mostraram que os resultados do gradiente morfológico são mais próximos da percepção humana, comparados aos resultados dos métodos tradicionais (Sobel, Laplace, Prewitt e LoG). Testes feitos usando nossa arquitetura com o banco de imagens BSDS resultaram em métricas equiparáveis aos trabalhos existentes disponíveis no sítio do BSDS.

**Palavras-chave—** segmentação, gradiente morfológico, *The Berkeley Segmentation Dataset and Benchmark,* filtro de suavização com preservação de bordas.

## 1 Introdução

O reconhecimento automático de objetos em imagens digitais em sistemas computacionais possui inúmeras aplicações, tais como indexação e recuperação automáticas de fotos e vídeos, uso em sistemas de navegação de robôs, monitoramento de poluição, uso em sistemas de segurança e em sistemas de verificação de qualidade de produtos.

Uma etapa prévia ao reconhecimento de objetos é a segmentação de regiões, ou detecção do contorno. O objetivo é prover um modo automático de obter uma decomposição da imagem de entrada em diversas sub-regiões, que guardam entre si, cada uma delas, alguma descrição comum. É um problema que já se estende por mais de três décadas, e ainda assim continua sendo tópico de grande interesse.

Há muitos desafios na tarefa de segmentação de imagens, como relatado em Conci e Nunes (2007), tais como a complexidade de texturas, a grande variação de condições de iluminação e imprecisões das regiões das bordas. Vários trabalhos sobre segmentação de imagens já foram propostos, como relatado em (Côco et. al 2007; Komati et. al 2010), mas ainda não há uma solução que possa ser considerada precisa e completa em todos os tipos de imagens.

Outra questão é a ambiguidade intrínseca na percepção visual. Imagens naturais possuem tantos elementos que muitas vezes, mesmo entre seres humanos, não há consenso na definição de quais são as regiões de uma imagem. Para resolver esta questão, utilizou-se o banco de dados "*The Berkeley Segmentation Dataset and Benchmark*" - BSDS (Martin et. al. 2001). Neste banco de imagens há duas bases, uma de treinamento composta por 200 imagens, e outra de teste com 100 imagens, tanto em cor quanto em escala de cinza, e utilizou-se apenas a versão em escala de cinza neste trabalho. Para cada imagem do banco de dados há o resultado da segmentação de pelo menos 5 anotadores humanos diferentes, compondo, assim, uma referência de comparação.

Mesmo usando um banco de dados com as imagens e as referências humanas, outro aspecto que pode ser discutido é como medir o quanto o resultado de uma segmentação está próximo da referência de segmentação humana. Há artigos que discutem qual seria o melhor método quantitativo, como em Wolf e colegas (2006) e em Zhang e colegas. (2008). Neste trabalho, optou-se por usar o método de Martin e colegas (2004), baseado nas métricas de *recall*, *precision* e *F-measure* (cujos termos serão mantidos em inglês). A escolha se deve ao fato de que são as métri-

cas padrão do BSDS, facilitando assim a comparação com outros algoritmos.

O método apresentado usa os resultados do trabalho de Tu e Zhu (2002), no qual observou-se que em imagens naturais há frequentemente quatro diferentes tipos de regiões. Em seguida, projetou-se uma arquitetura que usa diferentes canais para distinguir estes quatro tipos diferentes de região. Propõe-se uma abordagem que se inspira na estrutura clássica da abordagem baseada em banco de filtros (Randen e Husϕy, 1999), porém ao invés do uso de técnicas de filtragem de frequência, usa-se operadores da morfologia matemática (Soille, 1999).

Em cada canal, foram usados métodos de detecção de bordas e, somou-se as respostas dos vários canais, gerando uma imagem "*soft edge map*". Este tipo de imagem é aquele aceito pelo BSDS como imagem de segmentação. Não se construiu regiões conectadas limitadas pelas bordas, como em geral é feito nos métodos usuais de segmentação por descontinuidade, por exemplo, através de *watershed.*

Neste trabalho, propõe-se um método rápido e simples, com resultados que se equiparam aos resultados de outros algoritmos encontrados no BSDS, e que motivam a continuidade do trabalho.

O restante do artigo é organizado da seguinte forma: a seção 2 apresenta os tipos de regiões e como extrair cada uma delas da imagem, na seção 3 descreve-se a arquitetura proposta e, na seção 4 descreve-se os experimentos, apresentando seus resultados e a análise dos mesmos. Finalmente, fecha-se com a seção 5, onde se enfatiza os resultados e lista-se tarefas futuras.

## 2 A Estratégia Proposta

A abordagem de bancos de filtros (filtros multicanais) é muito utilizada no tratamento de texturas. Há trabalhos clássicos, como os de Bovik (1990, 1991) e Unser & Eden (1990). No trabalho de Randen e Husϕy (1999) é apresentado um ótimo estudo comparativo de diversas abordagens, com enfoque especial para o uso de banco de filtros no tratamento de texturas.

Nos trabalhos citados, descreve-se uma estrutura básica de análise baseada em banco de filtros. Há uma primeira etapa com filtros que são considerados detectores de características espaciais ou detectores espectrais da imagem. Há etapas posteriores, com processamentos não-lineares e de suavização, para ressaltar alguma característica estatística ou para auxiliar na detecção de micro-características presentes na textura.

Cada canal, isto é, cada filtro possui um processamento independente dos demais, até que ao final são integrados, podendo ser de uma forma simples com uma função de máximo, como Bovik (1990, 1991) ou com um classificador.

Inspirados nesta estrutura básica, esta proposta possui diferentes canais em paralelo, e em cada um deles, obtém-se respostas para tipos específicos de regiões, como mostrado na Figura 1. Não há necessidade de se criar canais para as regiões do tipo uniforme e sombreado. Apenas o uso do canal de gradiente é suficiente para identificar as regiões do tipo uniforme e sombreado.

Uma grande diferença entre a nossa proposta e a abordagem clássica por banco de filtros é que o primeiro passo por banco de filtros é sempre feito por um processamento linear e em nossa abordagem, todos os processos são não-lineares.

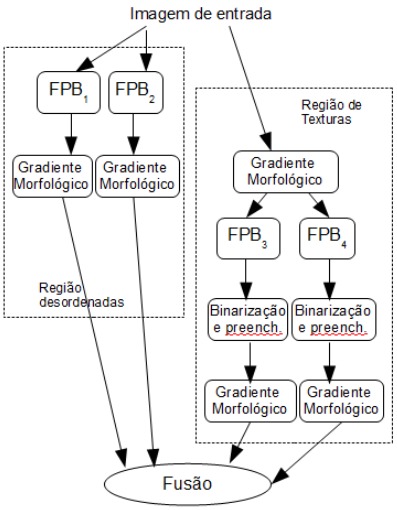



Figura 1. Diagrama de blocos da arquitetura proposta.

Na Figura 1, os retângulos tracejados estão apenas delineando o conjunto de processos que reconhecem a região desordenada e a região de textura. Para cada tipo de região, tem-se dois canais representando as diferentes máscaras do FPB, tratando a característica da granularidade das regiões. O resultado dos canais referentes à região desordenada são imagens em escala de cinza e os canais de textura respondem em imagens em preto e branco.

Cada imagem sempre é processada por todos os canais e depois a resposta de todos os canais é combinada por um processo de fusão. A fusão ao nível de sinal (Fonseca 1999; Freire et. al. 2004) refere-se à combinação dos sinais provenientes de um grupo de sensores com o objetivo de fornecer um sinal que é usualmente do mesmo tipo dos originais, mas com melhor qualidade.

Um exemplo das técnicas de fusão utilizadas ao nível de sinal é a média ponderada dos sinais. Avaliou-se toda a base de dados de treinamento e chegou-se empiricamente aos valores de peso para cada canal {0,3; 0,5; 0,1;0,1}. Estes valores estão na sequência dos canais da Figura 1, observando os canais da esquerda para direita.

## 3 Identificação de Tipos de Regiões

Esta proposta está baseada nos resultados de Tu e Zhu (2002), que após um estudo sobre um grande

conjunto de imagens de cenas naturais, definiu que imagens podem ser divididas em quatro tipos de regiões: uniforme, desordenado, textura e sombreado.

A Figura 2 mostra exemplos destes quatro tipos de região nos retângulos de bordas pretas em cada imagem. Na Figura 2a tem-se um exemplo da região uniforme, que é a neve, que possui uma cor branca com pouquíssima variação Na Figura 2b tem-se uma região desordenada que é a folhagem da árvore. Na Figura 2c tem-se uma textura, padrão, representada pela pele de um tigre e na Figura 2d tem-se um sombreado, que é uma variação gradativa da cor, em geral, nas imagens naturais, causada pela diferença de iluminação.

a) região uniforme b) região desordenada

c) região de textura d) região sombreada

Figura 2. Exemplos dos quatro de tipos de regiões estão marcados pelos retângulos de bordas pretas nas imagens.

### *3.1 Método de Detecção de Bordas*

O processo de detecção de bordas consiste na verificação da variação dos valores de luminosidade de uma imagem visando localizar e realçar os pixels de borda através da ampliação do contraste entre uma região e outra região vizinha.

Em testes preliminares, compara-se os métodos tradicionais de Sobel, Laplace, Prewitt, LoG e gradiente morfológico. Utiliza-se em todos os filtros espaciais, o tamanho 3x3, e o parâmetro alpha do filtro de Laplace foi configurado para 0,2. O elemento estruturante do gradiente morfológico foi de 3x3, em formato de cruz ou 4-conectado com valores 1.

Nestes testes, os detectores de borda foram aplicados diretamente nas imagens do BSDS e as métricas calculadas: *recall*, *precision* e *F-measure.* Como todos os resultados dos 5 métodos são imagens em escala de cinza, o algoritmo do BSDS efetua a binarização da imagem antes do cálculo das métricas. Passou-se como parâmetro o valor 30, o que significa que o BSDS usará 30 diferentes valores de limiares para a binarização. Depois, escolhe-se o limiar de binarização que gerou o melhor valor de *F-measure*.

Os resultados comparativos são apresentados na Tabela 1. Cada célula numérica da tabela representa a quantidade de imagens da base de testes que teve o maior valor da métrica da coluna com relação ao método da linha. Assim, entende-se que o gradiente morfológico apresentou maior valor de *recall* em 34 das 100 imagens e maior valor de *precision* em 63 das 100 imagens. O gradiente foi disparadamente melhor em questão de *F-measure*.

*Tabela 1. Tabela comparativa das métricas entre os métodos de detecção de bordas no BSDS.*

| | *recall* | *precision* | *F-measure* |
|---|---|---|---|
| Sobel | 13 | 11 | 7 |
| Laplace | 22 | 6 | 1 |
| Prewitt | 5 | 16 | 9 |
| LoG | 23 | 4 | 9 |
| Gradiente morfológico | **34** | **63** | **74** |

O gradiente morfológico (Soille 1999) é definido pela diferença aritmética entre a dilatação e a erosão de uma imagem por um elemento estruturante. Como a dilatação e a erosão representam, respectivamente, a filtragem máxima e mínima, cada ponto do gradiente morfológico é a diferença entre os valores máximo e mínimo do ponto central do elemento estruturante. Como toda operação morfológica, o gradiente depende da forma e tamanho do elemento estruturante.

Na morfologia matemática, imagens são funções mapeando um espaço Euclidiano (ou grade) *E* no conjunto dos reais. A dilatação de uma imagem *f(x)* por um elemento estruturante (ou função estruturante) *b(x)* é definida por:

$$(f \oplus b)(x) = \sup_{y \in E}[f(y) + b(x - y)] \quad (1)$$

onde, *sup* é a função de máximo.

A erosão de *f(x)* por *b(x)* é definida por:

$$(f \ominus b)(x) = \inf_{y \in E}[f(y) - b(y - x)] \quad (2)$$

onde *inf* é a função de mínimo.

O gradiente morfológico *G(f)* da imagem *f(x)* por um elemento estruturante *b(x)* é dado por:

$$G(f) = f \oplus b - f \ominus b \quad (3)$$

O melhor resultado do gradiente morfológico é devido às suas características de remoção de pequenos artefatos escuros e claros e de depender menos das direções das bordas quando se usa um elemento estruturante simétrico (Gonzalez & Woods 2000).

### *3.2 Regiões Uniformes e Sombreadas*

Para detectar uma região uniforme ou sombreada, optou-se por um elemento estruturante simples de tamanho 3x3 em formato de cruz, e aplicou-se o operador de gradiente morfológico em todas as imagens em escala de cinza do BSDS. Os resultados são mostrados na Figura 3, onde a coluna da direita se refere à imagem que possui regiões sombreadas e a coluna esquerda, à imagem que apresenta uma grande região do tipo uniforme.

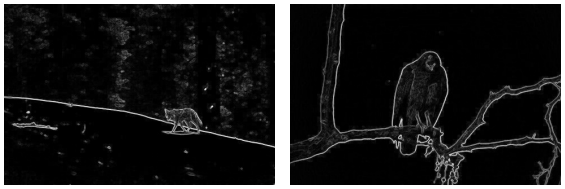
Na primeira linha, a Figura 3a, mostra a imagem original do BSDS de nome 167062 e a Figura 3b a imagem 42049. O termo *rank* significa a posição da imagem de acordo com o maior valor de *F-measure* apresentado por um algoritmo que tem seus dados publicados no sítio do BSDS. Logo, o *rank* oferece a informação do quão difícil é a segmentação da imagem: quanto maior o valor do *rank*, mais complexo é o seu processamento.

A segunda linha, Figuras 3c e 3d, mostram as imagens resultantes do gradiente morfológico, onde verifica-se que tanto a região uniforme, quanto a região sombreada não apresentam *pixels* de borda. É muito comum esta imagem apresentar um contorno circular por toda a imagem separando os cantos escuros, como mostrado em Komati et. al (2009).

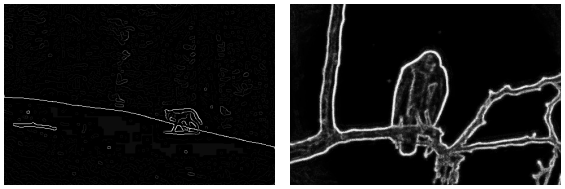
Os melhores resultados dentre os algoritmos disponíveis no sítio do BSDS são mostrados na terceira linha, Figuras 3e e 3f, com os respectivos valores do *F-measure* entre parênteses.

A última linha, Figuras 3g e 3h, mostra as imagens de referências feita pelos anotadores humanos, bem como o valor de *F-measure* associado. Os valores de *F-measure* humanos não são sempre o máximo (um), pois há diferenças entre os vários anotadores e estas diferenças diminuem o *F-measure*.

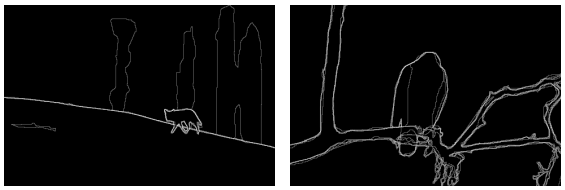
a) imagem 167062 (*rank* 27) b) imagem 42049 (*rank* 1)



c) resultado do gradiente morfológico (0,77) d) resultado do gradiente morfológico (0,92)



e) melhor resultado do BSDS com o algoritmo GM (0,78) f) melhor resultado do BSDS com o algoritmo BEL (0,92)



g) referência humana (0,67) h) referência humana (0,92)

Figura 3. Resultados do gradiente morfológico em imagens que contém região uniforme e região sombreada.

É importante observar que os valores alcançados de *F-measure* apenas com o gradiente morfológico são bem altos. No caso da imagem 42049, são iguais tanto aos dos anotadores humanos quanto ao melhor resultado de algoritmo. Para a imagem 167062, o valor fica um pouco abaixo do melhor algoritmo, 0,78 contra 0,77, e bem maior que o resultado humano.

### *3.3 Regiões Desordenadas*

Regiões desordenadas incluem exemplos de texturas, tais como a copa de uma árvore, grama, areia ou pelo de animais. Neste tipo de região, o gradiente morfológico, assim como outro tipos de gradientes, são muito sensíveis a qualquer alteração. É muito comum o uso de filtros de suavização para reduzir o "ruído" existente em regiões desordenadas.

Contudo, este procedimento pode comprometer a precisão na localização das bordas. Assim, existe uma dualidade entre a precisão na localização das bordas e a sensibilidade aos detalhes irrelevantes da imagem. Para que não se tenha este tipo de problema, potou-se por um filtro com preservação de bordas (FPB). Dentre os vários filtros FPB, usaremos o filtro de Kuwahara et. al. (1976), por ser um filtro simples e rápido, e porque sua implementação segue os conceitos de morfologia matemática.

A Figura 4 apresenta os resultados da aplicação do FPB e do gradiente morfológico numa imagem do BSDS. Na Figura 4a apresenta-se a imagem, de nome 69015, de *rank* 49. Tem-se uma região desordenada extensa representada pelo pêlo do coala. A Figura 4b é apenas a aplicação do gradiente direto na imagem de entrada, tal como foi feito nas imagens das Figuras 2c e 2d. Com esta imagem, é possível comparar os resultados da aplicação com e sem o FPB. Sem o FPB, o melhor valor de *F-measure* é de 0,59.

Seguindo os dois passos: aplicação do filtro de Kuwahara com máscara de tamanho 5x5 e depois o cômputo do operador de gradiente, te-se como resultado a Figura 4c. A Figura 4d é a binarização da imagem 4c para o melhor valor de *F-measure*, que foi de 0,75. É importante salientar a melhoria significativa do valor de *F-measure* com o FPB.

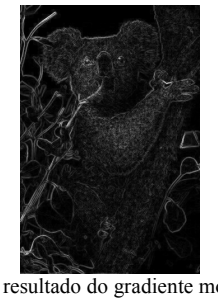
A Figura 4e mostra o melhor resultado dentre todos os algoritmos disponíveis no site do BSDS, o "*Global Probability of Boundary* (GPB)" (Maire et. al. 2008), que apresenta *F-measure* de 0,82. A Figura 4e mostra a segmentação humana de *F-measure* de 0,71. Embora nosso resultado não seja melhor que o GPB, ele quantitativamente é melhor que o resultado humano.

O grande resultado da Figura 4d foi que com a aplicação deste filtro, conseguiu-se identificar que os pêlos do coala formam uma região e não a identificação de borda para cada pêlo como na Figura 4b. Porém, outra característica importante numa região desordenada é a granularidade da região. A pele de um coala possui uma granularidade fina se comparada com as folhas de uma árvore, como a da Figura 2b. Assim, testes com diferentes tamanhos de máscaras

no filtro de Kuwahara foram feitos, e não se alterou o elemento estruturante do gradiente. Evita-se o aumento do tamanho do elemento estruturante no gradiente morfológico, pois ele pode ocasionar dois problemas (Soille 1999):

- a perda da localização das bordas, visto que gera-se bordas mais espessas e;
- a fusão de bordas onde as mesmas apresentam-se próximas geometricamente.

a) imagem 69015 (rank 11).



b) resultado do gradiente morfológico da imagem 69015 (0,59).

c) resultado do gradiente morfológico após aplicação de FPB de máscara de tamanho 5x5 na imagem de entrada.

d) resultado da binarização na imagem 3c no melhor *F-measure* (0,75)

e) melhor resultado do BSDS com o algoritmo GPB (0,82).

f) referência humana (0,71).

Figura 4. Resultados do gradiente morfológico e FPB na imagem 69015 com grande região desordenada.

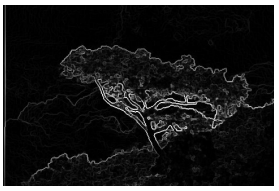
Nestes testes preliminares, conseguiu-se resultados interessantes para uma máscara de tamanho maior, no caso, 9x9. Na Figura 5 apresenta-se os resultados para a imagem 147091 de *rank* 39. Neste conjunto de imagens, mostra-se os resultados sem o filtro de Kuwahara (Figura 5b), com o filtro de Kuwahara para máscara de tamanho 5x5 (Figura 5c) e 9x9 (Figura 5e), facilitando a comparação de resultados.

Assim, constata-se que o valor de *F-measure* foi crescendo, inicialmente sem filtro tem-se o valor de 0,6, depois com o filtro de máscara menor, tem-se o valor de 0,72 e com o filtro de máscara maior o *F-measure* cresce para 0,75. Não só cresce, mas é importante notar como a copa da árvore é identificada como uma região quase única, mostrando que esse método é capaz de detectar regiões desordenadas.

O conceito de multi-escala está relacionado com a dimensão da máscara do FPB utilizado durante o processo.

a) imagem 147091 (rank 32).

b) resultado do gradiente morfológico (0,6).



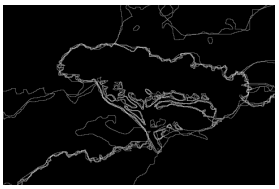
c) resultado do gradiente morfológico após aplicação do filtro de Kuwahara com máscara 5x5 na imagem 147091.

d) resultado da binarização na imagem 4c no melhor *F-measure* (0,72).

e) resultado do gradiente morfológico após aplicação do filtro de Kuwahara com máscara 9x9 na imagem 147091.

f) resultado da binarização na imagem 4e no melhor *F-measure* (0,75).

g) melhor resultado do BSDS com o algoritmo GPB (0,77).



h) referência humana (0,72).

Figura 5. Resultados do gradiente morfológico e FPB na imagem 147091 com grande região desordenada.

### *3.4 Regiões de Texturas*

A região de texturas é região mais complexa de se delimitar. Regiões como o pêlo de um tigre possuem variações abruptas de cor, de amarelo para preto, e embora tenha uma repetição dessas cores, elas não se apresentam na mesma escala, nem na mesma orientação. Isto pode ser facilmente conferido quando avalia-se o *rank* das imagens. As figuras a seguir, a 6 e a 7, estão posicionadas no final do *rank*, posições 94 e 90 respectivamente.

Em nossos testes preliminares, observou-se que a aplicação do FPB na imagem de gradiente agrupou

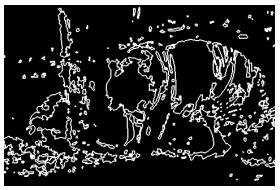
melhor as regiões de texturas, inversamente ao que foi feito para identificar regiões desordenadas.

O conjunto de imagens da Figura 6 usa o FPB com máscara de tamanho 5 e as imagens da Figura 7 usam máscara de tamanho 9. A Figura 6b mostra o resultado da aplicação do operador de gradiente direto na imagem 108005, gerando um *F-measure* de 0,4. Aplicando o filtro de Kuwahara com máscara de tamanho 5x5 na imagem de gradiente tem-se a imagem 5c que mantém *F-measure* de 0,4. Veja que as listras do tigre que antes estavam bem fortes na Figura 6b agora estão embaçadas na Figura 6c.

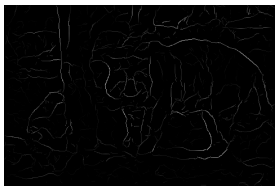
a) imagem 108005 (rank 94).

b) resultado do gradiente morfológico (0,4).

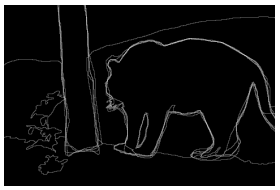
c) resultado do FPB de máscara de tamanho 5 no gradiente morfológico (0,4).



d) resultado da binarização, preenchimento de regiões e gradiente morfológico na figura 6c (0,47).



e) melhor resultado do BSDS com o algoritmo TG (0,54).



f) referência humana (0,78).

Figura 6. Resultados do processo de identificação de região de textura na imagem 108005.

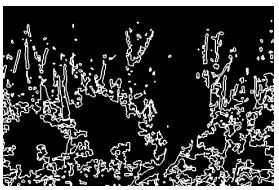
Sobre imagem, fez-se um processamento de binarização, usando o método de Otsu (1979) e em seguida, um algoritmo baseado em reconstrução morfológica que preenche regiões e cavidades da imagem (Soille 1999). A imagem resultante até este passo, é formada por regiões e não bordas, portanto processa-se de novo o gradiente morfológico. Diferentemente do processamento de identificação das regiões anteriores que finalizavam com uma imagem final em escala de cinza, a imagem deste processo é uma imagem em preto e branco, como o mostrado na Figura 6d, de *F-measure* 0,47.

Embora a Figura 6d não apresente uma segmentação perfeita e não tão concisa quanto a da referência humana, Figura 6f (*F-measure* 0,78), houve distinção do tigre com o fundo, e pode-se discernir cabeça e corpo do animal. O resultado final também não está longe do melhor resultado do BSDS que é o algoritmo "*Texture Gradient* (TG)" (Malik et. al 2001), de *F-measure* 0,54.

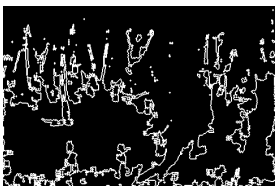
Da mesma forma que para regiões desordenadas, a granularidade em uma textura também é um aspecto importante. A Figura 7a mostra um canguru no meio de uma vegetação de galhos, isto é, uma região de granularidade bem grossa. O valor *F-measure* foi crescendo, sendo 0,35 apenas para a imagem de gradiente, Figura 7b, melhorando para 0,38 no processamento com máscara menor (Figura 7c e alcançando o valor de 0,43 para a máscara maior (Figura 7d). A segmentação da região de vegetação ficou bem menos ruidosa.

a) imagem 69040 (rank 90).

b) resultado do gradiente morfológico (0,35).



c) resultado do processo de identificação de regiões de textura com FPB de máscara de tamanho 5 (0,38).



d) r resultado do processo de identificação de regiões de textura com FPB de máscara de tamanho 9 (0,43).

e) melhor resultado do BSDS com o algoritmo *xren* (0,57).

f) referência humana (0,5).

Figura 7. Resultados do processo de identificação de região de textura na imagem 69040.

Embora estes valores de *F-measure* também foi aumentando a medida que se aplicavam as técnicas de processamento de imagens. Embora os valores de *F-measure* sejam baixos, o próprio resultado humano (Figura 7f) também apresentou *F-measure* baixo, de apenas 0,5. O melhor resultado de algoritmo foi o *xren* (Ren 2008) com *F-measure* de 0,57.

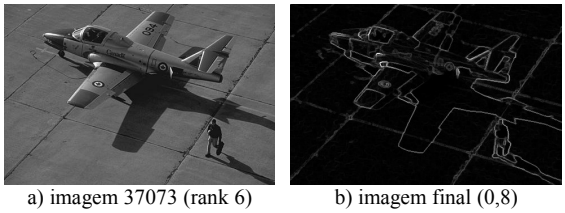
## 4 Experimentos, Resultados e Discussão

Este método é totalmente não supervisionado e não há nenhum parâmetro de entrada.

Levando-se em conta todas as 100 imagens da base de teste BSDS, o valor de *F-measure* final da proposta deste trabalho foi computado em 0,58. Den-

tre os 11 algoritmos que possuem resultados na página do BSDS, este trabalho se posicionaria em 6º lugar, empatado com outras duas técnicas.

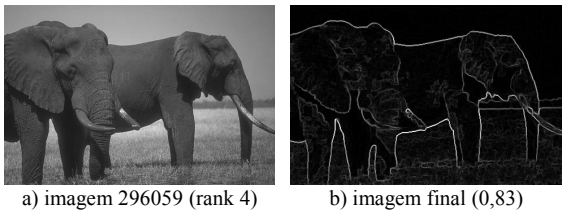
Apresenta-se mais alguns resultados desta proposta. A Figura 8a mostra a imagem 37073 de *rank* 6. Embora seja uma imagem relativamente simples, ainda não havia sido mostrado resultado com objetos de linhas retas e objetos feitos pelo ser humano. O resultado é mostrado na Figura 8b com *F-measure* de 0,8, onde as linhas estão bem definidas e é visualmente bem próximo à referência humana, Figura 8d (*F-measure* de 0,75). O melhor resultado do BSDS é mostrado na Figura 8c, alcançado pelo GPB, com *F-measure* de 0,84.



a) imagem 37073 (rank 6) b) imagem final (0,8)

c) melhor resultado do BSDS com o algoritmo GPB (0,84) d) referência humana (0,75).

Figura 8. Resultados para a imagem 37073.



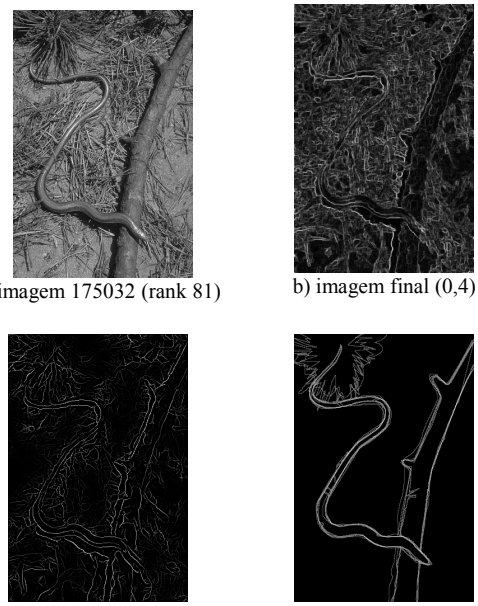
a) imagem 296059 (rank 4) b) imagem final (0,83)

c) melhor resultado do BSDS com o algoritmo GB (0,88) d) referência humana (0,95).

Figura 9. Resultados para a imagem 37073

Um outro resultado interessante é o mostrado na Figura 9, onde a Figura 9a, de nome 296059 está no *rank* 4. O resultado deste trabalho (Figura 9b) e o resultado do melhor algoritmo (Figura 9c) possuem bons valores de *F-measure*, porém, ainda diferente do resultado humano (Figura 9d). Isto se deve ao fato de que um ser humano consegue distinguir que há dois elefantes, dado o conhecimento prévio de que há olhos, presas, enfim que há duas cabeças distintas. A borda existente no meio dos elefantes só é feita tendo-se informação extra, pois baseado apenas na textura ou cor não é possível se fazer esta distinção.

Um exemplo que, infelizmente, não obteve bom resultado é mostrado na Figura 10. A Figura 10a é a imagem 175032, de *rank* 81. O nosso resultado não só teve um *F-measure* baixo, mas também não consegue distinguir a cobra do chão. Há uma grande complexidade em se chegar à segmentação humana (Figura 10d com *F-measure* de 0,85, porém, o algoritmo *xren* mostrou um resultado bem mais compatível (Figura 10c), com *F-measure* de 0,63.



a) imagem 175032 (rank 81) b) imagem final (0,4)

c) melhor resultado do BSDS com o algoritmo xren (0,63) d) referência humana (0,85).

Figura 10. Resultados para a imagem 175032.

## 5 Conclusão e Trabalhos Futuros

Neste trabalho, propôs-se uma arquitetura multicanal, onde cada canal responde melhor a um tipo de região. É também uma abordagem multi-escala, onde para cada região usou-se mais de um tamanho de máscara, fazendo uma análise de diferentes granularidades.

Com uma abordagem simples, verificou-se que os resultados de testes no BSDS ficam equiparados aos resultados dos algoritmos que se encontram publicados no sítio do BSDS.

A abordagem pode ser melhorada, uma extensão natural é projetar uma versão que manipule cores. Outra, é que a região do tipo textura precisa incorporar melhorias, pois os resultados mostrados nas Figura 5d, 6d ou 10d podem ser mais próximos da percepção humana.

A arquitetura é bem genérica, logo, cada bloco do diagrama da Figura 1 pode ser alterado sem alterar o restante do sistema, com isso, pode-se independentemente buscar o melhor filtro de suavização com preservação de bordas ou um gradiente melhor ou o uso de mais canais.

**Agradecimentos**

Os autores gostariam de agradecer à CAPES pelo suporte financeiro e ao PPGEE da UFES pelo suporte à pesquisa.

**Referências Bibliográficas**

Bovik, A. C.; Clark, M.; Geiler, W. (1990). Multichannel Texture Analysis Using Localized Spatial Filters, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 12, No. 1, pp. 55-73.

Bovik, A. C. (1991). Analysis of Multichannel Narrow-Band Filters for Image Texture Segmentation**,** *IEEE Trans. on Signal Processing*, Vol. 39, No. 9, pp. 2025-2043.

Côco, K. F.; Salles, E. O. T. and Sarcinelli-Filho, M. (2009). Topographic independent component analysis based on fractal and morphology applied to texture segmentation. *Lecture Notes in Computer Science*, Vol. 5441, pp.491–498.

Dollar, P.; Tu, Z. and Belongie, S. (2006). Supervised Learning of Edges and Object Boundaries, *IEEE Computer Vision and Pattern Recognition (CVPR)*, Vol. 2, pp.1964-1971.

Fonseca, J. F. C. (1999). *Aplicação de Técnicas de Fusão/Integração Sensorial de Dados no levantamento do Relevo de Objectos*. Doutor. Portugal. Universidade do Minho.

Freire, E. O; Bastos-Filho, T. F.; Sarcinelli-Filho, M. and Carelli, R. (2004). A New Mobile Robot Control Approach via Fusion of the Control Signals. *IEEE Transactions on Systems, Man and Cybernetics*, *Part B: Cybernetics*, Vol. 34, No. 1, pp. 419-429.

Gonzalez, R. C. and Woods R. E. (2000). Processamento de Imagens Digitais. Editora Edgard Blucher.

Komati, K. S.; Salles, E. O. T.; Sarcinelli-Filho, M. (2009). Fractal-JSEG: JSEG Using a Homogeneity Measurement based on Local Fractal Descriptor In: *XXIInd Brazilian Symposium on Computer Graphics and Image Processing (Sibgrapi 2009)*, Rio de Janeiro, Brazil 11-14 October 2009. Los Alamitos, CA: IEEE Computer Society, pp. 253-260.

Komati, K. S.; Salles, E. O. T.; Sarcinelli-Filho, M. (2010). Unsupervised Color Image Segmentation based on Local Fractal Descriptor and J-Images. In: *IEEE ICIT2010 International Conference on Industrial Tehnology 2010*, Viña del Mar, Chile. Piscataway, NJ, USA: IEEE.

Kuwahara, M.; Hachimura, K.; Ehiu,S. and Kinoshita, M. (1976). Processing of ri-angiocardiographic images. In: *Digital Processing of Biomedica Images*. NewYork: Plenum, pp.187–203.

Maire, M.; Arbelaez, P.; Fowlkes, C. and Malik, J. (2008). Using contours to detect and localize junctions in natural images. In: *IEEE Computer Vision and Pattern Recognition, 2008. (CVPR 2008*). Anchorage, Alaska. Los Alamitos, CA: IEEE Computer Society. pp. 1-8.

Malik, J.; Belongie, S.; Leung, T. K. and Shi, J. (2001) Contour and Texture Analysis for Image Segmentation. International Journal of Computer Vision, Vol. 43, No. 1, pp. 7-27.

Martin, D.; Fowlkes, C.; Tal,D. and Malik, J. (2001) A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: *Proc. 8th IEEE Int'l Conf .Computer Vision*, Vol. 2, July 2001, pp.416–423.

Martin, D.; Fowlkes, C. and Malik, J. (2004) Learning to detect natural image boundaries using local brightness, color, and texture cues. IEEE Trans. Pattern Anal. Mach. Intell., Vol.26, No. 5, pp. 530–549.

Nunes, E. O. and Conci, A. (2007). Segmentação por Textura e Localização do Contorno de Regiões em Imagens Multibandas. *Revista IEEE América Latina*, Vol. 5, pp. 185-192.

Otsu, N. (1979). A Threshold Selection Method from Gray-Level Histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 9, No. 1, pp. 62-66.

Randen, T.; Husϕy, J. H. (1999). Filtering for Texture Classification: A Comparative Study, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 21, No. 4, pp. 291-310.

Ren, X.F (2008). Multi-scale Improves Boundary Detection in Natural Images. In: *10th European Conference on Computer Vision (ECCV08)*. Marseille-France. Vol. 3. October 2008. pp. 533-545.

Soille, P. (1999) Morphological Image Analysis - Principles and Applications. Berlin Heidelberg: Springer-Verlag.

Tu, Z. and Zhu, S. C. (2002). Image Segmentation by Data-Driven Markov Chain Monte Carlo, *IEEE TPAMI*, Vol. 24, No. 5, pp. 657-673.

Unser, M. and Eden, M. (1990) Nonlinear Operators for Improving Texture Segmentation Based on Features Extracted by Spatial Filtering, *IEEE Trans. on Systems, Man, and Cybernetics*, Vol. 20, No. 4, pp. 804-815.

Wolf, C. and Jolion, J-M. (2006). Object count/Area Graphs for the Evaluation of Object Detection and Segmentation Algorithms. *International Journal on Document Analysis and Recognition*, Vol. 8, No. 4, pp. 280-296.

Zhang, H.; Fritts, J. E. and Goldman, S. A. (2008) Image segmentation evaluation: A survey of unsupervised methods. *Computer Vision and Image Understanding*, Vol.110, No 2, pp. 260-280.

# Weightings of Shannon Entropy's Additivity for Image Segmentation

Gallão, Celso Dênis e Rodrigues, Paulo Sérgio
Grupo de Inteligência Artificial e Processamento de Imagens e Sinais
Centro Universitário da FEI
São Bernardo do Campo, São Paulo, Brasil
celso.gallao@fatec.sp.gov.br
psergio@fei.edu.br

***Abstract*—Thresholding based on classical Shannon entropy doesn't find an optimal partition in images whose probability distribution of luminance can be approximated by bimodal Gaussians functions. Here we show that an appropriate balancing of Shannon additivity formalism allows to achieve a quality of segmentation identical to that observed when the optimal threshold is analytically calculated over these Gaussians. Thus, different weights are applied between the regions of interest. The quality of results remains the same even under parameter changing in the probability distribution.**

## I. Introdução

A separação de uma cena em regiões de interesse, conhecida como segmentação, é uma das técnicas mais importantes no contexto do processamento digital de imagens. Entre os métodos mais conhecidos está a limiarização, que consiste em separar a imagem em regiões que compartilham características comuns.

Shannon [1], em 1948, deduziu a Equação (1), unindo conceitos da mecânica estatística formulada por Boltzmann e Gibbs [2], e que mais tarde ficou conhecida como entropia de Boltzmann-Gibbs-Shannon, ou entropia BGS, dada por

$$S = -\sum p_i \log p_i. \tag{1}$$

Em 1981, T. Pun [3] propôs um método para limiarização de imagens baseada na propriedade aditiva dessa entropia [1]. De maneira formal, essa propriedade pode ser descrita da seguinte forma: considera-se a distribuição das probabilidades dos $n$ tons de cinza, dada por $P = \{p_1; p_2; ...; p_n\}$. O método proposto em [3] separa $P$ em duas distribuições normalizadas, sendo uma representando a região de interesse frontal ($P_P = \frac{p_1}{p_P}, \frac{p_2}{p_P}, ..., \frac{p_t}{p_P}$), e outra representando o fundo ($P_Q = \frac{p_{t+1}}{p_Q}, \frac{p_{t+2}}{p_Q}, ..., \frac{p_n}{p_Q}$), onde $p_P = \sum_{i=1}^{t} p_i$, $p_Q = \sum_{i=t+1}^{n} p_i$, e $t$ (*threshold*) é um valor que divide $P$ nas distribuições $P_P$ e $P_Q$. Em seguida, aplica-se a Equação (1). O limiar ótimo, através da propriedade aditiva da entropia BGS, é dado por

$$S(P,Q) = S(P) + S(Q), \tag{2}$$

onde $S(P)$ e $S(Q)$ são as respectivas entropias dadas pela Equação (1). Então, o valor de $t = T_S$, que melhor equilibra a quantidade de informação entre as duas regiões, é aquele que maximiza a Equação (2), ou seja,

$$T_S = max[\ S(P,Q)\ ]. \tag{3}$$

Esta técnica, por ser simples de entender e implementar, tornou-se frequente em diversas aplicações práticas e tem sido muito estudada. É aplicada aos mais variados domínios, com resultados promissores e que reforçam a eficiência da limiarização por entropia para segmentação de imagens.

No trabalho de Kapur et al. [4], maximizou-se o limiar superior da entropia máxima para obter o limiar ótimo. Este trabalho foi aprimorado por Abutaleb [5] através do método com entropias bidimensionais. No trabalho de Li e Lee [6] e Pal [7] usou-se a divergência direta de *Kullback-Leibler* [8] para selecionar o limiar ótimo, e Sahoo et al. [9] visou o mesmo objetivo utilizando a entropia de *Rényi*. Um estudo sobre essas metodologias pode ser visto em [10], que apresenta uma revisão sobre métodos de segmentação de imagens baseados em entropia.

Apesar da entropia BGS ser frequentemente aplicada a diversos problemas físicos há mais de um século, sabe-se, no entanto, que esse antigo formalismo é apenas uma aproximação quando se trata de sistemas físicos não-extensivos que ocorrem em fenômenos físicos observados como em turbulência, expansão de galáxias e interação de redes sociais, por exemplo [11].

A entropia BGS representa bem os sistemas extensivos, mas não se pode garantir que os sistemas não-extensivos também possam ser precisamente representados por ela. Para esses sistemas, em 1988 foi proposto por Constantino Tsallis a chamada *q*-entropia [12], uma generalização da entropia BGS que pôde ser comprovada em diversas áreas teóricas e experimentais. Em 2004, Albuquerque et al. demonstraram a aplicação da *q*-entropia para a limiarização de imagens digitais [13], através da pseudo-aditividade de Tsallis, uma característica da *q*-entropia equivalente à aditividade de BGS.

Pode-se notar a partir das Equações (1)-(3) que a limiarização calculada automaticamente através da entropia BGS é inflexível, uma vez que oferece apenas um único valor de limiar para separar as regiões. Desta forma, a medida da eficiência deste processo é dependente de análise subjetiva. Em aplicações reais, o limiar ótimo é de fato um elemento

subjetivo. No entanto, seu cálculo analítico quando depende de condições específicas, ainda é uma questão inconclusiva.

Sendo assim, propomos aqui uma alternativa à entropia BGS, de modo a contornar essa inflexibilidade. Trata-se do uso de uma ponderação entre os dois termos da aditividade - que em imagens representam a região de interesse e o fundo - através da inclusão de um parâmetro $0 \leq \alpha \leq 1$, de modo que seja possível alterar o equilíbrio para cada termo. Este novo formalismo é apresentado na Seção II.

Na Seção III são descritas as metodologias aplicadas para o cálculo do limiar ótimo, pois este é utilizado como valor para aferição da eficiência das metodologias. Tais metodologias são mostradas na Seção IV.

Para demonstrar a efetividade da ponderação da aditividade da entropia BGS para limiarização de imagens, foram conduzidos uma série de experimentos com distribuições Gaussianas artificiais, de modo a permitir controlar seus parâmetros de amplitude, desvio padrão e média, e realizar estudos do comportamento do método proposto sobre essas distribuições, sendo estes detalhados na Seção V, incluindo exemplos aplicados à limiarização de imagens reais. As discussões e considerações finais são apresentadas na Seção VI.

## II. Entropia de Shannon Ponderada ($S_w$)

Nesta seção introduzimos o formalismo da metodologia proposta para redefinição da entropia BGS aplicada à limiarização de imagens, da seguinte maneira: tomando a distribuição $P_P$ ponderada por um valor $\alpha$ e a distribuição $P_Q$ ponderada por um valor $(1-\alpha)$, onde $0 \leq \alpha \leq 1$ e $n = [1, ..., 254]$, temos

$$S_w(P) = -\sum_{i=1}^{n} \alpha p_i \log \alpha p_i, \tag{4}$$

$$S_w(Q) = -\sum_{j=1}^{255-n} (1-\alpha) q_j \log (1-\alpha) q_j; \tag{5}$$

sendo $p_i$ a ocorrência da distribuição $P_P$ na iteração $i$, e $q_j$ a ocorrência da distribuição $P_Q$ na iteração $j$. Assim, redefinimos a propriedade da aditividade de Shannon na forma de uma pseudo-aditividade de Shannon Ponderada, como sendo:

$$S_w(P,Q) = 2\ [\ \alpha S_w(P)\ +\ (1-\alpha) S_w(Q)\ ] \tag{6}$$

Na Equação (6), quando $\alpha = 0.5$, esta reduz-se à propriedade aditiva dada pela Equação (2). Convencionamos, então, chamar a propriedade aditiva apresentada na forma da Equação (6) de pseudo-aditividade-ponderada.

Analogamente ao trabalho de T. Pun [3], o limiar escolhido para a segmentação da imagem será o de maior valor na pseudo-aditividade-ponderada, dado por:

$$T_{SW} = max[\ S_w(P,Q)\ ] \tag{7}$$

Consideramos os dois subsistemas com pesos iguais ($\alpha = 0.5$) no cálculo da entropia de Shannon Ponderada sobre a distribuição $P$, através da Equação (4). Assim temos:

$$-\sum_{i=1}^{n} p_i \log p_i = -\sum_{i=1}^{n} 2\alpha p_i \log 2\alpha p_i$$

Desenvolvendo a equação da direita podemos concluir que, para a distribuição $P$, a entropia de Shannon Ponderada $S_w(P)$ é igual a entropia BGS quando $\alpha = 0.5$, ou seja,

$$-\sum_{i=1}^{n} p_i \log p_i = -2\sum_{i=1}^{n} (\alpha p_i \log \alpha p_i) - 2\alpha, \tag{8}$$

que se pode representar por $S(P) = 2S_w(P) - 2\alpha$, e resulta na variação da entropia de Shannon Ponderada da distribuição $P$, dada por:

$$S(P) = 2[\ S_w(P) - \alpha\ ] \tag{9}$$

Seguindo esta mesma linha de raciocínio, ao considerarmos o valor de $\alpha = 0.5$ no cálculo da entropia de Shannon Ponderada da distribuição $Q$, através da Equação (5), temos:

$$-\sum_{j=1}^{255-n} q_j \log q_j = -\sum_{j=1}^{255-n} 2(1-\alpha) q_j \log 2(1-\alpha) q_j$$

Desenvolvendo a equação da direita podemos concluir que, para a distribuição $Q$, considerando a ponderação $(1-\alpha)$, com $\alpha = 0.5$, a entropia de Shannon Ponderada $S_w(Q)$ é igual a entropia BGS da distribuição $Q$, ou seja,

$$-\sum_{j=1}^{255-n} q_j \log q_j = -2\sum_{j=1}^{255-n} [\ (1-\alpha) q_j \log (1-\alpha) q_j\ ] - 2(1-\alpha), \tag{10}$$

podendo ser representada por $S(Q) = 2S_w(Q) - 2(1-\alpha)$, e resulta na variação da entropia de Shannon Ponderada da distribuição $Q$, dada por:

$$S(Q) = 2[\ S_w(Q) - (1-\alpha)\ ] \tag{11}$$

Assim, a Equação (6), da pseudo-aditividade-ponderada sugere, analiticamente, que se pode considerar os dois subsistemas com importâncias distintas, ainda que dependentes um do outro. Isso abre a possibilidade de extrair informações com baixa probabilidade *a priori*, através de variações no balanceamento entre os respectivos subsistemas. Estas variações não se observam quando da segmentação de imagens baseada na entropia BGS, uma vez que esta considera seus subsistemas com ponderações iguais, sem margem para variações.

## III. Cálculo do Limiar Ótimo ($T_{opt}$)

Para estabelecer uma comparação entre as limiarizações baseadas na entropia BGS e na entropia de Shannon Ponderada, procuramos encontrar um valor de limiar ótimo ($T_{opt}$) como sendo o limiar de referência que otimamente separa a região de interesse de uma imagem. Assim, neste trabalho, representamos o sistema a ser analisado por duas funções Gaussianas, de maneira a obter-se uma única função Gaussiana bimodal, estando cada Gaussiana representando uma região distinta, simulando um histograma de luminância normalizado.

Formalmente, definimos as funções Gaussianas geradas como $G_1$ e $G_2$, discretizadas na faixa $i = [0, 1, .., 255]$. Definimos também $c_1$ e $c_2$ como sendo as respectivas amplitudes de $G_1$ e $G_2$ com valores iniciais $c_1 = c_2 = 1$, e também

$\sigma_1$ e $\sigma_2$ como sendo seus respectivos desvios padrões, com valores iniciais $\sigma_1 = \sigma_2 = 30$. Por fim, definimos as médias de $G_1$ e $G_2$ dadas por $\mu_1 = \frac{k}{3}\ 255$ e $\mu_2 = \frac{3-k}{3}\ 255$, onde $k = [0, .., 1.5]$, sendo a média normalizada entre $G_1$ e $G_2$ dada por $G = \frac{G_1(i)+G_2(i)}{\sum (G_1+G_2)}$.

Desta forma, consideramos que a Gaussiana $G$ representa o sistema a ser analisado, com variações destes 6 parâmetros de controle. O limiar ótimo analítico ($T_{opt}$) é identificado na interseção entre os subsistemas $G_1$ e $G_2$, um mínimo global. A representação gráfica de $G$ pode ser vista na Figura 1.

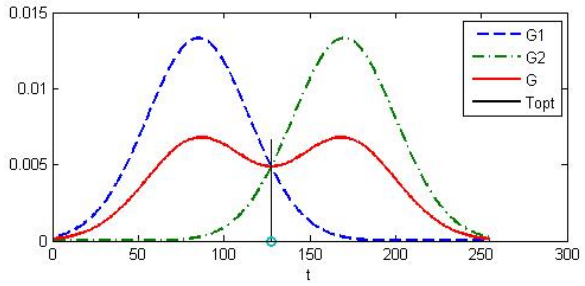



Figura 1. Funções Gaussianas $G_1$ e $G_2$ com seus parâmetros iniciais: amplitude $c1 = c2 = 1$; desvio padrão $\sigma_1 = \sigma_2 = 30$; médias $\mu_1 = \frac{1}{3}255$ e $\mu_2 = \frac{2}{3}255$. Função Gaussiana $G$, como sendo a média normalizada entre $G_1$ e $G_2$. O limiar ótimo $T_{opt}$ na interseção entre $G_1$ e $G_2$, por ser um mínimo global da função $G$, é o limiar ótimo buscado, também encontrado analiticamente.

Considerando o caso em que ambas Gaussianas, $G_1$ e $G_2$, possuem a mesma amplitude ($c_1 = c_2 = c$), mesmo desvio padrão ($\sigma_1 = \sigma_2 = \sigma$) e variando apenas as médias ($\mu_1$ e $\mu_2$), temos a igualdade $G_1 = G_2$ dada por

$$c\frac{\exp^{-\frac{(i-\mu_1)^2}{2\sigma^2}}}{\sqrt{2\pi}\sigma} = c\frac{\exp^{-\frac{(i-\mu_2)^2}{2\sigma^2}}}{\sqrt{2\pi}\sigma}, \quad (12)$$

de onde se obtém o limiar analítico (ótimo) dado por:

$$i = T_{opt}^{medias} = \frac{(\mu_1 + \mu_2)}{2} \quad (13)$$

No caso em que as amplitudes são distintas, mas com mesmos desvios-padrões ($\sigma_1 = \sigma_2 = \sigma$), temos a seguinte igualdade $G_1 = G_2$:

$$c_1\frac{\exp^{-\frac{(i-\mu_1)^2}{2\sigma^2}}}{\sqrt{2\pi}\sigma} = c_2\frac{\exp^{-\frac{(i-\mu_2)^2}{2\sigma^2}}}{\sqrt{2\pi}\sigma}, \quad (14)$$

que resulta na seguinte expressão para o limiar analítico ótimo:

$$i = T_{opt}^{amplitude} = \frac{ln(\frac{c_1}{c_2})2\sigma^2 \ - \ \mu_1^2 \ + \ \mu_2^2}{2(\mu_2 \ - \ \mu_1)} \quad (15)$$

Finalmente, para representar sistemas cujas distribuições de probabilidades são semelhantes a duas funções Gaussianas com desvios padrões distintos, e mantendo as amplitudes iguais ($c_1 = c_2 = c$), temos a igualdade $G_1 = G_2$ dada por:

$$c\frac{\exp^{-\frac{(i-\mu_1)^2}{2\sigma_1^2}}}{\sqrt{2\pi}\sigma_1} = c\frac{\exp^{-\frac{(i-\mu_2)^2}{2\sigma_2^2}}}{\sqrt{2\pi}\sigma_2} \quad (16)$$

Neste caso, há duas soluções para a equação, identificadas por $i_1$ e $i_2$. Calculadas as raízes, a escolha do melhor limiar entre elas é feita conforme o *Algorithm 1* a seguir, onde $T_1$ e $T_2$ são os valores máximos encontrados, respectivamente, nas funções Gaussianas $G_1$ e $G_2$.

**Algorithm 1** Algoritmo para escolha do limiar $T_{opt}^{desvpad}$, entre duas Gaussianas com desvios padrões distintos.

Input: $i_1$ e $i_2$, $\sigma_1$, $\sigma_2$, $T_1$, $T_2$.
Output: $T_{opt}^{desvpad} \in \{i_1, i_2\}$.
**if** $\sigma_1 < \sigma_2$ **then**
  **if** $i_1 \geq T_1$ **then**
    $T_{opt}^{desvpad} \leftarrow i_1$
  **else**
    $T_{opt}^{desvpad} \leftarrow i_2$
  **end if**
**else**
  **if** $i_1 \leq T_2$ **then**
    $T_{opt}^{desvpad} \leftarrow i_1$
  **else**
    $T_{opt}^{desvpad} \leftarrow i_2$
  **end if**
**end if**
**return** $T_{opt}^{desvpad}$

## IV. Metodologia proposta

De acordo com a seção anterior, experimentos com dados sintéticos em funções Gaussianas permitem controlar os parâmetros $\sigma_1$, $\sigma_2$, $\mu_1$, $\mu_2$, $c_1$ e $c_2$, que regem as distribuições de probabilidades, tornando possível estudar os seus efeitos sobre os algoritmos baseados em limiarização. Neste trabalho, os experimentos foram realizados como segue:

Para cada valor discreto $t$, onde $1 \leq t \leq 255$, calculam-se:

- O limiar ótimo analítico, $T_{opt}$, para cada variação nos parâmetros das distribuições, conforme a Seção III.
- A propriedade aditiva da entropia BGS, $S(P, Q)$, para cada variação nos parâmetros das distribuições, de acordo com a Equação (3), para encontrar o limiar de BGS ($T_S$).
- A pseudo-aditividade da entropia de Shannon Ponderada, $S_w(P, Q)$, para cada variação nos parâmetros das distribuições, inicialmente com o parâmetro de ponderação $\alpha = 0.5$, para encontrar o limiar de Shannon Ponderada ($T_{SW}$), de acordo com a Equação (7). Em seguida, aplicam-se variações em $\alpha$ buscando aproximar $T_{SW}$ ao $T_{opt}$ por meio de busca iterativa, limitada aos valores de $\alpha = [0, ..., 1]$, sendo que o melhor valor de $\alpha$ a ser escolhido é aquele que mais aproxima $T_{SW}$ ao $T_{opt}$ e que seja o mais próximo de $\alpha = 0.5$.

Comparam-se, então, os limiares $T_S$ e $T_{SW}$ com o $T_{opt}$, para aferir a eficiência do processo, onde se consideram como melhores resultados aqueles em que os limiares analisados, $T_S$ e $T_{SW}$, mais se aproximam do limiar ótimo analítico ($T_{opt}$), ponto de mínimo global de $G$.

Para calcular a quantidade de testes necessários a cada iteração, a fim de encontrar o valor de $\alpha$ que mantém

$T_{SW} = T_{opt}$, definimos o espaço de busca como sendo $SS_\alpha = (\alpha_{max} - \alpha_{min})10^\lambda$, onde $\alpha_{max}$ é o maior valor possível para $\alpha$; $\alpha_{min}$ é o respectivo menor, e $\lambda$ é a quantidade de casas decimais (discretização). Sendo $\alpha = [0, ..., 1]$, sabe-se que a quantidade de testes efetuados para encontrar o melhor $\alpha$ a cada iteração é dada por $SS_\alpha = 10^\lambda$.

## V. Experimentos Sintéticos

Para a realização dos experimentos, utilizamos a metodologia descrita na Seção IV, utilizando-se de funções Gaussianas com variações de amplitude, desvio padrão e média. Os resultados são aqui apresentados.

### A. Variação da Amplitude

Foram realizados 60 experimentos variando apenas a amplitude $c_2$ de $G_2$, onde foram calculados o valor de $T_{opt}$ para cada variação através da Equação (15). Para o cálculo da pseudo-aditividade-ponderada foram utilizados valores de $\alpha$ com 3 casas decimais, gerando $SS_\alpha = 10^3$; ou seja, 1000 iterações para cada variação de amplitude. O resultado de um destes experimentos é mostrado na Figura 2. Nota-se que $T_S$ afasta-se da interseção entre $G_1$ e $G_2$ seguindo para o lado do subsistema com a maior amplitude, enquanto $T_{SW}$ mantém-se na interseção, graças a um ajuste no parâmetro $\alpha = 0.521$.

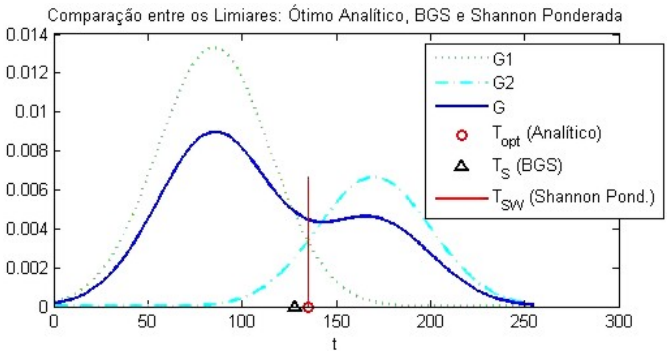



Figura 2. Exemplo dos limiares ótimo analítico, de BGS e de Shannon Ponderada, calculados sobre $G$, variando apenas a amplitude de $G_2$, $c_2 = 0.5$, mantendo os demais parâmetros com seus valores iniciais. Os resultados foram: $T_{opt} = 134.8392$ discretizado para $T_{opt} = 135$, $T_S = 128$, e $T_{SW} = 135$ com $\alpha = 0.521$.

Os gráficos da Figura 3 mostram resultados de 20 experimentos realizados com variações da amplitude de $G_2$ em $c_2 = [0.1, 0.2, ..., 2]$. Nota-se que em nenhum dos casos tivemos $T_S = T_{opt}$, mas há uma tendência de aproximação entre $T_S$ e $T_{opt}$ para valores de $c_2 = 1$. No entanto, constata-se que sempre há um valor de $\alpha$ que mantém $T_{SW} = T_{opt}$ em todos os testes.

Nos outros 40 experimentos realizados, constatou-se que em todos os casos houve um valor de $\alpha$ que manteve $T_{SW} = T_{opt}$, e em nenhum dos casos tivemos $T_S = T_{opt}$. Os resultados mostram a flexibilidade da entropia de Shannon Ponderada em relação à entropia BGS.

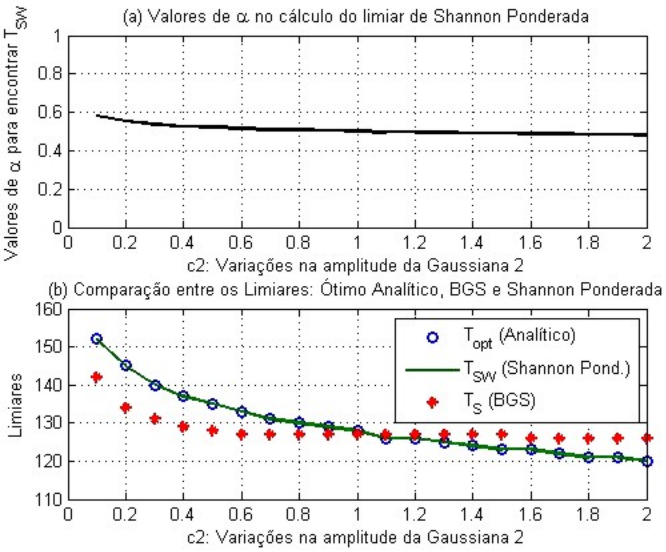



Figura 3. Testes com variações na amplitude de $G_2$: (a) Melhores valores de $\alpha$ encontrados em cada variação de amplitude, para manter $T_{SW} = T_{opt}$. (b) Limiares de $T_{opt}$, $T_{SW}$ e $T_S$, encontrados em cada variação de amplitude.

### B. Variação do Desvio Padrão

Foram realizados 60 experimentos variando apenas o desvio padrão $\sigma_2$ de $G_2$, onde o valor do $T_{opt}$ foi calculado para cada variação de acordo com *Algorithm 1*. Para o cálculo da pseudo-aditividade-ponderada, também utilizaram-se valores de $\alpha$ com 3 casas decimais, o que também representa 1000 iterações para cada variação do desvio padrão. O resultado de um destes testes é mostrado na Figura 4.

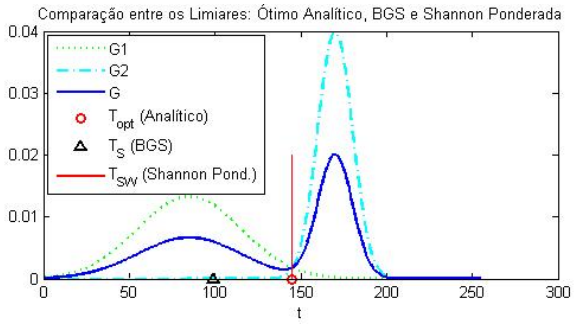



Figura 4. Exemplo dos limiares ótimos analíticos, de BGS e de Shannon Ponderada, calculados sobre $G$, variando apenas o desvio-padrão de $G_2$, $\sigma_2 = 10$. Os resultados foram: $T_{opt} = 145,0834$ discretizado para $T_{opt} = 145$, $T_S = 99$, e $T_{SW} = 145$ com $\alpha = 0.552$.

Nota-se que $T_S$ afasta-se da interseção entre $G_1$ e $G_2$, seguindo para o lado do subsistema com maior desvio padrão, enquanto $T_{SW}$ mantém-se na interseção graças a um ajuste no parâmetro $\alpha = 0.552$.

Os gráficos da Figura 5 mostram resultados de 20 experimentos realizados com variações no desvio padrão de $G_2$, com $\sigma_2 = [10, 20, ..., 200]$. Nota-se que em nenhum dos casos

tivemos $T_S = T_{opt}$, mas há uma tendência de aproximação entre $T_S$ e $T_{opt}$ para valores de $\sigma_2$ próximos a 30 e também para valores de $160 \leq \sigma_2 \leq 180$. No entanto, sempre houve um valor de $\alpha$ que manteve $T_{SW} = T_{opt}$ em todos os casos. Este comportamento pode ser observado no gráfico da Figura 5(b), pela sobreposição das curvas para $T_{opt}$ e $T_{SW}$.

Nos outros 40 experimentos realizados com variações apenas no desvio padrão, constatou-se que em todos os casos houve um valor de $\alpha$ que manteve $T_{SW} = T_{opt}$, e em nenhum dos casos tivemos $T_S = T_{opt}$. Os resultados reforçam a flexibilização da entropia de Shannon Ponderada em relação à entropia BGS.

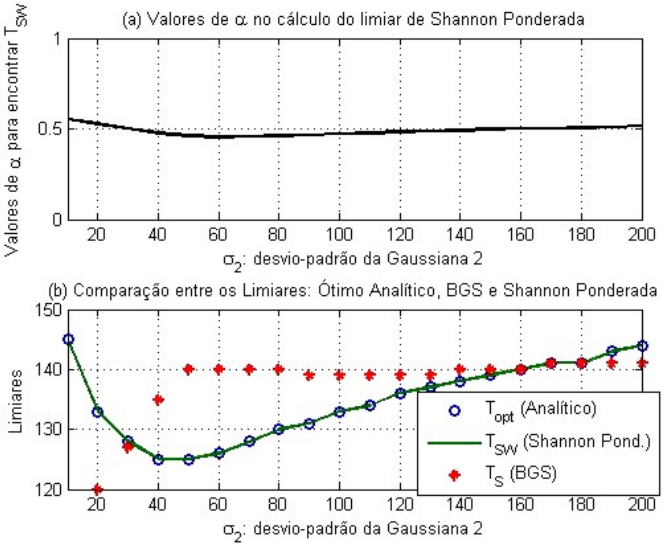



Figura 5. Testes com variações no desvio padrão de $G_2$: (a) Melhores valores de $\alpha$ encontrados em cada variação do desvio padrão, para manter $T_{SW} = T_{opt}$. (b) Limiares de $T_{opt}$, $T_{SW}$ e $T_S$, encontrados em cada variação do desvio padrão.

### C. Variação das Médias

Neste caso, foram realizados 31 experimentos com alterações no valor de $k = [0, 0.05, ..., 1.5]$ que é utilizado para o cálculo entre as médias $\mu_1$ e $\mu_2$ de $G_1$ e $G_2$. Para o cálculo do limiar ótimo analítico ($T_{opt}$) em cada variação, foi aplicada a Equação (13). Para o cálculo da pseudo-aditividade-ponderada, foram utilizados os valores de $\alpha$ com 6 casas decimais, no intervalo de $\alpha = [0.49, ..., 0.51]$, gerando $SS_\alpha = 0.02^6$, sendo 20.000 iterações para cada variação de $k$. O resultado de um destes experimentos é mostrado na Figura 6. Nota-se que $T_S$ afasta-se da interseção entre $G_1$ e $G_2$, enquanto $T_{SW}$ mantém-se na interseção graças a um ajuste no parâmetro $\alpha = 0.49807$.

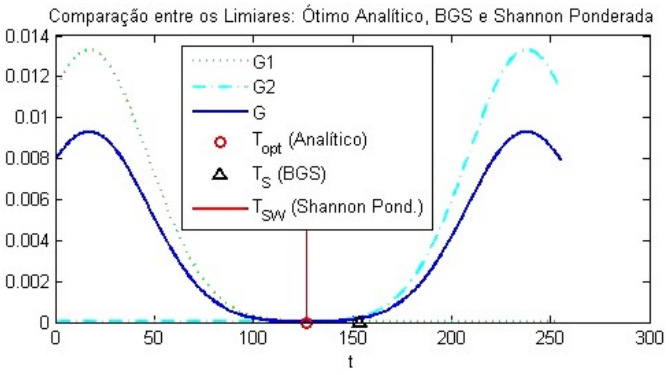



Figura 6. Exemplo dos limiares ótimo analítico, de BGS e de Shannon Ponderada, calculados sobre $G$, variando apenas o valor de $k = 0.2$, resultando nas médias $\mu_1 = \frac{0.2}{3}255$ e $\mu_2 = \frac{2.8}{3}255$. Os demais parâmetros foram mantidos com seus valores iniciais. Os resultados foram: $T_{opt} = 127$, $T_S = 154$, e $T_{SW} = 127$ com $\alpha = 0.49807$.

Os gráficos da Figura 7 mostram resultados dos 31 experimentos realizados com variação apenas do valor de $k = [0, 0.05, ..., 1.5]$. Nota-se que o valor de $T_S$ converge para o valor de $T_{opt}$, conforme o valor de $k$ aumenta. Observa-se também que, para valores de $k > 0.5$, há uma tendência de aproximação entre $T_S$ e $T_{opt}$. Para valores onde $k < 0.5$, apresentando um grande afastamento entre as Gaussianas, a tendência é de $T_S >> T_{opt}$.

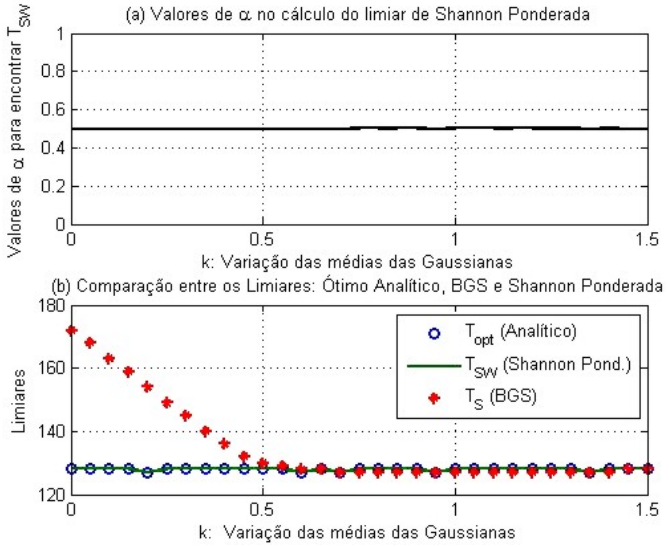



Figura 7. Testes com variações nas médias $\mu_1$ e $\mu_2$ através do valor de $k$, mantendo os demais parâmetros com seus valores iniciais: (a) Melhores valores de $\alpha$ encontrados em cada variação de $k$ para manter $T_{SW} = T_{opt}$ no cálculo da entropia de Shannon Ponderada. (b) Limiares de $T_{opt}$, $T_{SW}$ e $T_S$, encontrados em cada variação de $k$.

Pode-se notar que, para todos os valores de $k = [0, ..., 1.5]$, os ajustes no parâmetro $\alpha$ foram eficazes, pois mantiveram $T_{SW}$ = $T_{opt}$ em todos os experimentos.

Finalmente, a Figura 8 mostra exemplos de limiarização pela entropia de Shannon Ponderada em imagens reais, com

variações do parâmetro $\alpha = [0.1, 0.2, ..., 0.9]$. Nota-se que, para $\alpha = 0.5$, o resultado é o mesmo da entropia BGS.

Para valores de $\alpha < 0.5$ observa-se um peso maior à região de interesse da imagem (representada pela cor branca), e para $\alpha > 0.5$ observa-se um peso maior ao fundo da imagem (representada pela cor preta).

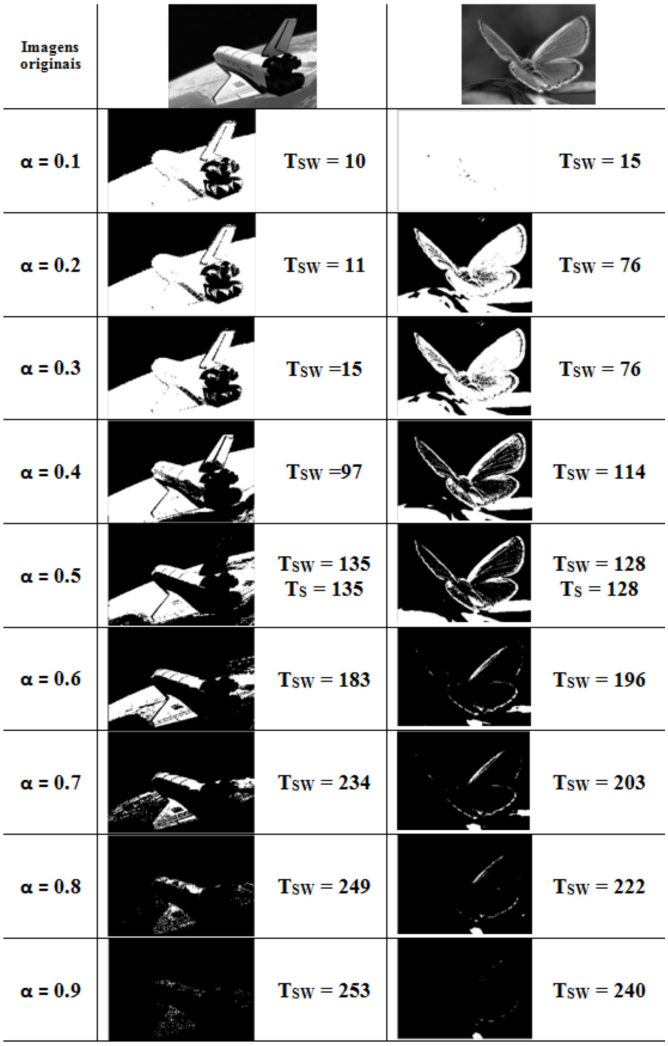



Figura 8. Exemplos de limiarização pela entropia de Shannon Ponderada com diversos valores de $\alpha$, a partir de imagens originais em 256 tons de cinza.

## VI. Conclusões

Os experimentos sintéticos realizados neste trabalho, com variações nas amplitudes, desvios padrões e médias entre funções Gaussianas, mostraram que, no cálculo da entropia de Shannon Ponderada, sempre há um valor de $\alpha$ que mantém o limiar $T_{SW}$ igual ao limiar ótimo $T_{opt}$, em todos os casos estudados. Isto sugere uma correlação entre o valor de $\alpha$ e um valor dependente dos parâmetros da distribuição.

Observou-se que, ao aplicar as variações nos parâmetros das funções Gaussianas, a propriedade aditiva da entropia BGS gera um limiar ($T_S$) que tende a afastar-se do limiar ótimo ($T_{opt}$) na maioria dos casos. Com a inflexibilidade da entropia BGS não há como promover ajustes para manter $T_S = T_{opt}$.

Os resultados mostram a eficiência da metodologia da entropia de Shannon Ponderada, através do ajuste do parâmetro $\alpha$ que pondera os subsistemas, abrindo assim a possibilidade de analisarmos informações que podem estar representadas por pequenas probabilidades, por exemplo. Isto significa, na aplicação da entropia de Shannon Ponderada aplicada na segmentação de imagens, poder atribuir pesos diferentes ao fundo em relação à região de interesse das imagens, sendo mais um recurso para análise de imagens a partir de segmentação.

Observando as variações nos valores de $\alpha$ nos testes efetuados, este manteve um comportamento estável entre os valores de $\alpha = [0.4, ..., 0.6]$. Este comportamento pode ser observado nas Figuras 3(a), 5(a) e 7(a).

Apesar de os experimentos realizados para encontrar o valor de $\alpha$ tenham sido efetuados de forma iterativa, a busca deste parâmetro na aplicação da entropia de Shannon Ponderada para segmentação de imagens não se caracteriza como um método iterativo, pois há uma tendência à estabilidade neste parâmetro, abrindo a perspectiva para novas investigações a fim de associá-lo a características físicas da imagem, e assim determiná-lo automaticamente ou por conhecimento *a priori*. Tornar-se-á, assim, mais um importante componente para análise e extração de informações no processo de segmentação.

## Referências


[1] C. Shannon, "A mathematical theory of communication," *The Bell System Technical Journal*, 1948.

[2] E. P. Borges, "Irreversibilidade, desordem e incerteza: Três visões da generalização do conceito de entropia," *Revista Brasileira de Ensino de Física*, vol. 21, no. 4, p. 453, Dezembro 1999.

[3] T. Pun, "Entropic thresholding: A new approach," *Computer Graphics and Image Processing*, vol. 16, no. 3, pp. 210–239, July 1981.

[4] J. N. Kapur, P. K. Sahoo, and A. K. C. Wong, "A new method for gray-level picture thresholding using the entropy of the histogram," *Computer Graphics Image Process*, vol. 29, pp. 273–285, 1985.

[5] A. S. Abutaleb, "A new method for gray-level picture thresholding using the entropy of the histogram," *Comput. Graphics Image Process*, vol. 47, pp. 22–32, 1989.

[6] C. H. Li and C. K. Lee, "Minimum cross entropy thresholding," *Pattern Recognition*, vol. 26, pp. 617–625, 1993.

[7] N. R. Pal, "On minimum cross entropy thresholding," *Pattern Recognition*, vol. 26, pp. 575–580, 1996.

[8] S. Kullback and R. Leibler, "On information and sufficiency," *The Annals of Mathematical Statistics 22(1), The George Washington University*, pp. 79–86, 1951.

[9] P. Sahoo, S. Soltani, A. Wong, and Y. Chen, "A survay of thresholding techniques," *Computer Vision Graphics Image Processing*, vol. 41, no. 1, pp. 233–260, 1988.

[10] C.-I. Chang, Y. Du, J. Wang, S.-M. Guo, and P. Thouin, "Survey and comparative analysis of entropy and relative entropy thresholding techniques," *The Institution of Engineering and Technology*, 2006.

[11] C. Tsallis, "Nonextensive statistical mechanics and its applications," *Lecture Notes in Physics*, vol. 560, 2001.

[12] C. TsalliS, "Possible generalization of boltzmann-gibbs statistics," *Journal of Statistical Physics*, vol. 52, pp. 479–487, 1988.

[13] M. P. Albuquerque, M. P. Albuquerque, I. Esquef, and A. Mello, "Image thresholding using tsallis entropy," *Pattern Recognition Letters, Elsevier B.V.*, 2004.

# A new approach to automate the seed vigor analysis on soybean seedling using digital images

Daniel Caio de Lima, Evandro L. L. Rodrigues
Escola de Engenharia de São Carlos – USP
São Carlos, São Paulo, Brasil
daniel.caio@usp.br, evandro@sc.usp.br

Lúcio André de Castro Jorge
Embrapa Instrumentação Agropecuária
São Carlos, São Paulo, Brasil
lucio.jorge@embrapa.br

***Abstract*— Soybean has an important role to brazilian economy, so it is necessary to estabilish some accurate methods that can help to determinate the quality of this seeds. Seed vigor is a value that describes the seedlot quality by analising the seedlings' growth. This paper shows advances in a previous work, introducing a different methodology to automate the seed vigor assessment, introducing a new segmentation method, a new way to removal of cotyledons and a new equation to calculate the seedlot uniformity. This experiment showed a better result than the previous one, but is still necessary to do some adjusts to the cotyledon removal method.**

***Keywords*— seedlings; image processing; segmentation; thinning; pruning**

## I. INTRODUÇÃO

Atualmente a cultura de soja ocupa cerca de 27,7 milhões de hectares de área plantada, apresentando rendimento de 2.934 kg por hectare [1], tendo grande importância econômica para o Brasil.

Devido à importância deste grão, devem ser tomados cuidados para que a qualidade dos grãos não seja afetada. Segundo [2], além de condições climáticas durante a maturação e a colheita, as principais causas do decréscimo da qualidade das sementes são o manejo inadequado dos grãos durante a colheita, o armazenamento prolongado em condições irregulares, a aplicação de substâncias químicas e os ataques de microrganismos e outras pragas. O teste de germinação de sementes é o método mais comum para avaliar a qualidade de sementes, tendo seu procedimento definido nas Regras para Análise de Sementes [3]. Após a realização dos testes de germinação, são apresentadas plântulas normais e plântulas anormais, sendo que as últimas, na maioria das vezes, são conseqüência das irregularidades nos grãos. Plântulas anormais apresentam deformidades ou ausência de estruturas vitais, como raiz primária, hipocótilo e plúmula, sendo as estruturas das plântulas de soja mostradas na Figura 1.

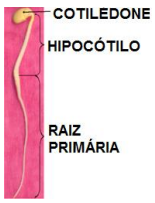



Figura 1 – Estruturas da plântula de soja.

Como conseqüência da má qualidade dos grãos, evidencia-se a diminuição do percentual de germinação e o baixo vigor do lote de sementes. A análise do vigor leva em consideração aspectos da relação entre a semente e o ambiente no qual será semeada, fornecendo assim informações complementares ao teste de germinação, diferenciando lotes de alto e baixo vigor, dentro de uma porcentagem adequada para a comercialização. A Associação de Analistas Oficiais de Semente (AOSA) define o vigor de sementes como propriedades que determinam o potencial para uma emergência rápida e uniforme e para o desenvolvimento de plântulas normais em um conjunto amplo de condições ambientais [4].

Métodos computacionais foram desenvolvidos para automatizar a avaliação do vigor em diferentes espécies de sementes, sendo os principais trabalhos descritos por [5], [6], [7] e [8]. Em um trabalho anterior, [9] propõe uma metodologia para analisar o vigor em plântulas de soja com três dias de germinação, utilizando técnicas convencionais de processamento de imagens e equações presentes na literatura. Neste trabalho serão apresentadas alterações na metodologia descrita por [9], alterações estas que compreendem o processo de segmentação da imagem, de extração de características e do cálculo do vigor.

Na seção II serão introduzidas de forma simplificada as técnicas e a teoria por trás de cada uma delas. Na seção III são mostrados os procedimentos utilizados para a construção do novo método para avaliação do vigor nas plântulas de soja. Na seção IV são mostrados os resultados do novo método em comparação ao método descrito em [9]. Por fim, na seção V será apresentada uma conclusão e uma discussão baseada nos resultados obtidos neste experimento.

## II. FUNDAMENTAÇÃO TEÓRICA

Nesta seção serão descritos, de maneira breve, algumas técnicas utilizadas para a elaboração dos procedimentos que serão descritos na seção 3.

### *A. Coeficiente de Uniformidade de Christiansen*

O coeficiente de uniformidade de Christiansen [10] é uma equação utilizada para calcular a uniformidade da distribuição de água em sistemas de irrigação, porém, uma adaptação desta seria uma boa alternativa à equação de uniformidade descrita por [7]. A equação sem modificações é mostrada a seguir.

$$CUC = 1 - \frac{\sum_{i=1}^{n} |x_i - \bar{x}|}{n \times \bar{x}} \quad (1)$$

Na equação temos $CUC$ como sendo o coeficiente de uniformidade de Christiansen, $x_i$ como sendo o comprimento da plântula a ser analisada, $\bar{x}$ o comprimento médio das plântulas do lote analisado e $n$ é a quantidade de plântulas presentes no lote.

### B. Afinamento e Pruning

O processo de afinamento consiste em afinar uma imagem binária passando por ela sucessivamente um conjunto de elementos estruturantes, sendo este processo repetido até que não ocorram mais alterações na estrutura do objeto binário. Uma das formas de realizar o afinamento por morfologia matemática é utilizando a transformação *Hit-Miss*, operação utilizada para buscar padrões desejados de *pixels* da imagem. O conjunto de elementos estruturantes é representado pela Figura 2.

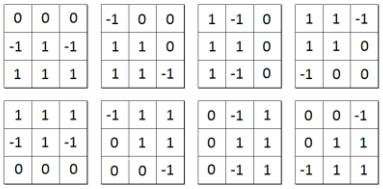


Figura 2 – Conjunto de elementos estruturantes para afinamento.

O processo de afinamento pela transformada *Hit-Miss* resulta no surgimento de muitas pontas, chamadas de forma genérica como "pés de galinha", resultantes do processo sobre as extremidades do objeto binário [11]. Uma forma variante do processo de afinamento é chamada de *pruning*, cujo objetivo é tirar os pontos extremos, as ramificações resultantes do afinamento. Para a realização do *pruning*, é possível utilizar um dos dois conjuntos de elementos estruturantes, representados pelas figuras 3 e 4.

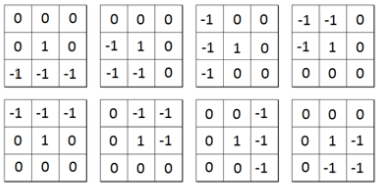


Figura 3 – Conjunto 1 de elementos estruturantes para *pruning*.

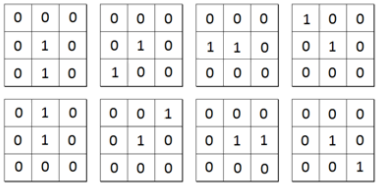


Figura 4 – Conjunto 2 de elementos estruturantes para *pruning*.

## III. METODOLOGIA

### A. Germinação e aquisição das imagens

Assim como no trabalho anterior, as sementes foram germinadas e submetidas ao processo de germinação, conforme prescrito pelas Regras de Análise de Sementes [3], no laboratório de análise de sementes do Departamento de Produção Vegetal da Escola Superior de Agricultura "Luiz de Queiroz" (USP-ESALQ).

Após o período de 3 dias de germinação, as imagens do lote de sementes foram capturadas por um *scanner* adaptado para esta finalidade.

### B. Segmentação

No método proposto em [9], foram testados métodos de segmentação baseados na aplicação da técnica de binarização nos canais G do espaço de cores RGB e H do espaço de cores HSV, além de descrever procedimentos de testes que utilizaram o modelo de contornos ativos (*Snakes*). Neste trabalho, foi testada uma nova abordagem para a segmentação das imagens, baseada em operações matemáticas entre imagens em escala de cinza extraídas dos canais dos espaços de cores RGB e HSV.

De posse da imagem capturada pelo scanner no espaço de cores RGB, assumiu-se cada canal como uma matriz em nível de cinza. A imagem capturada também foi convertida para o espaço de cor HSV e assim também foram assumidas matrizes. Depois de extraídas as matrizes dos canais dos dois espaços de cores, foram realizadas subtrações entre as mesmas, originando 30 imagens analisadas. O processo de subtração entre as matrizes é representado na Figura 5, na qual os círculos representam as matrizes dos espaços de cores e as setas representam a operação de subtração.

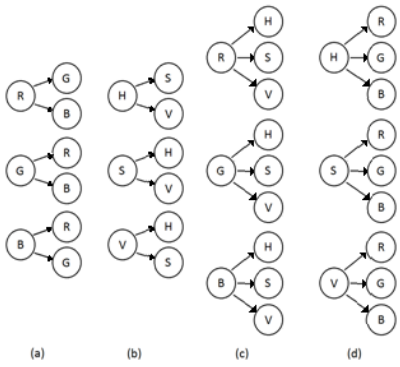



Figura 5 – Subtração entre canais RGB (a), subtração entre canais HSV (b), subtração dos canais RGB dos canais HSV (c) e subtração dos canais HSV dos canais RGB (d).

A partir da análise das imagens obtidas, foi possível observar que na imagem resultante da subtração entre a matriz do canal S pela matriz do canal G houve a remoção por completo de informação desnecessária para a análise do vigor de sementes, ou seja, o *background* da imagem foi removido, restando apenas às plântulas, conforme pode ser observado na Figura 6.

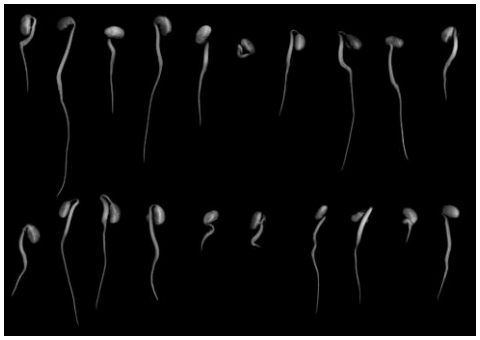


Figura 6 – Imagem resultante da subtração entre a matriz do canal S (HSV) e a matriz do canal G (RGB).

A partir da observação da Figura 5, é possível notar que alguns detalhes das plântulas ficaram ocultos, sendo necessário realizar novas operações matemáticas para realçar algumas características da imagem. Ao final dos testes, chegou-se na configuração representada por (2), que resulta em uma imagem com os detalhes realçados (Figura 7).

$$Img_{Segmentada} = (G - S) + (G - H) \quad (2)$$

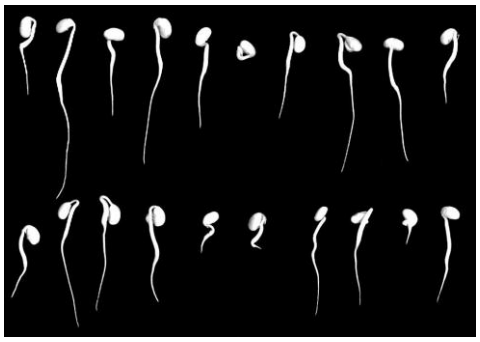


Figura 7 – Imagem segmentada.

Após o processo de segmentação, a imagem é submetida ao processo de *threshold*, transformando a imagens de níveis de cinza em uma imagem binária, necessária ao procedimento de extração de características.

*C. Extração de Características*

Após o processo de segmentação, cada objeto binário presente na imagem foi separado, de modo a extrair individualmente as características de cada plântula, realizando o mesmo processo descrito por [9].

O procedimento padrão para se avaliar o vigor de sementes é medir o comprimento das partes que compõem a plântula, sendo necessário remover a parte que compreende ao cotilédone. Para possibilitar a realização destas medidas, as imagens binárias das plântulas são submetidas ao processo de afinamento pela transformada *Hit-or-Miss*. Diferente do método utilizado anteriormente, que removia o cotilédone antes de submeter à imagem ao afinamento (utilizando uma estimativa da área que compreende ao cotilédone), a nova abordagem consiste em realizar a técnica de afinamento até que a imagem resultante fique com a característica evidenciada pela Figura 8.

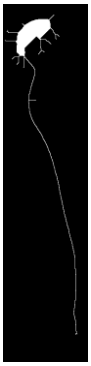


Figura 8 – Plântula após afinamento.

A parte que contém uma maior quantidade de pixels justapostos compreende a porção do cotilédone. Para removê-lo, foi utilizada a técnica de abertura por morfologia matemática, criando uma nova imagem que contém apenas a porção do cotilédone, assim, aplicando-se uma operação lógica XOR entre as duas imagens, obtém-se uma imagem contendo a apenas a raiz da plântula, porém, com algumas ramificações, removidas utilizando a técnica de *pruning* [10]. A Figura 9 ilustra o procedimento.

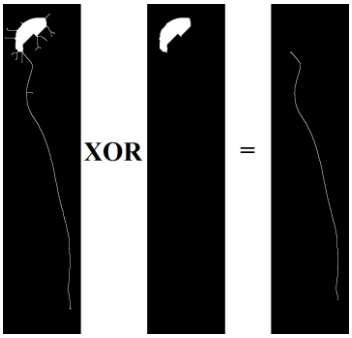



Figura 9 – Remoção do cotilédone.

Para encontrar a transição entre hipocótilo e raiz primária, foi utilizada uma adaptação da transformada de distância euclidiana para gerar um mapa de distâncias. Para visualização, cada distância foi separada em intervalos iguais de distância, e cada um deles foi apresentado graficamente por uma cor distinta. Um exemplo pode ser visto na Figura 10.

Figura 10 – Mapa de distâncias.

Foram comparadas a imagem do mapa de distância e a imagem do esqueleto, fazendo com que os pontos nos quais foram encontradas as partes com maior distância (maior número de cores distintas no eixo horizontal) fossem marcados no esqueleto automaticamente. O último desses pontos foi

considerado como sendo o ponto de transição entre as partes da plântula. O hipocótilo foi pintado de cor azul e a raiz primária foi pintada com a cor vermelha.

*D. Cálculo do índice de vigor*

Após a extração das características, é necessário medir o comprimento das partes para que seja calculado o índice de vigor. Para medir o comprimento das partes, foi desenvolvido um algoritmo semelhante ao descrito por [8], que analisa *pixels* na diagonal.

Para calcular o índice de vigor foram utilizadas algumas das equações descritas por [7], assim como foi utilizado no trabalho prévio em [9], porém, neste trabalho foi proposta uma substituição da equação do cálculo da uniformidade por uma adaptação do Coeficiente de Uniformidade de Christiansen (3), uma vez que a equação de uniformidade descrita por [7] apresenta resultados inferiores aos apresentados pelo sistema para análise de plântulas de soja descrito em [8].

$$CUC = MAX\left(0, \left(1 - \frac{\sum_{i=1}^{n} |x_i - \bar{x}|}{n \times \bar{x}}\right) \times 1000 - Penalização\right) \quad (3)$$

A penalização presente na equação (3) é dada em função do total de plântulas encontradas na imagem e do número de sementes que não germinaram, sendo sua equação mostrada a seguir.

$$Penalização = n_{mort\ as} \times \left(\frac{50}{n_{total}}\right) \quad (4)$$

## IV. RESULTADOS

Foram realizados testes com o método descrito por [9] e com o método descrito pelo presente trabalho. Serão apresentadas comparações por meio de tabelas e por meio de imagens resultantes dos dois procedimentos. Uma comparação visual entre os resultados dos dois métodos pode ser visto na Figura 11.

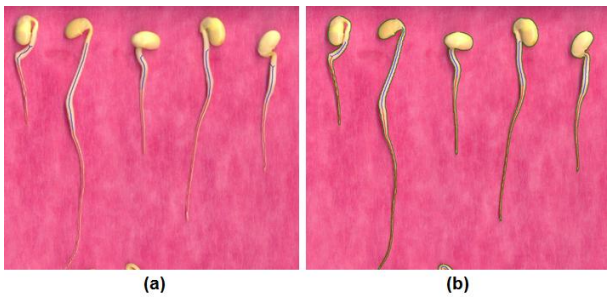



Figura 11 – (a) Resultado do método descrito em [9]. (b) Resultado do método descrito pelo presente trabalho.

Pode-se observar na Figura 10 é possível observar uma melhora em relação ao método de remoção do cotilédone, pois na Figura 10(a), o método removeu parte da raiz em todas as plântulas da imagem e na Figura 10 (b), o novo método de remoção reduziu o número de plântulas que tiveram parte da raiz removida. A Figura 12 mostra um exemplo comparando um exemplo de duas plântulas.

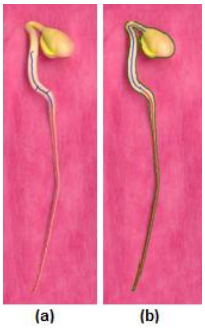



Figura 12 – Exemplo de remoção do cotilédone. (a) Método prévio. (b) Novo método.

Outro ponto a ser destacado foi a ausência de ramificações no esqueleto da imagem na Figura 11 (b), tornando evidente a eficácia do algoritmo de *pruning*. Com relação a separação entre hipocótilo e raiz primária, pode-se dizer que o método prévio [9] possui falhas, marcando o ponto fora da região na qual, segundo analistas de semente, se encontra o ponto de transição, conforme pode ser visto na Figura 13.

Figura 13 – Resultado prévio de separação de (a). Região esperada para localização do ponto de transição (b).

O novo método desenvolvido para encontrar o ponto de transição entre as partes da plântula apresentou resultados melhores que o método antigo, como pode ser observado na Figura 14, porém, em 30% das plântulas analisadas o ponto de transição ainda foi marcado indevidamente.

Figura 14 – Resultado do novo método de separação de estruturas (a). Região esperada para localização do ponto de transição (b).

Ao todo, testou-se um conjunto de 70 imagens de lotes de sementes, contendo 20 plântulas cada, porém, para ilustrar a eficiência das equações componentes do índice de vigor, foi escolhida uma imagem (Figura 15) como exemplo de um lote

de plântulas de soja previamente classificada como tendo alta uniformidade do comprimento e alto nível de crescimento, tendo como conseqüência um índice de vigor elevado.

Figura 15 – Imagem utilizada para testes do cálculo do índice de vigor.

As medidas úteis para a realização do cálculo do índice de vigor foram extraídas pelo método de mensuração descrito em [8]. Os cálculos foram feitos utilizando as equações do trabalho de [7] e os pesos da Tabela 1, sendo os resultados obtidos explicitados pela Tabela 2, assim como o teste da substituição da equação da uniformidade pelo Coeficiente de Uniformidade de Christiansen adaptado.

TABELA 1. Pesos utilizados nas equações

| Cálculo do Vigor | |
|---|---|
| Peso do Crescimento | 50% |
| Peso da Uniformidade | 50% |
| Cálculo do Crescimento | |
| Peso do Hipocótilo | 10% |
| Peso da Raiz primária | 90% |
| Cálculo da Uniformidade ([7]) | |
| Peso Desvio padrão Hipocotilo | 30% |
| Peso Desvio padrão Raiz primária | 30% |
| Peso Desvio padrão Raiz/Hipocotilo | 20% |
| Peso Plântulas Mortas | 20% |

TABELA 2. Comparação entre os resultados obtidos pela metodologia de [9] (Abordagem A) e pelo presente trabalho (Abordagem B)

| | Uniformidade | Crescimento | Índice de Vigor |
|---|---|---|---|
| Abordagem A | 294 | 635 | 464 |
| Abordagem B | 821 | 635 | 728 |

Como pode ser observado, a equação de uniformidade descrita por [7] resultou em um valor de 294 para a característica uniformidade (faixa 0 – 1000) para um lote previamente classificado como de uniformidade alta,

resultando em um índice de vigor baixo para um lote vigoroso. Com a equação da uniformidade de Christiansen foi obtido um valor de uniformidade de 821 (próximo de 1000), retratando a alta uniformidade e por conseqüência, o alto índice de vigor do lote analisado.

## V. Conclusões

A partir da análise dos resultados obtidos pelo novo método em comparação a metodologia do trabalho anterior, é possível concluir que houve uma melhora significativa com relação ao método de remover o cotilédone das plântulas, possibilitado pelo novo método de segmentação e pela aplicação da técnica de afinamento morfológico logo após a individualização das plântulas, mas ainda é necessário um pouco mais de trabalho nesta etapa, pois em muitos casos parte da raiz é excluída após a remoção desta estrutura.

Outra melhora notada foi em relação ao modo utilizado para encontrar a transição entre as estruturas, sendo localizada dentro da área considerada ideal por especialistas. Uma contribuição para o trabalho foi a substituição da equação da uniformidade pela adaptação do coeficiente de uniformidade de Christiansen, refletindo melhor a uniformidade do lote de sementes e, como conseqüência, gerando um índice de vigor mais próximo do esperado.

## VI. Referências


[1] Instituto Brasileiro de Geográfia e Estatística, "Levantamento Sistemático da Produção Agrícola", Outubro de 2013. Disponível em: < http://www.ibge.gov.br/home/estatistica/indicadores/agropecuaria/lspa/lspa_201310.pdf >. Último acesso em: 18 de março de 2014.

[2] C. C. Martins e N. M. Carvalho, "Fontes de Deterioração na Produção de Soja e Respectivas Anormalidades nas Plântulas", Revista Brasileira de Sementes, vol. 16, nº 2, 1994, p. 168-182.

[3] BRASIL, Ministério da Agricultura, Pecuária e Abastecimento, "Regras para Análise de Sementes", Brasília, 2009. 395p.

[4] A. C. S. Oliveira, G. N. Martins, R. F. Silva e H. D. Vieira, "Testes de Vigor em Sementes Baseados no Desempenho das Plântulas", Inter Science Place, ano 2, nº 4, 2009.

[5] A. C. McCormac, P. D. Keefe and S. R. Draper, "Automated Vigour Testing of Field Vegetables Using Image Analysis", SeedSci. &Technol, p.103-112. 1990.

[6] M. S. Howarth and P. C. Stanwood, "Measurement of Seedling Growth Rate by Machine Vision", Transactions of American Society of Agricultural Engineering, v.36, n.3, 1993, p.959-963.

[7] Y. Sako, M. B. McDonald, K. Fujimura, A. F. Evans and M. A. Bennett, "A system for automated vigor assessment", Ohio State University, 2001.

[8] A. L. Hoffmaster, M. B. Fujimura, M. B. McDonald and M. A. Bennett, "An Automated System for Vigor Testing Three-Day-Old Soybean Seedlings", The Ohio State University, 2003.

[9] D. C. Lima, E. L. L. Rodrigues, L. A. C. Jorge, "Application of Image Processing Techniques to Automate the Seed Vigor Assessment Process in Soybean Seedlings", IX Workshop de Visão Computacional, 2013.

[10] T. R. I. Rodrigues, H. S. Batista, J. M. Carvalho, A. O. Gonçalves, E. E. Matsura, "Uniformidade de distribuição de água em pivô central, com a utilização da técnica TDR na superfície e no interior do solo", Revista Brasileira de Engenharia Agrícola e Ambiental, vol. 35, nº 2, 2001. p. 187-191.

[11] J. Facon, "Morfologia Matemática: Teoria e Exemplos", Editora Universitária Champagnat da Pontifícia Universidade Católica do Paraná, 1996.

# The Impact of Compression Algorithms on Touchless-to-Touch Fingerprint Images

Alexandre Zaghetto, Pedro S. Franco e Daniel A. M. Sandoval
Departamento de Ciência da Computação
Universidade de Brasília, Brasília, Distrito Federal 70904–970
E-mail: zaghetto@image.unb.br, {pedro, daniel}@loopec.com.br

***Abstract*—Touchless multiview fingerprinting technology has recently been proposed as an alternative to overcome the intrinsic problems of traditional touch-based systems. However, since touch-based fingerprinting technology is undoubtedly the most used nowadays, images captured by touchless devices must be further processed in order to become similar (and therefore compatible) to those obtained by conventional scanners. Furthermore, considering a scenario where those images must be first stored for later transmission or processing, image compression is required. This paper discusses an algorithm that enables the touchless-to-touch compatibility and evaluates the impact of image compression standards (namely JPEG, JPEG2000, WSQ and H.264/AVC-I) on authentication systems that use the touch-equivalent fingerprints. Our results show that the compatibility algorithm successfully generates touch-equivalent images and that the analysed compression standards, specially JPEG2000, have minor impact on authentication.**

***Keywords*—Biometrics, fingerprint, touchless, compatibility, image compression**

## I. Introdução

A autenticação biométrica, ou simplesmente biometria, pode ser definida como a verificação ou o reconhecimento automático da identidade de um indivíduo a partir de características fisiológicas ou comportamentais [1]. Impressões digitais, geometria da mão, voz, íris, face, caligrafia e dinâmica de digitação são exemplos dessas características. Em geral, diferentes sistemas biométricos requerem tecnologia específica, dependendo da característica fisiológica ou comportamental considerada. No entanto, independentemente das especificidades dos diversos sistemas biométricos, há problemas comuns à maioria deles. Como exemplo pode-se citar: ($i$) a interoperabilidade entre os dispositivos que operam a partir de uma determinada biometria; e ($ii$) a transmissão e o armazenamento [2] da informação biométrica. Não é tarefa fácil determinar qual característica biométrica é a mais adequada, considerando a diversidade de aplicações e condições de operação. No entanto, em aplicações forenses, civis e comerciais [3], impressões digitais estão se tornado cada vez mais utilizadas.

A maior parte das atuais tecnologias que lidam com impressões digitais é baseada em contato, ou seja, requerem que o usuário pressione seus dedos contra uma superfície de aquisição. Recentemente, porém, soluções em biometria por impressões digitais que dispensam o contato têm sido propostas [4]–[6] com o objetivo de se superar os problemas intrinsecamente relacionados às tecnologias que exigem o contato. Dessa maneira, o presente artigo discute um algoritmo que permite a interoperabilidade entre sistemas biométricos baseados em impressões digitais adquiridas com e sem contato [7], bem como avalia o impacto de padrões de compressão de imagens no processo de autenticação biométrica que utiliza as impressões digitais sem contato processadas pelo algoritmo em questão. A seguir serão discutidas as impressões digitais com e sem contato, e o problema da compatibilidade entre as duas tecnologias.

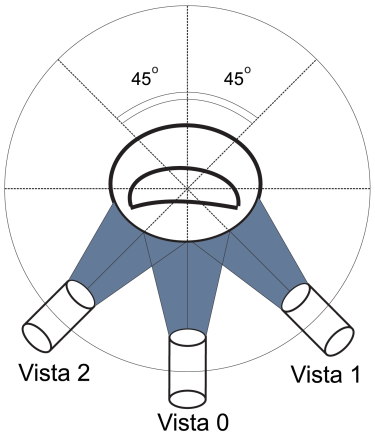



Figura 1. Ilustração esquemática de um sistema de biometria por impressão digital multivista sem contato.

### A. Impressões Digitais com Contato

O maior problema advindo dessa tecnologia são as distorções e inconsistências que podem ser introduzidas devido à elasticidade da pele. A qualidade das impressões digitais pode também ser seriamente afetada pelo contato não ideal causado por sujeira, suor, oleosidade, excesso de secura ou umidade, temperatura e impressões latentes. Além disso, partes diferentes do mesmo dedo podem ser capturadas a cada vez que o mesmo é apresentado ao sensor, resultando em amostras irreprodutíveis. Em alguns cenários tais desvantagens podem impor a necessidade de um operador qualificado durante a fase de registro e, em muitos casos, múltiplas tentativas são efetuadas por dedo até que sejam produzidas amostras com qualidade suficiente. Em aplicações governamentais como, por exemplo, o uso de biometria nas urnas eletrônicas empregadas pelo sistema eleitoral brasileiro[1], o processo de registro da população votante pode revelar-se extremamente lento. Apesar de nos últimos anos alguns algoritmos terem sido propostos

[1] http://www.tse.jus.br/eleicoes/biometria-e-urna-eletronica

(a) (b)

Figura 2. Comparação entre: (a) imagem adquirida com contato; e (b) imagem adquirida sem contato (vista 0).

com o objetivo de se compensar as limitações dos dispositivos com contato, esse paradigma de captura pode representar um gargalo no que tange à qualidade das imagens de impressões digitais.

### B. Impressões Digitais sem Contato

Essa abordagem procura atacar o problema da qualidade da impressão digital em seu nível mais fundamental. Uma vez que sistemas sem contato não obrigam o usuário a pressionar os dedos sobre uma superfície, tais sistemas não requerem, por exemplo, algoritmos que procuram compensar os artefatos decorrentes da elasticidade da pele ou do contato não ideal. Dentre as propostas de soluções sem contato, os dispositivos desenvolvidos pela empresa *Touchless Biometric Systems* (TBS)[3] utilizam uma abordagem bastante interessante e promissora [8]. Esses dispositivos combinam o imageamento baseado em reflexão com um sistema multivista construído a partir de três câmeras. Uma câmera é posicionada de maneira a capturar a parte do dedo em que normalmente núcleo e deltas estão localizados e, tomando-se essa câmera central como referência, as outras duas são posicionadas a 45 graus, uma no sentido horário e outra no sentido anti-horário. Se a sobreposição dos seus campos de visão é garantida, pode-se não apenas registrar fotograficamente uma área ampliada, como também gerar, por meio de algoritmos de costura [9], uma impressão digital equivalente a uma obtida por processo de rolagem (unha-a-unha). A Figura 1 mostra uma ilustração esquemática de um dispositivo de captura sem contato multivista.

### C. O Problema da Compatibilidade

As imagens capturadas por meio da tecnologia sem contato em geral não são compatíveis com as imagens obtidas em digitalizadores ópticos, principalmente aqueles em que as impressões digitais, para serem obtidas, requerem a rolagem do dedo sobre uma superfície plana, transparente e retro-iluminada. A Figura 2 mostra um exemplo comparativo.

Para desfrutarmos das vantagens operacionais dos dispositivos sem contato, as imagens resultantes desse tipo de captura precisam ser processadas para que se tornem similares àquelas obtidas nos digitalizadores convencionais. A seguir será apresentada uma solução para o problema da compatibilidade.

[3] http://www.tbsinc.com/

## II. Algoritmo de Compatibilidade

Inicialmente realizamos uma equalização local de histograma [10] utilizando blocos de $9 \times 9$ *pixels*. O segundo estágio do algoritmo proposto consiste em submeter a imagem $I_h$ a uma transformação gama [10], definida pela equação:

$$I_{h\gamma} = c \cdot I_h^{\gamma} \tag{1}$$

onde $c$ e $\gamma$ são constantes positivas. No presente artigo consideramos $c = 1$ e $\gamma = 1.5$.

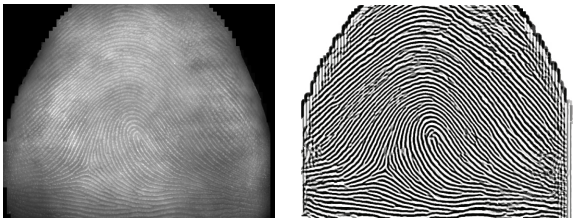
O passo seguinte consiste em filtrar a imagem $I_{h\gamma}$ com 3 filtros gaussianos de tamanhos $h_i = [3\ 5\ 7]$. Considerando que esses tamanhos de filtro representam as larguras a meia altura (*full width at half maximum* ou FWHM) de Gaussianas [11], os respectivos desvios padrões $\sigma_i$ são definidos a partir da equação,

$$\sigma_i = \frac{h_i}{2\sqrt{2\ln 2}}. \tag{2}$$

É importante ressaltar que o tamanho de bloco utilizado na equalização local de histograma, o valor de $\gamma$ utilizado no realce da imagem e a configuração dos filtros Gaussianos utilizados na filtragem das imagens foram definidos após exaustivos testes [12].

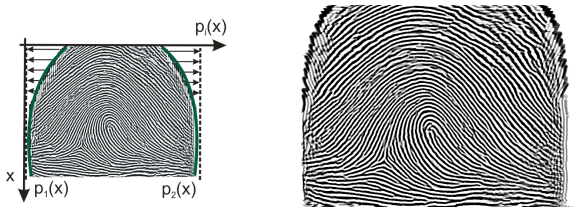
As três novas imagens, $I_{fi}$, $i = 1..3$, resultantes do processo de filtragem, são então binarizadas por meio do *software* $mindtct$ disponibilizado pelo NIST (*National Institute of Standards and Technology*) no pacote *NIST Biometric Image Software* (NBIS) [13]. O resultado são as imagens binárias $I_{bi}$, $i = 1..3$.

Gera-se, então, uma imagem $I_M$ a partir da média das imagens $I_{bi}$, conforme a Equação 3, e realiza-se um recorte exatamente nos limites da impressão digital. A Figura 3 mostra as imagens: (a) original $I$; e (b) processada $I_M$. A imagem original é uma imagem obtida a partir da costura das três vistas fornecidas pelo dispositivo de captura, ou seja, representa uma captura unha-a-unha. O aspecto de $I_M$ começa a se aproximar ao de uma imagem adquirida por meio de um dispositivo com contato. No entanto, a imagem apresenta-se afunilada em sua parte superior. Geralmente, em imagens adquiridas por meio de rolagem unha-a-unha, a diferença entre a largura da parte superior e a largura da parte inferior não é tão acentuada. Por isso, faz-se necessário um processamento cujo resultado é o alargamento da parte superior da impressão digital. O algoritmo proposto procura estimar por meio de uma regressão polinomial qual seria a função que aproxima as bordas direta e esquerda da impressão digital. Estamos assumindo que a distorção observada é simétrica. Uma vez encontrado o polinômio $p_1(x)$ que descreve a borda esquerda, por exemplo, assumimos que esse mesmo polinômio pode ser utilizado como referência para compensar também a borda direita, a partir de um polinômio simétrico $p_2(x)$. A correção geométrica proposta interpola cada linha da impressão digital



(a) (b)

Figura 3. Resultado das etapas iniciais de processamento: (a) imagem original $I$ ($712 \times 560$ *pixels*, 256 níveis de cinza); (b) imagem processada $I_M$, gerada a partir da média das imagens $I_{bi}$.




(a) (b)

Figura 4. Correção geométrica proposta: (a) cada linha da impressão digital é interpolada nas direções indicadas. O polinômio $p_1(x)$ é obtido a partir de regressão polinomial e descreve a borda esquerda da impressão digital. O polinômio $p_2(x)$ é obtido a partir da reflexão e deslocamento de $p_1(x)$; e (b) resultado da correção geométrica aplicada à imagem da Figura 3. Utilizaram-se polinômios de grau 5.

nas direções indicadas pela ilustração da Figura 4(a). Os eixos foram rotacionados para acompanhar a orientação de captura da imagem. A Figura 4(b) mostra a imagem $I_D$ resultante da correção geométrica. Utilizaram-se polinômios de grau 5.

$$I_M = \frac{1}{3} \sum_{i=1}^{3} I_{bi} \tag{3}$$

A penúltima etapa de processamento consiste em se aplicar uma textura sobre e imagem $I_D$. Para isso, projetam-se sinteticamente [14] elipses sobre $I_D$ com diagonais maiores (entre 1 e 5 *pixels*), diagonais menores (entre 1 e 3 *pixels*), ângulos (entre 0 e 180°) e níveis de cinza (entre 0 e 255) gerados aleatoriamente. O resultado é o observado na Figura 5(a). Com o objetivo de se descartar as margens da impressão digital por apresentarem ruídos causados pela baixa frequência de franjas, realiza-se um esmaecimento nas bordas da imagem. A Figura 5(b) mostra o resultado do esmaecimento, que igualmente representa o passo final do método proposto.

Na próxima seção serão descritos os padrões de compressão de imagens empregados na codificação das imagens adquiridas por meio da tecnologia de captura sem contato.

(a) (b)

Figura 5. Aplicação de textura (a) e esmaecimento das bordas (b).

## III. Padrões de Compressão de Imagens

O *Wavelet Scalar Quantizer* (WSQ), proposto pelo Federal Bureau of Investigation (FBI), é o atual padrão para compressão de imagens de impressões digitais com contato. No entanto, imagens produzidas por um sistema multivista sem contato apresentam características diferentes quando comparadas às obtidas por meio de dispositivos óticos com contato, conforme ilustrado na Figura 2. Nas próximas subseções apresentaremos brevemente os codificadores de imagem utilizados nos testes aqui propostos.

### A. WSQ

O WSQ é um algoritmo de compressão com perdas que consiste em três principais fases: ($i$) decomposição da imagem pela transformada *wavelet* discreta; ($ii$) quantização escalar dos coeficientes da transformada; e ($iii$) codificação de entropia baseada nas codificações *run-length* e Huffman. Este algoritmo tornou-se um padrão para armazenamento e transmissão de imagens de impressões digitais. Apesar do WSQ ser um codificador com alto grau de flexibilidade, diversos parâmetros são fixados com o fim de comprimir impressões digitais. A *wavelet* utilizada no WSQ é uma *wavelet* biortoognal de Cohen-Daubechies-Feauveu (CDF) 9-7 com decomposição complexa em 64 sub-bandas. Duas principais razões justificam o uso generalizado do WSQ. A primeira é a ausência de artefatos de bloco e a segunda é a preservação das características das minúcias inclusive em altas taxas de compressão.

### B. JPEG

O JPEG é o primeiro padrão de compressão de imagens em tons contínuos publicado e que foi rapidamente popularizado por conta dos bons resultados apresentados a altas taxas de compressão. O JPEG consiste em um padrão de compressão com perdas baseado na *Discrete Cosine Transform* (DCT). Para codificação: aplica-se a DCT bidimensional; são realizadas simultaneamente a normalização e a quantificação escalar uniforme [15] dos coeficientes da transformada; reordena-se a matriz $8 \times 8$ de coeficientes da transformada a partir da varredura *zigzag* [16] resultando em um vetor de 64 coeficientes; são, então, realizadas a quantização e a codificação de entropia, que toma proveito das sequências de zero que surgem dos processos anteriores.

### C. JPEG2000

Também baseado na transformada *wavelet*, o JPEG2000 é um padrão de codificação com e sem perdas. Para a codificação

com perdas é utilizado a transformada CDF 9-7, enquanto que para a compressão sem perdas utiliza-se a transformada CDF 5-3. Para decomposição em sub-bandas, o JPEG2000 utiliza a estrutura simples de Mallat. Após a transformação, os coeficientes da *wavelet* são quantizados utilizando a quantização escalar com zona morta (*scalar dead-zone quantization*) e em seguida aplica-se um codificador aritmético de entropia, o codificador binário MQ. Alguns resultados mostraram que o JPEG2000 oferece melhor compressão que o WSQ para imagens baseadas em contato, com menor impacto no desempenho do sistema biométrico [17].

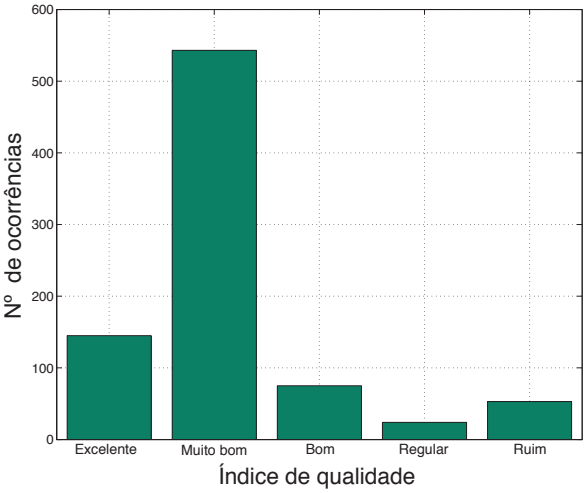
### *D. H.264/AVC*

H.264/AVC é um padrão de compressão de vídeo e a princípio não foi concebido para ser utilizado como uma ferramenta de compressão de imagens estáticas. No entanto, os avanços incorporados no H.264/AVC não só estabeleceram um novo patamar de desempenho para a compressão de vídeo, mas também o tornaram um compressor muito eficiente para imagens estáticas [18]. Nos referimos a este codificador como AVC-I. Ganhos do AVC-I sobre o JPEG2000 são tipicamente da ordem de 0.25 a 0.5 dB em PSNR para imagens em níveis de cinza. O AVC-I é um algoritmo baseado em blocos que utiliza a transformada discreta de cosseno. No entanto, incorpora um filtro de *deblocking* que pode minimizar artefatos de bloco indesejáveis.

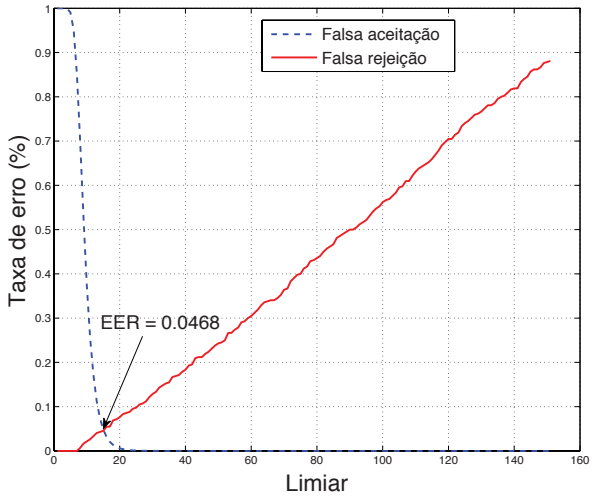
Na próxima Seção serão avaliados o algoritmo de processamento descrito na Seção II e o impacto dos padrões de compressão acima descritos no processo de casamento das impressões digitais processadas.

## IV. Resultados Experimentais

Nosso conjunto de testes é formado por 840 imagens adquiridas por um dispositivo sem contato multivista. As impressões digitais foram capturadas da forma mais natural possível, simulando um processo de aquisição real. Em outras palavras, procurou-se minimizar ao máximo a influência de um operador. O procedimento de captura foi descrito ao usuário e o operador apenas interferiu indicando ao usuário o momento em que o dedo deveria ser posicionado no dispositivo. Além disso, não foram descartadas quaisquer imagens, mesmo aquelas que pudessem ser classificadas pelo operador como sendo de baixa qualidade. As imagens representam 420 dedos diferentes, capturados duas vezes cada um. Todas as impressões digitais foram processadas pelo algoritmo descrito na Seção II. Em seguida computou-se, para cada impressão digital, um índice de qualidade a partir do *software* NFIQ (NIST *Fingerprint Image Quality*) [19]. Trata-se de um *software* baseado em redes neurais artificiais [20] que retorna um dos seguintes índices de qualidade: (1) Excelente; (2) Muito bom; (3) Bom; (4) Regular; e (5) Ruim. Detalhes a respeito de como o NFIQ estima a qualidade de imagens de impressões digitais podem ser consultados no documento proposto por Tabassi e colaboradores [19]. Porém, de uma maneira geral, o estimador baseia-se: (a) no mapa de qualidade onde são integradas informações de baixo contrastes, baixo fluxo de franjas, alta curvatura; (b) na área da impressão digital; (c) no número total de minúcias; e (d) no número de minúcias com qualidades superiores a 0.5, 0.6, 0.75, 0.8 e 0.9. A qualidade de uma minúcia é calculada a partir de dois fatores: (1) sua posição




(a)




(b)

Figura 6. (a) Índices de qualidade gerados pelo NFIQ para as 840 imagens do conjunto de testes: 90.8% das imagens receberam índices bom, muito bom ou excelente; (b) taxas de falsa aceitação e falsa rejeição geradas a partir do *software* de casamento de impressões digitais *bozorth3*: a *equal error rate* ficou em 4.68%.

no mapa de qualidade; e (2) média e desvio padrão dos *pixels* que constituem sua vizinhança imediata. A Figura 6(a) mostra os resultados gerados pelo NFIQ para o conjunto de testes considerado nesse trabalho, após ser processado pelo algoritmo de compatibilidade descrito na Seção II. Aproximadamente 90.8% das imagens receberam índices bom, muito bom ou excelente.

Após verificarmos a qualidade das impressões digitais, submetemos o conjunto de testes ao *software* de casamento *bozorth3* [13], com o objetivo de determinarmos a *equal error rate* (EER). As imagens são codificadas em níveis de cinza no formato *bitmap* (BMP) a $8\ bits/pixel$, ou seja, sem compressão. Para avaliarmos a taxa de falsa rejeição,

casamos cada uma das impressões digitais de cada dedo com a outra amostra do mesmo dedo. A taxa de falsa aceitação foi computada casando-se duas impressões digitais de cada dedo com todas as impressões digitais dos demais dedos. Ou seja, as duas amostras do dedo 1 foram casadas com todas as amostras dos dedos 2 a 420 (1676 casamentos). As duas amostras do dedo 2 foram casadas com todas as amostras dos dedos 3 a 420. E assim por diante, até que as duas amostras do dedo 419 foram casadas com as amostras do dedo 420 (4 casamentos). A definição dessas taxas depende também de um limiar estabelecido para o índice de similaridade retornado pelo *bozorth3*. A Figura 6(b) mostra as curvas de falsa aceitação e falsa rejeição para limiares entre 0 e 150. A EER ficou em 4.68% para o limiar 15.

É importante novamente ressaltar que as impressões digitais submetidas ao NFIQ e ao *bozorth3* são impressões digitais adquiridas por um dispositivo sem contato, processadas pelo algoritmo de compatibilidade discutido anteriormente. Os índices retornados por esses *softwares*, ou seja, $90.8\%$ das imagens processadas terem recebido índices bom, muito bom ou excelente, além de uma EER igual $4.68\%$, indicam que, para tais *softwares*, projetados para trabalharem com a tecnologia com contato, o fato de as imagens originalmente terem sido geradas a partir da tecnologia sem contato ficou irrelevante após processamento pelo algoritmo discutido.

Em um último experimento, submetemos todas as impressões digitais aos padrões de compressão descritos na Sessão III e recomputamos os valores da ERR e do índice de qualidade médio $Q$ para cada caso. O objetivo é verificar o impacto dos padrões de compressão no processo de autenticação biométrica que aplica o algoritmo de compatibilidade discutido anteriormente. Em trabalhos anteriores [21] foi avaliado o desempenho de algoritmos de compressão de imagens considerando um conjunto bastante restrito de 6 amostras. Cada amostra era composta pelas três vistas disponibilizadas pelo dispositivo de aquisição sem contato e o que se desejava era comparar o desempenho dos codificadores em termos de taxa-distorção (taxa de *bits* versus *peak signal-to-noise ratio*). O conjunto atual conta com 840 impressões digitais e não estamos trabalhando com as três vistas, mas sim com uma imagem única costurada a partir delas. Além disso, não estamos interessados em avaliar o desempenho dos codificadores de imagens em termos de taxa-distorção, mas sim o impacto desses codificadores em termos de ERR no processo de autenticação biométrica após o algoritmo de compatibilidade entre as tecnologias com e sem contato ter sido aplicado. Nesse novo cenário, os codificadores JPEG, JPEG2000 e WSQ foram ajustados para devolverem imagens comprimidas a uma taxa de aproximadamente $1\ bit/pixel$. Já o AVC-I foi ajustado para devolver imagens codificadas a aproximadamente $0.5\ bit/pixel$ (parâmetro de quantização $QP = 1$). Para o nosso conjunto de testes, o AVC-I não pôde devolver imagens codificadas a taxas acima de $0.5\ bit/pixel$. Os resultados podem ser observados na Tabela I. São comparados para cada codificador: a taxa de *bits* em $bits/pixel$ (bpp), a *equal error rate* (EER) e um índice de qualidade médio $Q$ gerado a partir dos dados fornecidos pelo NFIQ.

A seguir os resultados experimentais serão comentados.

Tabela I. RESULTADOS OBTIDOS

| Codificador | Taxa (bpp) | EER % | Q |
|---|---|---|---|
| BMP | 8.0000 | 4.68 | 2.1631 |
| JPEG | 0.9492 | 4.85 | 2.1940 |
| JPEG2000 | 0.9972 | 3.98 | 2.2536 |
| WSQ | 0.7925 | 4.96 | 2.2310 |
| AVC-I | 0.4543 | 4.83 | 2.1393 |

## V. CONCLUSÃO

O presente artigo discutiu um algoritmo que permite a compatibilidade entre sistemas biométricos que operam a partir de impressões digitais com e sem contato. Além disso, avaliou o impacto de padrões de compressão de imagens no processo de autenticação biométrica que utiliza as impressões digitais sem contato processadas pelo algoritmo em questão. Na avaliação da qualidade das impressões digitais sem compressão (formato BMP), $90.8\%$ das imagens sem contato processadas pelo algoritmo de compatibilidade receberam do NFIQ os índices de qualidade excelente, muito bom e bom, resultando numa qualidade média $Q$ de $2.1631$. Quanto ao teste de autenticação biométrica, a EER ficou em $4.68\%$. Esse dois índices mostram o funcionamento adequado do algoritmo proposto, uma vez que os *softwares* que os geraram foram pensados para serem aplicados a imagens geradas a partir de tecnologias com contato.

Quando à robustez do algoritmo de compatibilidade em relação aos padrões de compressão, definimos a métrica $C = 1/(EER \times BpP)$ como forma de comparação, onde $C$ é o índice de comparação, $EER$ é a já definida *equal error rate* e $BpP$ é a taxa de compressão média do conjunto de imagens em *bits* por *pixel*. Observamos que dentre os codificadores utilizados, o que gera o menor impacto é o AVC-C, com $C_{avc} = 45.57$, e o que gera o maior impacto é o JPEG, com $C_{jpg} = 21.72$. Em outras palavras, o algoritmo proposto é, a uma dada taxa, mais robusto às degradações geradas pelo AVC-I do que as geradas pelo JPEG. O JPEG2000 e o WSQ apresentaram desempenhos intermediários e equiparáveis, com $C_{jp2} = 25.19$ e $C_{wsq} = 25.44$, respectivamente.

É interessante observar que a ERR para o JPEG2000 ($3.98$) foi bem melhor que no caso em que as imagens nem sequer foram comprimidas ($4.68\%$). Isso se deve provavelmente à instabilidade do *software* de casamento de minúcias. Além disso, vale ressaltar o baixo impacto dos algoritmos de compressão no índice médio de qualidade $Q$ obtido por meio do NFIQ. Todos os valores obtidos ficaram entre $2.1$ e $2.3$.

Para trabalhos futuros pretende-se aprimorar o algoritmo de compatibilidade, além de se estudar a autenticação biométrica entre impressões digitais do mesmo indivíduo, porém adquiridas por ambas as tecnologias, com e sem contato, aqui estudadas.

## REFERÊNCIAS


[1] J. Wayman, "A definition of biometrics," in *National Biometric Test Center Colected Works 1997-2000*. San Jose State University, 2000.

[2] J. Wayman, A. Jain, D. Maltoni, and D. Maio, "An introduction to biometric authentication systems," in *Biometric Systems: Technology, Design and Performance Evaluation*. London: Springer, 2005, ch. 1.

[3] R. Allen, P. Sankar, and S. Prabhakar, "Fingerprint identification technology," in *Biometric Systems: Technology, Design and Performance Evaluation*, J. Wayman, A. Jain, D. Maltoni, and D. Maio, Eds. London: Springer, 2005, ch. 2.

[4] R. Labati, V. Piuri, and F. Scotti, "Neural-based quality measurement of fingerprint images in contactless biometric systems," in *Neural Networks (IJCNN), The 2010 International Joint Conference on*, july 2010, pp. 1 –8.

[5] R. D. Labati, V. Piuri, and F. Scotti, "A neural-based minutiae pair identification method for touch-less fingerprint images," in *Computational Intelligence in Biometrics and Identity Management (CIBIM), 2011 IEEE Workshop on*, april 2011, pp. 96 –102.

[6] V. Piuri and F. Scotti, "Fingerprint biometrics via low-cost sensors and webcams," in *Biometrics: Theory, Applications and Systems, 2008. BTAS 2008. 2nd IEEE International Conference on*, 29 2008-oct. 1 2008, pp. 1 –6.

[7] A. Zaghetto, P. S. Franco, and D. A. M. Sandoval, "Método para compatibilidade entre impressões digitais com e sem contato," in *Anais da Conferência Internacional de Ciências Forenses em Multimídia e Segurança Eletrônica*, 2012, pp. 44–48.

[8] G. Parziale, "Touchless fingerprinting technology," in *Advances in Biometrics: Sensors, Algorithms and Systems*, N. K. Ratha and V. Govindaraju, Eds. London: Springer, 2008, ch. 2.

[9] D. Milgram, "Computer methods for creating photomosaics," *IEEE Transactions on Computers*, vol. C-24, no. 11, pp. 1113–1119, Nov. 1975.

[10] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 3rd ed. Prentice Hall, 2007.

[11] E. W. Weisstein, "Gaussian Function," http://mathworld.wolfram.com/GaussianFunction.html, [Online; Acessado em 21-Junho-2012].

[12] P. S. Franco, "Método para compatibilidade entre impressões digitais com e sem contato," in *Trabalho de Conclusão de Curso (Engenharia de Computação) - Universidade de Brasília*, Brasil, 2014.

[13] C. Watson, M. Garris, E. Tabassi, C. Wilson, R. M. M. amd S. Janet, and K. Ko, *User's Guide to NIST Biometric Image Software*. NIST, 2008.

[14] R. Cappelli, A. Erol, D. Maio, and D. Maltoni, "Synthetic fingerprint-image generation," in *Pattern Recognition, 2000. Proceedings. 15th International Conference on*, vol. 3, 2000, pp. 471–474 vol.3.

[15] J. Yu, "Advantages of uniform scalar dead-zone quantization in image coding system," in *Communications, Circuits and Systems, 2004. ICCCAS 2004. 2004 International Conference on*, vol. 2, June 2004, pp. 805–808 Vol.2.

[16] W. Pennebaker and J. Mitchell, *JPEG: Still Image Data Compression Standard*, ser. Chapman & Hall digital multimedia standards series. Springer, 1993.

[17] M. A. Figueroa-Villanueva, N. K. Ratha, and R. M. Bolle, "A comparative performance analysis of jpeg 2000 vs. wsq for fingerprint image compression," in *Int. Conf. on Audio- and Video-Based Biometric Person Authentication, 4th*, 2003, pp. 385–392.

[18] R. L. de Queiroz, R. S. Ortis, A. Zaghetto, and T. A. Fonseca, "Fringe Benefits of the H.264/AVC," *Proc. of ITS*, pp. 208–212, Sept. 2006.

[19] E. Tabassi, C. L. Wilson, and C. I. Watson, *Fingerprint Image Quality*. NIST, 2004.

[20] S. Haykin, *Redes Neurais, Princípios e Prática*, 2nd ed. Pearson Education, 1999.

[21] N. Francisco, A. Zaghetto, B. Macchiavello, E. da Silva, M. Lima-Marques, N. Rodrigues, and S. M. M. De Faria, "Compression of touchless multiview fingerprints," in *Biometric Measurements and Systems for Security and Medical Applications (BIOMS), 2011 IEEE Workshop on*, Sept 2011, pp. 1–5.

# Face Recognition Using 3DLBP Method Applied to Depth Maps Obtained from Kinect Sensors

João Baptista Cardia Neto
Graduate Program in Computer Science
UNESP - São Paulo State University
Bauru,São Paulo, 17018-820
Email: joaobcardia@gmail.com

Aparecido Nilceu Marana
Department of Computing - Faculty of Sciences
UNESP - São Paulo State University
Bauru,São Paulo, 17018-820
Email: nilceu@fc.unesp.br

***Abstract*—Due to the difficulty in dealing with pose and illumination on 2D face recognition, 3D face recognition methods have been proposed in the last years, since 3D data is less sensible to illumination changes and, more important, is very useful to correct differences in pose. The main drawback of using 3D-based face recognition methods is the high cost of the traditional 3D sensors. One alternative to these expensive scanners are the Kinect devices that, besides being considerable cheaper, are able to capture quite precisely the depth information. The main goal of this work was to assess the performance of 3DLBP method for face recognition when using depth maps generated by Kinect devices. Another goal was to investigate the face regions in which the depth maps contribute most to increase the face recognition rates.**

## I. Introduction

Biometrics methods, which have been considered a solution for the problem of person identification, use physical/physiological (face, fingerprint, iris) or behavioral (gait, typing dynamics, signatures) traits [1], since they define what a person is, whilst the traditional identification methods are based on what a person has (identification cards, keys, tokens) or knows (password, personal data). The main advantages of biometrics identification methods are convenience, accuracy and security (they are more difficult to circumvent). For face based person identification systems another remarkable advantage is their ability to operate in covert mode, at a distance.

In controlled environments, the current 2D facial recognition methods can achieve quite satisfactory performances, better than human performances in many cases [2]. However, in more challenging environments, issues begin to appear regarding the illumination and pose variability.

By utilizing 3D face recognition methods, one can reduce the aforementioned problems [3]. However, one of the main limitations of such methods is the high cost of the traditional 3D face sensors, that can range from two thousand to fifty thousand dollars, as one can see in the Table I, which illustrates the cost and other characteristics of different 3D sensors.

One promising alternative to these expensive scanners are the Kinect devices that, besides being considerable cheaper, are able to capture quite precisely the depth information.

The face recognition method proposed in this paper replaces the input data obtained by traditional and expensive 3D sensors, such as the Minolta scanner, by depth images obtained by a cheaper Kinect sensor. This method applies the 3DLBP method [4] in order to obtain the face descriptors.

TABLE I. Comparison among different 3D scanners [6]. The speed is expressed in seconds, size in $inch^3$, price in USD, and the accuracy is an approximation expressed in mm.

| Device | Speed | Charge | Size | Price | Acc. |
|---|---|---|---|---|---|
| 3dMD | 0.002 | 10 sec | N/A | >$50K | <0.2 |
| Minolta | 2.5 | no | 1408 | >$50K | 0.1 |
| Artec Eva | 0.063 | no | 160.8 | >$20K | 0.5 |
| 3D3 HDI R1 | 1.3 | no | N/A | >$10K | 0.3 |
| SwissRanger | 0.02 | no | 17.53 | >$5K | 10 |
| DAVID SLS | 2.4 | no | N/A | >$2K | 0.5 |
| Kinect | 0.033 | no | 41.25 | >$200 | 1.5-50 |

In the experiments performed on the EURECOM Face Dataset [5], the proposed method reached 96% of correct classification at rank 1, a result much better than the one obtained by the traditional 2D LBP method, which obtained 78% of correct classification at rank 1 on the same database.

Another contribution of this paper is the identification of the best face regions from where the 3DLBP descriptors should be extracted. Some regions are very sensitive to facial expressions and their use can decrease significantly the performance of the biometrics identification system, as show the results obtained in our experiments (Section IV).

### A. Related Works

The work proposed by [7] also utilizes Kinect data for face recognition. On their work, an entropy map is generated for the depth and RGB images and a visual salience map is generated only for the RGB image. They utilize the HOG (Histogram of Oriented Gradients) for describing the entropies and the salience map. For the recognition, the HOGs from different images patches are concatenated, and a RDF (Random Decision Forest) classifier is trained and utilized to classify the subjects.

One important aspect about the aforementioned work is the utilization of two different databases for experiments: the IIIT-D RGB-D [7] and the EURECOM [5] face databases.

The IIIT-D RGB-D [7] is composed by 106 subjects with multiple RGB-D images, and the number of samples in the database for each subject varies from 11 to 254. This database was not utilized in the current work because it is not publicly available.

The EURECOM Face Dataset [5] is composed by 52 subjects, 14 females and 38 males. There are two sets of images captured in the interval of 15 days and the images in each set have nine variations. The conditions are: neutral, smiling, open mouth, left and right profile, occlusion of the eyes, mouth and half of the face, and illumination variation. The images have three different source of information: depth (in bitmap depth images and text files with all the values sensed by the Kinect), RGB image, and 3D .obj files. For each sample and for each format there are annotations for the left and right eyes, the tip of the nose, the left and right sides of the mouth and the chin.

Another work utilizing Kinect for face recognition is the one proposed by [6]. In this work, the authors utilize the depth map and the RGB image, the color from the RGB image goes through the DCS (Discriminant Color Space) transform before being utilized, since it has three channels they are stacked. The classification of the samples occurs with the SRC (Sparse Representation Classifier). The depth map is generated from a cloud of points resulted from a process called Symmetric Filling, in which the cloud of points is mirrored, after a pose correction. If the Euclidean distance between a point in the mirrored cloud and its closest point on the original image is smaller than a threshold that mirrored point is added to the original cloud of points. Since the simetrical characteristics of the human face, the Simetric Filling can be an interesting strategy to improve the quality of the clouds of points generated by the Kinect.

All the aforementioned works and this utilizes the Kinect device as a scanner for depth data. The main differences are the techniques and what is evaluated. The current work applies 3DLBP operator in a set of different regions and tries to conclude which strategy is better. Related works utilize different techniques and do not explore which set of regions have better results. Another difference is the focus on depth maps, the related works utilizes depth data but they also utilize the RGB images in their proposed method whilst in the current paper RGB is only utilized as a comparison and it is not part of the proposed method. The main relation between works is the Kinect.

### *B. Proposed Method*

In the proposed face recogniton method based on Kinect input data, three main steps are applied to the dataset: segmentation, feature extraction, and classification.

The segmentation is centered at the subject's nosetip, and then a square region with the width and height equal to the Euclidean distance of the subject's eyes is cropped, This cropped region is downsized to a 32x32 square region, which is utilized in the next steps for the feature extraction.

The feature extraction step is carried out by applying the 3DLBP on all 32x32 face region. Each region that goes through the feature extraction has a histogram describing it, in case of having multiple regions in the same face image the face descriptor is created by concatening all 3DLBP histograms.

The face descriptor obtained in the previous step are used to train a SVM (Support Vector Machine) classifier. The whole process is summarized in the Fig. 1.

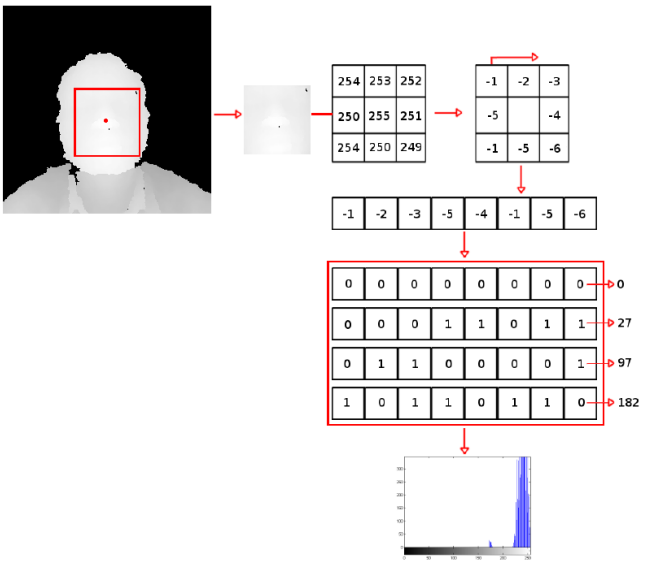



Fig. 1. Overview of the method utilized in the present paper.

## II. KINECT

The Microsoft Kinect device contains a depth sensor, a RGB camera, a microphone array, and a tilt engine. The sensor projects an infrared grid of 640X480 with a precision of 11 bits @ 30 Hz [8]. This grid is utilized to calculate the depth information. The way Kinect calculates the depth information was not revealed. However, based on [9], it is possible to speculate that the depth calculation is done by projecting a matrix of infrared dots in a group of objects and measuring the distortion caused by those objects in the infrared matrix.

The field of vision of the cameras are 57 horizontally and 43 vertically [10]. One important thing to notice about the Kinect hardware is that the angle is relative to the horizon and not to the base of the device. The possibility to control the tilt allows scanning the scene and finding the better angle to capture the biometrics characteristics.

Observing the Table I it is possible to see that Kinect is not very accurate, due to this the nose tip becomes the most reliable landmark of the face [6]. The very positive aspect of Kinect is its price and, since it is essentially a camera, the scanning of a subject becomes relatively easy and non-intrusive. The Fig. 2 shows a Kinect.

## III. 3DLBP

The original 2D LBP operator was proposed for texture description and can not deal with the behavior of depth values. This can be observed if two distinct points on different samples have the bigger and the lower depth values, respectively, than all their neighbor points. In this case they will have the same operator value, even though they are from different subjects [4]. This will happen commonly for nose tip points.

To deal with situations like this the 3DLBP was proposed [4]. The new approach of the 3DLBP considers, besides the signal of the difference betwen the depths, the difference value as well. In [4], the authors stated that more than 93% of

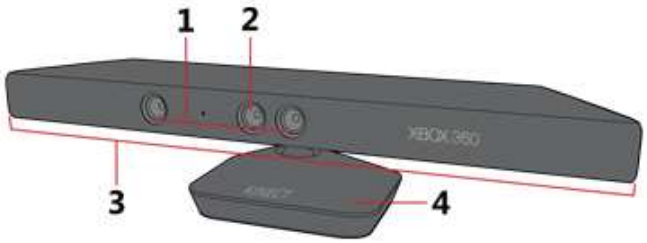



Fig. 2. The Kinect sensor: 1) 3D Depth Sensor, 2) RGB Camera, 3) Microphone array and 4) Motorized tilt [11].

all depth differences ($DD$) from points within a radius of two are smaller than 7. Due to this property, the absolute value of the $DD$ is stored in three binary units ($i_2 i_3 i_4$). Therefore, it is possible to affirm:

$$|DD| = i_2 \cdot 2^2 + i_3 \cdot 2^1 + i_4 \cdot 2^0 \quad (1)$$

There is also $i_1$, a binary unit defined by:

$$i_1 = \begin{cases} 1 & \text{if DD} \geq 0; \\ 0 & \text{if DD} < 0. \end{cases} \quad (2)$$

Those four binary units are divided into four layers and, for each layer, four decimal numbers are obtained: $P_1, P_2, P_3, P_4$. The value of the $P_1$ has the same value as the original LBP. For matching, the histograms of the local regions ($P_1, P_2, P_3, P_4$) are concatenated. The Fig. 3 shows the process for the generation of the 3DLBP face descriptors.

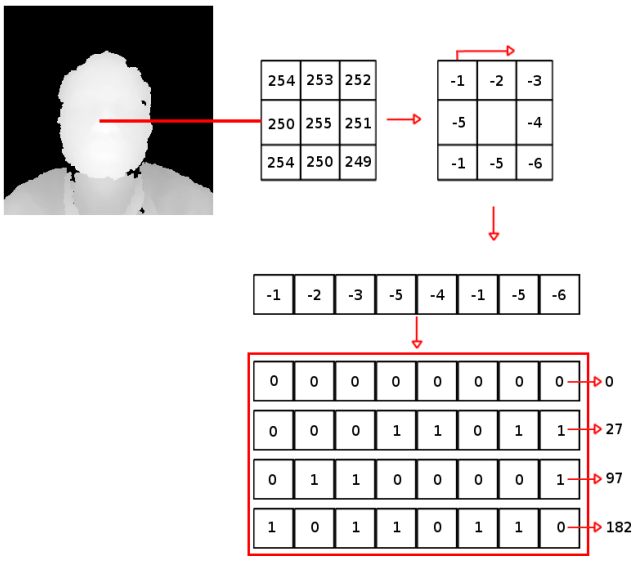
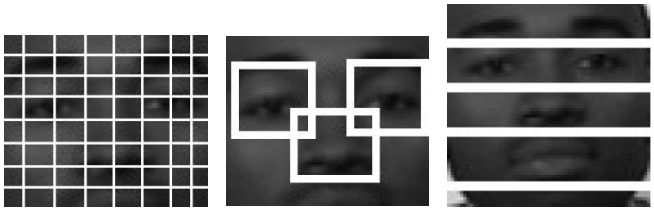



Fig. 3. The full process of the 3DLBP proposed by [4]. Each of the differences is encoded into the layers (layer 2, 3 and 4) and the signal into the layer 1.

## IV. Experiments

In this work, the experiments were carried out utilizing the EURECOM Face Dataset. For all the experiments, the gallery contains images with illumination difference, open mouth and smiling. The neutral face is taken as the probe. All those images are from the first session.

Four different types of experiments were made. Three of them used 3DLBP features, but extracted from different regions. The first one used the nose and eyes region. The second one used horizontal slices of the face extracted from 5 regions: forehead, eyes, nose, mouth and chin (which are the same set of regions utilized in [4]). The last region evaluated is defined by a 32X32 area approximately centered at the nose and excluding the mouth. The segmented area is then divided into 64 micro regions. For the face descriptor the histograms are fused. The Fig. 4 shows the different types of regions.



(a) The face divided into 64 regions (b) Eyes and nose regions. (c) Five slices of the face.

Fig. 4. Different regions in which the 3DLBP operator is evaluated.

The last experiment used the traditional 2D LBP operator. The regions were the same used in the third experiment on the 3DLBP, the difference is that the RGB images were used, instead of the depth maps. The images were converted to greyscale before the extraction of the operators. While the 3DLBP was fully implemented by the authors for the 2D LBP the implementation of [12] was utilized.

The Fig. 5 shows samples of images utilized in the experiments. For the experiments, the SVMTorch [13] implementation of SVM was utilized with RBF Kernel, the values of the parameters were not tweaked. The feature vector for each face was the concatenation of the histograms for each face region. The training set for the SVM classifier was composed by the faces in the gallery, the faces in the probe composes the testing set.

## V. Results

The Fig. 6 shows the CMC (Cumulative Match Characteristics) curve obtained in our experiments. For each subject the probe set was classified in descending order utilizing the score. Let $i$ be the genuine probe position in the ordered set (with $i$ ranging from 1 to the total number of subjects). Then, the Rank $n$ can be defined as:

$$Rank\, n = \begin{cases} true & \text{if } i \leq n; \\ false & \text{if } i > n. \end{cases} \quad (3)$$

The CMC Curve represents the occurrence percentage of Rank $n$, for all $n$, in the probe set.

It is possible to see that the traditional 2D LBP operator presented the lower performance. The original 3DLBP (the one

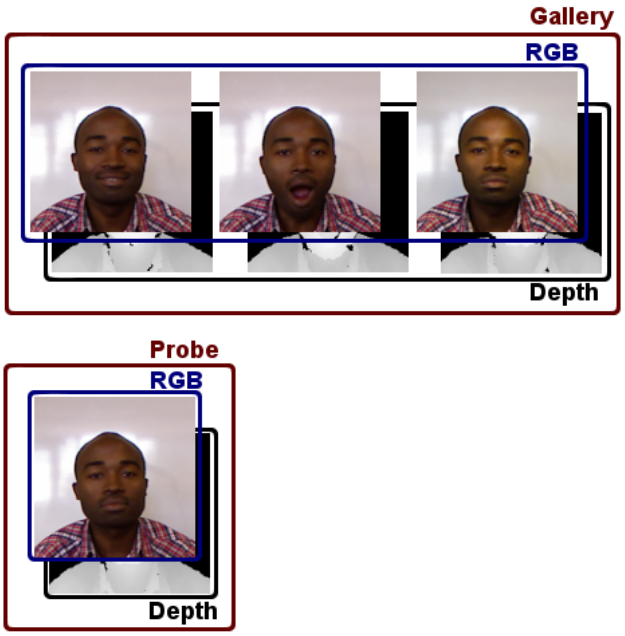



Fig. 5. Set of faces utilized for probe and gallery.

TABLE II. IDENTIFICATION ACCURACY FOR THE EXPERIMENTS.

| Technique | Rank 1 | Rank 2 | Rank 3 |
|---|---|---|---|
| 3DLBP - Nose and Eyes region | 82% | 92% | 98% |
| LBP | 78% | 86% | 92% |
| 3DLBP - Five slices from face | 78% | 94% | 96% |
| 3DLBP - Face cropped in 64 regions | 96% | 96% | 96% |

that utilizes the five slices of the face) had a poor performance at Rank 1, this was probably due to the use of the mouth region, which is highly sensible to facial expression deformations. The other experiments did not use the aforementioned region and showed better results when dealing with facial expressions. The Table II shows the identification accuracy obtained in the experiments.

3DLBP method with the face cropped in 64 regions starts with 96% at Rank 1. This means that, from 52 subjects, it misclassified two subjects. This is the reason that its rank remains continuous from 1 to 5. Utilizing 3DLBP with applied to the nose and eyes regions is the second better set of regions, starting with 82% Rank 1, followed by 3DLBP with five slices from the face and the original LBP method, both with 78% Rank 1.

Even though 3DLBP applied to the five regions of the face and the LBP starts with the same value at Rank 1, the 3DLBP method has an increase of 16% in Rank 2, jumping to the second best performance. At Rank 3, the 3DLBP performs better in the Nose and Eyes regions performs better than the 64 regions, at this point it is the best set of regions.

Finally, the LBP and the 3DLBP applied with 64 regions are the first to reach 100% in Rank 8.

## VI. CONCLUSIONS AND FUTURE WORK

From the experiment results, some conclusions can be pointed. First of all, it is very apparent that depth data takes

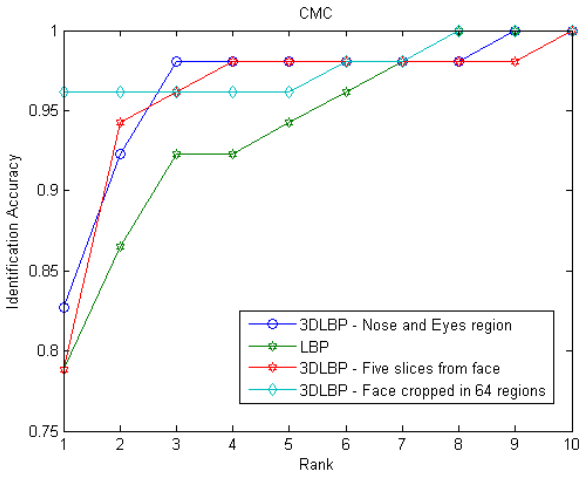



Fig. 6. CMC curve from experiments.

advantages in the chosen dataset, this can be explained due to the fact that illumination differences in RGB images generates more problems than in depth maps.

Another important conclusion is that, when not utilizing the mouth area, the techniques appears to be more resistant to expression variations. The region of the eyes and nose suffer less from deformations caused by expression. So, when focusing only on those regions, it is possible to get better results.

Finally, from the results obtained in this work, it is possible to conclude that the data generated by Kinect, even though of lower resolution when compared with other traditional 3D sensors, are discriminative enough to allow a correct identification among different subjects.

A possible extension to the current work is the evaluation of profile images by using the cloud of points also generated by Kinect.

## REFERENCES

[1] R. Bolle and S. Pankanti, *Biometrics, Personal Identification in Networked Society: Personal Identification in Networked Society*, A. K. Jain, Ed. Norwell, MA, USA: Kluwer Academic Publishers, 1998.

[2] S. Z. Li and A. K. Jain, Eds., *Handbook of Face Recognition, 2nd Edition.* Springer, 2011. [Online]. Available: http://dblp.uni-trier.de/db/books/daglib/0027896.html

[3] I. A. Kakadiaris, G. Passalis, G. Toderici, M. N. Murtuza, Y. Lu, N. Karampatziakis, and T. Theoharis, "Three-dimensional face recognition in the presence of facial expressions: An annotated deformable model approach," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 4, pp. 640–649, Apr. 2007. [Online]. Available: http://dx.doi.org/10.1109/TPAMI.2007.1017

[4] Y. Huang, Y. Wang, and T. Tan, "Combining statistics of geometrical and correlative features for 3d face recognition," in *Proceedings of the British Machine Vision Conference*. BMVA Press, 2006, pp. 90.1–90.10, doi:10.5244/C.20.90.

[5] T. Huynh, R. Min, and J.-L. Dugelay, "An efficient LBP-based descriptor for facial depth images applied to gender recognition using RGB-D face data," in *ACCV 2012, Workshop on Computer Vision with Local Binary Pattern Variants, Daejeon, Korea, November 5-9, 2012 / Published also as LNCS, Vol 7728, PART 1*, Daejeon, KOREA, DEMOCRATIC PEOPLE'S REPUBLIC OF, 11 2012. [Online]. Available: http://www.eurecom.fr/publication/3849

[6] B. Li, A. Mian, W. Liu, and A. Krishna, "Using kinect for face recognition under varying poses, expressions, illumination and disguise," in *Applications of Computer Vision (WACV), 2013 IEEE Workshop on*, 2013, pp. 186–192.

[7] G. Goswami, S. Bharadwaj, M. Vatsa, and R. Singh, "On rgb-d face recognition using kinect," *International Conference on Biometrics: Theory, Applications and Systems*, 2013.

[8] M. Zollhofer, M. Martinek, G. Greiner, M. Stamminger, and J. Sussmuth, "Automatic reconstruction of personalized avatars from 3d face scans," *Comput. Animat. Virtual Worlds*, vol. 22, no. 2-3, pp. 195–202, Apr. 2011. [Online]. Available: http://dx.doi.org/10.1002/cav.405

[9] A. Shpunt and Z. Zalevsky, "Depth-varying light fields for three dimensional sensing," 2008, uS Patent App. 11/724,068. [Online]. Available: https://www.google.com/patents/US20080106746

[10] J. Stowers, M. Hayes, and A. Bainbridge-Smith, "Altitude control of a quadrotor helicopter using depth map from microsoft kinect sensor," in *Mechatronics (ICM), 2011 IEEE International Conference on*, april 2011, pp. 358 –362.

[11] "Kinect sensor components," 2010. [Online]. Available: http://support.xbox.com/en-US/xbox-360/kinect/kinect-sensor-components

[12] L. P. Coelho, "Mahotas: Open source software for scriptable computer vision," *Journal of Open Research Software*, vol. 1, 2013. [Online]. Available: http://dx.doi.org/10.5334/jors.ac

[13] R. Collobert, S. Bengio, and C. Williamson, "Svmtorch: Support vector machines for large-scale regression problems," *Journal of Machine Learning Research*, vol. 1, pp. 143–160, 2001.

# Evaluating LBP and CLBP in classifying of mammograms lesions

Yan A. S. Duarte*, Marcelo Z. do Nascimento†, Sidon C. Duarte* e Leandro A. Neves ‡
*Centro de Matemática, Computação e Cognição (CMCC), Universidade Federal do ABC, Santo André, São Paulo
†Faculdade de Computação (FACOM), Universidade Federal de Uberlândia, Uberlândia, Minas Gerais
‡Departamento de Ciência da Computação e Estatística, Universidade Estadual de São Paulo, São José do Rio Preto, São Paulo
Email: yan_asd@hotmail.com, marcelo.zanchetta@gmail.com, sidoncd@gmail.com, leandro@ibilce.unesp.br

***Resumo*—In this work, computational methods were developed to analysis textural characteristics of mammographic images in order to obtain better results of rates in area under ROC curve (AUC) in CADX systems. We investigated robust techniques for extracting textural descriptors based on wavelet domain, LBP operator and a derivation. To make the feature extraction, the techniques Local Binary Pattern (LBP) and Completed Local Binary Pattern(CLBP) were evaluated applied in sub-images of wavelet domain. The images were obtained through the Digital Database for Screening Mammography (DDSM) The reducer of Analysis of Variance (ANOVA) was used to discard the redundant informations obtained through extraction techniques. The algorithms Random Forest (RaF) and Support Vector Machines (SVM) were used to the classification of cases. Quantitative analysis using AUC were used to analyse the proposed algorithms in textural analysis. The most significant results (AUC = 100 %) were obtained with the SVM classifier and the CLBP technique for the set of images of all classes and visions.**

## I. Introdução

Dentre os diversos métodos aplicados no diagnóstico do câncer de mama, o sistema mamográfico tela/filme se destaca devido ao baixo custo operacional e a fácil acessibilidade em nosso país. As imagens obtidas por esse equipamento são capazes de mostrar anormalidades com tamanho milimétrico [1]. A análise das diferenças de densidades ópticas presentes nos mamogramas permite ao especialista avaliar a presença de possíveis anormalidades e fornecer o diagnóstico ao paciente. No entanto, limitações ocorrem na detecção de lesões como o nódulo mamário, tendo em vista que estruturas mamárias com características densas podem ocultar essas anormalidades. Além disso, o nível de experiência do especialista e o extenso número de casos a serem avaliados na rotina diagnóstica podem influenciar o desempenho do número de acertos na fase do diagnóstico [2]. Segundo [3], uma forma de minimizar esses problemas é utilizar a dupla leitura, onde um outro especialista faz uma *"leitura suplementar"*. Essa estratégia eleva o custo operacional, o que em grande parte dos centro de radiodiagnóstico torna-se inviável.

Uma das formas de solucionar esse problema é o desenvolvimento de sistemas computacionais de auxílio ao diagnóstico (*Computer Aided Diagnosis* - CADx). Esse sistema tem como propósito extrair informações de regiões de interesse em imagens mamográficas digitalizadas com o intuito de classificá-los de acordo com a lesão presente. A etapa crucial para um bom desempenho dos algoritmos de classificação está relacionada à aplicação de algoritmos de extração de características [4]–[6]. Essas características podem ser categorizadas como morfológicas e não morfológicas. Os descritores morfológicos representam as extremidades da imagem como bordas e margens, enquanto os não morfológicos representam as características internas, os *pixels* que constituem a região.

Dentre as categorias, as características não morfológicas são consideradas problemáticas devido à dificuldade de interpretação das variações das informações internas pelo olho humano. Uma importante classe usada nessa representação pode ser obtida pela textura da imagem. Essa classe vêm sendo explorada na análise de imagens médicas para determinação de padrões na classificação de imagem [7]. Muitas aproximações vêm sendo propostas para extração dessa informação e, consequentemente, esse tópico de pesquisa vem ganhando destaque em parte dos estudos na área médica [8].

Estudos vêm demonstrando as vantagens da aplicação de técnicas baseadas em transformada *wavelet* na análise de lesões em mamogramas [9]–[11]. Os coeficientes *wavelets* são utilizados como importantes descritores das informações presentes em uma imagem. Um dos principais fatores de sua aplicação pode ser atribuído ao fato da técnica permitir separar os dados espaciais que contenham características específicas [12]–[14]

O padrão binário local (*Local Binary Patterns* - LBP) assim como o padrão binário local completo (*Completed Local Binary Pattern* - CLBP) são técnicas de extração de características não morfológicas de abordagem estática. O LBP vem sendo utilizado como um descritor promissor para obtenção de informações relacionados a textura de uma imagem. As principais vantagens são a não influência das variações de iluminação, baixa complexidade computacional, capacidade de codificar detalhes e permitir realizar análise local das informações [15]. Já o CLBP é uma técnica mais recente e elabora uma análise mais abrangente da imagem levando em consideração as informações de magnitude e a intensidade do *pixel* central em relação a imagem como um todo [16]. A utilização tanto da técnica LBP quanto da técnica CLBP sobre as subbandas *wavelets* conseguem obter informações que caracterizam melhor a imagem (em relação a textura). Isso se da devido ao fato das informações mais relevantes das texturas encontram-se nos canais de alta frequência [17].

O presente trabalho propõe um método para extração de características baseado em operadores texturais. O algoritmo propõe uma investigação do uso de operadores LBP e CLBP

em subbandas obtidas por meio de transformada wavelet. Os classificadores *Support Vector Machine* e *Random Forest* foram empregados na avaliação dos resultados obtidos com os descritores propostos. Um grupo de imagens mamográficas digitalizadas da base de dados *Digital Database for Screening Mammography* (DDSM) foi avaliado com os métodos propostos.

## II. METODOLOGIA

O diagrama de blocos da Figura 1 demonstra a estrutura da sequência utilizada na realização dos experimentos propostos.

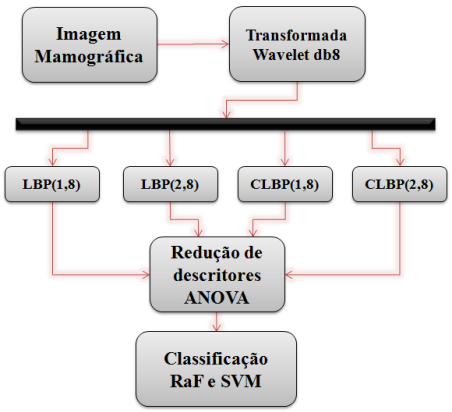



Figura 1. Diagrama em blocos do estudo proposto na comparação dos descritores de textura.

Nessa sequência, primeiramente as imagens mamográficas foram submetidas a transformada *wavelet* DB8. As subbandas de detalhe obtidas dessa abordagem foram combinadas e armazenadas. Logo após isso, essas subbandas foram submetidas a quatro diferentes experimentos com LBP e CLBP. Para diminuir o número de atributos de cada experimento, a técnica ANOVA foi empregada. Por último, os classificadores *Random Forest* e *Support Vector Machine* foram utilizados para avaliar as técnicas propostas.

A seguir são detalhadas cada etapa da metodologia utilizada neste artigo.

### A. Banco de Imagens

No desenvolvimento deste artigo foram utilizadas 720 regiões de interesse (*Regions of Interest* - ROI's) das imagens mamográficas, onde 240 não contêm nenhum tipo de anormalidade, 240 contêm anormalidade benigna e as últimas 240 contêm anormalidade maligna. A seleção das ROI's das imagens foram obtidas através do arquivo .ics, onde o mesmo contém um código em cadeia da posição da lesão. As imagens foram digitalizadas com o equipamento Lumisys em 50 $\mu$m e com 12 bits de resolução. As ROI's foram recortadas com tamanho 512×512 pixels, a qual o centro da imagem corresponde ao centro da anormalidade presente. Estas imagens estão disponíveis no banco de dados de domínio público, DDSM (*Digital Database for Screening Mammography*), da Universidade do Sul da Flórida - EUA. Este banco de dados é fruto de um estudo colaborativo onde participam pesquisadores de diferentes instituições dos Estados Unidos: *Massachusetts General Hospital* (MGH), *Wake Forest University School of Medicine* (WFUSM), *Sacred Heart Hospital* (SHH) e *Washington University* (WU). Esta base possui cerca de 2.500 casos, sendo que cada caso possui duas imagens das visões Crânio-Caudal (CC) e Médio-Lateral Oblíqua (MLO) referentes à mama esquerda e direita. Os casos foram obtidos entre os anos de 1988 e 1999 e digitalizados pelos equipamentos: Howtek (WU e MGH), Lumisys (WFUSM e SHH) e DBA (MGH). As imagens estão compactadas no padrão LJPEG e foram digitalizadas em níveis de cinza com resolução de 12 ou 16 *bits* por *pixels*. Informações como idade, época do estudo, código do paciente, classificação na categoria e na composição da densidade mamária de acordo com o padrão BI-RADS (*Breast Imageing Report and Data System*), tipo de anormalidade, entre outros, estão disponíveis no laudo mamográfico que acompanha cada imagem [18].

### B. *Transformada* wavelet

Para a extração de características foi, no primeiro momento, aplicado a transformada *wavelet* (TW) diretamente sobre as ROI's. A TW é uma técnica utilizada para se obter a representação de um sinal nos domínios da frequência e do tempo. A técnica executa o produto interno de um sinal $f(t)$ com uma função *wavelet-mãe* escalonada e transladada no eixo temporal $\psi_{a,b}(t)$. Ela é representada pela equação 1:

$$Wf(a,b) = \int_{-\infty}^{\infty} f(t)\psi_{a,b}(t)dt \quad (1)$$

onde, $a$ representa a escala (dilatação) e $b$ representa a transladação (deslocamento no tempo) da *wavelet-mãe* $\psi_{a,b}(t)$ definida pela equação 2:

$$\psi_{a,b}(t) = \frac{1}{\sqrt{a}}\psi(\frac{x-b}{a}) \quad (2)$$

Existem duas variações da TW, uma delas é a Transformada *Wavelet* Contínua (TWC) e a outra é a Transformada *Wavelet* Discreta (TWD). A TWC é utilizada para dividir uma função de tempo contínuo em *wavelets*. Com isto se consegue uma representação do tempo e do espaço desta função contínua. De acordo com [19], a utilização da TWC produz uma análise redundante e com muitas informações desnecessárias para o processamento de uma imagem. A TWD consegue proporcionar uma descrição mais eficiente para o processamento da mesma.

Em [20] foi demonstrado uma forma eficaz de se desenvolver a TWD. Esta solução baseia-se na utilização do *Quadrature Mirrors Filters* (QFM) e de operações de amostragem. O QMF é um banco de filtros que divide o sinal de entrada em bandas distintas, neste caso, formado pelo filtro passa-baixa e passa-alta, os quais foram as bases para essa transformada [21], [22].

Nos estudos de filtros aplicados a TWD existem duas importantes operações: a decomposição e a recomposição. A decomposição é o processo onde se descarta termos de uma sequência cujo o índice não seja múltiplo de um número $q$ e é representada pela equação 3:

$$(\downarrow q)(\overrightarrow{u}) = (\overrightarrow{u}_{nq}) \quad (3)$$

Geralmente, a aplicação da decomposição em uma TWD tem como valor da variável q = 2, pois permite eliminar os

elementos alternados de uma sequência. Na recomposição são adicionados valores 0 intercalados dentro de uma sequência de termos no sistema, conforme a equação 4:

$$(\uparrow q)(\overrightarrow{u}) = \begin{cases} u(k), & \text{se} \quad n = 2k, \\ 0, & \text{se} \quad n = 2k+1 \end{cases} \tag{4}$$

A relação entre o cálculo dos coeficientes *wavelet* e os bancos de filtros é demonstrada a partir de duas funções unidimensionais: a função escala e a função *wavelet-mãe*. A função escala relaciona a transformada ao aspecto de multirresolução, onde é definida pela equação 5:

$$\phi(x) = \sqrt{2} \sum_{k} l_k \phi(2x - k) \tag{5}$$

sendo, $l_k$ é o coeficiente de escala.

A *wavelet-mãe* expande as diferenças entre as resoluções, sendo definida pela equação 6:

$$\psi(x) = \sqrt{2} \sum_{k} h_k \phi(2x - k) \tag{6}$$

onde, $h_k$ é o coeficiente de detalhe.

A Figura 2 representa a TWD aplicada em uma ROI de uma imagem mamográfica com lesão. A *wavelet-mãe* utilizada nesse exemplo foi a Haar.

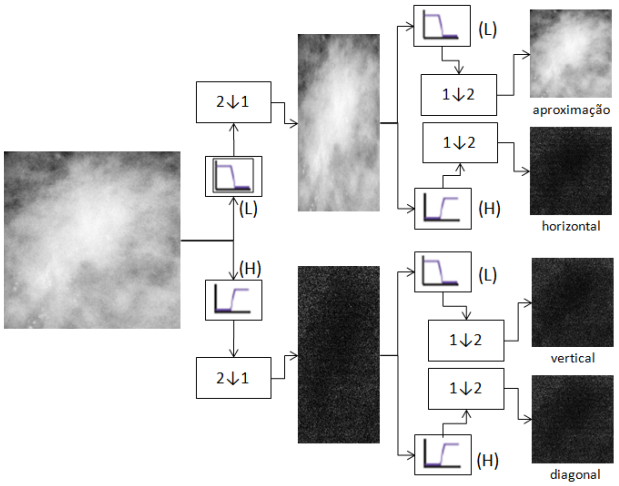



Figura 2. Transformada *wavelet* em uma ROI de uma imagem mamográfica. $(2 \downarrow 1)$ é a decomposição nas colunas e $(1 \downarrow 2)$ é a decomposição nas linhas.

O resultado desta operação apresenta quatro coeficientes, um de aproximação e outros três de detalhes em diferentes orientações: horizontal, vertical e diagonal. Para obter mais detalhes do sinal, é possível realizar a decomposição em mais níveis. Para isso, aplica-se o mesmo processo da TWD no coeficiente de aproximação resultante. A cada nível de decomposição quatro novos coeficientes são gerados, cada um contendo apenas metade da frequência de banda do sinal original.

Nesta etapa do estudo, as ROI's foram decompostas em dois níveis pela TWD a fim de se obter informações no domínio espacial e no domínio da frequência das mesmas. A função *wavelet-mãe* Daubechies 8 foi aplicada no processo de obtenção dos coeficientes *wavelets*. Essa função foi escolhida devido ao fato de ser ortogonal, de suporte compacto e diferenciável. Além disso, essa função já demonstrou bons resultados nos estudos [23], [24]. As análises seguintes foram feitas com todos coeficientes de detalhes resultantes desse processo. Um total de 245.760 coeficientes foram gerados para cada imagem (256x256x3 para os coeficientes de detalhe da primeira decomposição mais 128x128x3 para os coeficientes de detalhe da segunda decomposição).

### C. Local Binary Pattern *(LBP)*

Após obter os coeficientes de detalhe, os operadores de padrões binários locais (*Local Binary Pattern* - LBP), proposto por Wang e He em 1990 [25], foram aplicados para calcular o código LBP sobre os coeficientes *wavelet*.

O LBP é uma técnica de extração de características não morfológicas de abordagem estática. Ela elabora a extração de texturas de uma determinada imagem de forma robusta, exata e com baixa complexidade computacional. Seu funcionamento se baseia em percorrer cada *pixel* de uma imagem, fazendo uma análise que leva em consideração a diferença entre os níveis de cinza deste *pixel* com os seus *pixels* vizinhos e com isto obtém-se o código local daquele *pixel*. O valor LBP para o *pixel* central $(x_c, y_c)$ é dada pela equação 7:

$$LBP_{(P,R)} = \sum_{p=0}^{P-1} s(g_p - g_c) 2^p \tag{7}$$

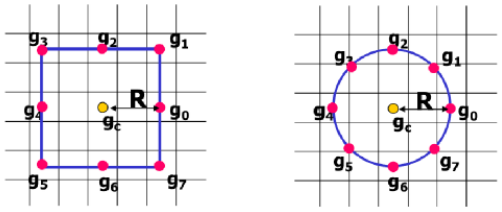
$$s(x) = \begin{cases} 0, & \text{se} \quad x < 0 \\ 1, & \text{se} \quad x \geqslant 0 \end{cases} \tag{8}$$

onde, $g_c$ é o valor de cinza do *pixel* central, $g_p$ é o valor de seus vizinhos, $P$ representa o número de vizinhos e $R$ representa o raio.

Caso a coordenada do *pixel* central $g_c$ seja $(0, 0)$, então as coordenadas de $g_p$ são dadas pela equação 9:

$$g_p = (x_p, y_p) = (R\cos(\frac{2\pi p}{P}), R\sin(\frac{2\pi p}{P})) \tag{9}$$

Quando um *pixel* estiver alinhado com o *pixel* central, o operador LBP é calculado conforme a figura 3(a). Caso contrário, o valor é calculado por meio de interpolação bilinear, como demonstra a Figura 3(b) [26].




(a) Quadrada (b) Circular

Figura 3. Obtenção de informações para vizinhança de 8 *pixels*.

Ao final de todo o procedimento, a extração de característica resultará em histograma de números binários que descrevem a textura da imagem.

Nesta etapa do trabalho, a técnica LBP foi aplicada de duas maneiras diferentes. O valor da variável de raio foi alterada a fim de se comparar os resultados obtidos. As configurações da LBP utilizadas foram:

- *Local Binary Pattern* com raio 1 e o número de vizinhos = 8
- *Local Binary Pattern* com raio 2 e o número de vizinhos = 8

### *D.* Completed Local Binary Pattern (CLBP)

Com a proposta de avaliação das técnicas de extração de características, o operador de Padrão Binário Local Completo (*Completed Local Binary Pattern* - CLBP) foi aplicado nos coeficientes *wavelets*. A CLBP é uma variação da LBP proposta em 2010 por Guo, Zhang e Zhang [16]. A função desse operador não é apenas analisar a informação do *pixel* central em relação a seus vizinhos, mas também a magnitude dessa distância e a intensidade do *pixel* central em relação a imagem completa. A utilização da magnitude desta distância é feita com a finalidade de tornar o processo mais discriminante. Esse operador é representado pela equação 10 :

$$m_p = |g_p - g_c| \tag{10}$$

O processo da CLBP retorna três operadores, que são:

- $CLBP_S$: considera o componente de sinal da diferença;
- $CLBP_M$: considera o componente de magnitude $m_p$;
- $CLBP_C$: considera a intensidade do *pixel* central $g_c$ em relação a imagem.

O operador $CLBP_S$ representa a operação LBP convencional. No caso, o operador $CLBP_M$ é definido de acordo com a equação 11:

$$CLBP_{M_{(P,R)}} = \sum_{p=0}^{P-1} s(m_p - c)2^p \tag{11}$$

onde, $s(x)$ está definido na equação 8 e $c$ representa a média dos valores absolutos das diferenças entre um pixel e seu vizinho. Por último, o operador $CLBP_C$ pode ser definido pela equação 12:

$$CLBP_{c_{P,R}} = s(g_c - \tau_1) \tag{12}$$

onde, $s(x)$ está definido na esquação 8 e $\tau_1$ representa a média dos valores de nível de cinza de todos os *pixels* da imagem.

Após obter os três operadores da CLBP, os mesmos são combinados formando um mapa de características que representa a imagem. A técnica CLBP foi aplicada nas subimagens dos coeficientes de duas maneiras diferentes. O valor da variável de raio foi alterada a fim de se comparar os resultados obtidos. As configurações da CLBP utilizadas foram:

- *Completed Local Binary Pattern* com raio 1 e o número de vizinhos = 8
- *Completed Local Binary Pattern* com raio 2 e o número de vizinhos = 8

### *E. Análise de variância*

Em seguida, para verificar a relevância dos atributos obtidos após a aplicações dos operadores LBP e CLBP a técnica de análise de variância foi aplicado.

A Análise de Variância (*Analysis of Variance* - ANOVA) é uma técnica que visa verificar se existe uma diferença entre as médias e se os fatores alteram em alguma variável dependente. Isto faz com que vários grupos sejam comparados simultaneamente. Esses fatores podem ser de origem qualitativa ou quantitativa, mas a variável dependente deverá necessariamente ser contínua. O teste é paramétrico, ou seja, a variável de interesse deve ter distribuição normal e os grupos tem que ser independentes. O método ANOVA utiliza um quadro de análise de variância, conforme apresentado na Tabela I. A coluna *"Fonte de Variação"* dessa tabela separa a variabilidade dos descritores, para amostras provindas de imagens diferentes, e a variabilidade residual, para amostras provindas ao acaso. A coluna *"Q"* representa a soma das variações, as colunas *"G.L."* e *"$S^2$"* representam, respectivamente, os graus de liberdade e os quadrados médios utilizados nos cálculos sobre os descritores e resíduos. Por último, a coluna *"Teste F"* representa a razão entre quadrado médio dos descritores sobre o quadrado médio dos resíduos.

Tabela I. QUADRO DE ANÁLISE DE VARIÂNCIA.

| Fonte de Variação | Q | G.L. | $S^2$ | Teste F |
|---|---|---|---|---|
| Descritores | $Q_e$ | $k-1$ | $S_e^2 = \frac{Q_e}{k-1}$ | $F_{calc} = \frac{S_e^2}{S_r^2}$ |
| Resíduos | $Q_r = Q_t - Q_e$ | $n-1$ | $S_r^2 = \frac{Q_r}{n-1}$ | |
| Total | $Q_t$ | $n-1$ | | |

As etapas de preenchimento do quadro e maiores detalhes podem ser vistos em [27]. Nesta etapa do estudo, foi aplicada a técnica ANOVA com o índice de significância igual a 5% sobre os resultados obtidos dos operadores LBP e CLBP.

### *F. Classificação*

A avaliação da extração de características necessita de um classificador para demonstrar a eficácia da técnica abordada. Para esta etapa, dois classificadores foram utilizados com intuito de fazer uma comparação entre os seus resultados. O primeiro classificador utilizado foi o *Random Forest*. Ele tem sua definição dada por um classificador de árvores de classificação. Seu conceito baseia-se em árvores de regressão, induzido por um método de amostragem com repetição, bootstrap, de um conjunto de dados de treinamento, usando descritores aleatórios selecionados no processo de indução de árvore. O processo de criação da árvore pode ser visto em [11].

O Segundo classificador utilizado foi o *Support Vector Machines-Sequential Minimal Optimization* (SVM-SMO). Sua técnica de aprendizagem de máquina é baseada na teoria de aprendizagem estatística. Esta teoria encontra um hiperplano com máxima separação entre as classes, assumindo que os dados são linearmente separáveis. Se este não for o caso, então o SVM mapeia os dados utilizando uma função *Kernel* em um espaço de características de dimensão mais elevada onde os dados se tornam linearmente separáveis [28]. Ambos os classificadores foram executados na plataforma WEKA (*Waikato Environment for Knowledge Analysis*) [29].

Estes dois classificadores foram escolhidos com o intuito de comparar uma técnica estática e outra linear, binária e não probabilística. Além disso, estes classificados vêm sendo explorados em imagens mamográficas e tem-se destacado conforme demonstram os estudos de [29]–[31].

Nesta etapa, foi utilizado o método de rotação (*n-fold cross validation*) com valor de $n = 10$. Este método utiliza 90% dos dados para efetuar o treinamento e os outros 10% para testar. Foram avaliados grupos das diferentes visões da mamografia, CC e MLO, e os seus diferentes tipos de anormalidades. A proposta foi avaliar o comportamento e influência de cada visão da imagem mamográfica em relação às técnicas de análise textural propostas. Os grupos investigados foram:

- Comparação de mamografias com anormalidades benignas e mamografias com anormalidade malignas (visões CC e MLO separadamente).
- Mamografias com anormalidades benignas contra mamografias sem anormalidade (visões CC e MLO separadamente).
- Mamografias sem anormalidades contra mamografias com anormalidade malignas (visões CC e MLO separadamente).

Avaliações quantitativas foram realizadas por meio de métricas como acurácia e área sobre a curva ROC *(Receiver Operating Characteristic)*. A curva ROC exibe as relações entre taxas de acertos e erros para cada grupo analisado, demonstrando assim o desempenho obtido em cada algoritmo proposto. O objetivo deste trabalho foi analisar a eficiência dos operadores LBP e CLBP na classificação das imagens.

## III. Resultados Experimentais

### A. Experimentos com operador LBP

Os resultados apresentados na Tabela II, são referentes a utilização do operador LBP, com sua variável de raio trocada, nas subbandas de imagens *wavelet* classificadas com os algoritmos RaF e SVM. Os resultados são apresentados de acordo com cada grupo de lesão e visão da mamografia. Neste experimento, podemos notar que o classificador SVM obteve um melhor desempenho em todos os casos, chegando a 84,17% de AUC com um desvio padrão de 8,05%, quando utilizado o operador LBP(1,8) e o grupo a ser classificado sendo os casos Benignos e Normais de visão CC.

A alteração feita na variável de raio do operador LBP não causou diferença nos resultados, sendo melhor quando o valor atribuído foi de "1". Para o classificador RaF, somente os grupos Benigna e Maligna com visão MLO, e Normal e Maligna com visão MLO obtiveram resultados melhores quando utilizado o operador LBP(2,8). Com o classificador SVM, somente os grupos Benigna e Normal com visão MLO, e Normal e Maligna com visão CC obtiveram resultados melhores com o operador LBP(2,8).

O grupo Benigna e Normal foi o que obteve uma melhor classificação em ambas as visões e em ambos classificadores, sendo menor apenas quando utilizado o operador LBP(1,8) e o classificador SVM nos casos de visão MLO, onde, para esta configuração, o grupo Benigna e Maligna foi mais eficaz. Em relação à comparação das visões mamográficas, os resultados obtidos foram mais eficazes, em sua maioria, nas imagens de visão MLO.

### B. Experimentos com operador CLBP

Os resultados apresentados na Tabela III, são referentes a utilização do operador CLBP, com sua variável de raio trocada, nas subbandas de imagens *wavelet* classificadas com os algoritmos RaF e SVM. Os resultados são apresentados de acordo com cada grupo de lesão e visão da mamografia. Neste experimento, podemos notar que o classificador SVM foi muito eficaz, tendo um índice de acerto de AUC = 100,00% para todos os experimentos realizados.

A alteração realizada na variável de raio do operador CLBP não gerou diferença nenhuma com o classificador SVM. Já com o RaF, os melhores resultados foram obtidos com a variável de raio igual a "2", somente nos grupos Benigna e Normal com visão CC e Normal e Maligna com visão CC a CLBP(1,8) demonstrou-se mais eficaz. Quando comparada as visões das imagens mamográficas, as técnicas aplicadas nas visões MLO obtiveram um melhor resultado com o classificador RaF.

Em relação aos operadores utilizados, notamos uma grande diferença nas classificações quando comparado o classificador LBP e CLBP, onde o CLBP demonstrou-se mais eficaz chegando a 100% de acerto em todos os casos quando utilizado o classificador SVM. Foi possível notar também que o classificador SVM foi mais eficaz em todos os resultados quando comparado ao RaF. Por fim, a visão da imagem mamográfica que obteve um melhor resultado ao ser analisada foi a MLO.

## IV. Conclusão

No trabalho foram utilizadas a *wavelet-mãe* Daubechies 8, os extratores de características texturais LBP e CLBP, ANOVA e os classificadores *Support Vector Machines* e *Random Forest* na avaliação de lesões em imagens mamográficas. Foi possível observar que a utilização de técnicas de extração de características tem um ótimo resultado quando utilizada nos coeficientes de detalhe obtidos através da transformada *wavelet* em conjunto com os descritores LBP e CLBP. Alguns experimentos resultaram em 100,00% de AUC quando utilizado o operador CLBP e o classificador *Support Vector Machines*. Isso permite afirmar que a técnica CLPB foi mais eficaz para classificação das lesões quando comparada com a LBP. Em trabalhos futuros, pretende-se avaliar outras técnicas para classificação como, por exemplo, a combinação do SVM com o algoritmo *Random Forest*.

## Referências


[1] J. Tang, R. Rangayyan, J. Xu, I. El Naqa, and Y. Yang, "Computer-aided detection and diagnosis of breast cancer with mammography: Recent advances," *Information Technology in Biomedicine, IEEE Transactions on*, vol. 13, no. 2, pp. 236 –251, 3 2009.

[2] R. Rangayyan, S. Prajna, F. Ayres, and J. Desautels, "Detection of architectural distortion in prior screening mammograms using gabor filters, phase portraits, fractal dimension, and texture analysis," *International Journal of Computer Assisted Radiology and Surgery*, vol. 2, pp. 347–361, 2008.

[3] S. V. Destounis, P. DiNitto, W. Logan-Young, E. Bonaccio, M. L. Zuley, and K. M. Willison, "Can computer-aided detection with double reading of screening mammograms help decrease the false-negative rate? initial experience." *Radiology*, vol. 232, no. 2, p. 578, 8 2004.

Tabela II. RESULTADOS OBTIDOS ATRAVÉS DOS OPERADORES LBP(1,8) E LBP(2,8) APLICADOS NOS COEFICIENTES DE DETALHE DA TW COM *wavelet-mãe* DAUBECHIES 8.

| | | LBP(1,8) | | | | LBP(2,8) | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | RaF | | SVM | | RaF | | SVM | |
| Grupo Classificação | Visão | AUC | $\sigma$ | AUC | $\sigma$ | AUC | $\sigma$ | AUC | $\sigma$ |
| Benigna x Maligna | MLO | 72,62% | 18,47% | 82,51% | 11,24% | 73,31% | 15,78% | 80,84% | 13,35% |
| Benigna x Normal | | 78,92% | 14,70% | 80,42% | 12,41% | 76,46% | 16,70% | 84,17% | 8,50% |
| Normal x Maligna | | 72,08% | 23,36% | 78,34% | 12,23% | 72,57% | 20,19% | 78,33% | 14,81% |
| Benigna x Maligna | CC | 69,61% | 17,66% | 79,17% | 11,62% | 65,94% | 21,30% | 72,93% | 18,97% |
| Benigna x Normal | | 77,22% | 19,10% | 84,17% | 8,05% | 71,63% | 18,72% | 79,17% | 11,27% |
| Normal x Maligna | | 69,24% | 24,15% | 75,83% | 14,13% | 68,08% | 24,12% | 79,17% | 14,30% |

Tabela III. RESULTADOS OBTIDOS ATRAVÉS DO OPERADORES CLBP(1,8) E CLBP(2,8) APLICADOS NOS COEFICIENTES DE DETALHE DA TW COM *wavelet-mãe* DAUBECHIES 8.

| | | CLBP(1,8) | | | | CLBP(2,8) | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | RaF | | SVM | | RaF | | SVM | |
| Grupo Classificação | Visão | AUC | $\sigma$ | AUC | $\sigma$ | AUC | $\sigma$ | AUC | $\sigma$ |
| Benigna x Maligna | MLO | 83,86% | 10,83% | 100,00% | 0,00% | 85,14% | 12,94% | 100,00% | 0,00% |
| Benigna x Normal | | 81,96% | 10,59% | 100,00% | 0,00% | 85,04% | 10,79% | 100,00% | 0,00% |
| Normal x Maligna | | 81,41% | 16,34% | 100,00% | 0,00% | 86,52% | 11,24% | 100,00% | 0,00% |
| Benigna x Maligna | CC | 80,24% | 13,59% | 100,00% | 0,00% | 84,74% | 14,10% | 100,00% | 0,00% |
| Benigna x Normal | | 83,79% | 10,81% | 100,00% | 0,00% | 79,38% | 12,10% | 100,00% | 0,00% |
| Normal x Maligna | | 80,59% | 15,90% | 100,00% | 0,00% | 80,53% | 14,67% | 100,00% | 0,00% |


[4] X. Llad Oliver, J. Freixenet, R. Martnd J. Mart'A textural approach for mass false positive reduction in mammography," *Computerized Medical Imaging and Graphics*, vol. 33, no. 6, pp. 415–422, 09 2009.

[5] R. de Souza Jacomini, M. do Nascimento, R. Dantas, and R. Ramos, "Comparison of pca and anova for information selection of cc and mlo views in classification of mammograms," in *Intelligent Data Engineering and Automated Learning - IDEAL 2012*, ser. Lecture Notes in Computer Science, H. Yin, J. Costa, and G. Barreto, Eds. Springer Berlin / Heidelberg, 2012, vol. 7435, pp. 117–126.

[6] L. Nanni, S. Brahnam, and A. Lumini, "A very high performing system to discriminate tissues in mammograms as benign and malignant," *Expert Syst. Appl.*, vol. 39, no. 2, pp. 1968–1971, 2 2012.

[7] X. Liu, X. You, and Y. ming Cheung, "Texture image retrieval using non-separable wavelets and local binary patterns," in *Computational Intelligence and Security, 2009. CIS '09. International Conference on*, vol. 1, 12 2009, pp. 287–291.

[8] F. Bianconi and A. Fernez, "On the occurrence probability of local binary patterns: A theoretical study," *Journal of Mathematical Imaging and Vision*, vol. 40, pp. 259–268, 2011.

[9] R. Dantas, M. Nascimento, R. Ramos, and D. Pereira, "Análise das variações da matriz de co-corrência em imagens derivadas da transformada wavelet haar em mamográfia," *XIV Congresso Brasileiro de Física Médica*, 2009.

[10] R. D. Dantas, M. Z. Nascimento, R. S. Jacomini, D. C. Pereira, and R. P. RAMOS, "Fusion of two-view information: S v d based modeling for computerized classification of breast lesions on mammograms," in *Mammography - Recent Advances*, N. Uchiyama and M. Z. do Nascimento, Eds. Intech, March 2012, pp. 261–278.

[11] R. P. Ramos, M. Z. do Nascimento, and D. C. Pereira, "Texture extraction: An evaluation of ridgelet, wavelet and co-occurrence based methods applied to mammograms," *Expert Systems with Applications*, vol. 39, no. 12, pp. 11 036 – 11 047, 2012.

[12] L. Du, X. You, H. Xu, Z. Gao, and Y. Tang, "Wavelet domain local binary pattern features for writer identification," in *Pattern Recognition (ICPR), 2010 20th International Conference on*, aug. 2010, pp. 3691 –3694.

[13] D. Huang, C. Shan, M. Ardabilian, Y. Wang, and L. Chen, "Local binary patterns and its application to facial image analysis: A survey," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 41, no. 6, pp. 765 –781, nov. 2011.

[14] B. C. Ko, S. H. Kim, and J. Y. Nam, "X-ray image classification using random forests with local wavelet-based cs-local binary patterns." *J Digit Imaging*, vol. 24, no. 6, pp. 1141–1151, 12 2011.

[15] L. Nanni, A. Lumini, and S. Brahnam, "Survey on lbp based texture descriptors for image classification," *Expert Syst. Appl.*, vol. 39, no. 3, pp. 3634–3641, 2 2012.

[16] Z. Guo, D. Zhang, and D. Zhang, "A completed modeling of local binary pattern operator for texture classification," *Image Processing, IEEE Transactions on*, vol. 19, no. 6, pp. 1657–1663, 2010.

[17] E. Azevedo and A. Conci, *Computa Grca: Teoria e Prca*, 11th ed., E. Campus, Ed. [S.1.]: Elsevier, Rio de Janeiro, Brasil, 2008, vol. 1.

[18] C. Balleyguier, S. Ayadi, K. Vannguyen, D. Vanel, C. Dromain, and R. Sigal, "Birads classification in mammography," *European Journal of Radiology*, vol. 61, no. 2, pp. 192–194, 02 2007.

[19] H. M. de Oliveira, *Anse de Sinais para Engenheiros: uma abordagem via Wavelets.* Rio de Janeiro: Brasport, 2007.

[20] S. Mallat, *A Wavelet Tour of Signal Processing*, Second, Ed. Academic Press, San Diego, CA, 1999.

[21] M. Misiti, Y. Misiti, G. Oppenheim, and J. Poggi, *Matlab Wavelet Toolbox - User's Guide*, S. s.n], Ed., 2008.

[22] H. Pedrini and W. Schwartz, *Anse de imagens digitais: princos algoritmos e aplicas*. Saulo: Thomson Learning, 2008.

[23] E. A. Rashed, I. A. Ismail, and S. I. Zaki, "Multiresolution mammogram analysis in multilevel decomposition," *Pattern Recognition Letters*, vol. 28, no. 2, pp. 286 – 292, 2007.

[24] M. M. Eltoukhy, I. Faye, and B. B. Samir, "A comparison of wavelet and curvelet for breast cancer diagnosis in digital mammogram," *Computer in Biology and Medicine*, vol. 40, no. 4, pp. 384–391, 2010.

[25] L. Wang and D. He, "Texture classification using texture spectrum," *Pattern Recognition, Elsevier*, vol. 23, no. 8, pp. 905–910, 1990.

[26] V. Takala, T. Ahonen, and M. Pietikäinen, "Block-Based Methods for Image Retrieval Using Local Binary Patterns," in *Image Analysis*. Berlin, Heidelberg: Springer Berlin / Heidelberg, 2005, vol. 3540, ch. 89, pp. 882–891.

[27] J. D. Fonseca, *Curso de estatica*, [S.1], Ed. Atlas, 1981.

[28] H. Osta, R. Qahwaji, and S. Ipson, "Comparisons of feature selection methods using discrete wavelet transforms and support vector machines for mammogram images," in *Systems, Signals and Devices, 2008. IEEE SSD 2008. 5th International Multi-Conference on*, july 2008, pp. 1 –6.

[29] L. Vibha, G. Harshavardhan, K. Pranaw, P. Shenoy, K. Venugopal, and L. Patnaik, "Classification of mammograms using decision trees," in *Database Engineering and Applications Symposium, 2006. IDEAS'06. 10th International*. IEEE, 2006, pp. 263–266.

[30] G. B. Junior, A. C. de Paiva, A. C. Silva, and A. C. M. de Oliveira, "Classification of breast tissues using moran's index and geary's coefficient as texture signatures and svm," *Computers in Biology and Medicine*, vol. 39, no. 12, pp. 1063 – 1072, 2009.

[31] J. Liu, X. Liu, J. Chen, and J. Tang, "Improved local binary patterns for classification of masses using mammography," in *Systems, Man, and Cybernetics (SMC), 2011 IEEE International Conference on*, oct. 2011, pp. 2692 –2695.

# Traffic Sign Detection and Recognition using the AdaBoost and SIFT algorithm

Francisco A. da Silva[1,2], Antônio C. P. Nascimento[1], Maria Stela Veludo de Paiva[2]
[1]Universidade do Oeste Paulista – Unoeste/FIPP
Presidente Prudente/SP, Brasil
[2]Universidade de São Paulo – USP/EESC
São Carlos/SP, Brasil
*chico@unoeste.br, acpn@unoeste.edu.br, mstela@sc.usp.br*

Almir Olivette Artero[3], Marco Antonio Piteri[3], Ricardo Luiz Barbosa[4]
[3]Universidade Estadual Paulista – Unesp/FCT
Presidente Prudente/SP, Brasil
[4]Universidade Estadual Paulista – Unesp
Campus de Sorocaba/SP, Brasil
{*almir, piteri*}*@fct.unesp.br, rluisbarbosa@usp.br*

***Abstract*—This paper presents the algorithm TSRAS (Traffic Sign Recognition using AdaBoost and SIFT) proposed for traffic sign detection and recognition using images from videos. Initially, the AdaBoost algorithm is used to detect the plates from a specific training for traffic signals, without considering all the information printed on the plates, but only the basic information, such as red contours, the presence or not of diagonal lines represent prohibition, yellow or red background. Different from several studies in the literature, this strategy reduces the amount of different plates needed for AdaBoost training. In addition, the HSV color space was used to eliminate regions incorrectly detected by AdaBoost. In the next, the algorithm proposed by Silva et al. [2] that uses the SIFT algorithm to extract keypoints in the candidate areas was applied. Finally, we performed the text recognition for the speed plates. The processing time of this new approach was 1/8 over the previous approach.**

***Keywords—traffic sign detection; traffic sign recognition; AdaBoost; SIFT***

## I. INTRODUÇÃO

O reconhecimento automático de placas de trânsito está se tornando uma área com muitas aplicações na indústria automotiva [3], por exemplo, possibilitando que um motorista seja avisado sobre ações impróprias e situações potencialmente perigosas, o que consiste em uma tarefa essencial para um sistema inteligente de auxílio a motoristas [4]. De fato, os textos e símbolos inseridos em um sinal de trânsito, geralmente, apresentam informação útil, como limite de velocidade, direção da via, e situações de trânsito, que ajudam os motoristas a dirigir com maior segurança [5].

Este trabalho apresenta uma metodologia para realizar a detecção e o reconhecimento de sinais de trânsito, a partir de quadros de vídeo, com potencial para operar em tempo real, pois utiliza o algoritmo AdaBoost [18], que tem sido muito usado para o reconhecimento de faces, para selecionar áreas candidatas contendo imagens de placas de sinalização de trânsito. Os sinais de trânsito tratados neste trabalho são as placas de sinalização afixadas em calçadas de ruas e avenidas.

As demais seções deste trabalho estão organizadas da seguinte maneira: Na Seção II são apresentados alguns trabalhos relacionados ao reconhecimento de sinais de trânsito; Na Seção III é apresentada uma breve revisão dos algoritmos usados; Na Seção IV é apresentado o algoritmo TSRAS, proposto neste trabalho, que realiza a detecção e o reconhecimento das placas; Na Seção V são apresentados os resultados obtidos com a aplicação da metodologia proposta em uma sequência de quadros de vídeo, além de apresentar uma comparação com um trabalho anterior, que usa um conjunto de imagens de alta resolução; Por fim, na Seção VI são apresentadas as conclusões e trabalhos futuros.

## II. TRABALHOS RELACIONADOS

Ao longo dos últimos anos foram propostas diversas abordagens para resolver o problema do reconhecimento de sinais de trânsito, como por exemplo: utilizando propriedades de cor dos sinais [6]; a partir da geometria das placas em imagens em tons de cinza [7]; baseada nas propriedades de forma dos sinais [8]; baseada em propriedades de cor e forma [9]; utilizando classificadores baseados em redes neurais [3]; analisando a curvatura da representação do sinal com contornos ativos (Snakes) [10]; usando *Support Vector Machine* (SVM) [11]; usando uma segmentação de sinais de trânsito baseado em conjuntos fuzzy [12].

Segundo Chen e Hsieh [5], é difícil detectar os sinais de trânsito diretamente a partir de vídeos, devido a diferentes alterações de condições ambientais (nebulosidade, iluminação, sombras, etc), o que ocasiona sinais de trânsito com diferentes mudanças na sua aparência. Além disso, para a câmera montada na frente de um carro em movimento, seus efeitos de perspectiva fazem o sinal de trânsito ter diferentes tamanhos, formas, contrastes e borramentos (ocasionado pelo deslocamento do carro). Em alguns casos, a placa com o sinal de trânsito pode ficar totalmente ou parcialmente encoberta por outros objetos como árvores, veículos, etc.

O trabalho de Silva et al. [2] apresenta como proposta para o reconhecimento de placas de sinalização de trânsito, o uso do algoritmo SIFT, para detectar pontos chave nas imagens e, em seguida, compara os descritores destes pontos com os descritores obtidos em templates contendo os sinais de trânsito. Por fim, aplica um algoritmo de reconhecimento de caracteres para reconhecer textos, tais como os limites de velocidade. Este algoritmo apresenta bons resultados, entretanto, como precisa aplicar o algoritmo SIFT na imagem inteira, a sua aplicação em tempo real se torna impraticável. De fato, este é o maior

diferencial da abordagem proposta neste trabalho, que sugere aplicar o algoritmo AdaBoost na imagem inteira e, então, usar o SIFT apenas nas regiões candidatas, detectadas pelo AdaBoost. A vantagem desta proposta é que o AdaBoost possui um tempo de processamento muito pequeno, sendo indicado para aplicações em tempo real.

## III. REFERENCIAL TEÓRICO

O detector de Viola e Jones [1] baseia-se em uma cascata de classificadores boosted Haar-like, que combina dois conceitos: (i) classificadores Haar-like e (ii) AdaBoost [18]. Os classificadores Haar-like são construídos com características retangulares simples, que representam as diferenças de somas de pixels específicos em uma imagem. Cada característica é emparelhada com um limiar, e a decisão do classificador é determinada por uma comparação do valor da característica com o limiar [13]. Esta simplicidade constitui um diferencial importante para minimizar o tempo de processamento. O algoritmo AdaBoost usa pesos em uma combinação linear de classificadores fracos, formando um classificador forte [5].

Viola e Jones seguem nesta linha, agrupando vários classificadores fortes, construídos por AdaBoost, em uma cascata que apresenta um processamento muito rápido. O classificador forte na primeira fase da cascata é escolhido de modo a descartar falsos positivos e, ao mesmo tempo, preservar quase todos os verdadeiros positivos do conjunto de treino. O processo de detecção de faces é realizado por uma janela deslizante de detecção através da imagem. Dentro da janela, a resposta da cascata é calculada. Depois de completar uma passagem sobre a imagem, o tamanho da janela de detecção é aumentado por algum fator (1.2 no OpenCV, significando que a escala da janela será aumentada em 20%). O tamanho da janela é aumentado até um tamanho pré-definido ser alcançado. Aumentando a janela de detecção por uma percentagem menor produz as melhores taxas de detecção, mas aumenta o número de falsos positivos e também o tempo total de processamento [13].

O sistema de detecção de faces proposto por Viola e Jones [1] utiliza características simples para classificar as imagens, baseadas nas funções de Haar. As características correspondem basicamente a retângulos que possuem duas regiões, sendo uma clara e outra escura. O cálculo do valor correspondente à característica é feito subtraindo o somatório da intensidade dos pixels da região escura do total acumulado pela intensidade dos pixels da região clara. Este cálculo pode ser feito rapidamente utilizando a representação de imagem integral. Na Fig. 1 são apresentadas as características básicas de Haar que foram utilizadas por Viola e Jones [1].

Fig. 1. Características Básicas de Haar.

Com o intuito de melhorar ainda mais a etapa de detecção de objetos, Lienhart e Maydt [20] estenderam a quantidade de tipos possíveis de características retangulares, acrescentando características às previamente sugeridas, mas desta vez rotacionadas em 45 graus. Com estas novas adições ao conjunto, consegue-se reduzir o número de falsos positivos em torno de 10%, aumentando ainda mais a eficiência do processo de detecção. Na Fig. 2 é apresentado o conjunto completo com as novas características retangulares sugeridas.

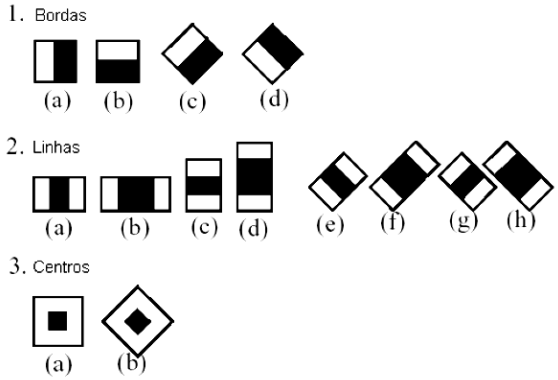



Fig. 2. Características Estendidas. Fonte: [20].

### A. *Algoritmo AdaBoost*

Boosting é um método usado para aprimorar o desempenho de um algoritmo de aprendizado, não sendo empregado de forma isolada, mas sendo aplicado de forma combinada com outras técnicas, como Árvores de Decisão e Redes Neurais. O objetivo é transformar classificadores "fracos" em "fortes", o que é feito combinando classificadores gerados por um mesmo algoritmo de aprendizado (algoritmo base ou fraco) e ajustando de acordo com os erros cometidos pelo classificador anterior. Deste modo, busca-se gerar um classificador melhor, com uma eficiência maior do que as obtidas pelos classificadores isolados [22]. O modelo Boosting, compreende uma família de algoritmos, dentre os quais o algoritmo Adaptive Boosting (AdaBoost) é o mais relevante e popular [22]. O funcionamento do algoritmo AdaBoost é descrito a seguir [21] [23]:

A primeira etapa consiste em um treinamento usando um conjunto de exemplos $S = \{(x_1, y_1), (x_2, y_2), ..., (x_m, y_m)\}$, onde:

- Cada $x_i$ representa $m$ vetores de atributos do sistema, ou seja, um conjunto de dados referentes aos parâmetros (ou propriedades) analisados;
- Dentre todos os possíveis grupos de classificação previamente definidos para o sistema analisado, cada $y_i$ representa o grupo de classificação associado ao $x_i$;
- O número total de exemplos da amostra de treinamento é igual a $m$.

O algoritmo base, que trabalha junto ao AdaBoost, fornece a cada iteração, uma distribuição de pesos referentes a cada um dos dados de treinamento. Os classificadores iniciam com pesos associados (Equação 1) igualmente distribuídos.

$$D_0 = \frac{1}{m} \quad (1)$$

Em cada ciclo de aprendizagem, o algoritmo base gera uma hipótese $h_t$, baseada nos pesos atuais, com o objetivo de priorizar a classificação correta dos dados que apresentam os

maiores pesos associados. Deste modo, o algoritmo base busca a hipótese que minimiza o erro de treinamento $e_t$.

$$e_t = \sum_{i:h_t(x_t)\neq y_t} D_t(i) \quad (2)$$

Em seguida, esses pesos são reavaliados, de forma a aumentar aqueles que são relacionados aos dados incorretamente classificados e, em seguida, uma nova iteração se inicia. Por fim, ao serem realizadas todas as $T$ iterações determinadas previamente, o AdaBoost combina todas as hipóteses intermediárias $h_t$ de modo a gerar uma única hipótese final $H(x)$. O método para se obter a hipótese final $H(x)$ é a combinação ponderada das diversas saídas das iterações. Assim, uma votação ponderada das $T$ hipóteses fracas é realizada, onde $\gamma_t$ é a importância associada à hipótese $h_t$. O esquema mostrado na Fig. 3 resume o funcionamento do algoritmo AdaBoost.

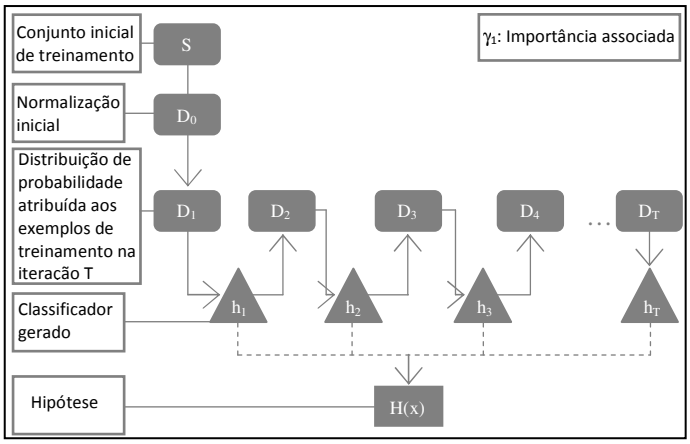



Fig. 3. Esquema do funcionamento do algoritmo AdaBoost.

Por alta velocidade de processamento e eficácia, o algoritmo AdaBoost é indicado neste trabalho para selecionar as áreas candidatas na imagem a terem uma placa de sinalização de trânsito, reduzindo as regiões a serem processadas pelo algoritmo SIFT, que apresenta tempo elevado de processamento.

## IV. TSRAS - Reconhecimento de Sinais de Trânsito usando AdaBoost e SIFT

O processo de reconhecimento de placas, geralmente, é dividido em duas etapas, que são a detecção e o reconhecimento. Neste trabalho, a detecção utiliza o algoritmo AdaBoost [1]. Apesar do AdaBoost ter sido usado originalmente na detecção de rostos, segundo Brkić [13], outras pesquisas têm aplicado com sucesso o detector em uma grande quantidade de outras classes de objetos. De fato, muitos autores já utilizam essa técnica para realizar o reconhecimento de placas de trânsito [3] [4] [5] [14] [15] [16] [17], sendo o principal diferencial deste trabalho, o uso de imagens de treinamento simplificadas, que omitem a maior parte das informações impressas nas placas, ou seja, com apenas as informações de bordas, como círculos vermelhos com ou sem as linhas diagonais que representam proibição, além de placas com fundo amarelo, ou placas com fundo vermelho. O objetivo desta simplificação é abranger a maior quantidade possível de diferentes tipos de sinais de trânsito, sem a necessidade de se usar exemplares de cada um deles durante o treinamento. Em geral, os autores se limitam a uma pequena quantidade de placas diferentes, para se obter um tempo baixo de processamento.

Após a aplicação da etapa de detecção, que deveria resultar apenas regiões das imagens que possuem uma placa de sinalização de trânsito, ainda é possível existir algumas regiões que não possuem tais elementos. Para que essas regiões sejam rapidamente eliminadas do processo, evitando o seu processamento desnecessário com o SIFT, foi utilizado o espaço de cor HSV (*Hue Saturation Value*), para se verificar as cores predominantes nessas regiões e, assim, descartar regiões detectadas incorretamente.

Em seguida, as regiões candidatas resultantes são enviadas para a etapa de reconhecimento, de modo semelhante à abordagem sugerida por Silva et al. [2], entretanto, neste trabalho, apenas as regiões candidatas a terem uma placa de trânsito são processadas pelo algoritmo SIFT, que localiza os pontos chave e seus respectivos descritores e, então, os compara com os descritores dos pontos chaves extraídos de templates. Por fim, usa o mesmo algoritmo de reconhecimento de caracteres, usados por Silva et al. [2], para reconhecer os limites de velocidades.

## V. Experimentos Realizados

Esta seção apresenta dois experimentos realizados com a proposta apresentada neste trabalho. O primeiro consistiu em aplicar o algoritmo TSRAS em um conjunto de 1.958 imagens, extraídas de um filme com nove minutos de duração, e uma taxa de captura de 29 quadros por segundo, sendo processados apenas os quadros múltiplos de oito (0, 8, 16, ..., 1.957). O segundo experimento consistiu em comparar os resultados obtidos com a abordagem proposta por Silva et al. [2] e os obtidos com o uso do algoritmo TSRAS.

### A. *Experimento I*

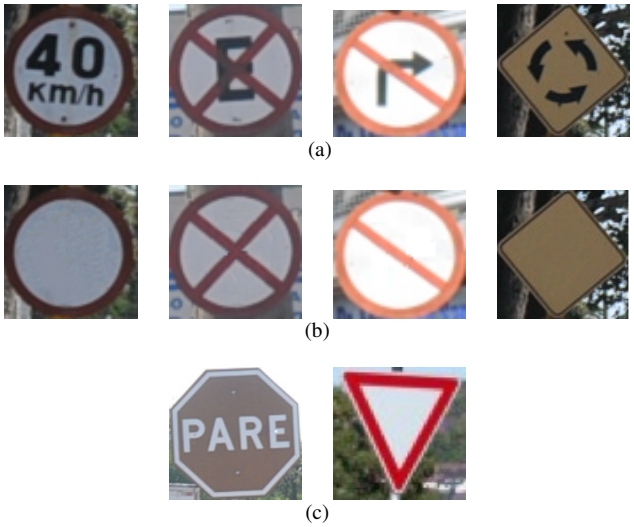
Para a realização do treinamento AdaBoost foram utilizadas 5.300 imagens positivas, ou seja, imagens que contém placas de trânsito, distribuídas em quatro classes, conforme mostra a Tabela I, que também apresenta o número de estágios usados durante o treinamento (trabalhos encontrados na literatura [1] [5] [22] sugerem de 20 à 30 estágios) e o tamanho das imagens. Também foram usadas 11.200 imagens negativas, ou seja, que não possuem placas de trânsito. As imagens foram obtidas percorrendo um trajeto de aproximadamente 4.000 metros, com uma câmera de vídeo JVC Full HD GZ-EX210BUB, fixada sobre o teto de um veículo, e transitando a uma velocidade média de 30 km/h.

TABELA I. Classes de Placas usadas no Treinamento.

| | Positivas | Negativas | Estágios | Tamanho |
|---|---|---|---|---|
| Circular | 2.892 | 6.000 | 25 | 24 x 24 |
| Alerta | 608 | 1200 | 20 | 24 x 24 |
| Pare | 900 | 2.000 | 20 | 24 x 24 |
| Preferência | 950 | 2.000 | 20 | 24 x 24 |

A Fig. 4(a) mostra alguns exemplos de imagens positivas, enquanto que em (b) tem-se as mesmas placas sem a

informação referente ao sinal de trânsito, com apenas as bordas circulares, sem ou com as linhas diagonais de proibição, e uma placa apenas com seu fundo amarelo. Essas placas sem a informação do sinal, representam exemplos de imagens que foram usadas no treinamento AdaBoost, adotado na etapa de detecção, cujo objetivo é identificar as regiões da imagem que podem ter uma placa de trânsito.




(a) (b) (c)

Fig. 4. Exemplos de imagens positivas: a) originais; b) usadas no treinamento, apenas com as informações básicas das placas; c) usadas diretamente no treinamento.

A Fig. 5 apresenta alguns exemplos de imagens negativas, usadas no treinamento, ou seja, imagens que não possuem placas de sinalização.

Fig. 5. Exemplos de imagens negativas (não possuem placas de trânsito) usadas no treinamento.

Os templates usados para se obter os pontos chave neste experimento são apresentados na Fig. 6. Estes templates foram escolhidos por serem os sinais de trânsito mais encontrados no trajeto percorrido.

Fig. 6. Templates das placas de trânsito extraídos do vídeo.

A Fig. 7 apresenta três exemplos de resultados obtidos com a metodologia apresentada neste trabalho, com o template à esquerda e o reconhecimento da placa de sinalização na imagem à direita.

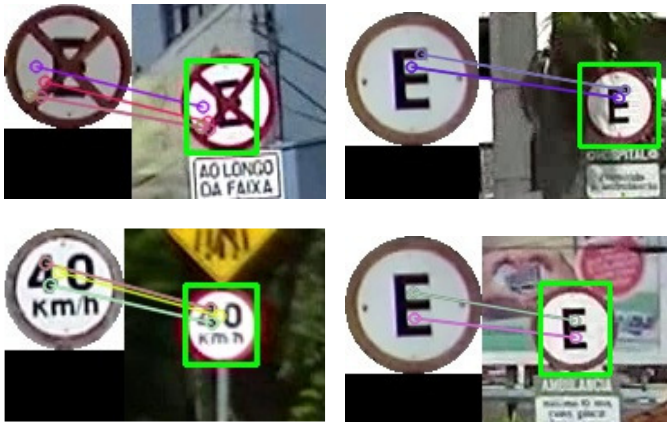



Fig. 7. Resultados do processamento do vídeo.

### B. *Experimento II*

Esta seção apresenta os resultados de um experimento com 60 imagens com resolução de 2144 x 1424 pixels, obtidas a partir de uma câmara Nikon D300S instalada sobre um veículo (Sistema de Mapeamento Móvel). As imagens foram obtidas com o veículo em movimento em condições normais de tráfego. Também é feita uma comparação com os resultados obtidos no trabalho de Silva et al. [2].

A Fig. 8 apresenta os *templates* usados neste experimento.

Fig. 8. Placas de sinais de trânsito usadas como template.

A Tabela II mostra o resultado obtido aplicando a metodologia proposta, sendo a primeira coluna referente aos índices numéricos existentes na Fig. 8.

Nas 60 imagens usadas neste experimento existem 73 placas de sinalização, de acordo com os templates apresentados na Fig. 8. Conforme apresentado na tabela, utilizando a abordagem sugerida por Silva et al. [2], foram reconhecidas corretamente 62 placas, obtendo-se uma taxa de acerto de 84.93%. Não ocorreram erros de identificação e 11 placas não foram reconhecidas (15.07%). Adotando a nova proposta apresentada neste trabalho, foram reconhecidas corretamente 58 placas, obtendo-se uma taxa de acerto de 79.45%. Ocorreram apenas 2 erros de identificação (2.74% de erros) e 13 placas não foram reconhecidas (17.81%).

Usando a abordagem sugerida por Silva et al. [2], o tempo médio de processamento destas 60 imagens em uma máquina com processador Intel Core i3 M330 2.13 GHz (RAM de 4 Giga Bytes) é de 90 segundos. Adotando a nova proposta apresentada neste trabalho, o tempo médio de processamento destas 60 imagens na mesma máquina é de apenas 7.4 segundos.

TABELA II. RESULTADOS NO RECONHECIMENTO DAS PLACAS.

| | Metodologia propsota por Silva et al. [2] | | | Metodologia proposta neste trabalho | | |
|---|---|---|---|---|---|---|
| *Templates* | Acertos | Erros | Não reconhecidas | Acertos | Erros | Não reconhecidas |
| 1 | 4 (100%) | 0 (0%) | 0 (0%) | 3 (75%) | 0 (0%) | 1 (25%) |
| 2 | 3 (100%) | 0 (0%) | 0 (0%) | 2 (67%) | 0 (0%) | 1 (33%) |
| 3 | 14 (73,68%) | 0 (0%) | 5 (26,32%) | 16 (84,21%) | 0 (0%) | 3 (15,79%) |
| 4 | 0 (0%) | 0 (0%) | 1 (100%) | 0 (0%) | 0 (0%) | 1 (100%) |
| 5 | 4 (100%) | 0 (0%) | 0 (0%) | 3 (75%) | 1 (25%) | 0 (0%) |
| 6 | 5 (100%) | 0 (0%) | 0 (0%) | 3 (60%) | 1 (20%) | 1 (20%) |
| 7 | 7 (77,78%) | 0 (0%) | 2 (22,22%) | 5 (55,56%) | 0 (0%) | 4 (44,44%) |
| 8 | 8 (100%) | 0 (0%) | 0 (0%) | 8 (100%) | 0 (0%) | 0 (0%) |
| 9 | 4 (66,67%) | 0 (0%) | 2 (33,33%) | 6 (100%) | 0 (0%) | 0 (0%) |
| 10 | 1 (100%) | 0 (0%) | 0 (0%) | 1 (100%) | 0 (0%) | 0 (0%) |
| 11 | 2 (100%) | 0 (0%) | 0 (0%) | 2 (100%) | 0 (0%) | 0 (0%) |
| 12 | 6 (100%) | 0 (0%) | 0 (0%) | 5 (83,33%) | 0 (0%) | 1 (16,67%) |
| 13 | 1 (100%) | 0 (0%) | 0 (0%) | 1 (100%) | 0 (0%) | 0 (0%) |
| 14 | 3 (75%) | 0 (0%) | 1 (25%) | 3 (100%) | 0 (0%) | 1 (25%) |

## VI. CONCLUSÕES

O reconhecimento de placas de sinalização de trânsito é uma tarefa importante na construção de sistemas de apoio ao tráfego de veículos e, fundamental na implementação de veículos autônomos, porém, a implementação de algoritmos que apresentam tempos de processamento razoáveis continua sendo um grande desafio. O trabalho de Silva et al. [2] apresenta taxas de acerto razoáveis, entretanto, a aplicação do algoritmo SIFT em todas as imagens torna o tempo de processamento inadequado para aplicações em tempo real. Assim, a adaptação do algoritmo AdaBoost para selecionar as áreas com maior potencial para conter placas de sinalização, e a aplicação do algoritmo SIFT, apenas nestas áreas, conseguiu manter a qualidade dos resultados e, ainda, diminuiu sensivelmente o tempo de processamento, o que indica que esta é uma alternativa que merece atenção, na solução deste problema.

## REFERÊNCIAS

[1] P. Viola and M.J. Jones, “Robust real-time object detection”, In: Second International Workshop on Statistical and Computational Theories of Vision – Modeling, Learning, Computing and Sampling, July 13, 2001.

[2] F.A. Silva, A.O. Artero, M.S.V. Paiva e R.L. Barbosa, “Uma Metodologia para Detectar e Reconhecer Placas de Sinalização de Trânsito”, In: VIII Workshop de Visão Computacional (WVC 2012). Anais... Goiás: UFG, 2012.

[3] A. Martinović, G. Glavaš, M. Juribašić, D. Sutić and Z. Kalafatić, “Real-time Detection and Recognition of Traffic Signs”, Proc. 33rd Inter. Convention on Information and Communication Technology, Electronics and Microeletronics – MIPRO 2010, v. III, 2010. p. 247-252.

[4] X. Baró, S. Escalera, J. Vitrià, O. Pujol and P. Radeva, “Traffic Sign Recognition using Evolutionary Adaboost Detection and Forest-ECOC classification”, In: IEEE Trans. on Intelligent Transportation Systems. 2009. p. 113-126.

[5] S.-Y. Chen and J.-W. Hsieh, “Boosted road sign detection and recognition”, In: International Conference on Machine Learning and Cybernetics, Proceedings… v. 7, pp. 3823–3826, July 2008.

[6] I.M. Creusen, R.G.J. Wijnhoven, E. Herbschleb and P.H.N de With, “Color Exploitation in Hog-Based Traffic Sign Detection”, Proc. IEEE 17th International Conference on Image Processing. Hong Kong, 2010, pp. 2669-2672.

[7] G. Loy and A. Zelinsky, “Fast Radial Symmetry for Detecting Points of Interest”, IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 25, no. 8, 2003, pp. 959-973.

[8] R. Belaroussi and J.-P. Tarel, “A real-time road sign detection using Bilateral Chinese Transform”, Proc. International Symposium on Visual Computing, ISVC, 2009, pp. 1161-1170.

[9] A. Arlicot, B. Soheilian and N. Paparoditis, “Circular Road Sign Extraction from Street Level Images Using Colour, Shape and Texture Database Maps”. In: Stilla U, Rottensteiner F, Paparoditis N (Eds) CMRT09. IAPRS, Vol. 38, France, 2009.

[10] C.F. Paulo and P.L. Correia, “Traffic Sign Recognition Based on Pictogram Contours”, Ninth Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS’2008), Klagenfurt, Austria, May 2008, pp. 67-70.

[11] N. Barnes, A. Zelinsky, “Real-Time Radial Symmetry for Speed Sign Detection”, Proc. Intelligent Vehicles Symposium, 2004, pp. 566–571.

[12] H. Fleyeh, “Road and Traffic Sign Color Detection and Segmentation - A Fuzzy Approach”, MVA2005 IAPR Conference on Machine VIsion Applications, Tsukuba Science City, Japan, May 16-18, 2005, pp. 124-127

[13] K. Brkić, “An overview of traffic sign detection methods”, Department of Electronics, Microelectronics, Computer and Intelligent Systems Faculty of Electrical Engineering and Computing Unska 3, 10000 Zagreb, Croatia, Tech. Rep., 2010.

[14] X. Baró and J. Vitria, “Fast traffic sign detection on greyscale images”, Recent Advances in Artificial Intelligence Research and Development, p. 69-76, October 2004.

[15] R. Timofte, K. Zimmermann and L. Van Gool, “Multi-view traffic sign detection, recognition, and 3d localization”, Snowbird, Utah, 2009, pp. 69–76.

[16] A. Ihara, H. Fujiyoshi, M. Takagi, H. Kumon and Y. Tamatsu, “Improved matching accuracy in Traffic Sign Recognition by using different deature subspaces”, Machine Vision Applications (MVA2009), 3-26, p. 130-133, 2009.

[17] K. Brkić, A. Pinz, and S. Šegvić, “Traffic sign detection as a component of an automated traffic infrastructure inventory system”, Stainz, Austria, 2009.

[18] Y. Freund and R.E. Schapire, “A Short Introduction to Booting”, Journal of Japanese Society for Artificial Intelligence, 1999, p. 771-780.

[19] T. Acharya, A.K. Ray, “Image processing: principles and applications”, John Wiley and Sons, 2005. 428 p.

[20] R. Lienhart, and J. Maydt, “An extended set of Haar-like features for rapid object detection”, In IEEE ICIP 2002, 2002, pp. 900–903.

[21] Y. Freund and R.E. Schapire, “Experiments with a New Boosting Algorithm”, Proc. Thirteenth International Conference on Machine Learning, p. 148-156, 1996.

[22] B.B. Chaves, “Estudo do Algoritmo AdaBoost de Aprendizagem de Máquina aplicado a Sensores e Sistemas Embarcados”. 2012. 119 f. (Mestrado em Engenharia Mecânica) – Escola Politécnica da Universidade de São Paulo, São Paulo, 2012.

[23] Oza, N. C. Online Ensemble Learning. Tese de Doutorado, Berkeley: University of California, 2001.

# Skin Color Segmentation in Face Images: An Approach for False Positive Reduction

João Paulo Brognoni Casati*, Diego Rafael Moraes† and Evandro Luis Linhari Rodrigues‡
Department of Electrical and Computer Engineering
São Carlos School of Engineering
University of São Paulo
*jpcasati@usp.br, †diego.moraes@usp.br and ‡evandro@sc.usp.br

***Abstract*—Skin segmentation is an important process for many kinds of application like face detection, face tracking, gesture recognition, forensic computing and others. One of the main problems found in this kind of application is the presence of objects which have skin color but are not part of actual skin segments, being wrongly classified as skin. Aiming to reduce the frequency of false positives, this work presents a method of skin segmentation in facial images which has a step that reduces the false positives without decreasing accuracy. This step is called FPAR, which removes segments classified as skin which are not continuous. To achieve this, the SFA image database was used, constituted of images retrieved from AR and FERET face databases, their respective ground truths of skin segmentation and skin and non-skin samples. The method presented in this work shows promising results, reaching up to 42.2% of false positive reduction and an improvement in the accuracy (0.7%).**

## I. Introduction

In computer vision, human skin detection in face digital images is used in a great variety of applications, as: face recognition [1], face detection [2], face tracking [3], forensic computing, facial surgery evaluation systems [4], and others.

Some works used color to detect human skin in digital images [5], [6], [7], [8], [9], [10], [11], [12], [13]. The skin color is an important feature for some methods. It means that if the detection has no good results the features will have much noise and the method's performance would be impaired.

One of the most important problems in skin detection is the high frequency of false positives. It occurs because it's possible that objects and pixels in image to have the same color of skin. These objects are usually wrongly classified as skin segments [14].

The SFA human skin face image database was created to assist research that uses human skin as useful feature. It is based on two classic face databases: The AR Face Database and FERET face database. It has images of people of different ethnic groups (Asian, Caucasian and Negro), age and sex (male and female). SFA has samples of various dimensions and it is composed of four distinct kinds of images: the original images, the ground truths of skin segmentation, samples of skin images and samples of non-skin images [15].

Colors are powerful features to describe digital images. It can simplify objects recognition, image segmentation and even their interpretation [16].

Colorful digital images has their colors represented by color spaces, which are coordinates systems where each color is represented by a single point.

It was chosen the most used color spaces in skin detection researched literature to be compared in this paper (RGB, HSV, YCbCr and Normalized RGB). The color space which achieved the highest accuracy was used in the second step of the proposed method.

After skin segmentation of the images using an ANN (Artificial Neural Network), it was applied FPAR procedure in order to decrease false positives. This procedure is called FPAR (False Positive Area Reduction).

The main objective of this paper was to develop a method of human skin detection in face images which the main goal is to minimize false positives without decreasing considerably the accuracy.

## II. Proposed Method

This section presents the proposed method, which is composed by two different steps (procedures).

The first step is the segmentation of human skin color in SFA images using various color spaces (RGB, YCbCr, HSV and NormRGB).

Then second step is the application of the procedure called FPAR to the segmented images using the color space which resulted better accuracy in order to reduce the amount of false positives.

The metrics achieved are the result of the comparison between the segmented image and the ground truths available in SFA image database. An example of these ground truths is shown on Figure 1.

### A. Skin Segmentation

The SFA images are originally in the RGB color space. It was made a conversion to the following color spaces in order to compare the results between them: HSV, YCbCr and normalized RGB (NormRGB). The output segmented image is the input of the next step (FPAR), but just for the color space which had the highest accuracy.

The procedure of skin segmentation was made using artificial neural networks (ANN). The architecture chosen was multi-layer perceptron, the most used architecture for pattern recognition problems [17].

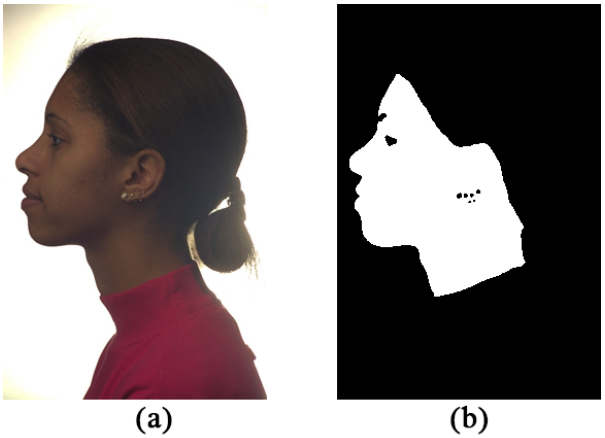



Figure 1. (a) Original image retrieved from FERET database and (b) binarized ground truth from SFA database.

The topology of the ANN is shown in Figure 2, which shows that the number of neurons in each the hidden layer varies between two and twenty, totalling 400 different topologies tested. There are three inputs, which are the values of R, G and B for RGB color space, Y, Cb and Cr for YCbCr color space, H, S and V for HSV color space and normalized R, normalized G and normalized B for NormRGB color space.

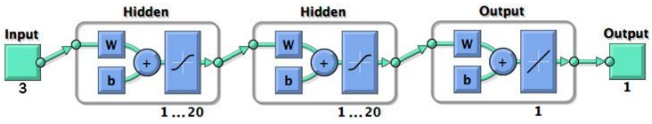



Figure 2. Topologies of the ANN used to segment skin in SFA images.

The best ANN topology was chosen using a method developed by [18], which considers the $r_{max}$ of the test sample group.

After the ANN process the images pixel-by-pixel, the output is a normalized grayscale image with values between 0 and 1, where values closer to 0 means a pixel closer to non-skin color and values closer to 1 means a pixel closer to skin color. After this, the output is binarized in order to find the best threshold. The range of threshold values is from 0.05 to 0.95, incrementing in 0.05 per iteration.

The best threshold is the value that achieved the highest average accuracy between the binarized outputs and the binarized ground truths, for the 1,118 tested images. The resultant images of this procedure were the inputs to FPAR application, which aims to reduce the false positive values without significantly reduction in the accuracy.

### B. FPAR Application

The FPAR procedure consists in remove small portions of pixels classified as skin which are not continous. It occurs because in the most of the cases, skin area is a big and continous area in face images [14].

The application of this procedure aims to reduce the occurrence of false positive in segmented images without significantly reduction in the accuracy achieved in the previous step.

The result of the previous step is the input to this procedure, considering only the color space which achieved the highest accuracy: the binarized image with the best threshold value, where the white color corresponds to skin and black to non-skin. The diagram presented in Figure 3 represents the FPAR procedure and its steps.

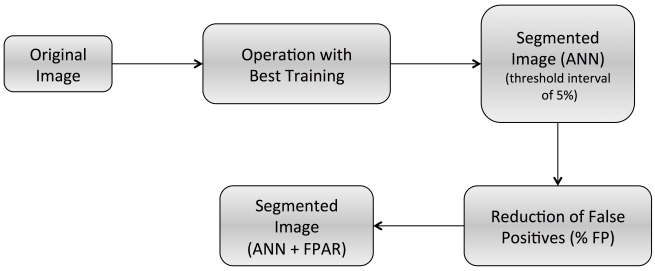



Figure 3. Steps of skin segmentation and FPAR application.

As Figure 3 shows, the input of the FPAR is the output image of the segmentation using ANN, it means a binary image where the pixels with value 1 were classified as skin and pixels with value 0 were classified as non-skin. After this, the number of pixels with value 1 are counted. This number is the quantity of true positives. The percentage of true positives is calculated for each image, following the pattern of the thresholding values used in the previous step (from 5% to 95%, varying in 5%). For each one of these percentage values, it was removed from the image (reclassified as non-skin) the portions of 8-connected pixels which has no more pixels than the result of the percentage multiplied by the numer of true positives, as shown in Equation 1.

$$px = tp * pc \tag{1}$$

where $pc$ is the percentage of the iteration, $tp$ is the number of true positives of the image and $px$ is the maximum size of the portions to be removed (reclassified as non-skin).

As an example, using a true positives fictional value of 1,000 and the step of percentage in 10%, it means that will be no 8-connected portion of pixels classified as skin smaller than 100 pixels.

For the analysis of the results, it is considered the average accuracy of the 1,118 images and the false positive values.

## III. Results and Discussion

The presented results is shown in two parts. The first one is the results concerning the step of skin segmentation using ANN and the second one shows the results of FPAR application, which aims to reduce the false positive values without significantly reduction in the accuracy.

### A. Skin Segmentation

The segmentation of skin color in SFA images was made for four different color spaces and its measured performance

considers the average accuracy of the 1,118 original images comparing with their ground truths.

Table I shows the results of accuracy for every tested color space considering the whole SFA database (1,118 images), only the 876 images from FERET database images (IM_FERET), only the 242 images of AR face database (IM_AR) and the average of the results of IM_FERET and IM_AR. These results were used to define the best color space for the segmentation, which FPAR will be applied later.

Table I. ACCURACY OF SEGMENTATION FOR IMAGES OF EACH TESTED COLOR SPACE.

| Color Space | SFA | IM_FERET | IM_AR | AVERAGE |
|---|---|---|---|---|
| RGB | 92.74% | 91.59% | 96.92% | 94.26% |
| HSV | 62.10% | 66.07% | 47.67% | 56.87% |
| YCbCr | 75.76% | 72.27% | 88.42% | 80.35% |
| NormRGB | 73.02% | 72.35% | 75.43% | 73.89% |

For the tested color spaces, it is worth to observe that the highest accuracy is achieved by the RGB color space, for every division of the SFA database. The results was 92.74% for the average of the whole SFA database, 91.79% for FERET images, 96.92% for AR images and 94.26% for the average between FERET and AR images.

Images of IM_FERET reached the lowest accuracy, it occurs because FERET images have more variety of environment, background, skin color and ethnicities. On the other hand, SFA images have the same background (white) and skin color varies less, because of this the results of IM_AR reached the highest accuracy.

It is important to add as a discussion the percentage of false positives of this segmentation using ANN for RGB color space, as observed in Table II.

Table II. RESULTS OF FALSE POSITIVES FOR THE APPLICATION OF ANN IN RGB COLOR SPACE.

| | |
|---|---|
| **SFA** | 11.42% |
| **IM_FERET** | 11.59% |
| **IM_AR** | 10.81% |
| **AVERAGE** | 11.20% |

These results of false positives will be compared with the results achieved after FPAR application in subsection III-B.

### B. FPAR Application

In this subsection the results are presented considering the average accuracy and the amount of false positives. This amount tends decrease because noncontiguous areas classified as skin were reclassified as non-skin, these areas usually do not belong to actual human skin segments in facial images.

It is possible to observe in Figure 4 that the accuracy values has little variation and starts to decrease considerably after the removal of the noncontiguous areas smaller than 60% of the total of true positives in the segmented image.

The highest accuracy value was achieved when the minimum size of 8-connected skin areas is of 15% of the total of pixels classified as skin. This result is presented in Table III. This is considered as the optimum value for whole SFA database, IM_FERET and IM_AR.

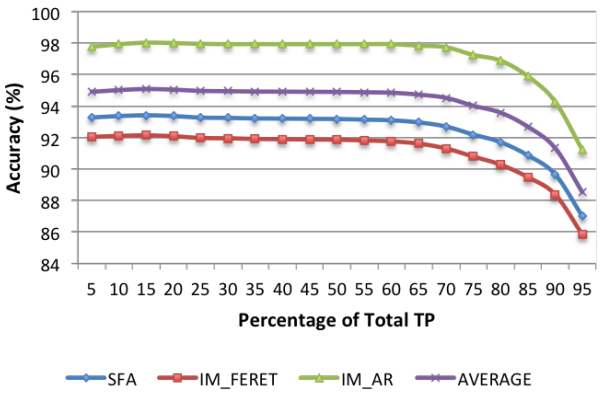



Figure 4. Results of accuracy after FPAR application in segmented images using ANN in RGB color space, considering every iteration related to total of true positives (TP).

Table III. ACCURACY WITH APPLICATION OF FPAR.

| Amount of TP | SFA | IM_FERET | IM_AR | AVERAGE |
|---|---|---|---|---|
| 5% | 93.28% | 92.04% | 97.76% | 94.90% |
| 10% | 93.37% | 92.11% | 97.93% | 95.02% |
| 15% | 93.42% | 92.14% | 98.03% | 95.09% |
| 20% | 93.37% | 92.09% | 98.01% | 95.04% |
| 25% | 93.27% | 91.88% | 97.96% | 94.97% |
| 30% | 93.25% | 91.96% | 97.94% | 94.95% |
| 35% | 93.22% | 91.91% | 97.94% | 94.93% |
| 40% | 93.21% | 91.90% | 97.94% | 94.92% |
| 45% | 93.19% | 91.88% | 97.94% | 94.91% |
| 50% | 93.18% | 91.86% | 97.94% | 94.90% |
| 55% | 93.15% | 91.82% | 97.94% | 94.88% |
| 60% | 93.10% | 91.76% | 97.94% | 94.85% |
| 65% | 92.97% | 91.62% | 97.84% | 94.73% |
| 70% | 92.70% | 91.31% | 97.74% | 94.53% |
| 75% | 92.19% | 90.80% | 97.25% | 94.02% |
| 80% | 91.71% | 90.28% | 96.89% | 93.59% |
| 85% | 90.88% | 89.49% | 95.90% | 92.69% |
| 90% | 89.67% | 88.39% | 94.29% | 91.34% |
| 95% | 87.01% | 85.84% | 91.22% | 88.53% |

The graphic presented in Figure 5 shows the behavior of FPAR application considering the false positive values where it decreases as the percentage of true positives to be removed increases. It is worth to notice that the reduction of false positives is more accented until it achieves 15% of true positives. This values is shown in Table IV.

Considering the parameter which most increased accuracy (15%) it is worth to observe that the false positives were significantly decreased (Table IV).

The application of FPAR showed efficiency both to increase accuracy and to decrease false positives, improving the results for this two metrics. The graphic presented in Figure 6 shows a comparison between false positive values achived by the segmentation with ANN and this values after FPAR application.

Table IV. FALSE POSITIVES WITH APPLICATION OF FPAR.

| Amount of TP | SFA | IM_FERET | IM_AR | AVERAGE |
|---|---|---|---|---|
| 5% | 9.09% | 9.67% | 7.01% | 8.34% |
| 10% | 8.54% | 9.21% | 6.14% | 7.68% |
| 15% | 8.03% | 8.77% | 5.34% | 7.06% |
| 20% | 7.87% | 8.60% | 5.22% | 6.91% |
| 25% | 7.82% | 8.54% | 5.22% | 6.88% |
| 30% | 7.65% | 8.32% | 5.22% | 6.77% |
| 35% | 7.62% | 8.29% | 5.22% | 6.76% |
| 40% | 7.62% | 8.28% | 5.22% | 6.75% |
| 45% | 7.62% | 8.28% | 5.22% | 6.75% |
| 50% | 7.62% | 8.28% | 5.22% | 6.75% |
| 55% | 7.62% | 8.28% | 5.22% | 6.75% |
| 60% | 7.53% | 8.16% | 5.22% | 6.69% |
| 65% | 7.43% | 8.04% | 5.21% | 6.63% |
| 70% | 7.36% | 7.96% | 5.18% | 6.57% |
| 75% | 7.13% | 7.73% | 4.98% | 6.36% |
| 80% | 6.92% | 7.53% | 4.69% | 6.11% |
| 85% | 6.64% | 7.27% | 4.38% | 5.83% |
| 90% | 6.17% | 6.73% | 4.12% | 5.43% |
| 95% | 5.11% | 5.65% | 3.15% | 4.40% |

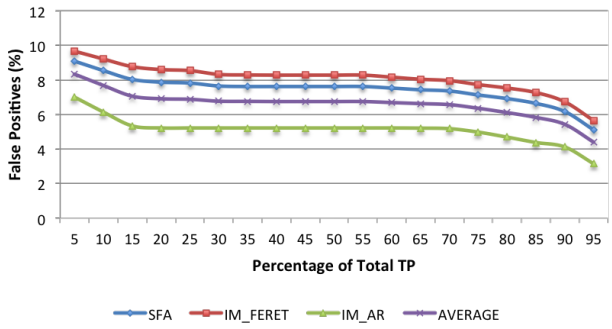



Figure 5. Results of false positives after FPAR application in segmented images using ANN in RGB color space, considering every iteration related to total of true positives (TP).

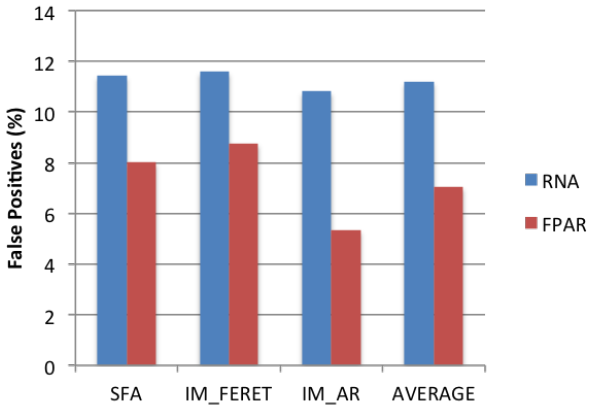



Figure 6. Comparison of false positive results between the output image of ANN and the output after FPAR application.

As shown in Figure 7, the ANN could segmente skin in the original image, but with the application of FPAR, the resultant image became closer to the ground truth, removing some pixels wrongly classified as skin which are not continous.

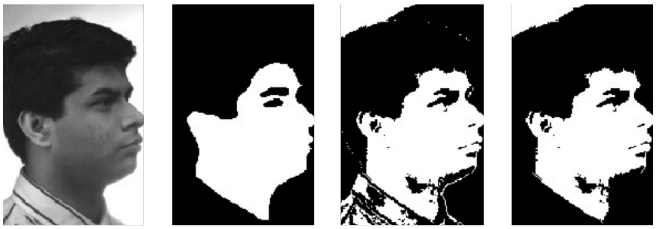



Figure 7. Original image (a), ground truth (b),image resultant of ANN (c) and image resultant after FPAR application (d).

The reduction of false positives with FPAR application was of 29.7% for whole SFA database using the parameter (15%) that achieved the highest accuracy, which also increased.

## IV. CONCLUSION

The main proposal of this paper was the development of a procedure which aims to decrease the incidence of false positives in problems of human skin segmentation in facial images.

The first step of the proposed method shows an artificial neural network based on information of skin and non-skin samples retrieved from SFA database which can segment skin color in face images, achieving average accuracy grater than 92% for the 1,118 SFA original images in RGB color space.

The efficiency achieved by FPAR (the proposed method) shows that the percentage of false positives can be reduced in cases where there are pixels wrongly classified as skin and it is not part of the principal skin segment.

FPAR procedure can be directly applied in problems which prioritize the reduction of lower false positives than higher accuracy, or even increase accuracy in some cases. It means that the objective of the proposed method was achieved as in problems where it is more important to do not have wrongly classified skin areas, the procedure can be adopted safely.

Promising results were achieved with the proposed method, reaching up to 42.2% of reduction in false positives and increasing even the accuracy (0.7%). Analising these results it is possible to assert that the proposed method could improve the results achieved by the segmentation with the ANN procedure.

As contributions of this paper, it was developed a procedure that can detect portions of skin color in face images and other that can be easily applied to human skin segmentation in face images which aims to remove non skin areas wrongly classified as skin.

As future work it is being tested algorithms concerning texture features using the SFA samples in order to improve the segmentation. FPAR procedure also can be tested in other kinds of images aiming to reduce the false positives of segmentation problems.

## REFERENCES


[1] J. M. Chaves-González, M. A. Vega-Rodríguez, J. A. Gómez-Pulido, and J. M. Sánchez-Pérez, "Detecting skin in face recognition systems: A colour spaces study," *Digit. Signal Process.*, vol. 20, no. 3, pp. 806–823, May 2010. [Online]. Available: http://dx.doi.org/10.1016/j.dsp.2009.10.008

[2] Z. Jin, Z. Lou, J.-Y. Yang, and Q. Sun, "Face detection using template matching and skin-color information." *Neurocomputing*, vol. 70, no. 4-6, pp. 794–800, 2007.

[3] R. Sarkar, S. Bakshi, and P. K. Sa, "A real-time model for multiple human face tracking from low-resolution surveillance videos," *Procedia Technology*, vol. 6, no. 0, pp. 1004 – 1010, 2012, 2nd International Conference on Communication, Computing and Security [ICCCS-2012]. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S2212017312006676

[4] M. Bakhshali, M. Shamsi, and A. Golzarfar, "Facial skin segmentation with varying illumination correction for evaluation of facial surgery systems," in *Industrial Electronics and Applications (ISIEA), 2012 IEEE Symposium on*, Sept 2012, pp. 305–309.

[5] P. Kakumanu, S. Makrogiannis, and N. Bourbakis, "A survey of skin-color modeling and detection methods," *Pattern Recognition*, vol. 40, no. 3, pp. 1106 – 1122, 2007. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0031320306002767

[6] N. Bourbakis, P. Kakumanu, S. Makrogiannis, R. Bryll, and S. Panchanathan, "Neural network approach for image chromatic adaptation for skin color detection," *International Journal of Neural Systems*, vol. 17, no. 01, pp. 1–12, 2007, pMID: 17393559. [Online]. Available: http://www.worldscientific.com/doi/abs/10.1142/S0129065707000920

[7] J. Brand and J. S. Mason, "A comparative assessment of three approaches to pixel-level human skin-detection," in *Pattern Recognition, 2000. Proceedings. 15th International Conference on*, vol. 1, 2000, pp. 1056–1059 vol.1.

[8] A. S. Abdallah, M. A. El-nasr, and A. L. Abbott, "A new color image database for benchmarking of automatic face detection and human skin segmentation techniques, to appear," in *In International Conference on Machine Learning and Pattern Recognition*, 2004, pp. 848–854.

[9] J. Y. Choi, Y.-M. Ro, and K. N. Plataniotis, "Boosting color feature selection for color face recognition," *Image Processing, IEEE Transactions on*, vol. 20, no. 5, pp. 1425–1434, 2011.

[10] J. C. Terrillon, M. N. Shirazi, H. Fukamachi, and S. Akamatsu, "Comparative performance of different skin chrominance models and chrominance spaces for the automatic detection of human faces in color images," in *Automatic Face and Gesture Recognition, 2000. Proceedings. Fourth IEEE International Conference on*, 2000, pp. 54–61.

[11] S. Mansoor Roomi, S. L. Virasundarii, S. Selvamegala, S. Jeevanandham, and D. Hariharasudhan, "Race classification based on facial features," in *Computer Vision, Pattern Recognition, Image Processing and Graphics (NCVPRIPG), 2011 Third National Conference on*, 2011, pp. 54–57.

[12] X. Wang, X. Zhang, and J. Yao, "Skin color detection under complex background," in *Mechatronic Science, Electric Engineering and Computer (MEC), 2011 International Conference on*, 2011, pp. 1985–1988.

[13] Z. Jiang, M. Yao, and W. Jiang, "Skin detection using color, texture and space information," in *Fuzzy Systems and Knowledge Discovery, 2007. FSKD 2007. Fourth International Conference on*, vol. 3, 2007, pp. 366–370.

[14] F. R. Siqueira, W. R. Schwartz, and H. Pedrini, "Adaptive detection of human skin in color images," in *IX Workshop de Visão Computacional (WVC), 2013 WVC*, Rio de Janeiro, RJ, BRA, 2013.

[15] J. P. B. Casati, D. R. Moraes, and E. L. L. Rodrigues, "Sfa: A human skin image database based on feret and ar facial images," in *IX Workshop de Viso Computacional (WVC 2013)*, 2013.

[16] R. C. Gonzalez and R. E. Woods, *Digital image processing*. Upper Saddle River, N.J.: Prentice Hall, 2008.

[17] I. Silva, D. Spatti, and R. Flauzino, *Redes Neurais Artificiais para Engenharia e Ciências Aplicadas - Curso Prático*. ARTLIBER.

[18] D. R. Moraes, J. P. B. Casati, and E. L. Linhari Rodrigues, "Analysis of polynomial behavior of the c3 cervical concavity to bone age estimation using artificial neural networks," in *Biosignals and Biorobotics Conference (BRC), 2013 ISSNIP*, 2013, pp. 1–6.

# Multiscale Detection of Convexities and Concavities Based on Local Computation of Weights

Antonio Louro
Universidade Estadual de Santa Cruz – Ilhéus – BA - Brasil.
Email: louro@uesc.br

Adilson Gonzaga
Universidade de São Paulo –EESC- São Carlos – SP – Brasil.
Email: agonzaga@usp.br

***Abstract* - This work proposes a new corner detector, which uses a binary image as input. This is based on computing the weight of the foreground and the background in a local neighborhood. The weight distribution is given by a convolution kernel, which is built from the sum of two concentric Gaussians. The kernel works on a multiscale fashion, and it does not suffer from the flattening effect. Consequently there are no localization problems and, more important, it allows an ordered relationship between weights and angles. Since the neighborhoods centered at the boundary forms an angular configuration between foreground and background, the angles (corners) can be represented by the weights of that configuration. The outputs of the proposed method are confronted to a ground-truth found in the literature, and to the methods that used the same set of images.**

***Keywords—vertex; corner; dominant points shape; convexity; concavity; multiscale; mixture of Gaussians.***

## I. INTRODUÇÃO

Em análise de formas bidimensionais é importante que apenas os detalhes relevantes sejam usados na representação da forma, a fim de tornar menos complexas as tarefas subsequentes, por exemplo, comparar para classificar.

Em experimentos psicofísicos fundamentados na teoria da informação, Fred Attneave expôs a importância das mudanças de direção (ângulos ou curvaturas) ao longo de contornos de imagens de objetos [1]. Os pontos onde as mudanças ocorrem são denominados "*corners*", pontos dominantes, pontos críticos, saliências, ou vértices. Todos são referidos equivalentemente na literatura como pontos de alta curvatura. No presente texto será utilizado o termo vértice.

Num dado segmento de contorno, um vértice é único, não redundante. Por isso, somente este necessita ser representado. Toda a sua vizinhança pode ser descartada e o respectivo trecho do contorno pode ser aproximado por um arco ou segmento de reta.

O trabalho de Attneave [1] é um marco para a pesquisa em análise de formas, a partir deste, mais de uma centena de técnicas de detecção de vértices já foram propostas [2].

Os vértices detectados em um contorno podem ser usados para alimentar aplicações de nível mais alto. Alguns exemplos são casamento de imagens [3,4], decomposição de curvas digitais [5-7], aproximação poligonal [8,9] e visão estéreo [10,11]. A detecção de tais pontos facilita a análise de imagens, pois reduz substancialmente a quantidade de dados a ser processada sem introduzir distorções significativas na forma do objeto [12].

As técnicas de detecção de vértices podem ser agrupadas em duas grandes classes. O que as separa é o tipo de imagem utilizada. Numa, os vértices são detectados diretamente na imagem de níveis de cinza e na outra, a detecção é realizada na imagem do contorno.

As técnicas de detecção de vértices enfrentam três tipos de dificuldades dependentes: a) encontrar os vértices significantes; b) suprimir ruídos e detalhes desnecessários e c) preservar a precisão na localização desses pontos.

Em geral, as soluções adotadas para extrair apenas os vértices significantes incluem uma ou mais suavizações com filtros Gaussianos. No entanto, isto introduz incerteza a respeito da localização dos pontos. Tal distorção obriga, após a detecção, o retrocesso na escala para determinar as posições exatas dos vértices. Esse efeito indesejável é consequência do achatamento da Gaussiana em virtude do aumento de seu desvio padrão. Uma filtragem com essas características tende a igualar os valores dos pixels de uma vizinhança da imagem, tornando difícil determinar as estruturas às quais pertencem.

Além dessas dificuldades, os métodos envolvidos com a detecção de vértices precisam lidar com a falta de uma definição de curvatura para curvas digitais.

Neste trabalho é proposto um novo detector de vértices, que supera os problemas de filtragem e estimação de curvaturas com um único "mecanismo", a computação de pesos em vizinhança local. Esta é baseada em um núcleo de convolução multiescala, construído a partir de mistura de Gaussianas (soma de duas Gaussianas concêntricas).

Este artigo é organizado da seguinte maneira: na seção 2 o detector proposto é descrito em detalhes; na seção 3 são apresentados os resultados experimentais e algumas comparações visuais; as conclusões são apresentadas na seção 4.

## II. O MÉTODO PROPOSTO

Para facilitar, a partir deste ponto o método proposto será chamado de CLP (computação local de pesos). A idéia fundamental em que o CLP está baseado pode ser resumida

pela figura 1, onde se representa o deslocamento de uma janela circular ao longo de uma imagem. Ao centrar a janela no contorno do objeto, setores circulares surgem em seu interior. Os ângulos desses setores dependem diretamente da quantidade de objeto presente na janela. Por exemplo, quanto menor for a concentração de objeto na janela, mais convexa é a região observada. É importante ressaltar que o centro da janela estando sobre o contorno do objeto gera os setores de interesse. Outras centralizações da janela geram setores que não pertencem ao contorno, os quais são desprezados pelo CLP.

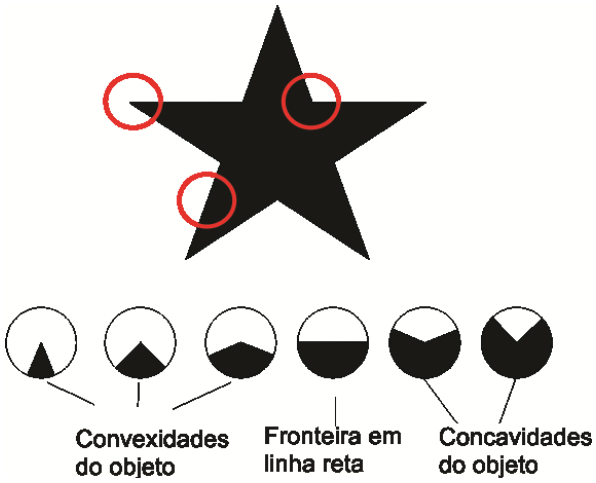



Fig.1. Idéia fundamental do CLP. Cada ponto da fronteira pode ser associado a um setor circular. A agudeza do setor identifica a forma como o objeto se relaciona com o fundo naquele ponto.

Todos os processos que compõem o CLP visam tornar factível a idéia representada na Fig. 1. Para quantificar os setores são usados os pesos de uma distribuição estatística. Esta necessita ser circularmente simétrica (tratamento isotrópico[1]) e possuir uma amplitude central bem mais alta do que o restante da distribuição. Tal relação deve ser mantida para qualquer escala utilizada. A diferença acentuada na amplitude central garante que o pixel em observação seja distinguível dos pixels de sua vizinhança (localização espacial precisa), que é a característica essencial para criar uma relação ordenada ângulo-peso, isto é, o peso de um ângulo agudo tem que ser menor que o peso de um ângulo menos agudo (ver Fig.1). Experimentalmente, observou-se que o peso central precisa ser igual ou maior que a metade do peso total da distribuição, ou seja, numa distribuição normalizada, o peso central deve ser, pelo menos, igual a 0.5.

Em escalas muito baixas (até σ=0.7), a distribuição Gaussiana é adequada. No entanto, para definir se um detalhe da forma é um ruído de digitalização ou uma pequena estrutura sem importância, deve-se observa-la em escalas maiores. A suavização com Gaussianas em escalas mais altas destrói a ordem na relação ângulo-peso.

[1] Tratar de forma isotrópica é não dar preferência a qualquer direção na imagem.

A princípio, a distribuição ideal foi gerada pela soma de duas Gaussianas concêntricas (mistura de Gaussianas em escala), porém, encontrar, em cada escala, duas Gaussianas que gerem a distribuição ideal é um trabalho tedioso. Optou-se, então, em construir a Gaussiana na escala desejada e modificá-la com o algoritmo descrito nas equações 1-3. Este redistribui os pesos da Gaussiana, igualando o peso central a 0.5. O aspecto final dessa distribuição é bastante próximoao da distribuição obtida com a mistura de Gaussianas, ou seja, apresenta abrangência circularmente simétrica e amplitude central elevada. Dado uma distribuição Gaussiana *G* com peso central *Wc*; substitui-se*Wc* por *Wt* e ao final normaliza-se a distribuição com o somatório dos pesos (com *Wt*ocupando o peso central), resultando na distribuição ideal *G'*.

$$\Delta = 1 - 2 \cdot W_c \quad (1)$$

$$W_t = W_c + \Delta \quad (2)$$

$$G' = \frac{G}{\sum W} \quad (3)$$

Na grade retangular, onde as imagens são representadas, a quantidade de ângulos distintoscom centro num pixel da fronteiraé dependente da escala de observação. A escala mais baixa utilizada neste trabalho é associada a uma janela 3x3.Nesta só podem ser gerados oito ângulos. Estes são mostrados na Fig. 2. Deve-se perceber que existem ângulos de mesmo valor, porém com configurações diferentesque possuem pesos próprios. Esta distorção é consequência do ajuste da distribuição circular na grade digital. Todos os ângulos formados por diagonais recebem, neste trabalho, o sufixo D, que é o caso dos itens (2) e (5) da Fig. 2 (90ºD e 180ºD).

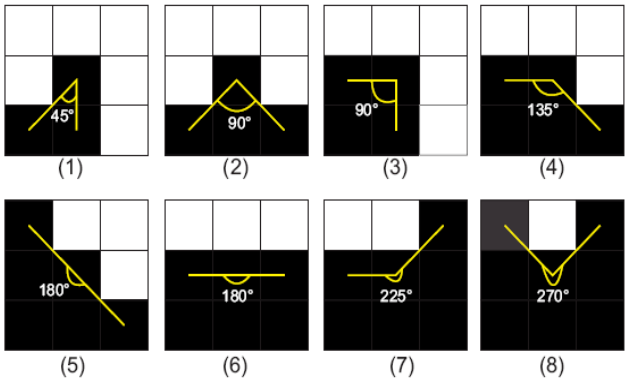



Fig. 2. Configurações angulares no contorno para uma janela 3x3. Observe que existem duas versões para os ângulos de 90º e 180º. Nas configurações (2) e (5) os ângulos são formados por diagonais e são referidos na Tabela 1 com o sufixo D.

Osângulos de contorno mostrados na Fig. 2 também foram construídos em janelas maiores. A Tabela I apresenta os oito ângulos e seus respectivos pesos em sete escalas diferentes. Os pesos desses ângulos são usados como referência na análise multiescala,a fim de determinar a variação de agudeza com o crescimento da escala.

As versões multiescala do núcleo de convolução são acomodadas em janelas quadradas com dimensões (em pixels)3, 5, 9, 17, 35, 71, 143.

TABELA I.Pesos dos ângulos de referência em cada escala considerada.

| **Ângulo** | **Tamanho do Núcleo (escala)** | | | | | | |
|---|---|---|---|---|---|---|---|
| | **3x3** | **5x5** | **9x9** | **17x17** | **35x35** | **71x71** | **143x143** |
| 45º | 0,6250 | 0.6205 | 0.6047 | 0.587 | 0.576 | 0.5694 | 0.566 |
| 90º D | 0,6399 | 0.6611 | 0.6567 | 0.6446 | 0.636 | 0.6307 | 0.6279 |
| 90º | 0,7351 | 0.705 | 0.6776 | 0.6545 | 0.641 | 0.6332 | 0.6291 |
| 135º | 0,7500 | 0.7455 | 0.7297 | 0.712 | 0.701 | 0.6944 | 0.691 |
| 180º D. | 0,7649 | 0.7861 | 0.7817 | 0.7696 | 0.761 | 0.7557 | 0.7529 |
| 180º | 0,8601 | 0.83 | 0.8026 | 0.7795 | 0.766 | 0.7582 | 0.7541 |
| 225º | 0,8750 | 0.8705 | 0.8547 | 0.837 | 0.826 | 0.8194 | 0.816 |
| 270º | 0,8899 | 0.9111 | 0.9067 | 0.8946 | 0.886 | 0.8807 | 0.8779 |

O CLP pode ser dividido em três estágios distintos. No primeiro estágio, a entrada é uma imagem binária (objeto = 0, fundo =1) e o objetivo é extrair todos os candidatos possíveis a vértices, incluindo ruídos e detalhes de pouca importância.

A Fig. 3 facilita a compreensão de como funciona o CLP neste estágio. A grade maior representa uma região qualquer de uma imagem binária. A grade menor é a janela 3x3 que contém os pesos da distribuição ideal para esta escala, a qual é deslocada pixel a pixel pela imagem de entrada. Os pixels da imagem de saída são computados pelo complemento do produto interno entre os valoresda janela 3x3 e os valoresda vizinhança coberta pela janela. Isto é, totaliza-se o peso do objeto contido na vizinhança. As exceções para esse cálculo ocorrem quando a vizinhança considerada apresenta apenas pixels de objeto, ou apenas pixels de fundo. Para ambos os casos é atribuído o valor zero para o pixel de saída.

Com o procedimento acima, a imagem de saída será o contorno do objeto, porém contendo uma espessura de três pixels, a qual está associada à largura da janela 3x3. Isto significa que pixels internos e externos ao objeto estão sendo considerados. No entanto, o interesse está apenas nos pixels do contorno. A faixa adicional de pixels (internos e externos) possui valores que não correspondem aos oito ângulos descritos na Fig. 2, o que torna fácil exclui-los com *thresholding*.

Os candidatos a vértices são todos os pixels de saída cujos pesos representam ângulos diferentes da linha reta, isto é, maiores que 180º e menores que 180ºD (ver Tabela I). Portanto, a saída do primeiro estágio é uma lista contendo as coordenadas espaciais 2D e os pesos dos candidatos.

No segundo estágio as entradas são a lista de candidatos e a imagem binária original. O objetivo é avaliar o comportamento dos candidatos ao longo de um conjunto de seis escalas adicionais. O processamento multiescala é feito pontualmente em cada candidato, gerando uma totalização de pesos para cada escala.

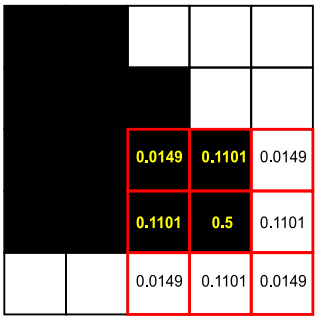



Fig. 3. Computação dos pesos na escala inicial (3x3). O resultado vai gerar o peso de saída correspondente ao ângulo de 90º.

O processamento multiescala pontual gera, para cada candidato, um vetor com seis pesos, um para cada escala considerada. O passo seguinte é combinar esses valores, de tal forma que permita a avaliação confiável sobre a evolução multiescala do candidato.Visto que cada janela é construída a partir de uma versão distinta do núcleo de convolução, com suas próprias aproximações e problemas de ajuste à grade digital, não é conveniente usar diretamente os pesos computados, requerendo, antes, algum tipo de normalização.

Os candidatos a convexidades (*CANDx*) são tratados separadamente dos candidatos a concavidades (*CANDv*). Cada classe com sua própria normalização. Para normalizaros candidatos convexos utiliza-se a distância convexa *Dx*. Para os candidatos côncavos utiliza-se a distância côncava *Dv*.

$$Dx_{escala} = Peso^{180D}_{escala} - Peso^{45}_{escala}$$

$$Dv_{escala} = Peso^{270}_{escala} - Peso^{180}_{escala}$$

Cada candidato *CANDx* e *CANDv* é um vetor com seis valores de peso (*p*), um para cada escala, onde escala = [5, 9, 17,35, 71,143] (ver Tabela I),

$$CANDx_{escala} = [p_{escala}]$$

$$CANDv_{escala} = [p_{escala}]$$

O candidato normalizado *CANDNx* ou *CANDNv* é obtido dividindo-o por *Dx* ou *Dv*. Isto é feito individualmente para cada escala,

$$CANDNx_{escala} = \frac{CANDx_{escala}}{Dx_{escala}}$$

$$CANDNv_{escala} = \frac{CANDv_{escala}}{Dv_{escala}}$$

Uma vez que os valores do candidato estão normalizados, deve-se produzir um número que represente sua evolução nas escalas, isto é feito multiplicando-se todos os pesos normalizados do candidato. Se o candidato é convexo, ele é representado por *Px* e por *Pv* se for côncavo.

$$Px = \prod_{i=1}^{6} CANDNx$$

$$Pv = \prod_{i=1}^{6} CANDNv$$

Para determinar se a importância (saliência) do candidato foi mantida (ou melhorada) ao longo das escalas é necessário comparar o seu número multiescala (*Px* ou *Pv*) com um limiar *Lx* ou *Lv*. Se *Px*<*Lx*, tem-se um bom candidato a vértice de convexidade. Se *Pv*>*Lv*, tem-se um bom candidato a vértice de concavidade. Os parâmetros *Lx* e *Lv* definidos pelo usuário são criados da mesma forma que *Px* e *Pv*. Escolhe-se um ângulo padrão (por exemplo, 135º ou 225º), normaliza-se os seus pesos em cada escala e computa-se o produtório.

A saída deste estágio é uma lista contendo as coordenadas dos bons candidatos e o número multiescala que representa cada um deles.

No terceiro estágio os bons candidatos são comparados entre si. Isto é feito, pois alguns candidatos podem estar localizados muito próximos uns dos outros, ocupando uma mesma região convexa ou côncava do objeto. Embora o vértice de uma dada região seja o ponto mais saliente desta, seus vizinhos são suficientemente salientes para serem considerados bons candidatos. A comparação é feita para determinar o vértice real da região.

A maior dificuldade neste estágio é definir o tamanho da região de disputa. Se for muito pequena, um ou mais vizinhos que estejam além desse limite poderão ser classificados como vértices, sem que o sejam. Se a região for maior do que deve, vértices reais de outras regiões poderão ser eliminados.

As técnicas de supressão de não máximos que usam janelas 2D, como a descrita em [13], não garante a detecção dos verdadeiros vértices. Dependendo das dimensões da janela e da forma da região é possível que ao eliminar pontos indesejáveis, sejam eliminados vértices verdadeiros.No CLP a região de disputa é determinada ao longo do contorno, ou seja, no espaço 1D.

As entradas para o terceiro estágio são a lista de candidatos gerada pelo segundo estágio e o contorno extraído da imagem original. A extração do contorno é necessária para definir as vizinhanças na dimensão 1D, para a comparação dos candidatos, a qual é feitaseparadamente para candidatos convexos e côncavos.

Para definir o tamanho da vizinhança 1D foi necessário introduzir um parâmetro *d*, que determina a distância de disputa entre os candidatos ao longo do contorno. Se dois ou mais candidatos estiverem dentro do limite definido por *d*, a comparação é realizada e o vencedor é eleito como vértice.

## III. RESULTADOS EXPERIMENTAIS

Nesta seção são apresentados alguns resultados experimentais de testes realizados em imagens retiradas da base de imagens MPEG-7 Part B [14]. O objetivo dos testes é avaliar a robustez a ruídos de contorno e invariância à rotação e mudanças de escala, bem como comparar os resultados do CLP com os resultados publicados de outros detectores. Embora não seja aparente nas figuras a seguir, todas as imagens utilizadas nos testes e comparações têm bastante ruído no contorno.

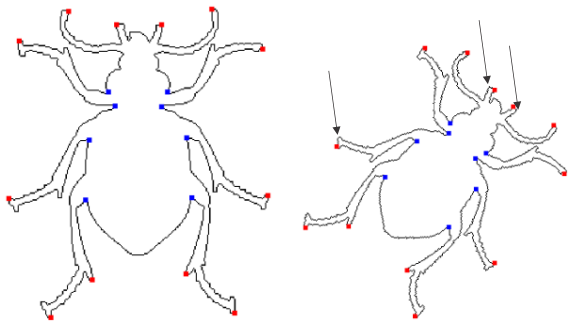


Fig. 4. Invariância à rotação e mudança de escala. Nas convexidades ocorreram três divergências em relação ao original, as quais estão indicadas por setas.

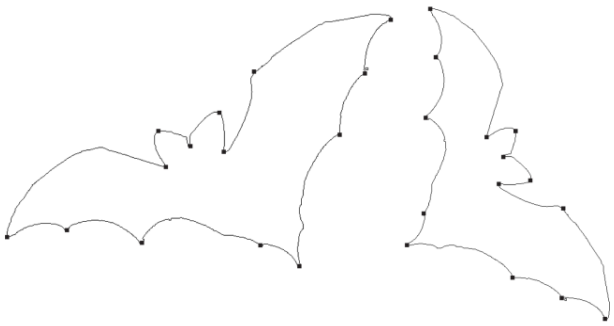


Fig. 5. Invariância à rotação e mudança de escala. Não ocorreram divergências entre os vértices detectados nas duas versões da imagem.

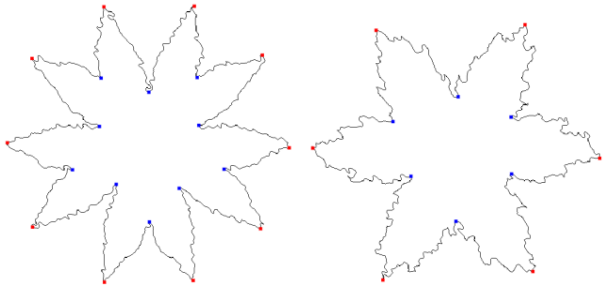


Fig. 6. O parâmetro *d* (tamanho da região de disputa) do CLP foi ajustado com um valor grande o suficiente para que apenas as concavidades e convexidades mais fortes prevalecessem.

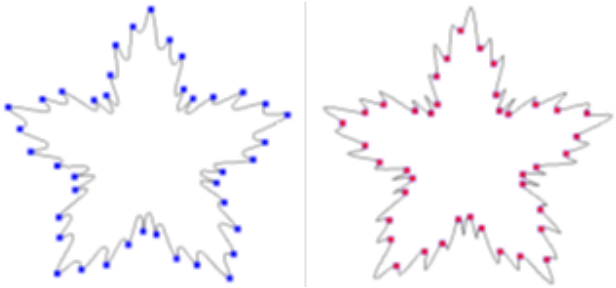


Fig. 7. O parâmetro *d* (tamanho da região de disputa) do CLP foi ajustado com um valor pequeno que permitisse mostrar todas as concavidades e convexidades.

A precisão de qualquer detector de vértices só pode ser julgada se as suas reais posições na imagem forem conhecidas. Em [7] foi publicado um conjunto de oito imagens, cujos vértices foram marcados por um conjunto de dez observadores humanos. Este mesmo conjunto tem sido usado por muitos algoritmos e estão disponíveis na literatura. Os resultados obtidos com o CLP são comparados com o *ground-truth* e com dez algoritmos que tiveram seus resultados publicados em [7,15-23]. Os resultados das comparações estão registrados na Tabela II.

Os resultados são avaliados com as medidas de efetividade *precisione recall*, as quais são representadas pelas equações (4) e (5).Onde, *MP* é a quantidade de pontos coincidentes entre o detector e o *ground-truth*, *TD* é a quantidade de vértices detectados e *GT* é a quantidade total de vértices do *ground-truth* (102 pontos). O valor perfeito para a *precision* é 1.0, o que significa que cada ponto detectado é relevante. No entanto, este valor não possui qualquer significadoconsiderando-se a detecção de todos os pontos verdadeiros, isto é, existe uma grande diferença entre afirmar que todos os pontos detectados são verdadeiros e afirmar que todos os pontos verdadeiros foram detectados. Da mesma forma, o valor perfeito para a *recall* é 1.0, significando que todos os pontos verdadeiros foram detectados, mas não diz nada a respeito da detecção de pontos falsos. Assim, utilizar apenas as medidas *precision* e *recall* para comparar o desempenho de detectores não é a forma ideal de comparação. Desta forma, outra medida *accuracy* utilizada em [24] é adotada aqui, a qual é calculada usando-se a equação (6).

$$precision = \frac{MP}{TD} \tag{4}$$

$$recall = \frac{MP}{GT} \tag{5}$$

$$accuracy = 100.\frac{precision + recall}{2} \tag{6}$$

A medida *accuracy* informa quão próximos os valores obtidos pelo detector estão dos valores verdadeiros. O Valor ideal para essa medida é 100%, o que indica que todos os pontos verdadeiros e nenhum ponto falso foram detectados. A fig. 8 mostra o *ground-truth* e a fig. 9 mostra os resultados do CLP.

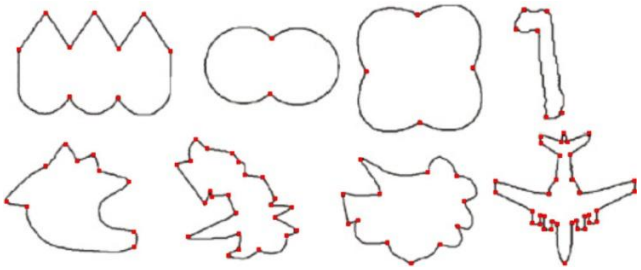


Fig. 8. *Ground-truth* reproduzido de [7]. Os pontos foram reforçados para maior visibilidade.

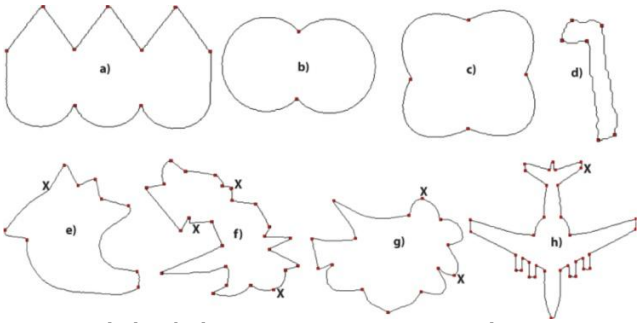



Fig. 9. Resultados obtidos com o CLP**.** Os pontos marcados com X são os falsos positivos e falsos negativos.

TABELA II. Comparação de detectores de vértices.

| Detector | Falso Positivo | Falso Negativo | Precision | Recall | Accuracy |
|---|---|---|---|---|---|
| CLP | 5 | 2 | 0.9524 | 0.980392 | 96.64% |
| [7] | 7 | 2 | 0.9346 | 0.980392 | 95.75% |
| [15] | 25 | 3 | 0.7984 | 0.970588 | 88.45% |
| [17] | 27 | 12 | 0.7692 | 0.882353 | 82.58% |
| [20] | 10 | 29 | 0.8795 | 0.715686 | 79.76% |
| [21] | 15 | 28 | 0.8315 | 0.72549 | 77.85% |
| [22] | 37 | 25 | 0.6754 | 0.754902 | 71.52% |
| [16] | 17 | 10 | 0.6909 | 0.791667 | 74.13% |
| [23] | 37 | 30 | 0.6606 | 0.705882 | 68.32% |
| [19] | 36 | 33 | 0.6571 | 0.676471 | 66.68% |
| [18] | 464 | 5 | 0.1729 | 0.95098 | 56.19% |

## IV. CONCLUSÕES

Neste artigo foi apresentado um novo detector de vértices (pontos dominantes) para imagens binárias, cujo fundamento principal é totalizar os pesos (dados por uma distribuição estatística) dos pixels pertencentes ao objeto. Os candidatos a vértices obtidos na escala mais baixa são observadosem seis escalas mais altas. O comportamento desses pontos através das escalas define quais são os bons candidatos. Estes se agrupam em torno das regiões côncavas e convexas do objeto. Através de uma comparação, similar à supressão de não máximos1D, são identificados os verdadeiros vértices de cada região.

O CLP introduz uma distribuição estatística própria, uma modificação da Gaussiana, que garante amplitude central elevada em qualquer escala utilizada, resolvendo o problema de perda de localização gerado por suavização com Gaussianas.

Foram feitos testes com o CLP para verificar invariância à rotação, à mudança de escala e a ruídos de contorno. Também foram feitas comparações com dez detectores diferentes, os quais foram escolhidos por utilizar um mesmo *ground-truth*. Como pode ser observado na Tabela II, o CLP se mostrou bastante promissor. O CLP ainda necessita de algumas modificações para torná-lo livre de parâmetros ajustáveis pelo usuário. O parâmetro *d* que determina a distância de disputa é bastante problemático, pois depende do tamanho do objeto em estudo e pode sofrer influência dos parâmetros de limiar (*Lx* e *Lv*). Quanto mais candidatos forem admitidos para a disputa, maior poderá ser o valor atribuído ao parâmetro*d*.

Estudos estão sendo conduzidos para determinar as escalas naturais de cada convexidade/concavidade do objeto, o que poderá facilitar a definição do vértice real de cada estrutura do objeto. A base desse estudo é compreender a interferência que as estruturas adjacentes, que compõem um objeto, exercem entre si.Ao aumentarmos a escala de uma vizinhança, esta podeenglobar outras estruturas do objeto, modificando os valores angulares esperados. A fig. 10 ilustra esse tipo de interferência.

A quantidade limitada de escalas (3x3 a 143x143) funcionou satisfatoriamente bem para as imagens usadas nos testes, porém pretende-se tornar essa quantidade adaptativa à área do objeto.

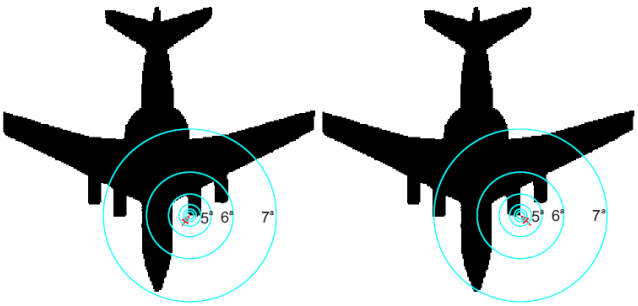



Fig. 10. Interferência de estruturas adjacentes à estrutura de interesse. As circunferências representam as escalas de observação dos vértices direito e esquerdo da turbina. O vértice esquerdo, que é mais próximo á fuselagem receberá uma importância reduzida em relação ao vértice direito, devido a escala 5 envolver parte da fuselagem e da asa. O mesmo só ocorrerá com o outro vértice a partir da escala 6. Ambos iniciam como um setor circular de 90º, o qual se alarga com a inclusão de regiões adjacentes.

## REFERÊNCIAS


[1] F. Attneave, “Some informational aspects of visual perception,” Psychological review, vol. 61(3),pp.183-193, 1954.

[2] A. Dutta, A. Kar, B.N.Chatterji, “Corner detection algorithms for digital images in last three decades,” IETE Tech Rev, vol. 25(3), pp. 123–132, 2008.

[3] P. Smith, D. Sinclair, R. Cipolla, K. Wood,“Effective corner matching,”inBritish machine vision conference, Citeseer, pp. 545–556, 1998.

[4] E. Vincent, R. Laganiere,“Detecting and matching feature points,” Journal of Visual Communication and Image Representation vol. 16(1), pp. 38–54, 2005.

[5] K. Abe, R. Morii, K. Nishida, T.Kadonaga,“Comparison of methods for detecting corner points from digital curves, a preliminary report,” in Document Analysis and Recognition, Proceedings of the Second International Conference on, IEEE, pp. 854–857, 1993.

[6] M. Marji, R.Klette, P.Siy,“Corner detection and curve partitioning using arc-chord distance,” in IWCIA'04 Proceedings of the 10th international conference on Combinatorial Image Analysis, vol. 3322 of Lecture Notes in Computer Science,Springer, 2004, pp. 512–521.

[7] M. Sarfraz, Interactive Curve Modeling With Applications to Computer Graphics, Vision and Image Processing, Springer, 2008.

[8] R.M.C. Junior, L. da Fontoura Costa L, “Piecewise linear segmentation of digital contours in O(N. log(N)) through a technique based on effective digital curvature estimation,” Real-Time Imaging, vol. 1(6), pp.409–417, 1995.

[9] M. Sarfraz, M.R. Asim, A. Masood, “Piecewise polygonal approximation of digital curves,”in Proceedings of theEighth International Conference onInformation Visualization, IV 2004,IEEE Computer Society, 2004, pp. 991–996.

[10] L. Haichau,“Feature matching based on corner and edgeconstraints,” Electronic imaging & signal processing,SPIE Newsroom, SPIE Newsroom, 6 june 2007, DOI: 10.1117/2.1200705.0541

[11] E. Vincent, R. Laganiere R, “Matching feature points in stereo pairs: A comparative study of some matching strategies,”Machine Graphics and Vision, vol. 10(3), pp. 237-259, 2001.

[12] H.C. Liu, M.D. Srinath,“Corner detection from chain-code,” Pattern Recognition, vol. 23(1-2), pp. 51–68, 1990.

[13] M. Brown, S. Winder, R. Szeliski,“Multi-Image Matching usingMulti-Scale Oriented Patches,”in Proceedings of the IEEE Computer Vision and Pattern Recognition, CVPR'05, vol.1, IEEE Computer Society, Washington, DC, USA, 2005, pp. 510-517

[14] The MPEG-7 Part B database, http://www.imageprocessingplace.com/ root_files_V3/image_databases.htm, Accessed on Mar 2012.

[15] X. Zhang, H. Wang, A. Smith, X. Ling, B. Lovell, D. Yang,“Corner detection based on gradient correlation matrices of planar curves,” Pattern Recognition, vol. 43(4), pp.1207–1223, 2010.

[16] A. Sobania, J. Evans,“Morphological corner detector using paired triangular structuring elements,” Pattern recognition, vol. 38(7), pp. 1087–1098, 2005.

[17] D. Chetverikov, Z. Szabo,“A simple and efficient algorithm for detection of high curvature points in planar curves,” in 23rd Workshop of Australian Pattern Recognition Group, vol. 23, 1999, 175–184.

[18] S. Smith, J. Brady,“SUSAN-A new approach to low level image processing,” Int. J. Computer Vision, vol. 23(1), pp.45–78, 1997.

[19] J. Shi, C. Tomasi, “Good features to track,”in Proceedings of Computer Vision and Pattern Recognition, CVPR 94, 1994, pp. 593–600.

[20] H. L. Beus, S.S.H. Tiu, “An improved corner detection algorithm based on chain-coded plane curves,” Pattern Recognition, vol. 20 (3), pp. 291–296, 1987.

[21] H. Freeman, L. Davis, “A corner-finding algorithm for chain-coded curves,” IEEE Transactions on Computers, vol. C-26(3), pp. 297–303, 1977.

[22] A. Rosenfeld, J. Weszka, “An improved method of angle detection on digital curves,” IEEE Transactions on Computers, vol. C-24(9), pp. 940–941, 1975.

[23] A. Rosenfeld, E. Johnston, Angle detection on digital curves, IEEE Transactions on Computers, vol. C-22(9), pp. 875–878, 1973.

[24] F. Mokhtarian, F. Mohanna, “Performance evaluation of corner detectors using consistency and accuracy measures,” Computer Vision and Image Understanding, vol. 102(1), pp.81–94, 2006.

# Towards 3D point cloud from thermal images using colormap transformations

Andriy G. Krefer, Mauren A. de Souza,
Tania M. Centeno, Humberto R. Gamba
Programa de Pós-graduação em Engenharia Elétrica
e Informática Industrial (CPGEI)
Universidade Tecnológica Federal do Paraná (UTFPR)
Curitiba, Brasil

Gustavo B. Borba
Programa de Pós-Graduação em
Engenharia Biomédica (PPGEB)
Universidade Tecnológica Federal do Paraná (UTFPR)
Curitiba, Brasil

***Resumo*—This paper presents a method for reconstructing a 3D point cloud from 2D infrared (IR) images. Methods for 3D reconstruction from conventional visible light are available. However, such algorithms encounter difficulties when dealing with IR images. IR images usually present less distinctive features when compared with their visible light counterpart. The Structure from Motion method can reconstruct 3D surfaces from 2D images and usually consists of the following steps: feature detection (extraction and matching, to indicate which points, in different views, are the same ones), point cloud reconstruction (through point triangulation) and surface reconstruction. The proposed method acts on the feature detection step. We have used colormap transformations in order to insert more distinctive features to the original images. As a result, a higher number of features has been found by the feature detector, allowing a denser 3D point cloud reconstruction.**

**Keywords— Thermal images, 3D reconstruction, structure from motion, point cloud.**

## I. Introdução

A análise termográfica é de grande aplicação na engenharia e medicina. Ela permite a visualização da energia térmica emitida por um corpo de maneira não intrusiva e não dependente de contato [1]. As imagens geradas por essa técnica são chamadas de imagens térmicas ou imagens infravermelhas (IV). Assim como a termografia, a reconstrução 3D também é de grande importância, possuindo aplicações em engenharia, medicina e na indústria. Diferentes técnicas de reconstrução 3D da superfície externa de um corpo (objeto qualquer) existem atualmente, entre elas: fotogrametria, sistemas de varredura laser, projeção de luz estruturada e máquinas de medição coordenada [2].

Não foi encontrado na literatura nenhum trabalho de reconstrução 3D térmica que utilize apenas imagens IV 2D como entrada. Em geral, a reconstrução da geometria do corpo é tratada separadamente e com sensores específicos para tal finalidade. Apenas em uma etapa posterior é feita a fusão dos dados térmicos com os dados geométricos, mas sem que haja tentativa de recuperação da estrutura espacial do corpo diretamente através das imagens IV.

Em [3], por exemplo, foi desenvolvido um dispositivo que integra uma câmera térmica e um scanner 3D em uma mesma armação mecânica. As coletas 3D e IV são feitas simultaneamente e as informações apanhadas são processadas e usadas na geração de um modelo 3D com textura térmica. Nesse tipo de solução podem surgir problemas de alinhamento no processo de combinação dos dados de diferentes fontes. Adicionalmente, a necessidade de mais de tipo de sensor aumenta o custo do sistema.

Uma família de técnicas para a reconstrução 3D de imagens é a conhecida como Structure from Motion (SFM) [4]. Estas técnicas podem ser classificadas como fotogramétricas e utilizam imagens bidimensionais de luz visível como entrada. Uma das etapas essenciais para a obtenção da malha 3D (3D mesh) final é a geração da nuvem de pontos (3D point cloud). A partir da nuvem é possível então, utilizando-se outras técnicas disponíveis [4] obter o modelo 3D.

Nesse artigo, apresentam-se técnicas para obtenção da nuvem de pontos a partir de imagens térmicas bidimensionais utilizando-se o SFM. Ressalta-se que esse é um processo difícil, visto que o SFM foi originalmente concebido para imagens de luz visível, nas quais o processo de extração de características (feature extraction) é facilitado pela presença de uma maior quantidade de pontos salientes (i.e. cantos) se comparado às imagens IV.

Como estratégia para a captura de mais pontos salientes, é proposto um pré-processamento baseado em transformações no mapa de cores (colormap) da imagem IV. Para os experimentos, foram utilizadas imagens térmicas de diferentes corpos de prova, adquiridas especificamente para este propósito. Diferentes transformações foram testadas e comparadas. Os resultados obtidos demonstram que a abordagem proposta promove nuvens de pontos de melhor qualidade para imagens IV se comparadas àquelas geradas a partir de imagens sem tratamento.

## II. Reconstrução 3D a partir de imagens térmicas

### A. Imagens térmicas

Todos os objetos com temperatura acima do zero absoluto emitem radiação térmica na faixa infravermelha do espectro eletromagnético, que compreende comprimentos de onda de 0,7 a 1000 μm [1]. As câmeras térmicas, também chamadas de câmeras IV, possuem sensores capazes de captar essa radiação. Como produto final, o dispositivo é capaz de gerar uma imagem térmica, na qual cada pixel corresponde a um valor de temperatura. Os valores podem ser mapeados em uma escala de

cinza, onde, em geral, o branco representa uma região quente e o preto uma região fria. A mesma imagem também pode ser representada em diferentes mapas de cores, conforme a Fig. 1 retrata. Apesar de cada pixel representar o mesmo valor de temperatura em todas as figuras, ele pode ser representado por diferentes cores [5]. As imagens térmicas possuem aplicações em diversas áreas, entre elas: medicina, engenharia, agricultura, vigilância, aplicações industriais, militares e utilização em ensaios não destrutivos [1], [5].

### B. *Extração e correspondência de características*

O método Scale-Invariant Feature Transform (SIFT) é utilizado no presente trabalho para identificar pontos correspondentes em imagens de um mesmo objeto adquirido de diferentes posições.

O SIFT é robusto para alterações de escala, rotação, variações de luminosidade, ruídos e certo grau de distorção afim (affine). Apesar de não ser invariante a distorções projetivas, na prática, localmente, distorções afim costumam ser aproximações satisfatórias [6], [7]. As marcações em vermelho na Fig. 6 (a ser ser discutida em mais detalhes na seção IV) exemplificam as características adquiridas de uma imagem. A escolha das regiões se dá principalmente pela detecção de cantos. Assim, a existência de texturas e áreas com componentes de alta frequência auxilia a extração de características suficientemente distintas.

As regiões escolhidas (blocos de pixels) são codificadas em descritores. Tal representação permite uma redução dimensional do bloco, pois imagens em geral, contém bastante redundância de informações. Os parâmetros extraídos conservam a propriedade de serem distintos uns dos outros. Além disso, permitem a ordenação e comparação dos descritores.

A correspondência de característica se dá pela comparação de diferentes blocos a fim de se reconhecer o mesmo ponto em diferentes imagens. Entretanto, a avaliação não é feita pixel a pixel, mas sim através dos descritores. Dessa maneira, consegue-se um ganho de desempenho e a invariabilidade à escala, rotação, ruído, luminosidade e distorção afim.

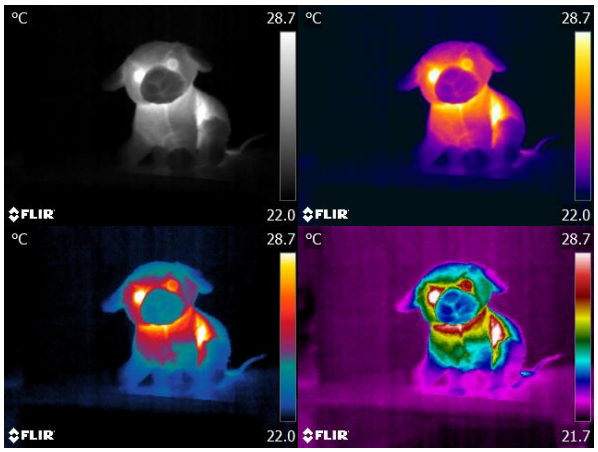



Figura 1. Imagem infravermelha com diferentes mapas de cores

### C. *Structure from Motion (SFM)*

Structure from motion (SFM) é um conjunto de técnicas utilizado para a reconstrução tridimensional de uma cena, dadas múltiplas imagens bidimensionais como entrada. Como resultado, o algoritmo provê tanto os pontos no espaço 3D quanto a posição das câmeras (translações e rotações) [4].

Seres humanos se utilizam do movimento de objetos e também do próprio movimento através de uma cena para estimar sua estrutura. Diversos ângulos de um mesmo corpo são informações úteis no processo. Computacionalmente, o problema é similar ao da reconstrução a partir de um par de imagens estéreo.

Após a aquisição das imagens, o primeiro passo é a utilização de um algoritmo de extração de características como o SIFT. As características são então comparadas entre si à procura de correspondências. Se forem encontradas, significa que correspondem a um mesmo ponto no espaço. Mesmo que a correlação entre os pontos seja positiva não significa que esteja necessariamente correta. Ou seja, podem existir ruídos, erros de amostragem, distorções não computadas ou correspondências incorretas. Para amenizar este problema, o SFM costuma utilizar a técnica Random Sample Consensus (RANSAC) para determinar os melhores valores através de tentativas aleatórias, desconsiderando os pontos fora da curva.

Após a triangulação e validação dos melhores pontos através do RANSAC, chega-se à reconstrução esparsa da cena. Nesse modelo, apenas alguns pontos-chave são encontrados. Com isso, tem-se a estrutura básica do corpo, ainda sem muitos detalhes. Pode-se perceber que esta etapa depende bastante da quantidade de características extraídas e correspondidas na fase anterior. São elas que irão determinar os pontos da nuvem esparsa.

A reconstrução esparsa é utilizada para a reconstrução densa. Além dos pontos-chave é feita a tentativa de correspondência de todos os pixels intermediários.

## III. Método proposto

O método proposto para a obtenção da nuvem de pontos para imagens térmicas atua na etapa de extração de características. O SIFT utiliza os cantos da imagem como candidatos a características. Sendo assim, uma imagem com componentes de alta frequência é mais rica em pontos salientes.

As imagens térmicas possuem diferentes propriedades em relação às imagens convencionais de luz visível. Possuem bordas suavizadas e menos componentes de alta frequência. Portanto, o SIFT, que foi originalmente desenvolvido para imagens convencionais, encontra maior dificuldade na extração de características em imagens térmicas [8].

Mapas de cores podem ser utilizados para transformar uma região de baixas frequências em uma região de altas frequências em uma imagem térmica.

A Fig. 3 ilustra esse conceito. Nas Fig. 3 (a) e (b) são mostrados os mapas de cores utilizadas, onde p é o valor original do pixel e V1(p) e V2(p) os valores após a transformação. O pixel p pode assumir valores de 0 a 255, onde 0 é preto e 255 é branco. Os mapas de cores utilizados incluem apenas níveis de cinza, visto que o SIFT não utiliza

cores para a extração de características. As Fig. 3 (c) e (d) mostram uma mesma imagem térmica com os mapas de cores V1 e V2, respectivamente. Pode-se observar que da Fig. 3 (c) para a Fig. 3 (d) há um incremento nas componentes de alta frequência da imagem, como, por exemplo, na região demarcada.

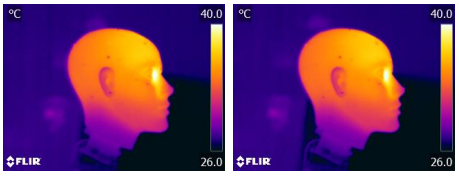
Uma observação pertinente é o fato de que imagens térmicas possuem a propriedade de serem praticamente invariantes a mudanças na iluminação [8]. A fotografia térmica da Fig. 2 (a) foi obtida com as luzes acesas, enquanto a da Fig. 2 (b) com as luzes apagadas. Não há diferenças perceptíveis entre elas.

Entretanto, em geral, esta mesma técnica proposta de manipulação do mapa de cores não seria bem sucedida em imagens de luz visível, devido à alta sensibilidade à variações de iluminação. As Fig. 3 (e) e Fig. 3 (g) são imagens de luz visível. Em Fig. 3 (g) ocorreu um aumento de iluminação em relação à Fig. 3 (e). Observando-se as transformações correspondentes, Fig. 3 (f) e (h), como por exemplo na região demarcada, nota-se que esta mesma região apresenta cores (e consequentemente características) diferentes após a aplicação do mapa de cores.

## IV. Experimentos e Resultados

Para a captura das imagens térmicas utilizou-se uma câmera IV modelo FLIR A325 com resolução de 320 x 240 pixels, com sensibilidade a comprimentos de onda de 7,5 a 13 µm, temperaturas de -20 a 120 ºC, resolução térmica de 0,05 ºC, com uma lente de distância focal de 18 milímetros. A câmera também possibilita a gravação de filmes a uma taxa de 30 quadros por segundo. Para a captura das imagens de luz visível foi utilizada a câmera amadora do aparelho celular Samsung Galaxy S (GT-I9000B) com a resolução de 1280 x 720 pixels e lentes com distância focal de 3,79 milímetros.

A fim de possibilitar um maior controle do posicionamento do dispositivo de aquisição de imagens, foi utilizado um braço rotatório (Fig. 4). Tal estrutura permite a fixação da câmera em sua extremidade e possibilita sua rotação em torno do corpo de prova, situado no centro da montagem. Adicionalmente, a altura ($h$), ângulos de inclinação ($\alpha$, $\beta$ e $\gamma$), ângulo de rotação ($\theta$) e distância ao centro ($d$) da câmera podem ser controlados. Além disso, o motor acoplado permite a rotação de uma volta completa com velocidade praticamente constante. Ao detectar o fim de curso (através uma chave mecânica) o motor é desligado. A velocidade de rotação também pode ser controlada através de um potenciômetro.




(a) (b)

Figura 2. Exemplo de imagens adquiridas. (a) Em torno do corpo. (b) Imagem descartada por apresentar distorções provenientes de trepidações.

Os experimentos foram conduzidos em uma sala de temperatura controlada, de modo que correntes de ar frio ou ar quente não pudessem provocar alterações de temperatura no corpo ou artefatos nas imagens térmicas.

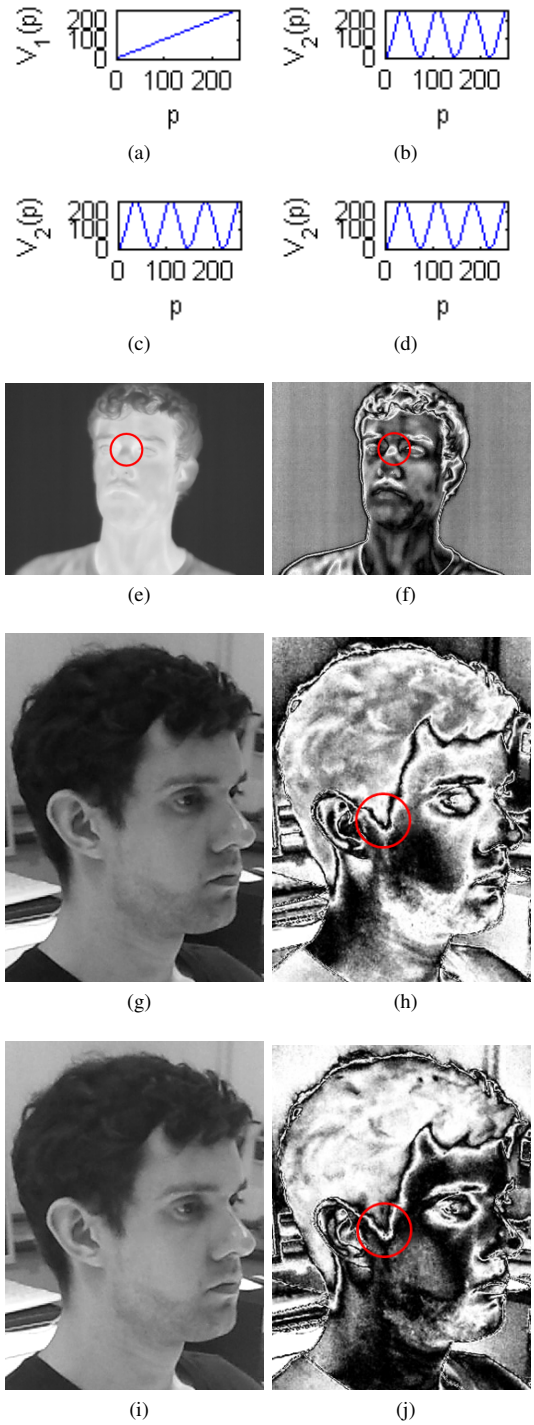



Figura 3. Transformação de mapas de cores. (a) e (b) Mapas de cores $V_1$ e $V_2$. (c) Imagem com o mapa de cores $V_1$. (d) Mesma imagem com o mapa de cores $V_2$, agora com acréscimo de componentes de alta frequência. (e) e (f) Foto de voluntário com os mapas de cores $V_1$ e $V_2$ respectivamente. (g) e (h) Foto com acréscimo de luminosidade e mapas de cores $V_1$ e $V_2$.

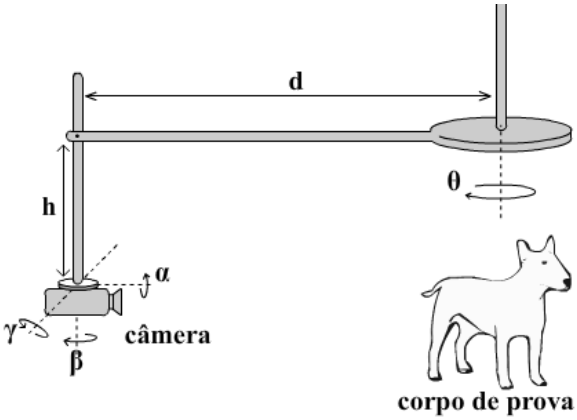



Figura 4. Braço rotatório

O primeiro passo do experimento foi o posicionamento do corpo de prova na posição central do aparato. A distância da câmera ao centro (d) e os demais parâmetros ($h$, $\alpha$, $\beta$ e $\gamma$) foram ajustados de maneira a permitir o melhor enquadramento possível do objeto.

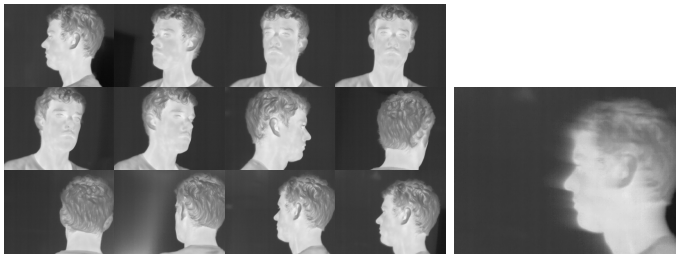
Em seguida, o foco da câmera foi ajustado e a gravação a 30 quadros por segundo é iniciada. O motor, após ter sua velocidade ajustada, é ligado. A captura de uma volta completa foi feita com o objeto parado. O mesmo procedimento foi repetido para todos os objetos e para a câmera de luz visível.

A Fig. 5 ilustra algumas das imagens coletadas. Após a gravação do filme foram selecionados 70 quadros de espaçamento aproximadamente uniforme. Existe uma certa trepidação em alguns pontos, podendo gerar alguns artefatos devidos à rápida movimentação. Os quadros com essas perturbações foram descartados manualmente(Fig. 5 (b)).

Os seguintes grupos de imagem foram coletados:

- Pessoa-lv – Aquisição de imagens de voluntário com câmera de luz visível.
- Pessoa-it: Aquisição do mesmo voluntário com câmera térmica.

Após a aquisição, as imagens térmicas foram submetidas a um ajuste de escala para que tivessem o melhor contraste possível, conforme (1).



(a) (b)

Figura 5. Exemplo de imagens adquiridas. (a) Em torno do corpo. (b) Imagem descartada por apresentar distorções provenientes de trepidações.

$$P_{val}(t) = round\left(\frac{255.(T - Tmin)}{Tmax - Tmin}\right) \quad (1)$$

O valor do pixel (Pval) é um número inteiro entre 0 e 255. Seu valor depende de sua temperatura (T), sendo que Tmin é o menor valor de temperatura dentro de um grupo de imagens do mesmo corpo e Tmax é o maior.

Após a aquisição das imagens, foi executado o algoritmo de extração de características SIFT convencional e o algoritmo com a alteração proposta. A Fig. 6 mostra os resultados em algumas imagens representativas. Em Fig. 6 (a), o algoritmo em uma imagem de luz visível, Fig. 6 (b) em imagem térmica e em Fig. 6 (c) e (d) a mesma imagem com diferentes mapas de cores. Os mapas de cores das Fig. 6 (c) e (d) são calculadas através de (2).

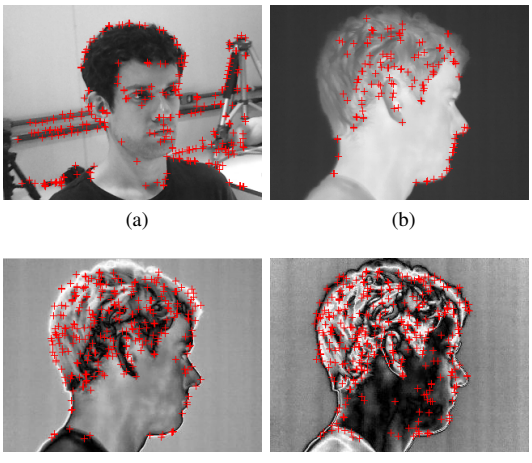
$$M(p) = \frac{255}{2}.\left(1 - \cos\left(\frac{2.\pi.f.p}{255}\right)\right) \quad (2)$$

A variável p representa o valor original do pixel e M(p) o seu valor (em escala de cinza, onde 0 é preto e 255 é branco) após a transformação. No mapa de cores utilizado na Fig. 6 (c) f = 3,5 (denominado mapa de cores 1). No mapa de cores utilizado na Fig. 6 (d) f = 1,5 (denominado mapa de cores 2).

Todas as características das imagens de um mesmo grupo são extraídas e comparadas a fim de se determinar quais são as regiões correspondentes. Para isso, foi utilizada a implementação do SIFT em GPU em [9].

Através da correspondência de características é possível identificar, entre diferentes fotografias de diversos ângulos, uma determinada região (que corresponde a um mesmo ponto no espaço tridimensional).

Após a correspondência de características efetua-se a triangulação e determinação dos principais pontos no espaço. A nuvem de pontos gerada nessa etapa é do tipo esparsa.



(a) (b)

(c) (d)

Figura 6. Detecção de zonas de interesse via SIFT. (a) Em uma imagem de luz visível. (b) Em uma imagem infravermelha. (c) e (d) Em uma imagem infravermelha com diferentes mapas de cores.

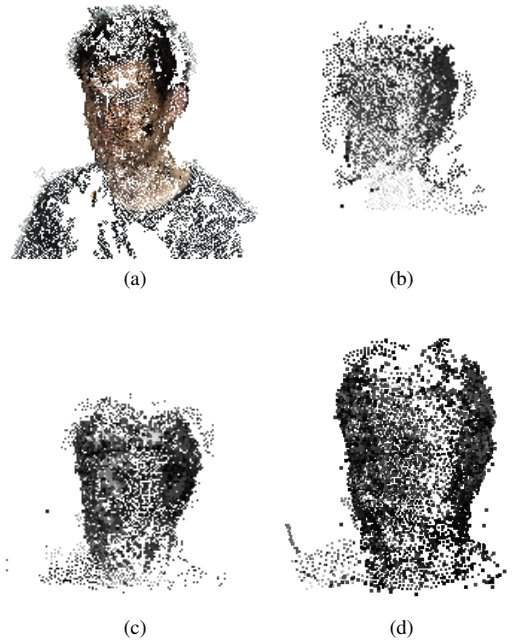


Figura 7. Detecção de zonas de interesse via SIFT. (a) Em uma imagem de luz visível. (b) Em uma imagem infravermelha. (c) e (d) Em uma imagem infravermelha com diferentes mapas de cores.

Ou seja, possui grande espaçamento entre os pontos, por ser formada apenas pela triangulação de alguns pontos-chave.

A reconstrução esparsa é utilizada como entrada para o algoritmo descrito em [10]. A saída do algoritmo é a reconstrução da nuvem de pontos do tipo densa. No processo, é feita a tentativa de correspondência do máximo de pixels possível, a fim de preencher os espaços vazios entre os pontos-chave da nuvem esparsa. A Fig. 7 mostra a reconstrução densa dos grupos de imagem pessoa-lv e pessoa-it (com mapas de cores 1 e 2).

## V. Discussão dos resultados

Observando a Fig. 6 pode-se perceber que o algoritmo proposto foi capaz de aumentar a número de características extraídas (representadas pelas marcações em vermelho). A imagem térmica sem tratamento (b) possui um menor número de características em relação às imagens com os mapas de cores 1 e 2 (c e d). O mapa de cores 2 foi o que registrou o maior número de pontos.

As etapas de extração e correspondência de características são de vital importância para a reconstrução da geometria do corpo. Através desta identificação do mesmo ponto da imagem em diferentes vistas, é possível fazer a triangulação e determinação de suas coordenadas no espaço tridimensional. Portanto, quanto maior o número de pontos triangulados corretamente, melhor será o modelo obtido.

A etapa de reconstrução densa da nuvem de pontos pode ser observada na Fig. 7. A Fig. 7 (a) representa o resultado do grupo de imagens pessoa-lv, que é utilizado como base de comparação, por se tratar de imagens de luz visível. Pode-se observar que a nuvem não é perfeita, apresentando ruídos no lado esquerdo da imagem (ombro) e na região do cabelo.

A Fig. 7 (b) representa o grupo pessoa-it sem nenhum tipo de tratamento (apenas o ajuste de contraste efetuado conforme (1). A reconstrução é disforme e apresenta bastante ruído, visto que o algoritmo tem dificuldade em triangular os pontos devido ao baixo número de características (ver Fig. 6 (b)).

As Fig. 7 (c) e (d) representam o grupo pessoa-it com os mapas de cores 1 e 2 respectivamente. O mapa de cores 1 apresenta um aumento no número de características extraídas e isso se mostra visível na nuvem de pontos obtida. O mapa de cores 2 produz um número ainda maior de características, e consequentemente gera uma nuvem de pontos ainda melhor.

É interessante destacar que a câmera térmica utilizada nos experimentos é de baixa resolução (apenas 320 x 240 pixels). Provavelmente, imagens térmicas de maior resolução gerariam nuvens de pontos de maior qualidade.

## VI. Conclusões

Comparando-se as nuvens de pontos geradas com (Fig. 7 (c) e (d)) e sem o auxílio do algoritmo proposto (Fig. 7 (b)), pode-se observar uma melhora nos resultados. Assim, pode-se afirmar que o método proposto é um recurso interessante para a reconstrução 3D a partir de imagens térmicas bidimensionais.

O método utiliza imagens térmicas convencionais, ou seja, não são necessários procedimentos específicos para a coleta das imagens térmicas. No entanto, o uso de marcadores durante a aquisição das imagens térmicas pode ser sugerido como um trabalho futuro, com o objetivo de fornecer ao algoritmo pontos salientes mais distinguíveis e confiáveis.

Outros trabalhos futuros incluem a aquisição de imagens com diferentes alturas do braço mecânico ($h$ na Fig. 4), o uso de uma câmera térmica de maior resolução, testes com diferentes mapas de cores e uma avaliação quantitativa da qualidade da nuvem de pontos a partir de um corpo de prova estático.

## Agradecimentos

Agradecimentos à Universidade Tecnológica Federal do Paraná (UTFPR) pela estrutura, e ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) pelas bolsas de pesquisa processos 133994/2013-9 e 310262/2011-9.

## Referências

[1] C. Meola and G. M. Carlomagno, "Recent advances in the use of infrared thermography," *Measurement science and technology*, vol. 15, no. 9, p. R27, 2004.

[2] M. A. de Souza, I. J. Sanches, and H. R. Gamba, "3D medical reconstruction: Overview and case studies," 2014.

[3] S. Barone, A. Paoli, and A. V. Razionale, "A biomedical application combining visible and thermal 3D imaging," in *XVIII Congreso internactional de Ingenieria Grafica, Barcelona*, 2006, p. 1–9.

[4] R. Szeliski, *Computer vision: algorithms and applications*. Springer, 2010.

[5] R. Gade and T. B. Moeslund, "Thermal cameras and applications: a survey," *Machine Vision and Applications*, vol. 25, no. 1, p. 245–262, 2014.

[6] J.-M. Morel and G. Yu, "ASIFT: a new framework for fully affine invariant image comparison," *SIAM Journal on Imaging Sciences*, vol. 2, no. 2, p. 438–469, 2009.

[7] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, Nov. 2004.

[8] Y. Yun and I.-H. Gu, “Multi-view face pose classification by boosting with weak hypothesis fusion using visual and infrared images,” in *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Mar. 2012, pp. 1949–1952.

[9] C. Wu, *SiftGPU: A GPU implementation of scale invariant feature transform (SIFT)*, 2007.

[10] Y. Furukawa and J. Ponce, “Accurate, dense, and robust multiview stereopsis,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 8, pp. 1362–1376, 2010.

# Application of texture analysis for differentiation of the *greening* from others pests

Patricia Pedroso Estevam Ribeiro
Departamento de Engenharia Elétrica
Escola de Engenharia de São Carlos - Universidade de São Paulo - USP,
São Carlos, Brasil
patriciapedrosoestevam@hotmail.com

Maria Stela Veludo de Paiva
Departamento de Engenharia Elétrica
Escola de Engenharia de São Carlos - Universidade de São Paulo - USP,
São Carlos, Brasil
mstela@sc.usp.br

Lúcio André de Castro Jorge
Embrapa Instrumentação Agropecuária
São Carlos, Brasil
lacjorge@gmail.com

*Abstract* — **The *greening* is one of the most serious diseases of citrus and in this paper is proposed a methodology based on texture analysis for assessment of symptomatic citrus leaves. To evaluate the methodology were used 120 images of citrus leaves containing leaves with diseases (*greening*, CVC and rubelose) and nutritional deficiencies (manganese, magnesium and zinc). Samples were extracted by leaf and quadrants in red and green colors channels and on grayscale. The samples were analyzed by LBP descriptors, LFP and Haralick. The results generated were classified by the $X^2$ distance and Weka, IBK and ANN MLP. The best results were for LBP descriptor and LFP-s using $X^2$ distance, with a sensitivity range between 97.02% and 99.94% and 93.39% and 99.52%, respectively, and LBP using IBK with a sensitivity range between 99.2% and 99.6% sensitivity.**

***Keywords— Greening, Texture analysis, LBP, LFP, Haralick, matriz de co-ocorrência.***

## I. INTRODUÇÃO

O *greening*, também conhecido como *Huanglongbing* (HLB), é considerado atualmente a mais grave doença dos citros no mundo [2]. Causada pela bactéria *Candidatus* Liberibacter *spp*., é transmitida pelo inseto psilídeo *Diaphorina citri*, que adquire e transmite a bactéria às demais plantas ao se alimentar de uma planta já contaminada. O controle do *greening* é realizado periodicamente com inspeção constante e a eliminação imediata de plantas com sintomas, e o controle do inseto transmissor é feito com a montagem de armadilhas. Atualmente, o método de inspeção visual e a análise da leitura do DNA da bactéria pela análise de *Polymerase Chain Reaction* (PCR) [4] são os mais utilizados para diagnosticar a doença. O método PCR é utilizado para diagnosticar o patógeno de plantas suspeitas, mas o custo elevado e o longo tempo para a análise o tornam proibitivo de ser aplicado em escala necessária para o controle. As inspeções visuais podem ser realizadas por inspetores caminhando ao lado das plantas cítricas ou em plataformas acopladas a um trator. Nos experimentos apresentados por [1] e [5] são destacados que a diferença de mensuração entre os inspetores depende da habilidade e condições físicas de cada um para discriminar a doença nas folhas. Em média esse tipo de inspeção detecta 47,61% das plantas sintomáticas.

O sintoma característico do *greening* aparece inicialmente, em alguns ramos apresentando folhas com clorose assimétrica (Figura 1). Estas manchas são facilmente confundidas com outras doenças e deficiências nutricionais que se assemelham às características visuais do *greening*, confundindo os inspetores na identificação destas doenças pelo método visual.

Fig. 1. Imagem de folhas de citros contaminada com o *greening*.

Este artigo propõe uma metodologia baseada em análise de textura para identificar folhas de citros sintomáticas, avaliando se estão contaminadas com o *greening* ou outras doenças e deficiências nutricionais.

O presente artigo está organizado nas seguintes seções: a seção 2 aborda a teoria relacionada ao trabalho, na seção 3 é apresentada a metodologia, na seção 4 os resultados e discussões, e finalmente na seção 5 a conclusão.

## II. ANÁLISE DE TEXTURA

### A. *Introdução sobre textura*

A textura é uma das características mais importantes para classificar e reconhecer objetos e cenas, sendo aplicada nas mais diversas áreas para: identificação de regiões por meio do padrão de cores, extração de informações de rugosidade ou suavidade de determinada superfície, e a sua regularidade, extração de características, segmentação, reconhecimento de padrões e classificação de atributos.

A textura pode ser descrita por diferentes abordagens [6], como a abordagem estatística, na qual se enquadram os descritores de Haralick [7], a estrutural, a espectral e a abordagem de micropadrões locais que analisa uma vizinhança local da imagem para extrair características que possibilitem a

descrição de textura local. Nessa última abordagem enquadram-se o descritor LBP (*Local Binary Pattern*) [8] e o LFP (*Local Fuzzy Pattern*) [9], que são descritos a seguir.

### B. *Local Binary Pattern*

O LBP é um descritor de textura local que analisa a imagem em relação às intensidades relativas entre os pixels em uma pequena vizinhança [8].

Com o descritor LBP uma imagem em tons de cinza é analisada, considerando uma vizinhança 3x3 ao redor do pixel central (Figura 2a). Essa vizinhança é limiarizada pelo valor do pixel central gerando uma matriz com valores binários, podendo ser observado na Figura 2 (b), possibilitando gerar $2^8$ (256) códigos. Cada pixel de vizinhança ($V_i$) adquire o valor ($E_i$), sendo atribuído a ele o valor '0', se estiver abaixo do valor do pixel central ($V_0$), caso contrário recebe o valor '1', como pode ser observado na Figura 2 (c). Os valores dos pixels na vizinhança limiarizada são multiplicados por uma matriz de pesos (Figura 2d). Realizando a soma desses valores, conforme a Equação 1, resulta no código LBP, como pode ser observado na Figura 2 (e) [9].

$$N_{LBP} = \sum_{i=1}^{8} E_i . 2^{i-1} \tag{1}$$

Um histograma é gerado para determinar as ocorrências dos diferentes códigos LBP, formando o vetor de característica LBP, caracterizando uma imagem de textura.

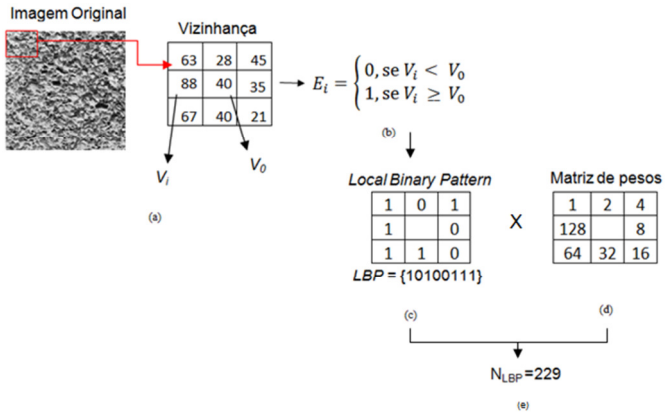



Fig. 2. Ilustração do cálculo do LBP [9].

### C. *Local Fuzzy Pattern*

O descritor de textura local LFP foi proposto como um novo descritor para análise de micropadrões de imagens digitais baseado na lógica *fuzzy* [9]. O diferencial desse descritor é que ao invés de utilizar regras *fuzzy* para se obter um código final que descreva um determinado padrão da imagem, são utilizados conjuntos *fuzzy* para tratar a vizinhança de pixels e um número *fuzzy* para tratar o nível de cinza de cada pixel. Como neste método não são utilizadas regras *fuzzy*, o seu custo computacional diminui.

A representação por números *fuzzy* é paramétrica e a definição de uma função de pertinência apropriada é heurística e não única. Assim, a definição de diferentes funções de pertinência é baseada nas propriedades da vizinhança do micropadrão $W$ x $W$ do pixel central $g(i,j)$ em uma imagem digital (Figura 3). No LFP cada distribuição de níveis de cinza dentro desta vizinhança é um conjunto *fuzzy,* e uma função de pertinência é responsável diretamente pelo código LFP.

O grau de pertinência $\hat{\mu}_{g(i,j)}$ do pixel central em relação à região definida pela amostra $W$ x $W$ pixels é dado pela média ponderada dos valores de pertinência individuais de cada pixel, gerando o código LFP. A Figura 3 demonstra o cálculo do código LFP, onde, $f_{g(i,j)}$ é a função de pertinência, gerando os valores das pertinências em relação ao pixel central e $P(k,l)$ é uma matriz de pesos para a vizinhança $A(k,l)$ com a mesma dimensão $W$ x $W$ pixels.

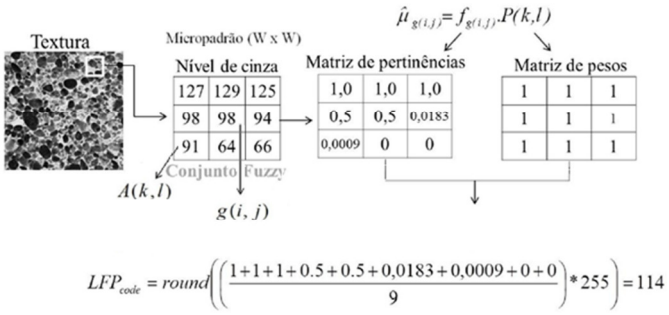



Fig. 3 - Ilustração do cálculo do código LFP [9].

### D. *Descritores de Haralick*

Os descritores de Haralick são compostos por um conjunto de 14 descritores para descrever a textura, em que são definidas diversas características advindas do cálculo de matriz de co-ocorrência. Os descritores são obtidos a partir de cálculos estatísticos de segunda ordem, considerando as ocorrências de cada nível de cinza em pixels diferentes, ao longo de diferentes direções, e são os seguintes: segundo momento angular, contraste, correlação, variância, momento de diferença inverso, média da soma, variância da soma, entropia da soma, entropia, variância da diferença, entropia da diferença, medidas de informações da correlação (com duas propriedades) e coeficiente de correlação máximo [7].

## III. METODOLOGIA

Este artigo propõe um método baseado em análise de textura para auxiliar a diferenciação do *greening* de outras doenças similares. A Figura 4 mostra o diagrama do desenvolvimento.

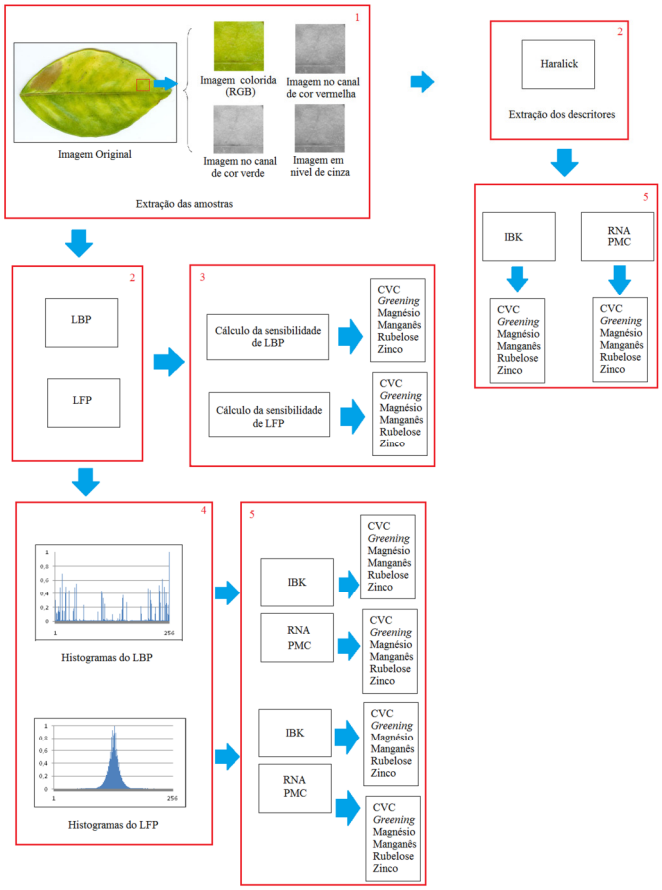



Fig. 4. Diagrama de desenvolvimento.

As etapas da metodologia proposta, ilustradas na Figura 4, são as seguintes: aquisição das imagens foliares, extração das amostras das folhas, aplicação dos descritores de textura LBP, LFP, e dos descritores de Haralick, cálculo da sensibilidade e classificação das classes de doenças pelos descritores LBP e LFP, extração dos histogramas do descritor LBP e LFP, classificação das classes de doenças utilizando o algoritmo IBK e RNA PMC.

### A. *Aquisição das Imagens Foliares*

Foram digitalizadas 150 folhas de citros fornecidas pela empresa do grupo Fischer [10], divididas em seis classes de doenças e deficiências nutricionais, contendo 25 amostras para cada tipo de doença (*greening*, CVC e rubelose) e deficiência nutricional (manganês, magnésio e zinco), sem deformidades geométricas. Estas amostras foram selecionadas por um técnico agrônomo, que identificou os sintomas, apenas por meio da inspeção visual, segundo instruções do manual técnico do *greening*, fornecido pela Fundecitrus [11]. As folhas de citros foram digitalizadas por um scanner de mesa, modelo SCANNER HP SCANJET G4050, no modelo de cores RGB, utilizando a parte frontal da folha. O lado do caule foi direcionado para o lado esquerdo e a ponta da folha no lado direito, para manter o mesmo padrão de análise.

Foram gerados dois bancos de imagens de folhas de citros, onde, no banco 1 foram digitalizadas as 150 folhas de citros, com resolução de 100 DPI e dimensão de 400 x 200 pixels das imagens originais em níveis de cinza. No banco 2 foram selecionadas e digitalizadas as 120 melhores folhas de citros, com resolução de 300 DPI e dimensão de 3000 x1800 pixels das imagens originais em RGB.

### B. *Extração das amostras*

Foram aplicados os descritores de textura LBP e LFP, e os descritores de Haralick nas amostras extraídas de cada folha. Inicialmente foi avaliado o número de amostras extraídas, o tamanho da amostra e o número de folhas para cada classe de doenças, considerando as melhores folhas sem deformidades geométricas.

O artigo [12] mostrou que as cores segmentadas influenciam na diferenciação das doenças. Devido a isso as amostras foram recortadas manualmente usando o Matlab, selecionando as áreas com manchas que caracterizam as doenças, gerando novas imagens recortadas nos canais de cores vermelho e verde, e em níveis de cinza usando a função *rgb2gray*. Não foi utilizado o canal de cor azul, pois este não apresentou propriedades discriminatórias nos testes. A Figura 5 ilustra um exemplo de uma amostra sendo extraída.

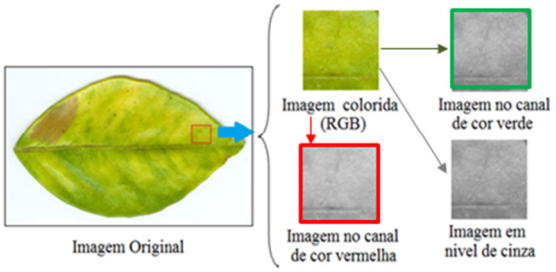



Fig. 5. Exemplo de amostra extraída por uma imagem de folha de citros nos canais de cores vermelho, verde e nível de cinza.

Foram feitas várias avaliações das amostras com os descritores com o objetivo de verificar a influência dos canais de cores e a influência da divisão em quadrantes (Figura 6).

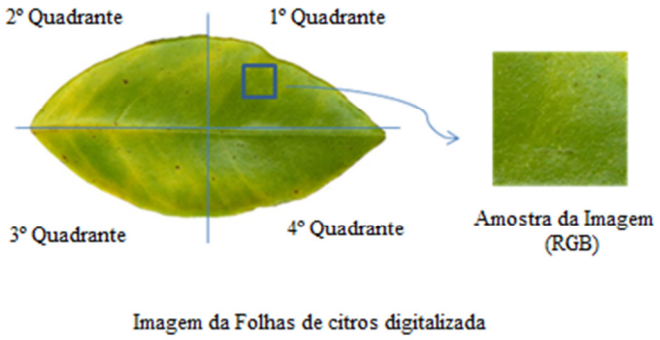



Fig. 6. Exemplo de amostras extraídas por uma imagem de folha de citros por quadrante.

### C. *Aplicação dos descritores de textura local LBP e LFP, e dos descritores de Haralick*

Nesta fase foram aplicados os descritores LBP, LFP e os descritores de Haralick nas amostras extraídas. Os resultados obtidos para os treze primeiros descritores de Haralick e também para os descritores LBP e LFP foram armazenados

em arquivo com extensão "arff" para posterior classificação das classes de doenças utilizando o *toolbox* Weka [3]. Para os descritores LBP e LFP foi também realizada a classificação das classes de doenças utilizando a métrica $X^2$, e calculada a sensibilidade, conforme detalhado na seção *D* seguinte.

### *D. Cálculo da sensibilidade e classificação das classes de doenças pelos descritores LBP e LFP*

O processo de classificação utilizado pelo descritor LFP com as funções de pertinências (sigmóide, triangular e gaussiana) e pelo descritor LBP, foi a métrica $X^2$, dada pela Equação (2).

$$\boldsymbol{X^2\ (P_i, Q_i) = \frac{1}{2}\sum_{i=0}^{255}\frac{(P_i - Q_i)^2}{(P_i + Q_i)}} \tag{2}$$

onde, *Qi* é a freqüência do nível de cinza da amostra que está sendo avaliada e *Pi* é a frequência do nível de cinza da amostra do conjunto, a qual está sendo comparada com a amostra avaliada. Utilizando validação cruzada *leave-one-out,* foram comparados os histogramas gerados para cada amostra com o restante das amostras do conjunto [9].

As amostras são comparadas e classificadas numa mesma classe, quando se obtêm a menor distância $X^2$entre as demais amostras da mesma classe. Os resultados da classificação são armazenados em uma matriz de confusão e o cálculo da sensibilidade (*S*) é dado pela Equação (3).

$$\boldsymbol{S = \frac{VP}{VP + FN}} \tag{3}$$

onde, *VP* são os verdadeiros positivos e *FN* os falsos negativos.

### *E. Extração dos histogramas do descritor LBP e LFP*

Após a aplicação dos descritores LBP e LFP, são gerados e salvos os histogramas correspondentes para cada amostra analisada, sendo também gerado um histograma para cada amostra de doença. Cada linha do histograma foi separada como atributo e salva em um arquivo com extensão "arff" para ser posteriormente avaliada pelos classificadores do *toolbox* Weka, utilizando validação cruzada de 10 *folds.*

### *F. classificação das classes de doenças utilizando o algoritimo IBK e RNA PMC*

Com os arquivos com extensão "arff", mencionados na seção E, foi realizada a classificação das classes de doenças com o algoritmo IBK e o RNA PMC do *toolbox* Weka. Esses resultados foram comparados com os resultados de sensibilidade obtidos com a métrica $X^2$(seção *D*). Os descritores de Haralick foram também armazenados como atributos e classificados utilizando o algoritmo IBK e RNA PMC para comparação de resultados.

### *G. Ferramentas*

Foram utilizadas as ferramentas: Matlab versão 7.6.0 R2008a para analisar as amostras com os descritores LBP, LFP e os descritores de Haralick e para converter as amostras nos canais de cores. O *toolbox* Weka 3.7.10 foi utilizado para classificar amostras extraídas dos descritores de Haralick e os histogramas gerados utilizando o algoritmo IBK e RNA PMC.

## IV. RESULTADOS E DISCUSSÕES

Inicialmente as amostras foram avaliadas em relação à influência do número de amostras extraídas por folha, o tamanho da amostra e a resolução para as imagens do banco 1.

Na Tabela 1 são apresentados os resultados de sensibilidade para os descritores LBP, LFP e os descritores de Haralick, considerando amostras com o tamanho de 30x30 e 100x100 pixels.

Tabela 1**.** Tabela da sensibilidade obtida com variação do tamanho da amostra.

| **Tabela com variação no tamanho da amostra W x W pixels** | | |
|---|---|---|
| **Descritores** | **30 x 30 pixels** | **100 x 100 pixels** |
| LBP | 32,70% | 43,20% |
| LFP-s | 34,30% | 37,80% |
| LFP-t | 25,30% | 31,70% |
| LFP-g | 26,80% | 32,80% |
| Teste 1 | 34,67% | 36,00% |
| Teste 2 | 34,17% | 36,17% |

Na Tabela 1 foram avaliadas 120 folhas de citros, sendo 20 imagens de folhas de citros para cada doença. Foram extraídas 5 amostras em nível de cinza de cada imagem digitalizada das folhas de citros, sendo 3 amostras da parte superior e 2 da parte inferior em relação à nervura central. Foram avaliadas as seis classes de doenças (CVC, *greening*, magnésio, manganês, rubelose e zinco), para o descritor LFP com função de pertinência (sigmóide, triangular e gaussiana) e para o LBP, considerando uma vizinhança de 3x3. Foram avaliadas também as seis classes de doenças para os descritores de Haralick.

A sensibilidade para o teste 1 e teste 2 dos descritores de Haralick foram calculadas usando o algoritmo IBK pelo *toolbox* Weka. Para o teste 1 foram utilizados treze descritores. Para o teste 2 foram utilizados onze descritores (segundo momento angular ou energia, contraste, correlação, variância, momento de diferença inversa, média da soma, variância da soma, entropia da soma, entropia, variância da diferença, entropia da diferença).

Observa-se na Tabela 1, que para as amostras com tamanho 100x100 pixels, os valores de sensibilidade foram superiores aos valores das amostras com tamanho 30x 30 pixels, para todos os descritores. Além disso, os descritores LFP-t (função de pertinência triangular) e LFP-g (função de pertinência gaussiana) apresentaram resultados de sensibilidade inferiores ao descritor LFP-s (função de pertinência sigmóide).

Foram realizados dois experimentos com o objetivo de avaliar a influência das classes de cores e das amostras

divididas nos quadrantes, onde inicialmente as amostras foram extraídas no canal de cor vermelho, no verde e em níveis de cinza. No primeiro experimento, as amostras são avaliadas considerando cada canal de cor separadamente, para todos os quadrantes. O objetivo desse experimento foi analisar a influência das cores. No segundo experimento são avaliadas todas as amostras de cada quadrante separadamente, o que inclui as amostras nos canais de cores vermelho, verde e níveis de cinza. O objetivo desse experimento foi analisar a influência da divisão por quadrantes.

Para o primeiro experimento (Tabela 2), foram utilizadas 5 folhas para cada doença, para as 6 classes de doenças, totalizando 30 imagens de folhas. Para cada folha foram extraídas 2 amostras em cada quadrante, com tamanho de 200 x 200 pixels, gerando 240 amostras extraídas para cada cor, totalizando 720 amostras extraídas para este experimento.

Tabela 2. Tabela da sensibilidade obtida pelo primeiro experimento.

| Tabela da sensibilidade do primeiro experimento | | | |
|---|---|---|---|
| | **Nível de cinza (%)** | **Vermelho(%)** | **Verde (%)** |
| LBP | 37,90 | 41,70 | 34,60 |
| LFP-s | 33,80 | 42,90 | 37,90 |
| Teste1 | 33,68 | 37,29 | 40,14 |
| Teste2 | 33,13 | 36,94 | 41,11 |

Os resultados da sensibilidade da Tabela 2 apresentados para os descritores de textura são abaixo de 50%, podendo-se concluir que a separação por canais de cores não evidencia a diferenciação das classes de doenças.

Para o segundo experimento (Tabela 3), foram utilizadas 5 folhas para cada doença, para as 6 classes de doenças, totalizando 30 imagens de folhas. Para cada folha foram extraídas 2 amostras em cada quadrante, com tamanho de 200x200 pixels, gerando 180 amostras por quadrante, totalizando 720 amostras extraídas para todos os quadrantes.

Tabela 3. Tabela da sensibilidade obtida pelo segundo experimento.

| Tabela do segundo experimento | | | | |
|---|---|---|---|---|
| | **1º Quad.(%)** | **2º Quad.(%)** | **3º Quad.(%)** | **4º Quad.(%)** |
| LBP | 95,00 | 98,30 | 97,20 | 94,40 |
| LFP-s | 71,10 | 82,80 | 84,40 | 76,70 |
| Teste1 | 35,37 | 31,48 | 35,56 | 36,48 |
| Teste2 | 36,11 | 30,56 | 38,89 | 38,24 |

Na Tabela 3, os maiores valores de sensibilidade foram para o descritor LBP, entre 94,40% e 98,30% seguido pelo descritor LFP-s com valores entre 71,10% e 84,40%, enquanto os descritores de Haralick obtiveram os menores valores de sensibilidade.

Portanto os resultados do segundo experimento com divisão por quadrantes evidenciam que o LBP e LFP-s, são os mais adequados para serem utilizados na diferenciação das classes das doenças.

### *A. Análise Geral dos Descritores*

Para analisar os descritores LBP e LFP-s, foram utilizadas as imagens do banco 2, e extraídas amostras por quadrante com tamanho e número variado de amostras, sendo gerados os resultados de sensibilidade em cada quadrante. A Tabela 4 apresenta os valores de sensibilidade do LBP e a Tabela 5 apresenta os valores de sensibilidade do LFP-s.

Tabela 4. Tabela de sensibilidade do LBP, por quadrante, variando o tamanho da amostra.

| Tabela de sensibilidade do LBP, por quadrante, variando o tamanho da amostra | | | | | |
|---|---|---|---|---|---|
| **Tamanho da amostra (pixels)** | **N° de Amostras** | **1º Quad.(%)** | **2º Quad.(%)** | **3º Quad.(%)** | **4º Quad.(%)** |
| 200x200 | 7 | 98,21 | 98,65 | 98,53 | 97,70 |
| 200x200 | 10 | 97,92 | 98,47 | 98,58 | 97,11 |
| 300x300 | 7 | 98,10 | 99,12 | 99,41 | 98,25 |
| 300x300 | 10 | 98,33 | 99,00 | 99,25 | 98,08 |

Observa-se que os melhores resultados da classificação das doenças na Tabela 4, foram para as amostras com o tamanho 300 x 300 pixels, com número de 7 e 10 amostras extraídas por quadrante, totalizando 2520 e 3600 amostras analisadas.

Tabela 5 - Tabela de sensibilidade do LFP-s, por quadrante, variando o tamanho da amostra.

| Tabela de sensibilidade do LFP-s, por quadrante, variando o tamanho da amostra | | | | | |
|---|---|---|---|---|---|
| **Tamanho da amostra (pixels)** | **N° de Amostras** | **1º Quad.(%)** | **2º Quad.(%)** | **3º Quad.(%)** | **4º Quad.(%)** |
| 200x200 | 7 | 96,75 | 97,38 | 97,02 | 95,91 |
| 200x200 | 10 | 93,58 | 95,50 | 94,69 | 93,25 |
| 300x300 | 7 | 97,38 | 97,81 | 97,74 | 96,98 |
| 300x300 | 10 | 96,81 | 98,11 | 97,39 | 96,67 |

O melhor resultado da classificação das doenças da Tabela 5 foi para as amostras com o tamanho 300 x 300 pixels, com número de 7 amostras extraídas por quadrante, totalizando 2520 amostras.

A Tabela 6 apresenta os resultados da comparação dos descritores LBP e LFP-s, considerando 7 amostras extraídas com tamanho de 300 x 300 pixels. Para os descritores de Haralick são apresentados os resultados que utilizam o algoritmo IBK.

Tabela 6. Tabela de comparação dos descritores.

| Tabela de sensibilidade com a métrica da distância $X^2$ | | | | |
|---|---|---|---|---|
| | **1º Quad. (%)** | **2º Quad. (%)** | **3º Quad.(%)** | **4º Quad. (%)** |
| LBP | 98,09 | 99,12 | 99,41 | 98,25 |
| LFP-s | 97,38 | 97,81 | 97,74 | 96,98 |
| Haralick | 38,73 | 37,54 | 45,67 | 40,63 |

Os resultados de sensibilidade apresentados na Tabela 6 evidenciam que os valores de sensibilidade para os descritores LBP e LFP-s são superiores a 97% para LFP-s e 98 % para o LBP. Os descritores de Haralick resultaram em valores de sensibilidade abaixo de 45,67%.

A Tabela 7 apresenta a sensibilidade média para as classes de doenças para os descritores LBP, LFP-s e descritores de Haralick. Observa-se que o descritor LBP classifica as classes de doenças com valores de sensibilidade acima de 97,02%, enquanto que o descritor LFP-s classifica as classes de doenças com valores de sensibilidade acima de 93,39%. No entanto para as classes de doenças CVC, *greening*, magnésio,

manganês, tanto o descritor LBP quanto o descritor LFP-s apresentaram valores acima de 98% de sensibilidade, mostrando serem adequados para a diferenciação das doenças.

Tabela 7. Tabela de sensibilidade média das doenças pelo cálculo da distância $X^2$.

| Tabela de sensibilidade das doenças | | | |
|---|---|---|---|
| **Doenças** | **LBP(%)** | **LFP-s(%)** | **Haralick (%)** |
| CVC | 99,76 | 99,52 | 33,65 |
| *Greening* | 99,94 | 99,46 | 40,28 |
| Magnésio | 99,41 | 98,33 | 32,63 |
| Manganês | 98,63 | 99,05 | 41,13 |
| Rubelose | 97,56 | 93,39 | 38,45 |
| Zinco | 97,02 | 95,12 | 58,73 |

A Tabela 8 apresenta os resultados de sensibilidade média na classificação dos valores extraídos dos histogramas. O descritor LBP com o algoritmo IBK apresentou o melhor resultado na diferenciação das doenças, com valores de sensibilidades acima de 99,2% .

Tabela 8 – Tabela de sensibilidade média das classes de doenças dos valores extraídos dos histogramas.

| Tabela de sensibilidade das doenças | | | | |
|---|---|---|---|---|
| **Doenças** | **LBP IBK(%)** | **LBP RNA PMC(%)** | **LFP-s IBK(%)** | **LFP-s RNA PMC(%)** |
| CVC | 99,5 | 91,4 | 86 | 50,3 |
| *Greening* | 99,3 | 93,6 | 78,9 | 38,3 |
| Magnésio | 99,2 | 88,4 | 81,6 | 38,6 |
| Manganês | 99,4 | 89,4 | 87,8 | 37,9 |
| Rubelose | 99,6 | 90,8 | 79,5 | 23,2 |
| Zinco | 99,6 | 90,3 | 74,4 | 23,1 |

Ao comparar os resultados das Tabelas 7 e 8, observa-se que os descritores LBP e LFP-s com a distância $X^2$ seguido pelo LBP com o algoritmo IBK obtiveram os melhores valores de sensibilidade.

## V. CONCLUSÕES

Foi avaliada a metodologia baseada nos descritores de textura LBP, LFP e os descritores de Haralick, identificando se as folhas de citros estão contaminadas com o *greening* ou outras doenças e deficiências nutricionais. Foram construídos dois bancos de imagens digitalizadas de folhas de citros, contendo seis tipos de doenças (CVC, deficiência de manganês, deficiência de magnésio, deficiência de zinco, *greening* e rubelose). Os melhores resultados foram obtidos avaliando as amostras nos quadrantes, incluindo todas as amostras nos canais de cores vermelho, verde e níveis de cinza.

Os resultados da configuração que apresentou a melhor sensibilidade de diferenciação das classes de doenças foi a avaliação por quadrante, com 7 amostras extraídas para cada quadrante, para 20 imagens de folhas para cada doença, totalizando 120 imagens analisadas, nas seis classes de doenças avaliadas, para um tamanho de amostra de 300 x 300 pixels, com total de 2520 amostras.

Para o descritor LFP, a função de pertinência que apresentou uma melhor eficiência na diferenciação das amostras das imagens analisadas foi à função sigmóide.

Os descritores que apresentaram os melhores resultados foram o LBP e LFP-s pela distância $X^2$, sendo que o descritor LBP classificou as classes de doenças com valores de sensibilidade entre 97,02% e 99,94%, enquanto que para o descritor LFP-s os valores de sensibilidade ficaram entre 93,39% e 99,52%.

Os melhores resultados da classificação pelo *toolbox* Weka foram para os descritores LBP e LFP-s, utilizando o algoritmo IBK. O LBP apresentou os valores de sensibilidade entre 99,2% e 99,6% , mostrando ser eficiente na diferenciação das classes de doenças. Para o LFP-s os valores de sensibilidade ficaram entre 74,4% e 87,8%.

Os descritores de Haralick mostraram não ser adequados para a diferenciação das classes de doenças em todos os testes realizados, com valores de sensibilidade abaixo de 50%.

## AGRADECIMENTOS

Agradecemos à Embrapa Instrumentação Agropecuária e a Universidade de São Paulo - USP de São Carlos pela infra-estrutura, e ao grupo FISHER pelo fornecimento das amostras de folhas cítricas.

## REFERÊNCIAS

[1] MARTINS, M.C.; GUERZONI, R.A.; CÂMARA, G.M.S., MATTIAZZI, P., LOURENÇO, S.A. & AMORIM, L. Escala diagramática para a quantificação do complexo de doenças foliares de final de ciclo em soja. Fitopatologia Brasileira, vol.29, p.179-184, 2004.

[2] BOVÉ, J. M. Huanglongbing: a destructive, newlyemerging,century-old disease of citrus. Journal of Plant Pathology, Pisa, v. 88, p.7-37, 2006.

[3] WEKA. Disponível em: < http://www.cs.waikato.ac.nz/ml/weka/ >. Acesso em: 29 abr. 2011.

[4] INNIS, M. A.; GELFAND, D. H.; SNINSKY, J. J.; WHITE, T. J. (eds.), PCR Protocols: A guide to methods and applications, Academic Press: San Diego, 1990.

[5] NUTTER, JR., F.W.; SCHULTZ, P.M. Improving the accuracy and precision of disease assessments: selection of methods and use of computer-aided training programs. Canadian Journal of Plant Pathology, Ottawa, v.17, n.1, p.174-184, 1995.

[6] GONZALEZ, R. C.; WOODS, R. E. Processamento digital de imagens; 3 ed., São Paulo: Ed. Pearson Hall, 2010.

[7] HARALICK, R. M.; SHANMUGAN, K.; DINSTEIN, I. H. Textural Features for Image Classification, IEEE Transactions on Systems, Man and Cybernetics,Vol. 3, Issue: 6, p: 610-621, Nov. 1973.

[8] OJALA, T..; PIETIKÄINEN, M.; HARWOOD, D. A. Comparative study of texture measures with classification based on feature distribuions. **Pattern Recognition**. v.29, n., p.51-59, 1996.

[9] VIEIRA, R. T.; CHIERICI, C. E. O.; FERRAZ, C. T.; GONZAGA, A. Local Fuzzy Pattern: A New Way for Micro-pattern Analysis, IDEAL 2012, H. Yin et al. (Eds.): LNCS 7435, Springer-Verlag Berlin Heidelberg 2012, pp. 602–611, 2012.

[10] FISCHER. Disponível em: <http://www.citrosuco.com.br/fischer/fischer/sites/fischer/portal_grupo/home/index.html>. Acessado em: 28 Set. de 2011.

[11] FUNDECITRUS (2009a). Greening: manual técnico, Araraquara, 2009. Disponível em: <http://www.fundecitrus.com.br/ImageBank/PageFlip/pageflip.aspx?idPage=143>. Acesso em: 29 abril de 2011.

[12] RIBEIRO, P. P. E . ; PAIVA, M. S. V. de ; INAMASU, R. Y. ; JORGE, L. A. de C. . Aplicação de técnicas de processamentos de imagens para diferenciação do Greening de outra pragas. In: INAMASU, R. Y.; NAIME, J. de M.; RESENDE, A. V. de; BASSOI, L. H.; BERNARDI, A. de C. (Ed.).. (Org.). Agricultura de Precisão: um novo olhar. 1ed.: CUBO, 2011, v. 1, p. 73-77.

# An automatic facial expression recognition system evaluated by different classifiers

Caroline Silva*, Andrews Sobral* and Raissa Tavares Vieira†
*Programa de Pós-Graduação em Mecatrônica
Universidade Federal da Bahia,
Email: lolyne.pacheco@gmail.com
Email: andrewssobral@gmail.com
†Department of Electrical Engineering - EESC/USP
Email: raissa@ieee.org,

***Abstract*—This work proposes an automatic human-face expression recognition system that classifies seven different facial expressions: happiness, anger, sadness, surprise, disgust, fear and neutral. The experimental results show that the proposed system achieves the best hit hate using a linear discriminant classifier, 99.71% and 99.55% for MUG and FEEDTUM databases respectively.**

## I. Introduction

The automatic analysis of facial expressions has drawn attention of several researchers in different fields such as psychology, computer science, neuroscience and others [1]. In the last years, the researchers has introduced and evaluated several approaches for face detection, recognition and analysis [2]. The facial expression recognition can be used for human-computer interface (HCI) [3], computer graphics animation [4], security and surveillance [5], [6] and driver safety [7].

Paul Ekman and Friesen [8] carrying out more in-depth Darwin studies [9], have concluded that there are six transcultural and prototypic emotional expressions. These basic expressions are: happiness, anger, sadness, surprise, disgust and fear. Some authors consider the neutral face as a seventh expression. In [10], [11], [12] the authors describe the facial expression changes through a set or subset of prototypic emotional expressions.

Typically an automatic facial expression analysis (AFEA) system has three stages: face detection, feature extraction and classification. First, the face is detected, then facial features are extracted. Next, the extracted features are classified to determine their corresponding facial expression. Tian et al. [13] show that two approaches can be used for facial expression classification: frame-based or sequence-based.

The frame-based approach tries to classify the facial expression using only one image. This approach is also called frame-based - the AFEA system classifies the face expression for each video frame. Otherwise, the sequence-based approach uses temporal (or dynamical) information to estimate the facial expression. In this work we focus on the frame-based approach.

## II. Related Works

In [14] the authors have used a multiple set of Gabor features to extract the facial information. Then, the features' set were used to train an Artificial Neural Network (ANN) classifier to predict the facial expression. In [15], the authors use also an ANN classifier to classify six face expressions. The ANN is also used in the works of [16], [17], [18], [19], [20], [21]. In [12], Chuang et al. uses the Independent Component Analysis (ICA) to extract the facial features, and next, a Support Vector Machines (SVM) classifier tries to predict the facial expressions. The SVM classifier is also used in the works of [22], [23], [24], [25], [26]. Khanam et al. [27] used the fuzzy approach to classift five facial expressions: fear, surprise, happiness, sadness and anger. [28], [29], [30] shows how to perform the facial expression recognition by rule-based methods. A recent survey of AFEA systems can be found in Tian et al. [13] and De la Torre and Cohn [31].

In this work we propose a new frame-based system for automatic facial expression recognition. A block diagram of the proposed system is shown in Fig. 1. Seven facial expressions are recognized: happiness, anger, sadness, surprise, disgust, fear and neutral. The proposed system is evaluated with three classifiers: ANN, Linear Discriminant Analysis (LDA) and K-Nearest Neighbor (KNN). In the next sections we describes each step of the proposed system. First we start with the face detection method, then the features extraction process and classification. Finally, in the last section, the experimental results are shown as well as conclusions.

## III. Facial Expression Framework

First, the face detection method is presented, next the facial feature extraction and normalization are shown. In the last section, the classification step is presented.

### A. Face Detection

The first preparatory step consists to locate the face in the input image. The face detection is performed by the traditional Viola-Jones object detection framework. The Viola-Jones framework consists of two main steps: a) Haar-like features extraction and b) Adaboost classifier [32].

### B. Facial Feature Extraction

The next step consists to extract the facial features using the Active Shape Models (ASM) proposed by Cootes et al. [33]. Typically the ASM works as follows: each structured object or target is represented by a set of landmarks manually placed

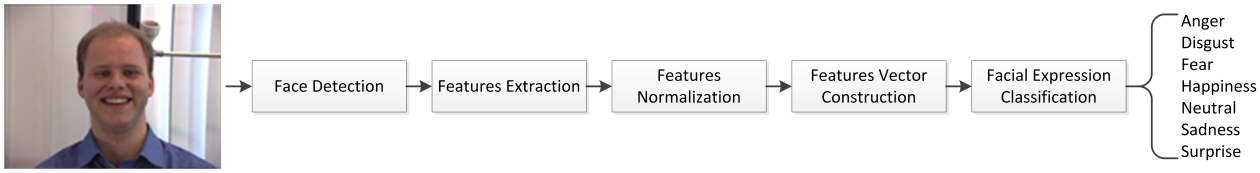
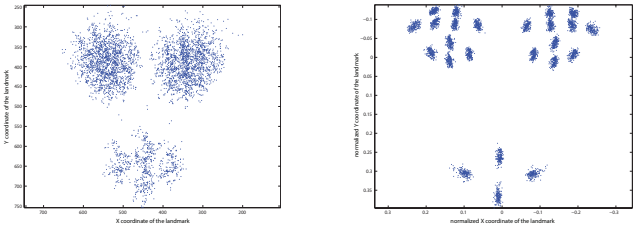



Fig. 1: Proposed system. From left to right: face detection, feature extraction, features normalization, feature vector construction and facial expression classification.

Fig. 2: The 24 facial landmarks extracted by ASM.



(a) Raw data (b) Normalization with GPA

Fig. 3: Landmarks normalization: a) shows the raw data, and b) the normalized data with GPA.

in each image of the training set. Next, the landmarks are automatically aligned to minimize the distance between their corresponding points. The ASM creates a statistical model of the facial shape which iteratively deform to fit the model in a new image.

In this work, the ASM is used to extracts 24 facial landmarks: four landmarks for the mouth; four landmarks for each eye; and six landmarks for each eye brow. An example of the facial extracted features is shown in Fig. 2. In Section IV-B we explains the method used to determine the optimal number of facial landmarks.

### C. Feature Normalization

Following the feature extraction process, the Generalized Procrustes Analysis (GPA) is used to normalize the extracted landmarks. The GPA is a statistical algorithm that eliminates the effects of scale, rotation and translation in the landmarks' set, decreasing the variations between the corresponding landmarks and searching for the best fit. The GPA was proposed by Gower [34], and lately modified by [35]. Fig. 3 shows the extracted features from the "surprise"expression in the MUG database. The Fig. 3a shows the raw data, and Fig. 3b shows the normalized data with GPA.

### D. Feature Vector Construction

To train the classifier for predict the facial expressions, first a feature vector is built by:

$$\vec{F}_{i,j} = (f_1, f_2, \ldots, f_n), f = (x, y)$$

where $i$ and $j$ denotes the i-th landmark set for the j-th facial expression, $n$ the number of extracted landmarks, and $f = (x, y)$ the Cartesian coordinate for the landmark $f$. In this work $n = 24$.

### E. Facial Expression Classification

As previously shown in the related works, several classifiers have been used to predict facial expressions. In this work, the proposed system is evaluated with three different classifiers: ANN, LDA and KNN. The goal is to determine which of the three classifiers achieves the best results for the seven facial expressions: happiness, anger, sadness, surprise, disgust, fear and neutral. In the next section, the experimental results of the proposed system are shown.

## IV. Experimental Results

First the used databases are described. Next, the experimental results for each classifier is shown.

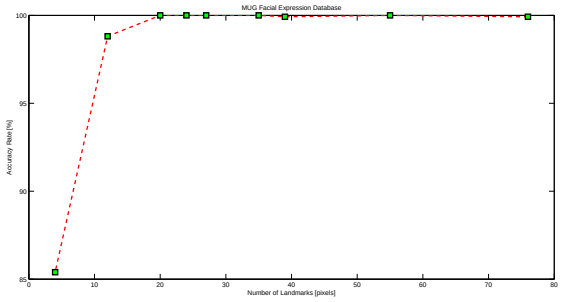
### A. Database

In this work, two facial expression databases are chosen: MUG Facial Expression database proposed by [36], and FEEDTUM database proposed by [37].

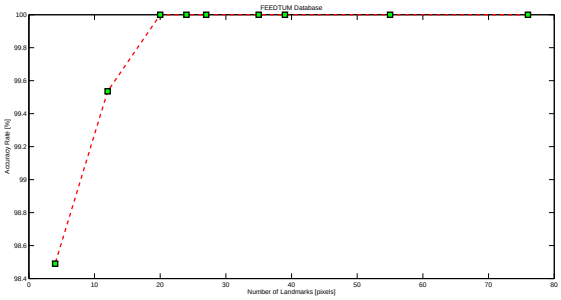
The MUG Facial Expression database contains 86 people, where 35 are women and 51 men, between 20 and 35 years. But only 52 subject images are accessible to authorized users. For each expression, a total of 50 to 160 images sequence are available.

The FEEDTUM database consists of 18 different persons (9 male and 9 female). For each of them we have seven different emotions (anger, disgust, fear, happiness, sadness, surprise and neutral) in a short video sequence.

Fig. 4: From left to right. Different sets of facial landmarks with: 4, 12, 20, 24, 27, 35, 39, 55 and 76 landmarks respectively.




(a) Results for MUG database.




(b) Results for FEEDTUM database.

Fig. 5: Optimal number of facial landmarks for a better facial expression recognition performance. The maximum hit rate is reached with 24 landmarks for both databases.

In this work, we have used 1260 images of the MUG database, 180 images for each facial expression. We have also selected 861 images form the FEEDTUM database, 123 images for each facial expression.

All selected images were divided into two parts: $2/3$ for training, and $1/3$ for testing.

### B. *Evaluation of Extracted Features*

In this section we explains the method used to determine the optimal number of facial landmarks. The goal is to demonstrate which the smaller number of facial landmarks is sufficient for a better facial expression recognition performance. First, 9 sets of facial landmarks are extracted, containing: 4, 12, 20, 24, 27, 35, 39, 55 and 76 landmarks respectively (see Fig. 4). All landmarks were extracted using ASM tool. Subsequently the LDA classifier is used to classify seven facial expressions: happiness, anger, sadness, surprise, disgust, fear and neutral. The LDA was chosen for its best performance explained in the next section. Figs. 5a and 5b shows the results with MUG and FEEDTUM databases respectively. As can be seen in Figs. 5a and 5b, the maximum accuracy ratio is reached with 24 landmarks.

### C. *Features Classification*

In this work, the proposed facial expression recognition framework is evaluated with three classifiers: ANN, LDA and KNN. In the next sections, the results for each classifier are shown.

TABLE I: Results from MUG and FEEDTUM databases using ANN classifier. Yellow row shows the best hit rate for each database.

| **MUG database** | | **FEEDTUM database** | |
|---|---|---|---|
| **# of HN** | **Hit rate (%)** | **# of HN** | **Hit rate (%)** |
| 3 | 56.58 | 3 | 97.90 |
| 10 | 93.17 | 10 | 90.59 |
| 12 | 99.04 | 12 | 92.10 |
| 14 | 84.60 | 14 | 94.30 |
| 16 | 69.12 | 16 | 88.96 |
| 18 | 96.66 | 18 | 89.77 |
| 20 | 99.28 | 20 | 90.70 |

* HN - Hidden Neurons.

*1) Multi-Layer Perceptrons:* To perform the classification with neural networks, the feed-forward MLP network was chosen. With hidden layers, the MLP networks are able to solve non-linearly separable data.

The MLP network was configured as follows: a) the input layer has 48 neurons: 24 landmarks times 2 (x and y positions of the Cartesian coordinate system); b) the hidden layer evaluation are made with 3, 10, 12, 14, 16, 18 and 20 neurons; c) the output layer contains 7 neurons, one for each facial expression. The training algorithm used in this work is the CGP (Conjugate gradient backpropagation with Polak-Ribiere updates) proposed by Polak and Ribiere (1969) [38]. In our experiments, the CGP has achieved better results

TABLE II: Results from MUG and FEEDTUM databases using LDA classifier. Yellow row shows the best hit rate for each database.

| MUG database | | FEEDTUM database | |
|---|---|---|---|
| **DF** | **Hit Rate (%)** | **DF** | **Hit Rate (%)** |
| linear | 99.71 | linear | 99.55 |
| diagLinear | 62.34 | diagLinear | 46.34 |
| diagQuadratic | 60.07 | diagQuadratic | 45.52 |

* DF - Discriminative Function.

than tradicional backpropagation algorithm. All neurons use the same sigmoid activation function.

Table I shows the results for MUG and FEEDTUM databases respectively. As can be seen in Table I, for each database, different results are shown for each number of hidden neurons. For MUG database the best results is achieved with 20 hidden neurons. However, for FEEDTUM database the best hit rate is achieved with 3 hidden neurons. In addition, for MUG database the ANN have large hit rate variation relative to FEEDTUM database.

*2) Linear Discriminant Analysis:* The Linear Discriminant Analysis is a multivariate statistical tool used for feature classification. The LDA tries to identify which variables (or their combinations) get the best discriminate boundary between two or more groups [39].

Table II shows the results for MUG and FEEDTUM databases respectively. As can be seen in Table II, for each database the best result is given using a linear discriminative function. Differently from ANN classifier, the hit rate variation is more constant over the two databases. In addition, the LDA classifier get superior results than ANN classifier.

*3) K-Nearest Neighbor:* The KNN is a non-parametric tool used for classification and regression. For classification case, an object is classified by a majority vote of its neighbors, with the object being assigned to the class most common among its k nearest neighbors, where k is a positive integer. The best choice of k depends upon the data. Generally, larger values of k reduce the effect of noise on the classification.

In this work, the Euclidean distance is chosen. Table III shows the results for MUG and FEEDTUM databases respectively. As can be seen in Table III, the hit rate is smaller than ANN and LDA classifiers. The best hit rate for MUG and FEEDTUM databases is given for $k = 3$ and $k = 1$, respectively.

## V. Discussion

The experimental results have shown that the LDA classifier has the best hit hate: 99.7% for MUG database and 99.5% for FEEDTUM database. In addition, LDA is less sensitive than ANN classifier. As we saw in experimental results, the ANN shows higher hit hate variations given the number of hidden neurons, an issue that is absent in LDA classifier. Moreover, in Section IV-B we have shown that LDA gets high hit rate starting with 24 landmarks. However, KNN has left a great deal to be desired, getting inferior results than ANN and LDA classifiers.

TABLE III: Results from MUG and FEEDTUM databases using KNN classifier. Yellow row shows the best hit rate for each database.

| MUG database | | FEEDTUM database | |
|---|---|---|---|
| **k** | **Hit Rate (%)** | **k** | **Hit Rate (%)** |
| 1 | 75.23 | 1 | 54.70 |
| 3 | 76.42 | 3 | 48.78 |
| 5 | 75.71 | 5 | 49.59 |
| 7 | 75.23 | 7 | 51.68 |
| 9 | 75.95 | 9 | 50.75 |

* k - number of neighbors.

## VI. Conclusions

This paper has presented a frame-based facial expression recognition system evaluated with three well-know classifiers: ANN, LDA and KNN. First, 24 facial landmarks are extracted after face detection step. Next, all landmarks are normalized with GPA tool. The next step consists to classify all extracted features in seven facial expression classes: happiness, anger, sadness, surprise, disgust, fear and neutral. As we saw in experimental results, the LDA classifier has the best hit hate than ANN and KNN classifiers. In addition, we have shown that LDA gets a high and stable hit rate starting with 24 landmarks.

## References


[1] A. C. Koutlas and D. I. Fotiadis, "An automatic region based methodology for facial expression recognition," in *IEEE Int. Conf. on Systems, Man and Cybernetics*, 2008, pp. 662–666.

[2] S. Moore and R. Bowden, "The effect of pose on facial expression recognition," in *Proc. of the British Machine Vision Conf.*, 2009, pp. 1–1.

[3] M. Pantic and L. Rothkrantz, "Towards an affect-sensitive multimodal human-computer interaction," *Proc. of the IEEE*, vol. 91, no. 9, pp. 1370–1390, 2003.

[4] Z. Zeng, M. Pantic, G. I. Roisman, and T. S. Huang, "A survey of affect recognition methods: Audio, visual, and spontaneous expressions," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 1, pp. 39–58, 2009.

[5] G. Bhaumik, T. Mallick, K. S. Chowdhury, and G. Sanyal, "Analysis and detection of human faces by using minimum distance classifier for surveillance," *Int. Conf. on Recent Trends in Information, Telecommunication and Computing*, pp. 265–267, 2010.

[6] A. A. M. Al-modwahi, O. Sebetela, L. N. Batleng, B. Parhizkar, and A. Lashkari, "Facial expression recognition intelligent security system for real time surveillance," in *Proc. of World Congress in Computer Science, Computer Engineering, and Applied Computing*, 2012.

[7] E. Vural, M. Cetin, A. Ercil, G. Littlewort, M. Bartlett, and J. Movellan, "Automated drowsiness detection for improved driver safety-comprehensive databases for facial expression analysis," in *Proc. of the Int. Conf. on Automotive Technologies*, 2008.

[8] P. Ekman and W. V. Friesen, "Constants across cultures in the face and emotion," *Journal of Personality and Social Psychology*, vol. 17, no. 2, pp. 124–129, 1971.

[9] C. Darwin, *The Expression of the Emotions in Man and Animals*, anniversary ed., P. Ekman, Ed. Harper Perennial, 1872.

[10] Y.-l. Tian, T. Kanade, and J. F. Cohn, "Recognizing action units for facial expression analysis," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 23, no. 2, pp. 97–115, feb 2001.

[11] P. Michel and R. E. Kaliouby, "Real time facial expression recognition in video using support vector machines," 2003.

[12] C.-F. Chuang and F. Y. Shih, "Recognizing facial action units using independent component analysis and support vector machine," *Pattern Recognition*, vol. 39, no. 9, pp. 1795–1798, 2006.

[13] Y.-L. Tian, T. Kanade, and J. Cohn, "Facial expression recognition," in *Handbook of Face Recognition*, S. Z. Li and A. K. Jain, Eds. Springer, 2011, pp. 487–519.

[14] Y. Wang, H. Ai, B. Wu, and C. Huang, "Real time facial expression recognition with adaboost," in *17th Int. Conf. on Pattern Recognition*. IEEE Computer Society, 2004, pp. 926–929.

[15] H. Kobayashi and F. Hara, "Facial interaction between animated 3d face robot and human beings," in *IEEE Int. Conf. on Systems, Man, and Cybernetics*, vol. 4, oct 1997, pp. 3732–3737.

[16] M. Wang, Y. Iwai, and M. Yachida, "Expression recognition from time-sequential facial images by use of expression change model," in *Third IEEE Int. Conf. on Automatic Face and Gesture Recognition*, 1998, pp. 324–329.

[17] Z. Zhang and Z. Zhang, "Feature-based facial expression recognition: Sensitivity analysis and experiments with a multilayer perceptron," *Int. Journal of Pattern Recognition and Artificial Intelligence*, vol. 13, pp. 893–911, 1999.

[18] B. Fasel, F. Monay, and D. Gatica-Perez, "Latent semantic analysis of facial action codes for automatic facial expression recognition," in *Proc. of the 6th ACM SIGMM Int. workshop on Multimedia information retrieval*, ser. MIR '04. New York, NY: ACM, 2004, pp. 181–188.

[19] N. Sebe, M. Lew, Y. Sun, I. Cohen, T. Gevers, and T. Huang, "Authentic facial expression analysis," *Image and Vision Computing*, vol. 25, no. 12, pp. 1856 – 1863, 2007.

[20] Y. Sun, Z. Li, C. Tang, W. Zhou, and R. Jiang, "An evolving neural network for authentic emotion classification," in *Proc. of the 2009 Fifth Int. Conf. on Natural Computation*, ser. ICNC '09. Washington, DC: IEEE Computer Society, 2009, pp. 109–113.

[21] A. A. A. Youssif and W. A. A. Asker, "Automatic facial expression recognition system based on geometric and appearance features," *Computer and Information Science*, pp. 115–124, 2011.

[22] G. Littlewort, I. Fasel, M. S. Bartlett, and J. R. Movellan, "Fully automatic coding of basic expressions from video," California Institute of Technology, Tech. Rep., May 2002.

[23] M. S. Bartlett, G. Littlewort, I. Fasel, and J. R. Movellan, "Real time face detection and facial expression recognition: Development and applications to human computer interaction," in *Proc. of the CVPR Workshop on Computer Vision and Pattern Recognition for Human-Computer Interaction*, 2003.

[24] I. Kotsia, N. Nikolaidis, and I. Pitas, "Facial expression recognition in videos using a novel multi-class support vector machines variant," in *IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, 2007.

[25] G. Lei, X. Li, J. Zhou, and X. gang Gong, "Geometric feature based facial expression recognition using multiclass support vector machine," in *IEEE Int. Conf. on Granular Computing*. IEEE Int. Conf. on Granular Computing, pp. 318–321.

[26] P. Lucey, J. Cohn, T. Kanade, J. Saragih, Z. Ambadar, and I. Matthews, "The extended cohn-kanade dataset (ck+): A complete dataset for action unit and emotion-specified expression," in *EEE Computer Society Conf. on Computer Vision and Pattern Recognition Workshops*, june 2010, pp. 94–101.

[27] A. Khanam, M. Shafiq, and M. Akram, "Fuzzy based facial expression recognition," in *Image and Signal Processing, 2008. CISP '08. Congress on*, vol. 1, may 2008, pp. 598 –602.

[28] M. Pantic, S. Member, and L. J. M. Rothkrantz, "Automatic analysis of facial expressions: The state of the art," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 22, pp. 1424–1445, 2000.

[29] M. Pantic and L. J. Rothkrantz, "Facial action recognition for facial expression analysis from static face images," *Trans. Sys. Man Cyber. Part B*, vol. 34, no. 3, pp. 1449–1461, Jun. 2004.

[30] M. Mufti and A. Khanam, "Fuzzy rule based facial expression recognition," in *Int. Conf. on Intelligent Agents, Web Technologies and Internet Commerce*, 2006, pp. 57–57.

[31] F. De la Torre and J. F. Cohn, "Facial expression analysis," in *Visual Analysis of Humans*. Springer, 2011, pp. 377–409.

[32] P. Viola and M. Jones, "Robust real-time object detection," in *Int. Journal of Computer Vision*, vol. 57, 2001, pp. 137–154.

[33] T. F. Cootes, G. Edwards, and C. J. Taylor, "Comparing active shape models with active appearance models," in *Proc. of British Machine Vision Conf.*, 1999, pp. 173–182.

[34] J. Gower, "Generalized procrustes analysis," *Psychometrika*, vol. 40, pp. 33–51, 1975.

[35] J. Ten Berge, "Orthogonal procrustes rotation for two or more matrices," *Psychometrika*, vol. 42, pp. 267–276, 1977.

[36] N. Aifanti, C. Papachristou, and A. Delopoulos, "The MUG facial expression database," in *11th Int. Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS)*, april 2010, pp. 1–4.

[37] F. Wallhoff, "Database with Facial Expressions and Emotions from Technical University of Munich (FEEDTUM)," 2006.

[38] R. G. Polak, E., "Note sur la convergence de methodes de directions conjuguees," *ESAIM: Mathematical Modelling and Numerical Analysis - Modelisation Mathematique et Analyse Numerique*, vol. 3, no. R1, pp. 35–43, 1969. [Online]. Available: http://eudml.org/doc/193115

[39] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. fisherfaces: Recognition using class specific linear projection," 1997.

# Facial expression recognition in static images by generalized procrustes analysis

Caroline Silva*, Andrews Sobral* and Raissa Tavares Vieira†
*Programa de Pós-Graduação em Mecatrônica
Universidade Federal da Bahia,
Email: lolyne.pacheco@gmail.com
Email: andrewssobral@gmail.com
†Department of Electrical Engineering - EESC/USP
Email: raissa@ieee.org,

***Abstract*—This work proposes a framework for facial expression recognition based on generalized procrustes analysis. The proposed system classifies seven different facial expressions: happiness, anger, sadness, surprise, disgust, fear and neutral. The proposed system was evaluated with the MUG Facial Expression database. Experimental results shows that the proposed method achieves 97.78% of recognition rate.**

## I. Introdução

Nos últimos anos, sistemas automáticos para a análise de expressões faciais tornaram-se cada vez mais importantes na comunidade de pesquisa de visão computacional. Em muitos casos, observa-se que as expressões faciais descrevem os estados emocionais de uma pessoa, com maior precisão do que as palavras. O ser humano possui essa capacidade de interação com seu semelhante independente da utilização de uma linguagem em comum através da expressão. Consequentemente, sistemas automáticos de reconhecimento de expressões faciais podem trazer melhorias no que diz respeito à interação homem-máquina, de modo que, por exemplo, máquinas possam perceber o comportamento humano e reagirem proativamente. Outras possíveis aplicações encontram-se na área de segurança e vigilância, realidade aumentada, diagnóstico médico, sistema de percepção, entre outras [1], [2].

O britânico Charles Darwin foi o precursor em relacionar as expressões faciais com o ser humano, evidenciando a importância da face como um indicativo social para a garantia da sobrevivência do homem no processo de seleção natural. As ideias inovadoras de Darwin [3] despertaram o interesse de muitos pesquisadores que começaram a desenvolver pesquisas neste campo. Um dos psicólogos que se destaca é Paul Ekman, autor do livro "*A Linguagem das Emoções*" e seu colega Friesen, que em 1971, aprofundando os estudos de Darwin, concluíram que existem emoções básicas universais e transculturais, ou seja, que não precisam ser ensinadas e que não variam entre grupos sociais. Essas expressões básicas são: felicidade, raiva, tristeza, surpresa, desgosto e medo; que são inatas na natureza humana e, independem da cultura, do país, da raça ou dos aprendizados adquiridos.

A fim de padronizar o processo de reconhecimento das expressões faciais, foram desenvolvidos diversos sistemas de codificação da ação facial. Dentre estes se destaca o *Facial Action Coding System* (FACS) criado por Ekman e Friesen em 1978 que descreve as expressões faciais em unidades de ação (*Action Units* - AU's). Das 44 AU's definidas, 30 são anatomicamente relacionadas com a contração de músculos faciais específicos (12 AU's para a parte superior da face, e 18 AU's para a parte inferior da face), que podem corresponder a um músculo específico ou a um grupo muscular. O FACS vem sendo bastante utilizado como uma ferramenta descritiva e tem sido empregado em vários trabalhos científicos relacionados a micro-expressões.

Segundo [4], uma das primeiras investigações sobre reconhecimento automático de expressões faciais foi apresentada por [5]. Os autores analisaram as expressões faciais, rastreando o movimento de 20 pontos identificados em uma sequência de imagens. A partir deste trabalho, muitos progressos têm sido realizados, devido ao desenvolvimento de novos métodos de processamento de imagens, novas abordagens para detecção e reconhecimento facial, bem como o aumento da capacidade computacional.

Desde 1978, muitos sistemas de reconhecimento de expressões faciais têm sido desenvolvidos, tais como em [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], [14], [16], [17]. Estes sistemas são similares no sentido de que a maioria inicialmente extrai características faciais, e em seguida as utilizam para classificar diferentes categorias de expressões. O diferencial destes sistemas encontra-se no tipo de informação discriminativa extraída da imagem e pelo procedimento de classificação utilizado.

Segundo [18] os métodos de classificação de expressões podem ser normalmente baseados em imagens estáticas ou baseados em sequência de imagens. Métodos baseados em imagens estáticas utilizam apenas o *frame* corrente, com ou sem uma imagem de referência para reconhecer as expressões apresentadas nas imagens. Por outro lado, os baseados em sequências de imagens utilizam informações temporais de pelo menos dois exemplos de uma face no mesmo estado emocional. Uma revisão de diferentes abordagens para sistemas de reconhecimentos de expressões faciais são encontrados em [18] e [19].

Neste trabalho é proposto um sistema para reconhecer expressões faciais baseado em imagens estáticas. O sistema classifica as expressões universais definidas por [20]: felicidade, raiva, tristeza, surpresa, desgosto, medo; incluindo a expressão neutra. Nas próximas seções, são descritas as etapas desenvolvidas neste trabalho para classificar cada uma das

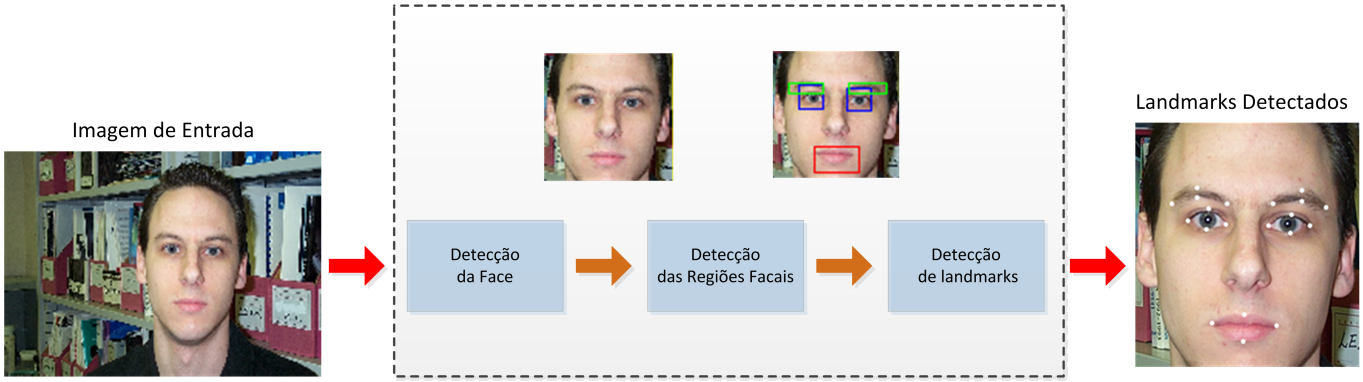



Fig. 1: Processo para localização dos *landmarks* faciais.

expressões. Na última seção, os resultados experimentais são apresentados, bem como as conclusões.

## II. Extração de Características Faciais

Neste trabalho o tipo de característica extraída para representar cada uma das expressões foi um conjunto de *landmarks* faciais localizados na região da face. Estes *landmarks* foram extraídos a partir de um conjunto de imagens.

A primeira etapa para detectar os *landmarks* é localizar a face na imagem. Para isto foi utilizado um detector de Viola-Jones que é baseado em Haar-like-features como extrator de características (features) e Adaboost como classificador [21]. A face detectada representa uma região de interesse na imagem para o detector de landmarks. O método de [21] também foi utilizado para localizar a região dos olhos e da boca. A região da sobrancelha foi estimada a partir da localização da região dos olhos.

Após a detecção dos ROIs, a próxima etapa foi localizar os *landmarks* na face. Para localizar pontos de interesse na face, inicialmente foi preciso melhorar e extrair informações relevantes na imagem. Para isto, neste trabalho assim como em [22] foram utilizadas diferentes técnicas de processamento de imagens. No trabalho de [22] foram utilizadas as seguintes técnicas: ajuste de contraste, limiarização, subtração de plano de fundo, detecção de contorno e filtros Laplaciano e Gaussiano para em seguida, extrair pontos de interesse. Porém, neste trabalho além de utilizar algumas técnicas utilizadas por [22] outras técnicas de processamento de imagens também foram aplicadas com intuito de obter melhores resultados.

As técnicas de processamento de imagens utilizadas neste trabalho foram: (a) equalização de histograma para uniformizar os valores dos níveis de cinza na imagem; (b) um filtro gaussiano foi utilizado com a finalidade de suavizar a imagem para descartar pequenos ruídos como falhas na figura e linhas pouco evidentes; (c) o contraste foi ajustado para realçar a imagem; (d) para segmentar uma imagem em regiões (conjunto de pixels) foram utilizados dois métodos de segmentação: limiarização e operações morfológicas; (e) para extrair as bordas dos olhos, sobrancelhas e da boca foi utilizado o detector Canny e em seguida, foi realizada uma busca em cada uma das bordas resultantes, com o objetivo de detectar 20 *landmarks* faciais. As técnicas de processamento de imagens utilizadas e a busca pelos *landmarks* são descritas detalhadamente em um artigo publicado pelos autores. A Figura 1 ilustra o processo para localização de 20 *landmarks* faciais distribuídos por toda a face para descrever as formas das regiões faciais (das sobrancelhas, dos olhos e da boca).

## III. Classificação de Expressões Faciais

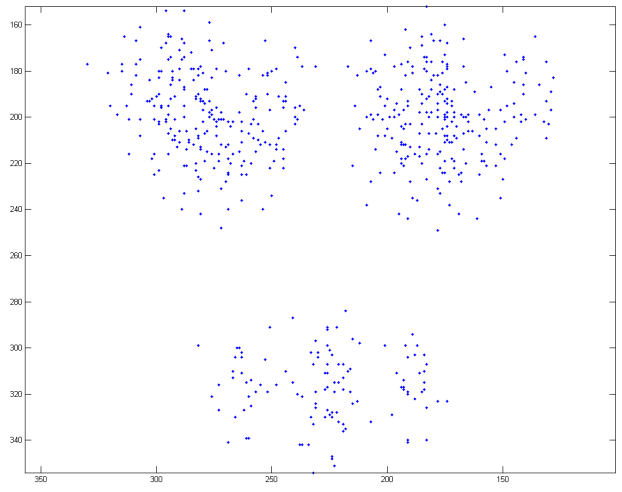
Neste trabalho a abordagem escolhida para classificar as expressões é chamada Análise Generalizada de Procrustes. Inicialmente, é criado um modelo médio para cada uma das seguintes expressões: felicidade, raiva, tristeza, surpresa, desgosto, medo e neutra. Em seguida, é calculado o grau de similaridade entre os modelos estimados. A seguir o método de classificação é descrito em detalhes.

### A. *Análise Generalizada de Procrustes (AGP)*

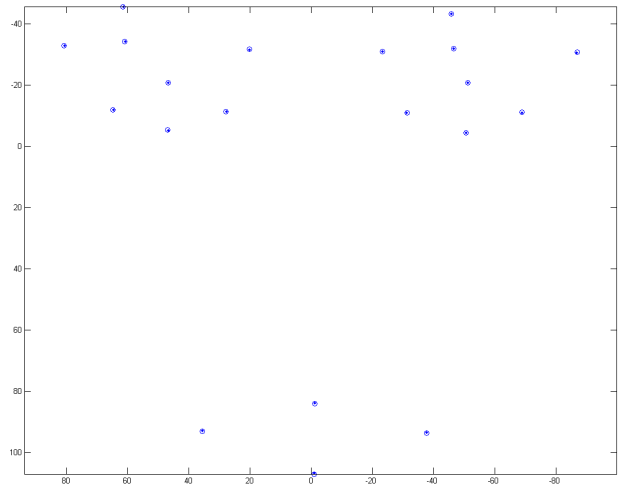
Análise Generalizada de Procrustes (AGP) desenvolvida por [23], e aprimorada por [24] é um algoritmo estatístico que elimina os efeitos de escala, rotação e translação dos objetos, conseguindo quantificar as variações entre os *landmarks* correspondentes, buscando o ajuste para a melhor sobreposição das configurações geométricas. A configuração consiste em um conjunto de *landmarks* de um determinado objeto [25].

O AGP baseia-se em alinhar sequencialmente pares de formas utilizando uma forma de referência (a forma média) e alinha as outras formas para a forma média. Inicialmente uma forma aleatória do conjunto é escolhida para ser a forma média. Após o alinhamento, uma nova estimativa para a forma média é calculada, e novamente, as formas são alinhados para a forma média. Este procedimento é realizado repetidamente até que a forma média não mude significativamente durante as interações [26]. A Figura 2 mostra os resultados deste procedimento de alinhamento. O lado esquerdo da imagem mostra exemplo de dados brutos extraídos de uma sequência de imagem de uma única expressão e o lado direito corresponde a forma média após a aplicação do AGP.

Após os modelos médios serem estimados, é calculado o grau de similaridade entre a forma média (de cada uma das expressões) e a forma obtida da imagem de entrada. O modelo que apresentar a menor distância de *Procrustes* é considerado a expressão apresentada na imagem. O valor da distância de *Procrustes* varia entre 0 e 1, ou seja, quando



(a)



(b)

Fig. 2: (a) Exemplo de dados brutos extraídos de uma sequência de imagem da expressão neutra (b) a forma média após a aplicação do AGP.

mais próximo de zero mais similar é a forma comparada. Para calcular o valador da distância de *Procrustes* é realizada uma transformação linear em uma forma para encontrar a melhor correspondência entre duas formas. O objetivo é encontrar o grau de similaridade entre duas formas [25].

### B. Base de Dados

A base de dados MUG Facial *Expression* [27] é composta por sequências de imagens de 86 indivíduos que executam diferentes expressões faciais. As imagens possuem uma resolução de 896x896 *pixels* e iluminação uniforme. Dos 86 indivíduos, 35 são mulheres e 51 homens, todos de origem caucasiana entre 20 e 35 anos de idade. Alguns homens apresentam barbas e não existem oclusões (exceto uma parte do cabelo sobre a face). Porém dos 86 indivíduos, apenas imagens de 52 indivíduos são acessíveis para usuários autorizados,

Fig. 3: Base de dados MUG Expression Facial.

sendo que imagens de 25 indivíduos estão disponíveis mediante solicitação e 9 estão disponíveis apenas ao laboratório MUG. Esta base de dados consiste de duas partes. Na primeira parte, os participantes foram orientados a realizar as *expressões básicas universais* que são: felicidade, raiva, tristeza, surpresa, desgosto, medo; incluindo a expressão neutra. A segunda parte é composta emoções laboratoriais induzidas.

### C. Metodologia Utilizada

A metodologia utilizada para avaliar a classificação consiste em inicialmente formar três conjuntos de imagens. Sendo que cada conjunto foi composto por um total de 30 imagens para cada uma das sete classes de expressões (felicidade, raiva, tristeza, surpresa, desgosto, medo e neutra). Na sequencia, dois terços das imagens foram selecionadas para compor o conjunto de treinamento e um terço para compor o conjunto de teste. Esta metodologia é chamada de validação cruzada. A Figura 4 ilustra a metodologia para avaliar a etapa de classificação do presente trabalho.

Os três conjuntos foram representados pelas letras A, B e C e foram utilizados para treinar e avaliar o desempenho da abordagem utilizada neste trabalho para classificar sete diferentes expressões. Na próxima subseção são apresentados os resultados obtidos utilizando a metodologia descrita.

### D. Resultados da Classificação

Para avaliar o sistema proposto, foram realizados testes em relação à taxa média de acerto entre 3 a 7 expressões, e através dos resultados experimentais, foi possível observar que a taxa de acerto deste método diminuiu à medida que aumentou-se a quantidade de expressões. Por exemplo, para as expressões: felicidade, neutra e surpresa o sistema atingiu uma taxa média de acerto de 97,78%. Para 4 expressões (felicidade, neutra, surpresa e tristeza) a média foi de 73,33% de acertos, enquanto

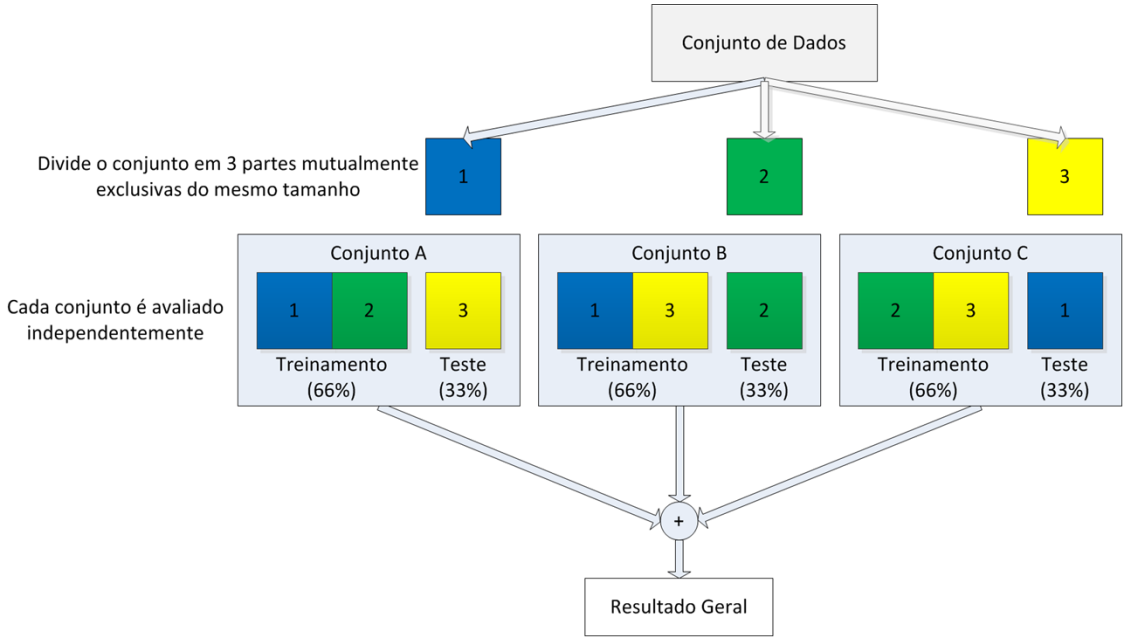



Fig. 4: Metodologia para avaliar a etapa de classificação.

TABLE I: Resultados da classificação baseada em correspondência entre modelos utilizando a base MUG *Expression Facial*

| | | Expressões | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **# de Expressões** | **% de Acerto** | **Neutra** | **Felicidade** | **Surpresa** | **Tristeza** | **Raiva** | **Desgosto** | **Medo** |
| **3** | 97,78 | 96,67 | 100,00 | 96,67 | - | - | - | - |
| **4** | 73,33 | 83,33 | 100,00 | 96,67 | 13,33 | - | - | - |
| **5** | 69,33 | 73,33 | 100,00 | 96,67 | 13,33 | 63,33 | - | - |
| **6** | 66,67 | 73,33 | 93,33 | 96,67 | 13,33 | 56,67 | 66,67 | - |
| **7** | 52,38 | 70,00 | 93,33 | 26,67 | 13,33 | 56,67 | 66,67 | 40,00 |

que para 5 expressões (felicidade, neutra, surpresa e tristeza e raiva) a média foi de 69,33%, e para 6 expressões (felicidade, neutra, surpresa e tristeza, raiva e desgosto) uma média de 66,67% de acertos. Entretanto, quando uma sétima expressão é adicionada, no caso o medo, o sistema alcançou uma taxa média de reconhecimento de 52,38%. Conforme mostrado na Tabela 1.

Na próxima seção são apresentadas as considerações finais do presente trabalho.

## IV. Conclusão

Neste trabalho é apresentado um sistema automático de reconhecimento de expressões faciais que reconhece sete diferentes classes de expressões (felicidade, raiva, tristeza, surpresa, desgosto, medo e neutra). O sistema foi treinado e testado utilizando a base de dados MUG *Facial Expression* em que as imagens apresentam iluminação uniforme e planos de fundo neutro. As imagens utilizadas pelo sistema estão restritas a ambientes fechados.

No presente trabalho, foi utilizada a abordagem AGP para classificar as expressões. Os resultados experimentais demonstram que a melhor taxa de reconhecimento apresentada pelo sistema foi obtida quando as expressões: felicidade, neutra e surpresa foram utilizadas atingindo uma taxa média de acerto de 97,78%, porém a medida que novas expressões são adicionadas ao sistema a taxa de reconhecimento diminui. Isso, pode ter occorrido devido ao fato que, o processo de estimação do modelo médio pode suavizar alguns detalhes faciais eliminando algumas características importantes de cada expressão. Além disso, o método de extração de características utilizado apesar de apresentar uma taxa de precisão acima de 90% para os *landmarks* de algumas regiões da face (conforme apresentado no artigo publicado pelos autores), ainda assim algumas melhorias devem ser realizadas para que o método apresente melhores resultados, por exemplo, nos *landmarks* detectados na região dos olhos.

Como trabalho futuro, pretende-se avaliar a abordagem de classificação proposta utilizando outros métodos de extração de características, por exemplo, Point Distribution Models (PDMs), Active Shape Model (ASM), Active Appearance model (AMM), entre outros. Além disso, testar e treinar o sistema com outras bases de dados.

## Referências

[1] C. Shan, "Inferring facial and body language," Tech. Rep. RR-08-01, February 2008, submitted for the degree of Doctor of Philosophy of the University of London.

[2] N. Hesse, "Multi-view facial expression classification," Ph.D. dissertation, Karlsruhe Institute of Technology, March 2011.

[3] C. Darwin, *The Expression of the Emotions in Man and Animals*, anniversary ed., P. Ekman, Ed. Harper Perennial, 1872.

[4] B. Fasel and J. Luettin, "Automatic facial expression analysis: a survey," *Pattern Recognition*, vol. 36, pp. 259–275, 2003.

[5] M. Suwa, N. Sugie, and k.Fujimora, "A preliminary note on pattern recognition of human emotional expression," *International Joint Conference on Pattern Recognition*, pp. 408–410, 1978.

[6] M. J. Black and Y. Yacoob, "Recognizing facial expressions in image sequences using local parameterized models of image motion," *International Journal of Computer Vision*, vol. 25, pp. 23–48, 1997.

[7] G. Donato, M. S. Bartlett, J. C. Hager, P. Ekman, and T. J. Sejnowski, "Classifying facial actions," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 10, pp. 974–989, 1999.

[8] M. Pantic and L. J. M. Rothkrantz, "Expert system for automatic analysis of facial expressions," 2000.

[9] Y.-l. Tian, T. Kanade, and J. F. Cohn, "Evaluation of gabor-wavelet-based facial action unit recognition in image sequences of increasing complexity," in *Proceedings of the Fifth IEEE International Conference on Automatic Face and Gesture Recognition*, ser. FGR '02. Washington, DC, USA: IEEE Computer Society, 2002, pp. 229–.

[10] P. Michel and R. E. Kaliouby, "Real time facial expression recognition in video using support vector machines," 2003.

[11] C.-F. Chuang and F. Y. Shih, "Recognizing facial action units using independent component analysis and support vector machine," *Pattern Recognition*, vol. 39, no. 9, pp. 1795–1798, Sep. 2006.

[12] Z. Zeng, Y. Fu, G. I. Roisman, Z. Wen, Y. Hu, and T. S. Huang, "Spontaneous emotional facial expression detection," *Journal of Multimedia*, vol. 1, pp. 1–8, 2006.

[13] M. S. Bartlett, G. C. Littlewort, M. G. Frank, C. Lainscsek, I. R. Fasel, and J. R. Movellan, "Automatic recognition of facial actions in spontaneous expressions," 2006.

[14] A. Ryan, J. Cohn, S. Lucey, J. Saragih, P. Lucey, F. De la Torre, and A. Rossi, "Automated facial expression recognition system," in *Security Technology, 2009. 43rd Annual 2009 International Carnahan Conference on*, oct. 2009, pp. 172 –177.

[15] K. Saket, R. Narender, and H. S, "Facial expression (mood) detection from facial images using committee neural networks," *Biomedical Engineering Online, V8, 16*, 2009.

[16] B. Jiang, M. F. Valstar, and M. Pantic, "Action unit detection using sparse appearance descriptors in space-time video volumes," in *IEEE International Conference on Automatic Face & Gesture Recognition and Workshops, FG 2011*. USA: IEEE Computer Society, March 2011, pp. 314–321.

[17] P. Visutsak, "Emotion classification using adaptive svms," *International Journal of Computer and Information Engineering*, 2012.

[18] Y. li Tian, T. Kanade, and J. F. Cohn, *Handbook of Face Recognition*. Springer, 2005.

[19] F. De la Torre and J. F. Cohn, *Guide to Visual Analysis of Humans: Looking at People*. Springer, 2011, ch. Facial Expression Analysis.

[20] P. Ekman and W. V. Friesen, "Constants across cultures in the face and emotion," *Journal of Personality and Social Psychology*, vol. 17, no. 2, pp. 124–129, 1971.

[21] P. Viola and M. Jones, "M. robust real-time object detection," in *International Journal of Computer Vision*, 2001.

[22] A. Sayeed, M. Sohail, and P. Bhattacharya, "Detection of facial feature points using anthropometric face model," 2006.

[23] J. Gower, "Generalized procrustes analysis," *Psychometrika*, vol. 40, pp. 33–51, 1975.

[24] J. Ten Berge, "Orthogonal procrustes rotation for two or more matrices," *Psychometrika*, vol. 42, pp. 267–276, 1977.

[25] I. Dryden and K. Mardia, *Statistical shape analysis*, ser. Wiley series in probability and statistics. Chichester [u.a.]: Wiley, 1998.

[26] F. L. Bookstein, "Landmark Methods for Forms Without Landmarks: Localizing Group Differences in Outline Shape," in *MMBIA '96: Proceedings of the 1996 Workshop on Mathematical Methods in Biomedical Image Analysis (MMBIA '96)*. Washington, DC, USA: IEEE Computer Society, 1996.

[27] N. Aifanti, C. Papachristou, and A. Delopoulos, "The mug facial expression database," in *Image Analysis for Multimedia Interactive Services (WIAMIS), 2010 11th International Workshop on*, april 2010, pp. 1 –4.

# Roads and Trajectories Detection in Agricultural Environment Using CatadioptricVision System

Victor de Assis Rodrigues[1], Vinicius Bergoli Trentini[2]
Programa de Pós Graduação em Ciência da Computação
Universidade Estadual Paulista "Júlio de Mesquita Filho"
São José do Rio Preto, SP, Brasil
victorassisrd@gmail.com[1], vinicius_trentini@hotmail.com[2]

Luciano Cássio Lulio[3], Mário Luiz Tronco[4]
Escola de Engenharia De São Carlos
Universidade de São Paulo – EESC/USP
São Carlos, SP, Brasil
luciano.lulio@gmail.com[3], mltronco@sc.usp.br[4]

***Resumo*— Esse trabalho utiliza um sistema de visão catadióptrico para capturar imagens de um cenário natural agrícola e realizar o levantamento de regiões que possam auxiliar diversas aplicações na área da Robótica Móvel Agrícola. Os sistemas de visão catadióptricos buscam capturar uma imagem 360° do ambiente a partir da combinação de lentes e espelhos. A imagem omnidirecional é retificada e seus quadrantes são extraídos, originando quatro novas imagens que representam lados de visão do veículo. Uma etapa de segmentação por cor é proposta utilizando como base o algoritmo Otsu Threshold. No final do processo, é possível visualizar as regiões de interesse de cada quadrante.**

***Palavras Chave—Visão omnnidirecional, robôs agrícolas móveis, sistemas catadióptricos, ambientes externos agrícolas.***

***Abstract*— This paper have aimed images in agricultural environment using a catadioptric vision system and performed the recognition of regions that can help many applications in mobile robotic area. The catadioptric vision systems have aimed a 360-degree vision of the environment using combination of lenses and mirrors. The omnidirectional image obtained is unwrapped and yours quadrants are generated, representing each side of the vehicle. The step of segmentation using color band is applied with base in the Otsu Threshold algorithm. In the last step of process, is possible to see the regions of interested for each quadrant.**

***Keywords—Omnidirectional vision, agricultural mobile robots, catadiptric systems, agricultural outdoor environments.***

## I. INTRODUÇÃO

Sistema de Visão Omnidirecional fornecem técnicas que possibilitam obter imagens em 360º do ambiente [3]. Dentro dessas técnicas estão os sistemas omnidirecionais catadióptricos, criados a partir da combinação de lentes e espelhos. Em alguns problemas da área da Visão Computacional é necessário ampliar o campo de visão para aumentar a eficiência das aplicações. O aumento de informações permite melhorar a eficiência em tomadas de decisões e obter assim resultados mais precisos. Na Agricultura de Precisão tarefas como vigilância de ambientes, rastreamento, navegação e mapeamento de ambientes [1] exigem um grande volume de informações para que seus processos funcionem adequadamente.

Esse trabalho pertence ao módulo de visão computacional do projeto AgriBOT. O AgriBOT é uma plataforma robótica modular com o objetivo de locomoção em ambientes típicos da área agrícola, possuindo como finalidade a aquisição de dados e pesquisa de novas tecnologias para o sensoriamento remoto para a área agrícola [9]. O projeto teve a atuação de professores e estudantes de graduação e pós-graduação da EESC/USP (Escola de Engenharia de São Carlos da Universidade de São Paulo). Além disso, diversas entidades brasileiras importantes participaram do projeto, como a FINEP ("Financiadora de Estudos e Projetos"), Embrapa ("Empresa Brasileira de Pesquisa Agropecuária") e Jacto S.A. company [10].

O objetivo desse trabalho dentro do projeto AgriBOT consiste na captura e identificação de regiões em imagens omnidirecionais relacionadas ao percurso efetuado pela plataforma robótica em cenários agrícolas. A identificação de regiões presentes no ambiente permite a precipitação de situações que possam ocorrer durante a navegação. Essas informações resgatadas do ambiente servem de base para outros módulos do veículo, sendo este intimamente ligado ao sistema de controle de navegação da plataforma robótica base [9]. Entretanto, o sistema proposto é uma aplicação modular, onde as informações geradas no final do processo podem vir a auxiliar outros sistemas da Agricultura de Precisão a alcançarem seus objetivos, tanto de forma total como parcial.

A captura das imagens foi realizada em plantações de pomares de laranja, uma das culturas de interesse do projeto AgriBOT, no mês de novembro em uma fazenda particular do município de São Manuel/SP. Como a captura foi realizada em um período entre as fases de coleta do fruto, foi possível identificar nos pomares frutos de diferentes tamanhos e estágios de crescimento, consequentemente diferentes tonalidades. Além disso, todas as imagens foram registradas com o veículo em movimento, sendo esse guiado por um operador humano. Devido processos burocráticos, para agilizar o desenvolvimento desse trabalho, o veículo utilizado para os testes é uma caminhonete de propriedade particular, porém é um projeto futuro a implementação desse sistema na plataforma AgriBOT.

O trabalho é organizado como segue: a Sessão II apresenta a metodologia do trabalho e os conceitos utilizados, a Sessão III apresenta os resultados obtidos e na Sessão IV a conclusão.

## II. METODOLOGIA DO TRABALHO

### *A. Construção do Sistema de Visão*

Para os experimentos foi utilizada uma câmera Ximea xiQ USB 3.0 modelo MQ013CG-E2. As imagens adquiridas possuem resolução de 640x480 pixels. Durante as coletas, o sistema de exposição de luz e balanço de branco foram acionados para obter uma melhor qualidade das imagens. Apenas o foco automático da câmera foi desligado, uma vez que a distância entre câmera e espelho não se alteram durante a captura das imagens. Na Fig. 1 é possível observar a configuração do sistema utilizado e na Fig. 2 a alocação desse sistema no veículo.

Fig. 1 – Sistema de visão catadióptrico

O perfil de espelho hiperbólico, como o utilizado nesse trabalho, apresenta uma resolução equilibrada em toda superfície e permite obter um centro único de projeção, propriedade que garante transformar as imagens omnidirecionais em imagens panorâmicas nítidas sem distorções [2][4].

### *B. Extração do veículo da imagem*

Dependendo da configuração utilizada pelo sistema de visão e também das dimensões do veículo, a imagem capturada pode apresentar constantemente uma área destinada à reflexão do veículo. Essa região pode atrapalhar o processo de segmentação se for mantida na imagem e por isso tornou-se necessário a exclusão dessa região.

Para realizar esse processo, uma máscara de corte foi configurada com o intuito de definir qual a área de interesse para as demais etapas do sistema. A Fig. 3 (a) apresenta uma imagem omnidirecional originalmente obtida pelo sistema de visão, em (b) é possível observar um exemplo de máscara para essa imagem e em (c) a aplicação da máscara na imagem omnidirecional.

Fig. 2 – Alocação do sistema de visão no veículo

### *C. Retificação da imagem*

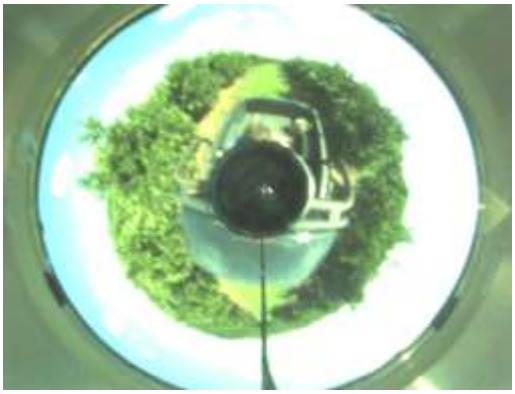
A etapa de retificação consiste na geração de imagens panorâmicas a partir de imagens omnidirecionais. A técnica utilizada para esse processo é o mapeamento de coordenadas utilizando a propriedade de centro único de projeção, descrito em [2]. Essa propriedade está presente em espelhos parabólicos e hiperbólicos e permite a utilização de um mapeamento polar linear, gerando no processo de retificação uma imagem livre de distorções verticais. Na Fig. 4 é possível verificar o resultado dessa etapa de processamento.

### *D. Extração do Céu da Imagem*

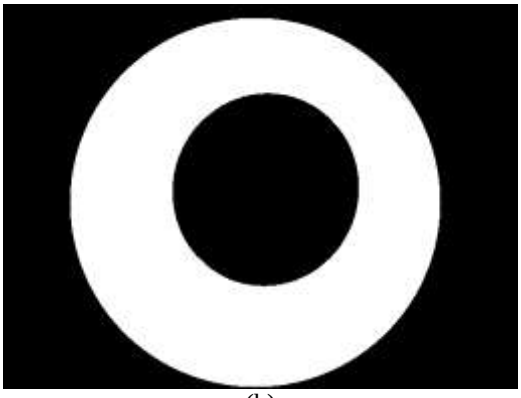
De modo parecido ao que acontece na etapa da extração da área do veículo, é necessário também retirar a região da imagem que representa o céu do ambiente. Para isso, foram aplicados dois limiar no espaço de cores HSV [5] apenas na metade superior da imagem panorâmica. Foi possível observar a eficiência dessa técnica para céus nublados, com pouca presença de nuvens e com bastante presença de nuvens, situações de interesse presentes no problema estudado.

O primeiro limiar tem como objetivo retirar a área azul do ambiente. Os limiares para segmentar essa região foram mínimo de 150º e máximo de 270º para o canal Matiz e mínimo de 110 e máximo de 255 para o canal de Iluminação. O canal precisou ser verificado para ignorar pixels pretos que possam ter valores dentro dos limites estabelecidos no canal Matiz.

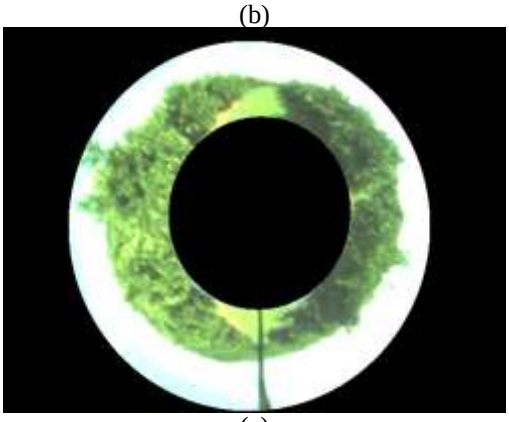
O segundo limiar tem como objetivo segmentar áreas que contenham nuvens. Nessa técnica é utilizado o intervalo de 0 à 20 para o canal de Saturação e intervalo de 240 à 255 para o canal de Iluminação. Na Fig. 5 é possível verificar um resultado dessa etapa de processamento.



(a)



(b)



(c)

Fig. 3 – Aplicação da máscara de interesse nas imagens omnidirecionais

Ao navegar por cenários naturais desestruturados, a imagem frequentemente contém informações de diferentes colorações de plantas e solos. Quando o veículo estiver no final de uma linha de navegação, a imagem capturada pode conter informações de várias outras ruas, árvores de vários tipos, sombras e mudança repentina da coloração do terreno. O isolamento da visão de cada lado do veículo em imagens separadas pode ser uma abordagem interessante, uma vez que o algoritmo de segmentação se baseará apenas nas informações próximas e é possível assim avaliar um limiar para cada tipo de cenário ou visão. Assim, essa abordagem visa melhorar a eficiência da etapa de segmentação.

### *E. Adaptação do Otsu Threshold*

O algoritmo de segmentação por cor Otsu Treshold [8] tem como objetivo encontrar o melhor valor para se dividir um histograma de uma imagem em escala de cinza. Para isso, o método busca encontrar o maior grau de variância entre diferentes grupos de pixels. Esse algoritmo é muito utilizado em cenários naturais e está presente em vários trabalhos da área [6] [7].

No projeto, o algoritmo Otsu Threshold foi adaptado para que pudesse ser executado em grupos de pixels específicos do histograma da imagem. O objetivo dessa adaptação é encontrar o pixel que representa o maior grau de variância dentro de um grupo de pixels. Nas duas técnicas utilizadas o algoritmo é executado duas vezes nos canais especificados, onde a primeira refere-se ao histograma como um todo e a segunda verificando apenas o grupo de pixels definido como limite inicial pelo primeiro limiar calculado.

### *F. Participação de verde nas imagens*

No presente trabalho são utilizadas duas técnicas diferentes para encontrar as ruas de navegação na imagem. Contudo, antes de aplica-las é necessário verificar qual técnica deve ser utilizada no quadrante que está sendo analisado, uma vez que cada técnica serve para determinada situação.

A condição utilizada para determinar qual ação será executada é analisar a porcentagem de pixels verdes presentes no quadrante. Para isso, uma nova limiarização no espaço HSV

Fig. 4 – Imagem retificada pela técnica de centro único de projeção

Fig. 5 – Etapa de retirada do céu da imagem

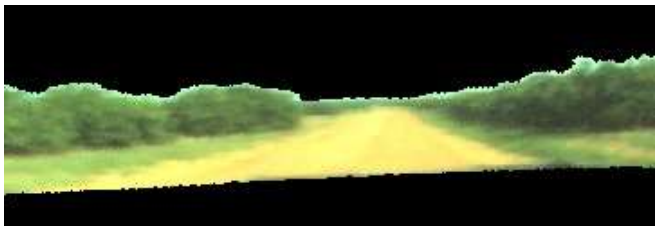
é executada utilizando o intervalo de limiares 90º à 150º para o canal Matiz e o intervalo de 50 à 255 para o canal *de* Iluminação. O canal de Iluminação é utilizado como filtro desse processo para não abranger regiões de pixels pretos.

Após selecionar a região da imagem que pertencem aos pixels verdes, é possível verificar a porcentagem de pixels verdes presentes na imagem.

*G. Otsu Threshold para o canal Matiz*

Nos quadrantes que possuem uma participação de verde inferior a 80%, é aplicada a técnica do Otsu Threshold no canal Matiz do espaço de cores HSV. Esse método será aplicado em cenários onde as ruas para navegação destacam-se das áreas representadas pelos pomares. Como o algoritmo de segmentação é adaptado para ser executado mais de uma vez no intuito de maximizar a eficiência dessa etapa, o limiar calculado na primeira execução do algoritmo é utilizado como parâmetro de entrada para uma segunda execução.

Na Fig. 6 (a) é possível analisar um quadrante com pouca predominância de verde e em (b) é possível verificar um resultado dessa etapa de segmentação.



(a)

(b)

Fig. 6 – Aplicação do Otsu Threshold no canal Matiz

*H. Otsu Threshold para canal de Iluminação*

Quando a participação de verde nos quadrantes for superior a 80%, ou seja, quando as regiões dos pomares possuírem cores bastante próximas das ruas de navegação, a segmentação por limiarização aplicada no canal Matiz do espaço de cores HSV torna-se pouco eficiente. Para segmentar as imagens com essa condição utilizamos o algoritmo Otsu Treshold Adaptado nas informações do canal de Iluminção. Assim, o canal de cores (Matiz) é utilizado para filtrar os pixels verdes e, com a técnica Otsu, a relação de luminosidade é aproximada para obter as regiões desejadas.

Na Fig. 7 (a) está ilustrada um quadrante predominantemente verde e em (b) é possível analisar os resultados obtidos por essa técnica.

*I. Aplicação da máscara para detecção de caminhos*

Após o processamento da segmentação, é necessário aprimorar os resultados obtidos até o momento. A primeira etapa para esse procedimento é a aplicação de uma máscara que limita a região em que se pode encontrar a rua de navegação. Essa máscara deve ser definida fora da etapa de execução do sistema, e pode ser única para cada quadrante da imagem. O tamanho e formato da máscara podem também variar de acordo com cada aplicação.

(a)

(b)

Fig. 7 – Aplicação do Otsu Threshold em regiões predominantemente verdes

Na Fig. 8 é possível verificar um exemplo de máscara para detecção de caminhos. Na Fig. 9 é possível observar o resultado dessa técnica, onde em (a) é possível observar um threshold resultante da etapa anterior e em (b) o resultado dessa etapa.

Fig. 8 – Exemplo de máscara de corte para detecção de caminhos

*J. Limpeza da região de interesse*

Mesmo com a aplicação da máscara, a imagem ainda pode apresentar regiões pequenas ou insatisfatórias na imagem que podem atrapalhar os procedimentos de pós-processamento. É necessário eliminar essas regiões que não possuem importância para o problema tratado. Um método foi criado para realizar a limpeza dessas regiões. Nesse algoritmo, o objetivo é encontrar regiões pequenas de pixels brancos e eliminá-los.

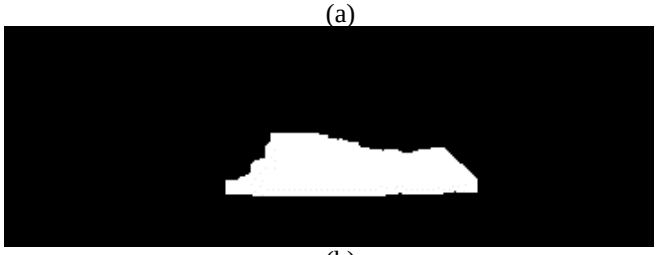
(a)

(b)

Fig. 9 – Aplicação da máscara para detecção de caminhos

O algoritmo percorre a imagem da esquerda para a direita e de cima para baixo. Após, calcula-se a quantidade de pixels brancos consecutivos para todos os lados do pixel analisado e, caso a quantidade for menor que um valor pré-definido em pelo menos três lados do pixel, o pixel torna-se preto. Na Fig. 10 (a) é possível observar um quadrante após a aplicação da máscara de detecção de caminhos e em (b) é possível analisar os resultados obtidos pelo algoritmo de limpeza de região de interesse.

(a)



(b)

Fig. 10 – Eficiência do algoritmo de limpeza de regiões

## III. RESULTADOS

A construção do sistema de visão foi a etapa mais complexa do sistema, devido a dificuldade em construir um sistema que a câmera e o espelho ficassem completamente alinhados. Além disso, essa etapa possui uma grande importância para a qualidade dos resultados obtidos. Com a coleta de imagens em ambiente natural é possível perceber como as características do ambiente interferem na qualidade do ambiente, como por exemplo a constante exposição do sol, vibração do motor e terreno irregular, além de sombras e outros objetos. Muitos desses problemas podem ser corrigidos no momento da escolha do material que será utilizado nos equipamentos e também na forma que eles são alocados no veículo. Outras configurações podem ser feitas na própria câmera, como a exposição de luz permitida.

Nas coletas de imagens foram obtidas 4.326 imagens omnidirecionais. Como cada imagem omnidirecional gera 4 quadrantes, foram obtidos no total 17.308 quadrantes possíveis de serem analisados. Os quadrantes foram separados em grupos de acordo com a participação de pixels verdes presentes na imagem. Assim, as imagens presentes no grupo 1 possuem 0% à 20% de pixels verdes; no grupo 2, 20% à 40%; no grupo 3, 40% à 60%; no grupo 4, 60% à 80%; e no grupo 5, 80% à 100%. Foram selecionadas 421 imagens para o grupo 1, 1071 imagens para o grupo 2, 2617 imagens para o grupo 3, 3578 imagens para o grupo 4 e 9617 imagens para o grupo 5.

Cada imagem foi processada utilizando as duas técnicas de segmentação apresentadas nesse trabalho. Selecionando de maneira randômica 40 imagens de cada grupo, foram avaliados a eficiência das etapas de extração de quadrantes, threshold, máscara de corte e algoritmo de limpeza. Assim, foi possível obter as tabelas I e II.

TABLE I. RESULTADOS THRESHOLD COR

| **Grupos Analisados** | **Etapas de Processamento** | | | |
|---|---|---|---|---|
| | ***Quadran-tes*** | ***Etapa Treshold*** | ***Máscara de corte*** | ***Limpar área*** |
| 0% - 20% | 0,95 | 0,552631 | 0,578947 | 0,526315 |
| 20% - 40% | 0,95 | 0,789473 | 0,789473 | 0,789473 |
| 40% - 60% | **1** | **0,925** | **0,9** | **0,875** |
| 60% - 80% | **1** | 0,825 | 0,8 | 0,8 |
| 80% - 100% | **1** | 0,375 | 0,475 | 0,475 |

TABLE II. RESULTADOS THRESHOLD CANAL *DE ILUMINAÇÃO*

| **Grupos Analisados** | **Etapas de Processamento** | | | |
|---|---|---|---|---|
| | ***Quadran-tes*** | ***Etapa Treshold*** | ***Máscara de corte*** | ***Limpar área*** |
| 0% - 20% | 0,95 | 0,026315 | 0,026315 | 0,026315 |
| 20% - 40% | 0,95 | 0 | 0 | 0 |
| 40% - 60% | **1** | 0,25 | 0,125 | 0,075 |
| 60% - 80% | **1** | 0,4 | 0,375 | 0,425 |
| 80% - 100% | **1** | **0,9** | **0,925** | **0,925** |

A maioria das imagens pertencentes ao grupo 1 possuem uma resolução bem inferior aos demais grupos e são imagens que ficaram muito claras e pouco definidas as linhas de navegação. O grupo 2 é representado por imagens mais nítidas se comparadas ao grupo anterior, porém muitas delas possuem ruas muito largas, onde é bastante difícil definir os limites das ruas de plantação. O grupo 3 e o grupo 4 apresentam imagens

bastante definidas e parecidas. As ruas normalmente se destacam das demais regiões da imagem e é possível entender a eficiência da primeira técnica do threshold nesses dois grupos, uma vez que a técnica foi escolhida para funcionar em imagens com essas características. Contudo, o grupo 4 apresenta também imagens com vegetação não uniforme, onde existe regiões pequenas dentro da área da rua com textura diferente das demais áreas que a compõe.

O grupo 5 é representado por ruas totalmente verdes e que se confundem com a área de plantação das culturas. Além disso, imagens que representam os pomares sem a presença de qualquer área navegável também estão presentes.

Com esses resultados, é possível dizer que nenhuma técnica utilizada nesse trabalho é perfeita para todas as situações que possam ocorrer no ambiente. É possível criar um sistema em que utilize as duas técnicas de segmentação com base na participação de verde de cada quadrante, e é possível definir que o critério para utilização da técnica seja o valor de 80%, buscando contemplar cada grupo com a técnica que possui uma melhor eficiência.

O sistema de visão utilizado conseguiu ter bons resultados em ambientes que possuem uma grande distância entre linhas de plantio. Nas imagens omnidirecionais com menor participação de verde, as ruas eram identificadas como elementos de terra ou areia, e a técnica utilizada para esse grupo não é afetado por sombras. Nas imagens com grande participação de verde, a cor não é uma característica única de definição e devido a isso a técnica é sensível a luz. Contudo, o algoritmo Otsu Threshold mostrou eficácia em muitas imagens desse grupo. A máscara para detecção de caminhos é um dos pontos chaves do processo, pois sua aplicação aperfeiçoa o resultado da segmentação.

No final do processo, os quadrantes possuem informação da localização das ruas para navegação. Caso houver necessidade, é possível obter as regiões pertencentes à cultura plantada realizando a subtração das áreas do céu, robô e ruas, sobrando apenas a área das culturas.

## IV. Conclusão

Apesar das dificuldades na construção do sistema de visão e demais particularidades do ambiente, é possível evoluir e a aplicar conceitos de visão omnidirecional na agricultura de precisão. O sistema de visão e as técnicas utilizadas se mostraram eficientes, apesar de demonstrar sensibilidade a algumas características do ambiente. As imagens resultantes podem auxiliar diversos processos da agricultura de precisão e robótica móvel agrícola, sendo aplicável para diversas culturas diferentes e com um tempo de processamento bastante rápido.

## References


[1] A. Bonarini., P. Aliverti and M. Lucioni. *An omnidirectional vision sensor for fast tracking for mobile robots*. Instrumentation and Measurement Technology Conference, 1999. IMTC/99. Proceedings of the 16th IEEE. 2000.

[2] V.G. Junior and J.O. Junior, “Development of an omnidirectional vision system,” Mechatronics and Machine Vision in Practice, 2007.

[3] S.K. Nayar, “Catadioptric omnidirectional camera,” IEEE Computer Society Conference, 1997.

[4] H. Ishiguro, “Development of low-cost compact omnidirectional vision sensors and their applications,” Departament of Electrical & Computer Engineering, University of California, San Diego, 1998.

[5] B., Banchev and L. Georgieva, “Comparison of pixel based skin recognition techniques”. In: НАУЧНИ ТРУДОВЕ НА РУСЕНСКИЯ УНИВЕРСИТЕТ, 2012.

[6] G. Jiang and C. Zhao. 2010, “A Vision Based Crop Rows for Agricultural Mobile Robot”. In: International Conference on Computer Application and System Modeling - ICCASM 2010.

[7] H. Jiang et al., “Curve path detection of unstructured roads for the outdoor robot navigation”, Mathematical and Computer Modelling, 2013, Vol. 58, pp. 536-544.

[8] N. Otsu, “A threshold selection method from gray-level histograms”. IEEE Trans. on Systems Man and Cybernetics, 9(1):63–66. 1979.

[9] NEPAS, *Núcleo de Ensino e Pesquisa em Automação e Simulação: Agribot*, http://www.simulacao.eesc.sc.usp.br/roboticaagricola/index .php/ robos-agricolas/agribot. (current Aug. 3, 2014).

[10] J.F. Diaz et al., "Agribot - Mobile Robot To Support of Agricultural Precision, Activities". In: 22nd International Congress of Mechanical Engineering - COBEM 2013, Ribeirão Preto, 2013.

# Investigating the use of Block-Matching 3D Denoising Algorithm to Reduce Radiation Dose in Digital Mammography

Helder C. R. de Oliveira, Polyana F. Nunes, Lucas R. Borges, Marcelo A. da C. Vieira
Departamento de Engenharia Elétrica, Universidade de São Paulo, São Carlos, Brasil
{heldercro, polyananunes, lucas.rodrigues.borges}@usp.br
{mvieira}@sc.usp.br

*Abstract*—**This paper investigates the use of the Block-Matching 3D algorithm (BM3D) to remove noise in mammographic images. The main objective is to study the possibility of using the BM3D technique to allow a radiation dose reduction in digital mammography, by removing the extra noise incorporated in the image when the photon-counting rate is reduced. Results were obtained by denoising synthetic mammograms acquired with different levels of radiation exposure, where the BM3D parameters were adjusted to provide better attenuation of noise without blurring or insertion of artifacts at the final image. We also tested the BM3D under Anscombe domain in order to evaluate whether the noise is better removed or not. Evaluation of denoising performance were conducted by comparing mammographic images acquired with normal dose of radiation to a set of images acquired at lower doses levels, after processed by BM3D method.**

**Keywords—Image processing; image denoising; digital mammography; quantum noise; block-matching 3D; Anscombe transformation.**

## I. INTRODUÇÃO

A mamografia digital é considerada o exame mais indicado para a detecção precoce do câncer de mama, com desempenho significativamente maior que a mamografia analógica [1]. No entanto, as doses de radiação utilizadas na prática clínica com os sistemas digitais atuais tem sido geralmente as mesmas que eram usadas com os sistemas analógicos antigos. Isto ocorre porque a relação entre o ruído na imagem e a precisão do diagnóstico ainda não foi muito bem estabelecida para a mamografia digital [2].

Sabe-se que a exposição à radiação causada pela mamografia pode acabar induzindo o aparecimento do câncer de mama em algumas mulheres radiografadas [3,4]. Por outro lado, a principal dificuldade envolvida em uma possível diminuição da dose de radiação nas mamografias é o aumento do ruído quântico da imagem. O ruído quântico, resultante do baixo número de fótons utilizados no processo de formação da imagem mamográfica, causa uma diminuição do contaste entre o fundo e as estruturas de interesse na imagem [5]. Assim, a diminuição das doses de radiação e o consequente aumento do ruído nas imagens mamográficas tornaria ainda mais difícil a detecção de lesões mamárias pelos radiologistas, principalmente em seu estágio inicial.

Nesse sentido, o objetivo desse estudo é de investigar a possibilidade de se utilizar técnicas de filtragem de ruído para remover o ruído quântico que é incorporado à imagem em uma eventual redução da dose de radiação nos exames mamográficos, garantindo que a qualidade da imagem não seja comprometida. A técnica de filtragem de ruído utilizada nesse estudo foi o *Block-Matching 3D* (BM3D) [6], considerado atualmente pela literatura como o "estado da arte" na filtragem de ruído [7]. Nesse estudo, o BM3D foi aplicado em um conjunto de imagens mamográficas antropomórficas sintéticas geradas com diferentes níveis de exposição à radiação, adquiridas a partir de um simulador de anatomia mamária desenvolvido na *University of Pennsylvania* [8,9]. Além disso, como o BM3D foi originalmente proposto para filtragem de ruído branco, gaussiano, aditivo e independente da intensidade da imagem, foi também avaliado o uso da Transformada de Anscombe [10,11] para a estabilização da variância do sinal da imagem antes da aplicação da filtragem de ruído pelo BM3D, já que o ruído predominante na mamografia digital é o ruído quântico [5], que segue a distribuição de Poisson, não é aditivo e depende da intensidade de radiação utilizada.

## II. MATERIAIS E MÉTODOS

### A. Filtragem do Ruído

A observação de uma imagem $g(x)$ corrompida pelo ruído branco gaussiano aditivo, onde $x$ é uma coordenada espacial 2D pertencente à imagem, pode ser escrita da forma:

$$g(x) = f(x) + n(x)\,, \tag{1}$$

onde $f(x)$ é a imagem sem ruído e $n(x)$ o ruído aditivo.

O BM3D é constituído de duas grandes etapas, onde na primeira é feita uma pré-estimativa da imagem sem ruído e na segunda etapa, essa pré-estimativa é utilizada como referência para a filtragem da imagem ruidosa utilizando o filtro de Wiener. As partes principais da primeira etapa são ilustradas na Figura 1.

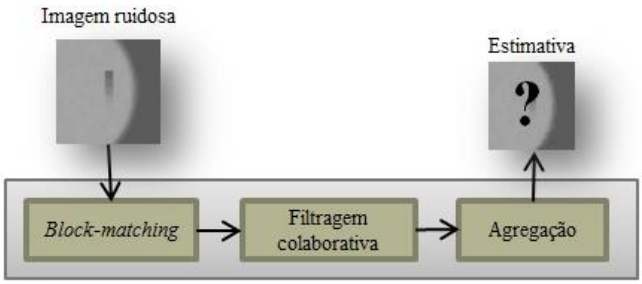



**Figura 1**: Primeira etapa do BM3D.

Para calcular uma pré-estimativa da imagem restaurada, inicialmente o algoritmo assume que se conheça o desvio padrão do ruído ($\sigma_n$). Sendo assim, esse valor é inserido manualmente. Na etapa *block-matching*, uma janela de busca de dimensão n é definida, onde em seu centro uma subárea P (*patch*) de tamanho k é utilizada como referência, conforme ilustrado na Figura 2.

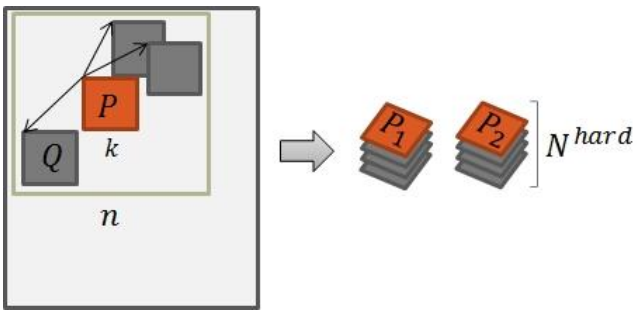



**Figura 2**: Janela de busca e *patches* selecionados dentro da imagem.

A janela de busca percorre toda a imagem. Em cada posição que ela assume é feita uma busca por *patches* $Q$ que sejam semelhantes ao de referência $P$. Assim, é montado um conjunto $\mathcal{P}(P)$ de todos os patches que possuem distância euclidiana $d(P,Q)$ menor que um limiar (*threshold*) definido por $\tau^{hard}$, conforme Equação (2):

$$\mathcal{P}(P) = \{Q: d(P,Q) \leq \tau^{hard}\}. \quad (2)$$

Os $N^{hard}$ patches selecionados são empilhados, formando uma estrutura 3D, onde ocorre a filtragem colaborativa. Essa filtragem é feita aplicando uma transformada *wavelet* 2D (os autores sugerem o uso da *wavelet* Biortogonal 1.5) em cada *patch*, produzindo uma representação esparsa da pilha [6]. Na terceira dimensão da pilha é aplicado o *hard-thresholding* nos coeficientes *wavelets* e aplicando a transformada inversa. Ao fim desta etapa tem-se uma estimativa de cada *patch* sem ruído e um conjunto de estimativas para cada pixel da imagem.

Na etapa de agregação, os *patches* são analisados e processados, com a finalidade de reduzir os efeitos de borda e a remoção de artefatos. Depois de processados os *patches* estimados são agrupados em um *buffer*. A imagem final $u^{basic}(x)$, estimada nessa etapa, é então calculada pela Equação (3):

$$u^{basic}(x) = \frac{\sum_P w_P^{hard} \sum_{Q \in \mathrm{P}(P)} \chi_Q(x) u_{Q,P}^{hard}(x)}{\sum_P w_P^{hard} \sum_{Q \in \mathrm{P}(P)} \chi_Q(x)}, \quad (3)$$

onde $w_P^{hard}$ é um peso dado no processo de agregação para dar prioridade aos *patches* mais homogêneos; $u_{Q,P}^{hard}$ é a estimativa do pixel $x$ pertencente ao patch $Q$ obtida durante a filtragem colaborativa do patch de referência $P$; e $\chi_Q(x)$ é uma função binária do tipo: $\chi_Q(x) = 1$ se $x \in Q$, senão $\chi_Q(x) = 0$.

Na segunda etapa é realizada uma filtragem colaborativa Wiener, onde a estimativa obtida na etapa anterior $u^{basic}(x)$ e a imagem ruidosa $u(x)$ são processadas seguindo a mesma metodologia utilizada na etapa anterior, como pode ser observado na Figura 3.

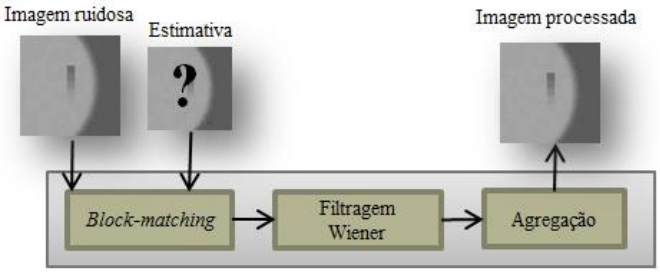



**Figura 3**: Segunda etapa do BM3D, filtragem Wiener.

A necessidade das duas imagens está relacionada ao fato de que o filtro de Wiener executa a filtragem de uma imagem ruidosa $u(x)$ buscando minimizar o erro quadrático médio (MSE) entre esta e uma imagem de referência, que nesse caso é a estimativa inicial da imagem sem ruído $u^{basic}(x)$ obtida na primeira etapa.

No caso na mamogafia digital, o ruído predominante é o ruído quântico, devido à baixa contagem de fótons que é utilizada no processo de aquisição da imagem. Esse ruído não é aditivo, depende da intensidade do sinal (da dose de radiação, nesse caso) e pode ser modelado por uma distribuição de Poisson [12].

Assim, para a filtragem das imagens mamográficas pelo BM3D, foi considerado também o uso da transformada de Anscombe [10,11], que permite a aproximação do ruído Poisson por um ruído gaussiano aditivo de média zero e variância unitária. Nesse caso, a transformada aplicada à imagem ruidosa $g(x)$, é :

$$z(x) = 2\sqrt{g(x) + \frac{3}{8}} \quad (5)$$

Após a aplicação da transformada de Anscombe é possível utilizar qualquer técnica convencional para a remoção de ruído gaussiano para filtragem de ruído quântico no domínio de Anscombe. Após a filtragem do ruído, a transformada inversa de Anscombe deve ser utilizada para se obter a imagem filtrada no domínio do espaço. Existem diversas aproximações matemáticas para a transformada inversa de Anscombe. Nesse

trabalho, foi utilizada a transformada inversa exata não-enviesada, apresentada em [11]. Com a finalidade de realizar comparações entre os resultados, testes experimentais foram realizados utilizando o BM3D dentro e fora do domínio de Anscombe.

### *B. Imagens Mamográficas*

Testes com o BM3D foram realizados em um banco de imagens mamográficas sintéticas geradas a partir de um simulador computacional da anatomia mamária desenvolvido na *University of Pennsylvania* [8,9]. Esse simulador permite a geração de imagens mamográficas antropomórficas com excepcional realismo, incluindo regiões de tecido adiposo e fibroglandular e diferentes lesões mamárias de interesse clínico. Usando este software simulador, foram selecionados aleatoriamente diferentes distribuições de tecidos adiposos e fibroglandular e um agrupamento de microcalcificações foi adicionado no modelo em uma posição aleatória, simulando uma lesão mamária de grande importância clínica, pois é um dos indícios de câncer de mama em seu estágio inicial [13]. Assim, foram geradas imagens mamográficas com a dose normal de radiação usada em exames clínicos de mamografia (chamado aqui de 100% da dose) e também imagens com dose correspondente à 50% e 25% da dose normal do exame. No total, foram utilizadas seis imagens mamográficas com as dosagens clínicas citadas anteriormente, sendo três delas ruidosas e outras três sem ruído, cada uma com $1792 \times 4096$ pixels, 14 bits e tamanho de pixel de 70 μm. As imagens sem ruído foram utilizadas a fim de fornecer padrões de referência (*ground-truth*) que permitiram a avaliação objetiva do desempenho do filtro. A Figura 4 mostra um exemplo de imagem mamográfica sintética utilizada nesse trabalho.

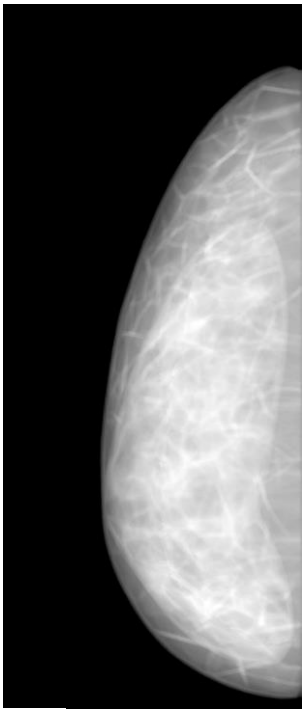


**Figura 4**: Exemplo de imagem mamográfica utilizada nesse estudo.

### *C. Avaliação do BM3D na redução da dose da mamografia*

A avaliação de desempenho do BM3D na filtragem do ruído quântico em diferentes doses de radiação foi realizada usando medidas quantitativas e qualitativas. Medidas quantitativas foram obtidas pelo cálculo do pico da relação sinal-ruído (PSNR - *peak of signal-to-noise ratio*) [14] e do índice de similaridade estrutural (SSIM - *structural similarity index*) [15]. Medidas qualitativas foram realizadas analisando o do perfil radiométrico de alguns recortes extraídos das imagens após a filtragem do ruído.

## III. Resultados e Discussão

A filtragem e os cálculos de similaridade foram aplicados às imagens mamográficas considerando a seguinte abordagem:

1. A filtragem de ruído pelo BM3D foi aplicada em todas as imagens mamográficas ruidosas adquiridas com 50% e 25% da dose, tanto no domínio do espaço como no domínio de Anscombe;
2. Para cada uma das imagens foram extraídos cinco recortes, $256 \times 256$ pixels, que apresentavam uma variação de contraste e riqueza de detalhes;
3. As medidas de qualidade da imagem (PSNR e SSIM) foram aplicadas em cada um dos cinco recortes, antes e após a filtragem pelo BM3D, utilizando como referência a imagem sem ruído (*ground-truth*);
4. Os resultados foram obtidos comparando-se as medidas de qualidade dos recortes extraídos das imagens obtidas com 50% e 25% da dose, após a filtragem pelo BM3D, com os recortes obtidos com 100% da dose.

Como o objetivo do trabalho é avaliar a possibilidade de se reduzir a dose de radiação nos exames mamográficos, os parâmetros do BM3D foram ajustados para que as imagens adquiridas com menor dose (50% e 25%) fossem filtradas, tendo como objetivo ser o mais similar possível à imagem obtida com 100% da dose, que é representativa do caso clínico normal de um exame mamográfico.

A Figura 5 mostra os cinco recortes extraídos de cada uma das imagens mamográficas geradas para esse trabalho. Como as imagens possuem exatamente a mesma estrutura, os recortes são idênticos, independente da dose.

Cada recorte ruidoso nas doses mencionadas anteriormente (50% e 25%) foi comparado com o seu respectivo recorte sem ruído, ambos sem qualquer processamento. Esse levantamento se fez necessário para que tivéssemos um objetivo numérico no processamento, permitindo um ajuste dos parâmetros do BM3D. Os valores obtidos através dos cálculos de similaridade, antes da filtragem de ruído, são apresentados na Tabela 1. Como se esperava, os valores de PSNR e SSIM pioram a medida que a dose é reduzida, mostrando um aumento no ruído da imagem.

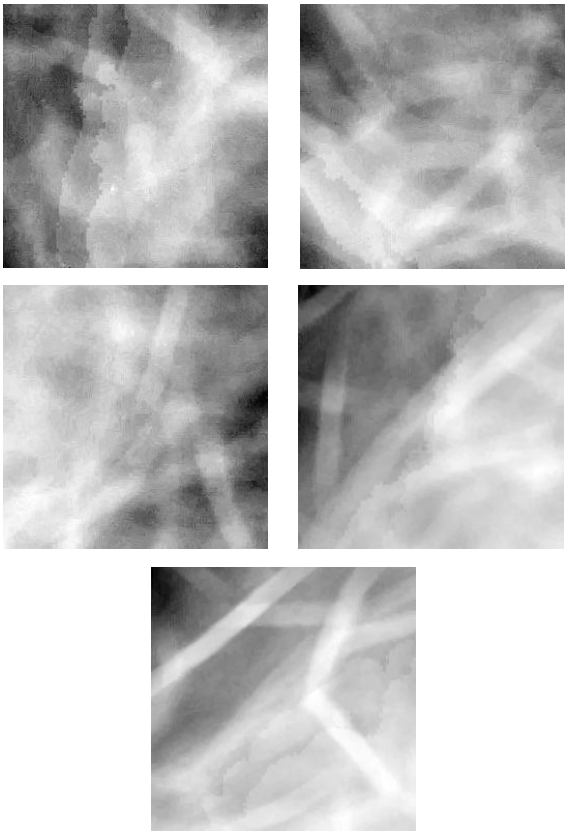


**Figura 5**: Recortes utilizados nos testes.

**Tabela 1**: Valor médido dos parâmetros de qualidade da imagem calculados para os recortes, antes do processamento pelo BM3D.

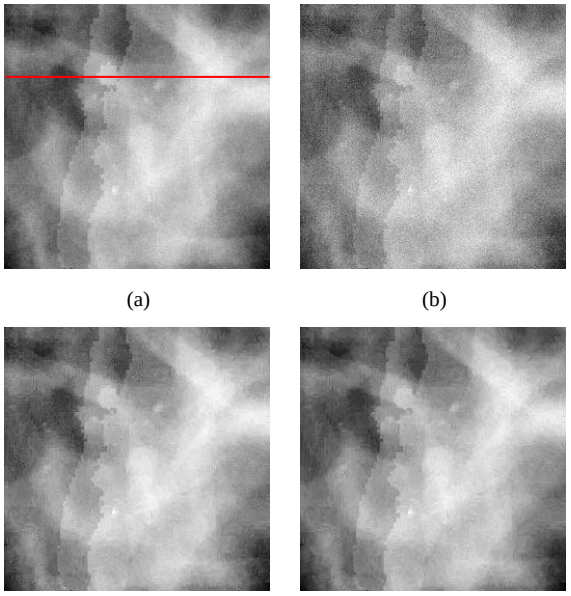
| Doses | PSNR (db) | SSIM |
|---|---|---|
| **100%** | 41,53 | 0,9425 |
| **50%** | 38,49 | 0,8923 |
| **25%** | 35,53 | 0,8111 |

A filtragem do ruído com o BM3D, tanto no domínio do espaço como no domínio de Anscombe, foi realizada para todas as imagens mamográficas, nas dosagens de 25% e 50%. Os parâmetros do filtro foram otimizados de modo a aproximar os valores de PSNR e SSIM, após a filtragem, aos valores calculados anteriormente para as imagens obtidas com 100% da dose. Os valores do desvio padrão do ruído para o algoritmo BM3D foram obtidos experimentalmente: $\sigma_n = 4$ e $\sigma_n = 2{,}3$, para o BM3D na filtragem das imagens com 25% e 50% da dose, respectivamente; e $\sigma_n = 2{,}6$ e $\sigma_n = 1{,}35$, para BM3D+Anscombe na filtragem das imagens com 25% e 50%, respectivamente, de maneira a se ter uma boa remoção de ruído com o mínimo de borramento.

A Figura 6 mostra um exemplo de recorte extraído de uma imagem mamográfica adquirida com 25% da dose normal de radiação. A imagem (a) corresponde à imagem original, obtida com 100% da dose; (b) mostra o mesmo recorte obtido com uma redução de 25% na dose original (note o aumento no ruído da imagem). As imagens da Figura 6(c) e (d) mostram, respectivamente, os resultados da filtragem da imagem adquirida com 25% da dose original pelo algoritmo BM3D e pelo BM3D no domínio de Anscombe.

Comparando-se visualmente as imagens filtradas com a imagem 25% ruidosa, observa-se que houve uma boa remoção do ruído, pouco borramento e quase nenhuma inserção de artefatos. Também observamos que as imagens filtradas ficaram bastante parecidas com a 100% ruidosa (Figura 6a), que é a imagem que o radiologista normalmente teria em mãos no exame mamográfico. Não é possível observar diferenças significativas entre as imagens processadas pelo BM3D no domínio de Anscombe e no domínio do espaço.

Os valores de PSNR e SSIM calculados para as imagens obtidas com 25% e 50% da dose, após a filtragem do ruído pelo BM3D e BM3D+Anscombe podem ser visualizados na Tabela 2. Nota-se, comparando a Tabela 2 com os dados da Tabela 1 para 100% da dose, que em todos os casos o BM3D conseguiu aproximar a qualidade das imagens com dose reduzida, com as imagens adquiridas com a dose total de radiação.



(a) (b) (c) (d)

**Figura 6**: Recorte extraído de uma imagem mamográfica: (a) adquirida com 100% da dose de radiação; (b) adquirida com 25% da dose; (c) 25% da dose filtrada com o BM3D; e (d) 25% da dose filtrada com o BM3D+Anscombe. O traço vermelho mostra o local onde foi extraído um perfil radiomético dessa imagem.

**Tabela 2**: Resultados do processamento das imagens com dose reduzida com o BM3D no domínio do espaço e no domíno de Anscombe.

| | Doses | PSNR (db) | SSIM |
|---|---|---|---|
| **BM3D** | **50%** | 42,75 | 0,9553 |
| | **25%** | 41,57 | 0,9490 |
| **BM3D + Anscombe** | **50%** | 42,31 | 0,9484 |
| | **25%** | 41,76 | 0,9512 |

Perfis radiométricos foram traçados em uma região com grande variação de contraste nas imagens, no local indicado pela linha em vermelho na Figura 6a. A Figura 7 apresenta o perfil de duas imagens, uma com 100% da dose e outra com 50% da dose, mostrando claramente o aumento no ruído da imagem quando a dose de radiação é reduzida à metade do valor original.

Também foram calculadas a distância Euclidiana entre os vetores que armazenam o perfil radiométrico das imagens de 100% e 50% da dose. O valor obtido nesse caso foi 21,73 (mostrado na parte superior da Figura 7). Ao traçar o perfil radiométrico na mesma linha mostrada anteriormente, agora nas imagens com 100% e 25% da dose, é possível observar, na Figura 8, que a imagem com 25% da dose possui uma quantidade bem mais acentuada de ruído, se comparada com a 100% ou até mesmo com a 50%. A distância entre os vetores, nesse caso, aumentou para 29,90.

**Figura 7**: Perfil radiométrico das imagens ruidosas com 100% e 50% da dose normal de radiação.

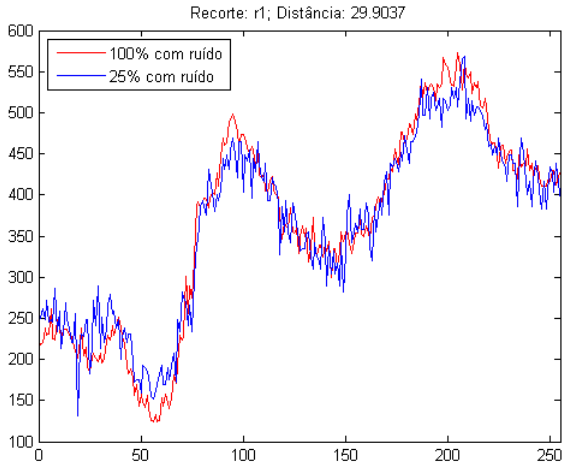



**Figura 8**: Perfil radiométrico das imagens ruidosas com 100% e 25% da dose normal de radiação.

Após a filtragem do ruído das imagens pelo BM3D, os perfis radiométricos foram novamente traçados, agora comparando as imagens filtradas com a imagem ruidosa obtida com 100% da dose. O objetivo é avaliar o quando a imagem processada se aproximou da imagem original obtida com a dose total de radiação.

Iniciando pela análise da imagem adquirida com 50% da dose, processada pelo BM3D no domínio do espaço e no domínio de Anscombe, o seu perfil é apresentado na Figura 9.

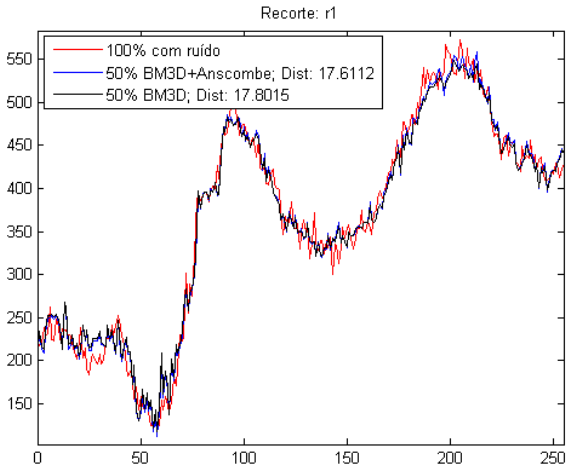



Figura 9: Perfil radiométrico de um recorte extraído da imagem adquirida com 50% da dose, filtrado com BM3D dentro e fora do domínio de Anscombe. O perfil da imagem adquirida com 100% da dose foi incluído para comparação.

Na avaliação dos perfis mostrados na Figura 9, pode-se ver que os perfis estão bem próximos, mostrando que o algoritmo BM3D removeu o ruído da imagem com menor dose aproximando-a da imagem original adquirida com 100% da dose, inclusive preservando os detalhes da imagem. O uso da transformada de Anscombe não alterou significativamente o resultado da filtragem. No entanto, percebe-se uma leve melhora na aproximação junto ao sinal original, pois a distância entre os vetores variou de 17,80 para 17,61 com o uso do Anscombe.

Também foi traçado o perfil da imagem adquirida com 25% da dose após o processamento (Figura 10). Novamente o uso do BM3D aproximou a imagem com dose reduzida à imagem original com 100% da dose (a distância para esse perfil passou de 29,90 para 17,02). A transformada de Anscombe apresentou um resultado pouco significativo do desempenho do BM3D na remoção do ruído nesse caso.

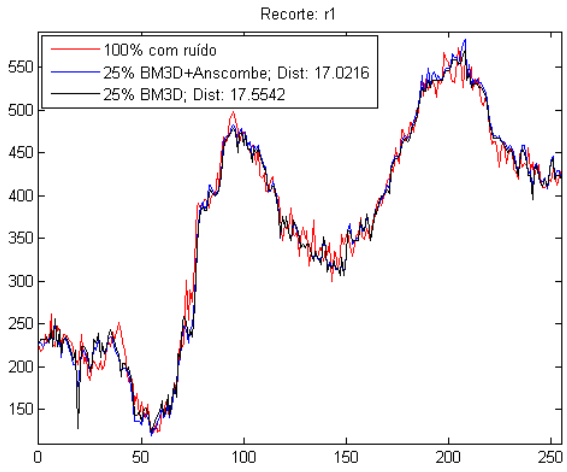



**Figura 10**: Perfil radiométrico de um recorte extraído da imagem adquirida com 25% da dose, filtrado com BM3D dentro e fora do domínio de Anscombe. O perfil da imagem adquirida com 100% da dose foi incluído para comparação.

## IV. Conclusões

Neste trabalho, foi investigado o uso do BM3D para redução da dose na aquisição de imagens mamográficas digitais. Apesar de se tratar de um estudo inicial sobre a possibilidade de redução nas doses de radiação da mamografia por meio de técnicas de filtragem de ruído, os resultados obtidos nesse trabalho foram promissores. Resultados preliminares com imagens sintéticas mostraram que é possível filtrar o ruído quântico que é incorporado na imagem mamográfica quando a dose de radiação é reduzida para 50% e 25% da dose original. Tanto a avaliação quantitativa, com os parâmetros PSNR e SSIM, e os perfis radiométricos, mostraram que as imagens adquiridas com doses reduzidas, depois de filtradas pelo BM3D, tornam-se muito similares as imagens adquiridas originalmente com 100% da dose.

O uso da transformada de Anscombe antes da filtragem pelo BM3D não alterou significativamente os resultados. Isso deve ter ocorrido pois mesmo para as imagens com 25% da dose, a taxa de contagem da distribuição de Poisson ainda é alta (>1000 fótons por pixel). Sabe-se que quanto mais alta a taxa de contagem de uma distribuição de Poisson, mais ela se aproxima da distribuição gaussiana [16]. Por isso, a contribuição no desempenho da filtragem do ruído com a transformada de Anscombe, no caso específico da mamografia digital, foi muito pequena.

Uma das limitações desse trabalho é o fato de que ele foi desenvolvido utilizando apenas imagens sintéticas. A opção por imagens sintéticas se deu pelo fato de que esse é o único meio de se obter as imagens de referência sem ruído (*ground-truth*), imprescindível para a análise quantitativa utilizando medidas objetivas como PSNR e SSIM. Embora essas imagens sintéticas tenham sido obtidas por um software 3D antropomórfico, capaz de fornecer simulação realista das imagens DBT da mama, um estudo utilizando imagens clínicas deve ser conduzido para uma avaliação mais adequada da viabilidade do uso do BM3D para redução das doses de radiação nos exames de mamografia digital.

## Agradecimentos

Os autores gostariam de agradecer à FAPESP (Proc. 2013/18915-5) e à CAPES (Proc. 88881.030443/2013-01) pelo apoio financeiro.

Agradecemos ao professor Predrag R. Bakic, da *University of Pennsylvania*, por ter fornecido as imagens mamográficas sintéticas utilizadas nesse trabalho.

## Referências

[1] Glynn CG, Farria DM, Monsees BS, Salcman JT, Wiele KN Hildebolt CF: “Effect of transition to digital mammography on clinical outcomes”. Radiology 260(3):664-670, 2011.

[2] Samei E, Saunders RS, Baker JA, Delong DM: Digital mammography: effects of reduced radiation dose on diagnostic performance, Radiology 243(2):396-404, 2007.

[3] Hendrick RE: “Radiation doses and cancer risks from breast imaging studies”, Radiology 257(1):246-253, 2010.

[4] Yaffe MJ, Mainprize, JG: “Risk of radiation-induced breast cancer from mammographic screening”, Radiology 258(1):98-105, 2011.

[5] Yaffe MJ: “Digital Mammography”. In: Beutel J, Kundel HL and Van Metter RL (Eds). Handbook of Medical Imaging Vol 1. Physics and Psychophysics, Bellingham, WA: SPIE Press, pp 329-372, 2000.

[6] Dabov K, Foi A, Katkovnik V, Egiazarian K: “Image denoising by sparse 3D transform-domain collaborative filtering”. In: IEEE Trans Image Process 16(8): 2080-2095, 2007.

[7] Milanfar P: A tour of modern image filtering: new insights and methods, both practical and theoretical, IEEE Signal Processing Magazine 30(1):106-128, 2013.

[8] Bakic PR, Zhang C, Maidment AD: Development and characterization of an anthropomorphic breast software phantom based upon region-growing algorithm. Med Phys 38(6):3165-3176, 2011.

[9] Pokrajac DD, Maidment ADA, Bakic PR: Optimized generation of high resolution breast anthropomorphic software phantoms. Med Phys 39(4): 2290-2302, 2012.

[10] Anscombe FJ: The transformation of Poisson, binomial and negative-binomial data. Biometrika 35:246-254, 1948.

[11] Mäkitalo M, Foi A: A closed-form approximation of the exact unbiased inverse of the Anscombe variance-stabilizing transformation. IEEE T Image Process 20(9): 2697-2698, 2011.

[12] Vieira MAC, Bakic PR, Maidment ADA, Schiabel H, Mascarenhas NDA: Filtering of Poisson Noise in Digital Mammography Using Local Statistics and Adaptive Wiener Filter, Lectures Notes in Computer Science 7361: 268-275, 2012.

[13] Veronesi U, Boyle P, Goldhirsch A, Orecchia R, Viale G: Breast Cancer. The Lancet 365:1727-1741, 2005.

[14] Wang Z, Bovik AC: Mean squared error: Love it or leave it? A new look at signal fidelity measures. IEEE Signal Proc Mag 26(1):98-117, 2009.

[15] Wang Z, Bovik AC, Sheikh HR, Simoncelli, EP: Image quality assessment: from error visibility to structural similarity. IEEE T Image Process 13(4):600-612, 2004.

[16] Starck J.-L., Murtagh F: Astronomical image and signal processing: Looking at noise, information, and scale. IEEE Signal Process. Mag., 18(2):30–40, 2001.

# Human Iris Segmentation on Videos Obtained via Natural Lighting from SmartPhones

Pinheiro, H.M.; Costa, R. M.; Laureano, G. T.; Romero, R.J.; Soares, F. A. A. M. N; Galdino, L. O.
Instituto de Informática/Universidade Federal de Goiás, Goiânia, Brasil
Goiânia, Brasil
{hedenir, ronaldomc12, gustavoengdm, renan.j.romero}@gmail.com; {fabrizzio, leandroluis}@inf.ufg.br

*Abstract*— **The use of human iris images is considered one of the safest ways to provide biometrical recognition. On iris images obtained with natural lighting, the presence of reflection in its surface difficult the process of segmentation and extraction of the patterns present in its texture. This present work has as objective studying and applying techniques for segmentation of the iris region on videos of human eyes obtained via natural lighting. To do this, methods focusing on improving the location of the internal and external iris contour were used in images obtained on these conditions. Some procedures were also made to the removal of present noise over the iris region, such as eyelids and reflections. The results showed that the search of the internal iris region via estimation of its shape provided high matching rates on the segmentation of this region on the video analysis**

***Keywords— Iris Segmentation; Biometric Recognition; Natural Lighting;***

## I. INTRODUÇÃO

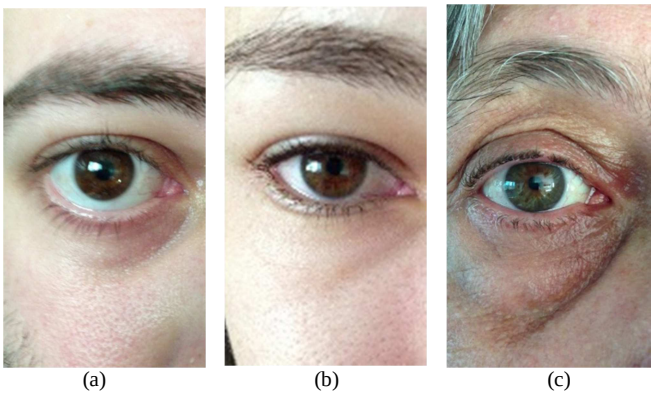
Nos últimos anos, as pesquisas na área de reconhecimento biométrico tiveram grande avanço, métodos de reconhecimento tornaram possível realizar a segmentação e extração da região da íris de forma bastante eficiente [1]. Entretanto, a maioria desses métodos exige que as imagens sejam capturadas por câmeras que trabalham na faixa de frequência próxima ao infravermelho (*Near Infra-Red* - NIR). Essa restrição referente à captura das imagens exigidas por estes métodos tornam muitas vezes o reconhecimento pela íris um processo inviável em ambientes onde não é possível seguir tal restrição. Por essa razão, mesmo sendo considerada uma das mais precisas [2], a biometria que utiliza as marcações presente na íris não é a forma mais utilizada de reconhecimento biométrico em sistemas de autenticação.

O objetivo do presente trabalho é analisar e aplicar técnicas de segmentação da íris em vídeos obtidos com uma câmera digital comum, capturados em ambientes com iluminação natural, de modo que a qualidade da segmentação obtida contribua para a construção de sistemas que utilizem esse tipo de imagem.

## II. MATERIAIS E MÉTODOS

Para a realização deste trabalho, são utilizados vídeos do olho humano obtidos com a câmera de um iPhone 5 com resolução de 1080 por 1920 pixels.

Foram utilizados 3 vídeos, de indivíduos diferentes; obtidos em ambiente fechado com a incidência de luz natural. Esses vídeos serão chamados de A, B e C. Na figura 1 são apresentados exemplos de frames retirados destes vídeos.



(a) (b) (c)

Figura 1: Frames Extraídos de Cada Um dos Vídeos Utilizados

Durante as fases da segmentação o tamanho dos frames dos vídeos sofreu variações dependendo do método utilizado. Os frames em algumas fases foram divididos em canais RGB, em outras foram convertidos para escala de cinza.

Os seguintes passos foram executados para segmentação e normalização:

A. Detecção do olho;
B. Detecção da íris e da pupila;
C. Remoção das pálpebras superiores;
D. Remoção de reflexos;
E. Normalização

### *A. Detecção do olho*

A figura 1 mostra que nos vídeos gravados, outras partes do rosto além do olho foram capturadas. Tendo em vista que a detecção da íris nas imagens depende da análise do contraste entre íris, pupila, pálpebras e esclera; a presença de outras regiões de contraste como rugas, sobrancelhas ou até mesmo reflexos na pele podem influenciar negativamente a qualidade da segmentação. Um exemplo disso pode ser observado figura 1c, em que nota-se a presença de várias regiões de alto contraste ao redor do olho.

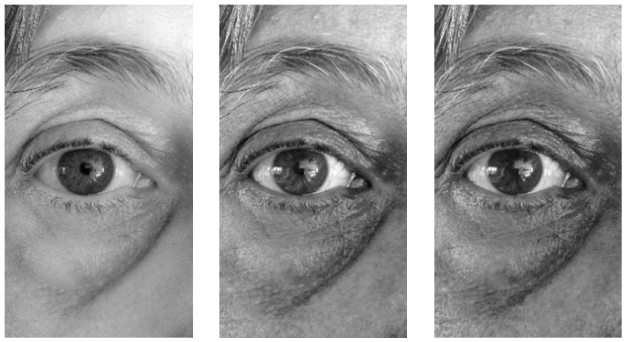
A remoção dessas regiões indesejadas é necessária para evitar que elas influenciem negativamente na detecção das regiões de interesse. Ao remover essas regiões aumenta-se a

área proporcional da íris em relação às dimensões de altura e largura das imagens, que podem então ser utilizadas como referência para a detecção dessa região. Outra vantagem da redução da área das imagens é de que um número menor de pixels será processado nas diversas etapas que constituem as fases seguintes de segmentação da íris.

*1) Transformações nas Imagens*

Para o processamento da segmentação do olho, os vídeos foram redimensionados para um tamanho proporcional a 20 % do tamanho original, o que resulta em um tamanho aproximado de 216 x 384 pixels. A quantidade da redução foi ajustada experimentalmente de tal forma que não se perdesse informações necessárias para a detecção e segmentação da região olho. Este redimensionamento não é aplicado nas imagens resultantes, pois o recorte do olho é realizado sobre a imagem com suas dimensões originais.

O processamento realizado nesta fase foi feito exclusivamente no canal vermelho do espaço de cores RGB. Pode-se observar na figura 2 um frame do vídeo C redimensionado decomposto em seus 3 canais de cores.



(a) (b) (c)

Figura 2: Decomposição da Imagem nos Canais RGB

Pode-se notar que no canal vermelho, figura 2a, a incidência de sombras diminuiu consideravelmente em relação aos outros canais. Existe também uma diminuição grande das áreas de reflexo incidentes sobre a região do olho. A redução destas sombras e dos reflexos faz com que se tenha uma imagem do olho com a região da íris mais bem definida e com um menor grau de intensidade de pixels se comparada às demais regiões da imagem.

*2) Técnica Utilizada*

Para realizar a segmentação do olho, tomou-se como base o proposto em [3], onde os pixels da imagem são agrupados em níveis de intensidade através do processo conhecido como k-means.

Assume-se a partir dessa técnica que a região que contêm o olho é aquela constituída por pixels vizinhos (conectados) pertencentes ao grupo de pixels com menor intensidade e que esteja mais próxima ao centro da imagem. Na figura 1 observa-se que a região do olho nos vídeos capturados está próxima ao centro, logo no presente cenário é possível utilizar esta técnica para detectar o olho nas imagens.

*3) Aplicação da Técnica*

A aplicação do k-means nas imagens foi realizada através da função k-means fornecida pelo OpenCV, onde os centróides iniciais dos clusters são gerados de maneira aleatória. De posse das informações geradas pelo k-means é possível construir uma nova imagem para cada frame dividindo os pixels de acordo com os seus índices. Na nova imagem, os pixels de um bloco recebem o valor de seu centroide, que guarda a média de intensidade de todos os pixels que constituem aquele bloco. A figura 3 demonstra a técnica para detecção e extração do olho aplicada a cada um dos vídeos analisados. Nela pode-se observar o resultado da divisão dos pixels gerada pelo processamento do k-means.

Figura 3: Resultado do k-means em 4 Níveis

Com o objetivo de tornar mais compacto o bloco da íris foram realizados vários experimentos com operações morfológicas nas imagens geradas pelo k-means.

Ao final destes experimentos chegou-se a uma configuração em que as operações morfológicas conhecidas como closing (dilatação seguida de erosão) e opening (erosão seguida de dilatação) são aplicadas em sequência nas imagens. Os resultados dessas operações podem ser observados na figura 4. O tamanho dos elementos estruturantes utilizados nessas operações foi definido experimentalmente como sendo de 9x9.

O conjunto de operações morfológicas reduziu o tamanho do bloco contendo a íris e a maioria dos blocos indesejados resultantes do k-means. A figura 5 demonstra o olho segmentado obtido pela seleção do bloco mais escuro de pixels que está mais próximo ao centro.

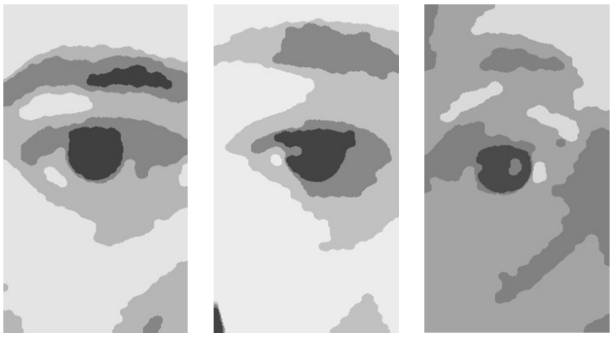


Figura 4: Resultado das Operações Morfológicas

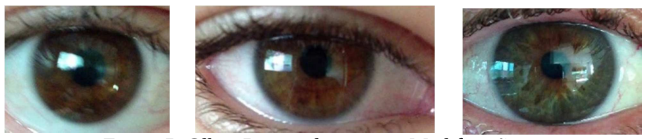


Figura 5: Olhos Recortados com as Modificações

### B. *Detecção da Pupila e a Íris*

Por se tratarem de regiões com características próximas a de uma circunferência, a detecção da íris e da pupila consiste em encontrar um círculo que seja compatível com suas dimensões e localização nas imagens analisadas.

Em [4, 5, 6, 7, 8, 9] foi possível perceber que a localização desses círculos é feita na maioria das vezes através de dois passos básicos: aplicação do filtro de *Canny* [10] para a detecção de bordas seguida da *transformada de Hough* [11] para a detecção dos círculos.

Nas imagens da figura 5 percebemos que o contraste entre a íris e a esclera é em geral maior que o contraste entre a pupila e a íris. Logo, a primeira abordagem utilizada para a detecção da íris e da pupila foi a de se localizar primeiramente o círculo da íris. Entretanto, visto que nesse cenário não foi possível chegar a uma configuração que apresentasse resultados satisfatórios de segmentação para os 3 vídeos; foi testada uma nova forma de detecção da pupila e inversão na ordem de detecção entre íris e pupila. Em ambas as abordagens, as informações do primeiro círculo encontrado foram utilizadas na localização do segundo.

#### 1) *Detectando a Pupila*

Para detectar a pupila, a suavização e a binarização foram aplicadas nas imagens. Na binarização, o limiar foi definido em 20. O cuidado ao ajustar o valor do limiar foi o de isolar o contorno da região da pupila das outras partes da imagem sem que com isso houvesse uma grande perda deste contorno.

Os contornos foram detectados através da função *findCountours* fornecida pelo OpenCV. Diferentemente da função *Canny*, a *findCountours* retorna para cada contorno detectado, um vetor com todos os pixels da imagem que o constitui. A partir desse vetor de pixels é possível obter as suas informações através da aplicação da função *boundingRect* do OpenCV, que retorna um retângulo que envolve a área daquele contorno.

O contorno que representa a pupila é considerado como sendo aquele mais próximo ao centro da imagem. Logo para cada retângulo encontrado essa distância deve ser calculada. O retângulo que possui a menor distância é então selecionado para a construção do círculo que representará a pupila.

A partir do retângulo selecionado é utilizada a função *minEclosingCircle* fornecida pelo OpenCV para a detecção de um círculo que se aproxime das dimensões daquele retângulo. As imagens da figura 6 demonstram os círculos desenhados juntamente aos retângulos que lhes deram origem.

Figura 6: Círculos Desenhados junto aos Retângulos

As imagens da figura 7 demostram a pupila detectada com estes círculos.

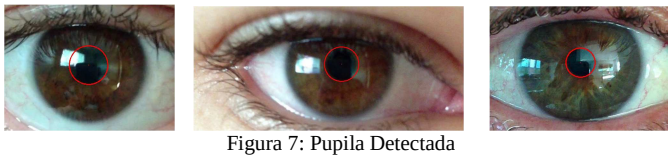


Figura 7: Pupila Detectada

#### 2) *Detectando a Íris*

Para a detecção do círculo da íris foi aplicado: suavização, binarização, detecção de contornos com *Canny* e busca por círculos através da *transformada de Hough*, utilizando-se as informações da pupila como parâmetros de restrição.

Foi definido através dos experimentos que o raio do círculo da íris deve ter um valor entre duas e cinco vezes o valor do raio do círculo da pupila. A distância máxima entre os centros do círculo da íris e o círculo da pupila foi definida como sendo 15% do valor do raio do círculo da pupila. Com uma restrição maior em relação ao raio e ao posicionamento do círculo da íris, os parâmetros da função *HoughCircles* foram ajustados para que esta retornasse um número menor de círculos candidatos. O menor desses círculos é então escolhido como sendo o círculo que representa a íris. A figura 8 demonstra os círculos gerados pela *transformada de Hough* e aquele selecionado segundo as restrições estabelecidas.

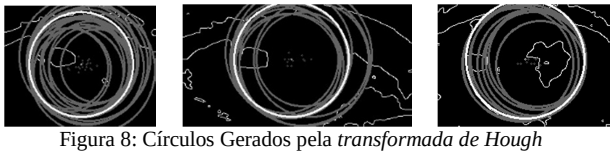


Figura 8: Círculos Gerados pela *transformada de Hough*

A figura 9 mostra a imagem resultante com a íris e a pupila detectadas.

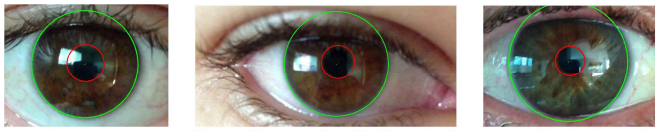


Figura 9: Iris e Pupila Detectadas

*C. Detecção da Pálpebra Superior*

A pálpebra em uma imagem do olho possui um formato semicircular. Para detectá-la deve-se encontrar o círculo em que este semicírculo está contido. Para detectar este círculo primeiramente detectam-se as bordas da pálpebra nas laterais da íris. Ao detectar as bordas da pálpebra, detectam-se dois pontos que se encontraram no perímetro do circulo desejado.

Para encontrar estes pontos, constroem-se dois retângulos com largura igual a um terço do raio da íris e altura igual ao raio da íris adicionado do raio da pupila. O primeiro retângulo é posicionado à esquerda da íris, tendo como posição da sua base a mesma do círculo da pupila. O segundo retângulo é posicionado a direita da íris, com a base na mesma linha do retângulo anterior. A figura 10 demonstra estes retângulos desenhados nas imagens geradas na segmentação da íris.

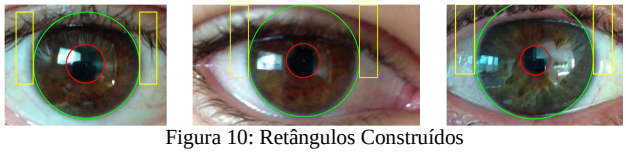


Figura 10: Retângulos Construídos

Para cada retângulo, uma nova imagem será então gerada para que a busca pelas bordas da pálpebra possa ser realizada.

Ao analisar os retângulos gerados nas imagens da figura 11, percebe-se que a região da borda da pálpebra tende a ter um grande contraste com a esclera devido à presença dos cílios. Esse contraste ocorre devido à equalização do histograma.

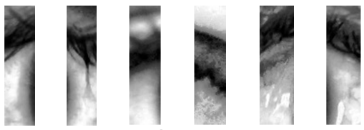
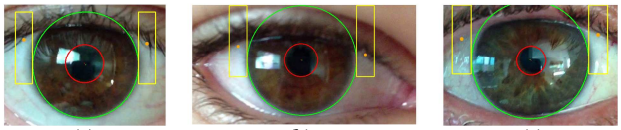


Figura 11: Retângulos Equalizados

O passo seguinte consiste em aplicar a binarização nos retângulos com o objetivo de destacar o limite entre a pálpebra e a esclera. O limiar da binarização foi ajustado entre os três vídeos para haver um equilíbrio entre a eliminação de sombras e ao mesmo tempo a não eliminação dos limites entre a pálpebra e a esclera. O valor encontrado foi o de 115. As Imagens da figura 12 apresentam os resultados da binarização.

Figura 12: Retângulos Binarizados

Observa-se que os procedimentos realizados conseguiram destacar as bordas da pálpebra na maior parte dos frames analisados. A figura 13 apresenta o realce dos pontos identificados como limites entre a pálpebra e a esclera.



(a) (b) (c)

Figura 13: Pontos da Borda das Pálpebras Encontrados

A figura 13(a) mostra que os pontos encontrados estão sobre os cílios presentes nessa região. Este fato ocorreu em outros frames do mesmo vídeo, mas pela proximidade do cílio para a pálpebra isso não se mostrou um problema para o restante do processo.

A partir da localização destes pontos, é possível encontrar o círculo que passa por eles. Define-se que o ponto encontrado a esquerda da íris será chamado de E e que o ponto encontrado localizado a direita será chamado de D. O objetivo final dessa fase é encontrar o centro de um círculo (ponto P) que tenha os pontos E e D em seu perímetro. Um terceiro ponto é utilizado durante o processo para encontrar o centro do círculo, que é o ponto médio entre E e D, chamado de C.

Sahmoud e Abuhaiba em [3] demonstram que entre o ponto P e ponto M existe uma distância de duas vezes o raio do círculo da íris, e que a reta formada pelos pontos C e D é perpendicular a reta formada pelos pontos E e D. Com base nessas informações utiliza-se propriedades da trigonometria para encontrar as coordenadas do ponto P. Chamando-se o raio da íris de R, pela equação de distância entre pontos tem-se que:

$$2R = \sqrt{(\mathrm{M.x} - \mathrm{P.x})^2 + (\mathrm{M.y} - \mathrm{P.y})^2}. \quad (1)$$

E através da condição de perpendicularidade entre retas tem-se:

$$\left(\frac{Dy-Ey}{Dx-Ex}\right) * \left(\frac{Py-My}{Px-Mx}\right) = -1. \quad (2)$$

Com essas duas equações é possível montar um sistema. Resolvendo esse sistema, tem-se dois valores possíveis para a coordenada Dx e dois valores para a coordenada Dy, que são as raízes da equação de segundo grau gerada.

Os pares de valores (x1,y1) e (x2,y2) encontrados representam pontos que estão na reta perpendicular a reta ED, em uma distancia 2R do ponto M. Logo, entre esses dois pontos seleciona-se aquele que esta abaixo dos demais pontos como sendo o centro do círculo que representa a pálpebra. O raio desse circulo é dado pelo cálculo da distância entre o ponto P encontrado e um dos pontos que estão no perímetro desse círculo, que são A ou B.

A figura 14 demonstra o circulo encontrado desenhado nas imagens. É possível perceber que o método proposto em [3] consegue de fato realizar a detecção das pálpebras superiores.

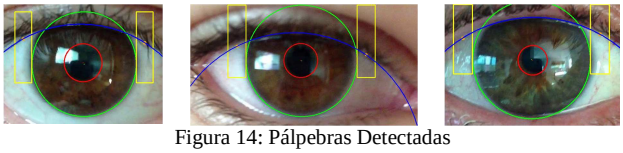


Figura 14: Pálpebras Detectadas

*D. Detecção de Reflexos*

Um reflexo pode ser definido como um ponto de luz que incida diretamente sobre a região da íris [12].

A técnica proposta por Sahmoud e Abuhaiba [3] para remover reflexos das imagens consiste em realizar uma média dos valores dos pixels para cada canal RGB. Têm-se então três médias, uma para cada canal. Como os pixels nas regiões de reflexo tendem a ser mais claros, estabelece-se um limiar com base nessas médias de modo que se um pixel no canal da imagem tiver valor maior que a média dos valores dos pixels naquele canal mais uma constante, esse pixel é considerado pertencer à região do reflexo. A constante é necessária para evitar que pixels mais claros que compõe a textura da íris sejam considerados parte da região do reflexo. Durante os experimentos o valor dessa constante foi ajustado em 30. A figura 15 demonstra o resultado da detecção de reflexos aplicada nas imagens mascaradas.

Figura 15: Reflexos Detectados

*E. Normalização*

O método proposto por Daugman em [1] é o mais utilizado para a normalização de imagens da íris. Neste trabalho foi realizada a segmentação da íris com base na idéia principal deste modelo, conhecido como Rubber Sheet Model, onde é realizada a divisão dos pixels em fatias angulares e radiais em relação ao centro da íris detectada.

Para realizar a normalização, primeiramente é feita a conversão das coordenadas da imagem do plano cartesiano para o plano polar. Na representação polar um pixel deixa de ser identificado por linha e coluna e passa a ser identificado por um ângulo e uma magnitude em relação ao pixel de origem, que foi definido como o pixel do centro da pupila. A magnitude representa então a distância de um pixel para o centro da íris, e o ângulo representa a inclinação da reta formada por aquele ponto e o centro da íris.

Uma vez feita a transformação na representação dos pixels, o próximo passo consiste em dividir os pixels da região da íris em fatias, onde a divisão das fatias toma como base o ângulo que foi atribuído a cada pixel. Na figura 16 tem-se uma ilustração dessa divisão. Observa-se que o centro da íris é a origem do plano polar. Divide-se então a região da íris em k fatias de Θ graus cada uma. Na ilustração percebe-se que existem 8 fatias, logo o Θ é de 45 graus. O número de fatias em que a íris é dividida representa o número de linhas que existirão na imagem normalizada. Foi utilizado um k com valor igual a 2. Ou seja, a região da íris é divida em fatias de 2 graus cada uma, resultando em uma imagem normalizada com 180 linhas.

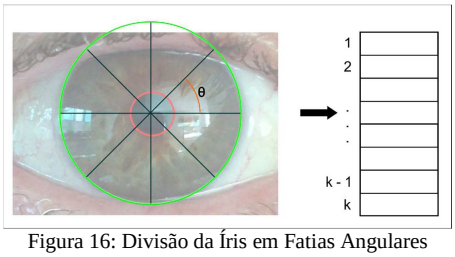



Figura 16: Divisão da Íris em Fatias Angulares

Feita a divisão das fatias, o próximo passo consiste em dividir os pixels de cada fatia em L blocos. A divisão dos blocos toma como base a distância dos pixels para o centro da íris. O número de blocos representa o número de colunas que existirão na imagem normalizada. Na implementação realizada, L recebeu o valor de 50. A figura 17 ilustra este processo de divisão.

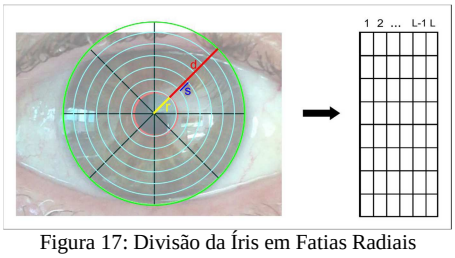



Figura 17: Divisão da Íris em Fatias Radiais

O valor r é dado pela distância do centro da íris para o perímetro da pupila em um dado ângulo e representa a magnitude mínima de um pixel para que este seja normalizado. Pelo fato da íris e da pupila não serem concêntricas o valor de r nem sempre será igual em todos os ângulos. Pensando nisso o círculo da íris foi dividido em 8 novas regiões de 45 graus cada uma, onde cada região possui um valor diferente para r. O cálculo de r para cada região foi feito com base no ângulo de início daquela região. Por exemplo, na região que começa no ângulo 0, r recebeu a distância entre o centro da íris e o perímetro da pupila no ângulo 0. Na região que começa em 45 graus, r recebeu o valor da distância em 45 graus, e assim sucessivamente.

Cada região possui um valor d que é dado pela diferença entre o raio da íris e o valor r. O valor d representa a distância do perímetro da íris para o perímetro da pupila no ângulo de início daquela região. O valor s representa o intervalo de magnitude de pixels que um bloco engloba e é dado pela divisão de d por L. Com base nessas informações os seguintes procedimentos são realizados para determinar a qual fatia, região e bloco um dado pixel pertence:

1. dado o ângulo α de um pixel, descobriu-se a qual fatia f aquele pixel pertence, onde f = α /Θ;

2. para descobrir qual dos 8 valores de r e s utilizar para o pixel, precisa-se localizar à qual região t ele pertence. Tem-se que t = f /45;

3. descoberta a fatia e a região do pixel, resta descobrir em qual bloco daquela fatia o pixel esta localizado. Um pixel pertencerá ao bloco em que sua magnitude estiver entre r + i * s e r + (i + 1) * s, onde i varia de 0 a L.

Agora que todos os blocos foram construídos, resta construir a imagem normalizada. Cada bloco da imagem original será transformado em um pixel na imagem normalizada. Essa transformação é feita através do cálculo da média entre todos os pixels que pertencem àquele bloco. Um pixel participará do cálculo da média somente se ele não estiver mascarado. Logo um bloco que contém todos os pixels mascarados resultará um ponto preto na imagem normalizada. A figura 18 ilustra este cálculo das médias e atribuição destes valores na imagem normalizada.

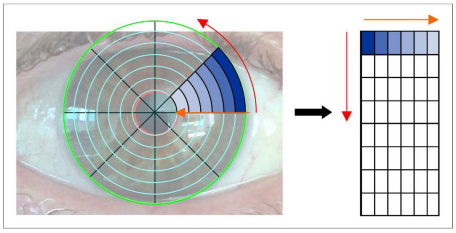


Figura 18: Construção da Imagem Normalizada

A figura 19 demonstra o resultado dessas operações aplicadas nas imagens segmentadas. Com a aplicação da normalização é possível visualizar a região da íris segmentada no mesmo formato utilizado pela maioria dos trabalhos estudados que realizam o reconhecimento biométrico pela íris.

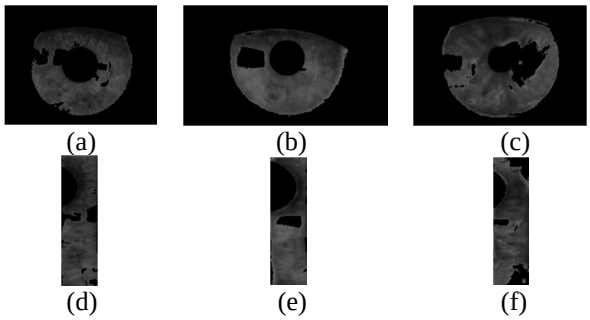



Figura 19: Exemplos de Imagens Normalizadas: (a) (b) (c): Regiões da íris Segmentadas; (d) (e) (f).

## III. RESULTADOS

Para realizar os testes de segmentação, foram utilizados 200 frames de cada vídeo, com o objetivo de padronizar os dados gerados em um número fixo de observações que permitissem a análise de seus resultados.

Embora 3 vídeos possam parecer insuficientes para se medir a efetividade do método, na verdade o que se tem são 600 imagens. Dentre as quais existem variações de iluminação, de posicionamento do olho, de diâmetro da íris; movimentações, perdas de foco, diferenças estruturais entre os olhos e outros fatores que geram uma série de cenários interessantes para se testar a precisão do método demonstrado. Fatores esses dificilmente encontrados em fotos. Pois em geral, ao se capturar uma foto, toma-se o cuidado de posicionar corretamente a câmera, focalizar, controlar iluminação e assim por diante. Fatores que em um vídeo dificilmente são controlados. Desta forma mesmo um fragmento de um vídeo, representa uma diversidade bastante grande de informações.

### A. *Segmentação do Olho*

Para avaliar o desempenho das fases de segmentação do olho, calculou-se a redução média das dimensões das imagens geradas por esse processo. O cálculo foi realizado utilizando a divisão dos pixels da imagem em 4 níveis de intensidade pelo k-means. A tabela 1 demonstra as porcentagens de redução obtidas na largura, altura e área total das imagens.

Tabela 1

| **Propriedade** | **Redução Media** | | | |
|---|---|---|---|---|
| | ***Video A*** | ***Video B*** | ***Video C*** | ***Média*** |
| Altura | 80,5% | 75,9% | 81,1% | 79,2% |
| Largura | 56,1% | 12,7% | 40,3% | 40,3% |
| Área | 91,4% | 78,9% | 90,9% | 87,1% |

Redução Média da Área das Imagens com o k-means em 4 Níveis.

Os resultados obtidos demonstram uma redução média de 80,5% na área das imagens resultantes. Com base na resolução original dos vídeos de 1080x1920. Esse valor representa em média que 1.670.000 pixels desnecessários para a detecção da íris foram retirados das imagens, o que se mostra bastante interessante tendo em vista os diversos processos iterativos necessários para realizar a segmentação da íris.

Dos 600 frames processados utilizando o k-means, não houve caso onde o olho não tenha sido localizado ou tenha sido recortado de maneira que parte da íris fosse eliminada.

### B. *Detecção da Íris e da Pupila*

Por se tratar de uma análise manual, a precisão da análise não pode ser considerada absoluta, apenas uma aproximação, pois em alguns casos é difícil estabelecer visualmente o sucesso ou não de uma segmentação.

A tabela 3 apresenta os resultados obtidos realizando-se a segmentação partindo da detecção da pupila através das dimensões de seu contorno.

Com essa abordagem a taxa média de acertos foi de cerca de 70%, pois com uma detecção precisa da pupila, foi possível aumentar as restrições de detecção da íris, o que possibilitou uma significativa taxa de segmentações corretas.

Tabela 2

| | **Segmentação da Íris** | | |
|---|---|---|---|
| | ***Video A*** | ***Video B*** | ***Video B*** |
| Nº de frames processados | 200 | 200 | 200 |
| Círculo da íris e da pupila encontrados | 195 | 195 | 158 |

| | Segmentação da Íris | | |
|---|---|---|---|
| | ***Video A*** | ***Video B*** | ***Video B*** |
| Segmentações corretas | 157 | 146 | 118 |
| Erros somente na pupila | 1 | 0 | 9 |
| Erros somente na íris | 38 | 44 | 22 |
| Erros em ambos | 0 | 5 | 9 |

Resultados da Segmentação da Íris.

### *C. Detecção das Pálpebras*

Nos testes realizados a detecção das pálpebras superiores se mostrou bastante satisfatória. Dos 195 frames gerados na detecção da íris e da pupila no vídeo A a pálpebra foi detectada em 181 destes. No vídeo B a detecção ocorreu em 193 dos 195 frames processados. No vídeo C a pálpebra foi localizada em 156 de 158 frames.

### *D. Resultado Final*

Considerando finalmente os casos em que em um frame houve sucesso na detecção da íris, pupila e pálpebras; tem-se que a taxa média de sucesso de segmentação dos 600 frames analisados foi de 67,6%, valor esse que corresponde às expectativas de se conseguir uma boa qualidade na segmentação da íris em vídeos obtidos em condições naturais de iluminação.

## IV. CONCLUSÃO

Através das técnicas e métodos implementadas neste trabalho, foi possível alcançar o objetivo principal de segmentar a região da íris de forma precisa em grande parte dos frames dos 3 vídeos obtidos em condições naturais de iluminação. Dado que métodos tradicionais de segmentação exigem que a imagem da íris processada seja capturada sob iluminação infravermelho, estes resultados encorajam o aprimoramento e desenvolvimento de técnicas de segmentação que possibilitem que imagens da íris obtidas em condições não ideais de captura e iluminação sejam utilizadas na construção de sistemas mais acessíveis de reconhecimento biométrico.

Possíveis trabalhos futuros poderiam: 1) medir a eficiência de um sistema biométrico que utilize os métodos de segmentação e normalização aqui desenvolvidos; 2) aplicar métodos de fusão de imagens nos frames dos vídeos normalizados, tais como o proposto em [13] para amenizar a perda de informações causadas pelos ruídos presentes sobre a íris; 3) utilizar e a aprimorar a técnica de detecção da pupila através das dimensões de seu contorno para a análise dos seus movimentos de contração e dilatação em vídeos; 4) Utilizar os resultados da segmentação como um validador de qualidade de vídeos para pesquisa baseando-se no número de frames segmentados com sucesso.

## REFERENCIAS


[1] Daugman, J. How Iris Recognition Works. 2004.
[2] Mansfield, T.; Kelly, G.; Chandler, D.; Kane, J. Biometric Product Testing Final Report. Disponível em http://www.sas.el.utwente.nl/open/courses/intro_biometrics/Mansfield01.pdf. Acesso em: Apr 2014
[3] Sahmoud, S. A.; Abuhaiba, I. S. Efficient iris segmentation method in unconstrained environments. 2013.
[4] Daugman, J. The importance of being random: statistical principles of iris recognition. 2003.
[5] Wildes, R. P. Iris recognition: an emerging biometric technology. 1997.
[6] Santos, G.; Hoyle, E. A fusion approach to unconstrained iris recognition. 2012.
[7] Li, P.; Liu, X. An incremental method for accurate iris segmentation. 2008.
[8] Sahmoud, S. A.; Abuhaiba, I. S. Efficient iris segmentation method in unconstrained environments. 2013.
[9] Hollingsworth, K.; Peters, T.; Bowyer, K.; Flynn, P. Iris Recognition Using Signal-Level Fusion of Frames From Video. 2009.
[10] Canny, John. A computational approach to edge detection. Pattern Analysis and Machine Intelligence, IEEE Transactions on, n. 6, p. 679-698. 1986.
[11] Hough, Paul V.; Powell, Brian W. A method for faster analysis of bubble chamber photographs. Il Nuovo Cimento, v. 18, n. 6, p. 1184-1191. 1960.
[12] Bowyer, K. W.; Hollingsworth, K.; Flynn, P. J. Image understanding for iris biometrics: A survey. 2008.
[13] Jillela, R.; Ross, A.; Flynn, P. J. Information fusion in low-resolution iris videos using Principal Components Transform. 2011.

# Methods to Musical Notes Recognition from Sheet Music: Preliminary Results

Caio Jin Aoqui, Almir Olivette Artero, Marco Antonio Piteri, Danilo Medeiros Eler
UNESP – Univ Estadual Paulista, Faculdade de Ciências e Tecnologia,
Departamento de Matemática e Computação, Presidente Prudente/SP, Brazil
caio_jin@hotmail.com, {almir, piteri, daniloeler}@fct.unesp.br

***Resumo*—This paper presents preliminary studies performed for developing a method to recognize musical notes in sheet music by employing image projections and mathematical morphology operations. Using these operations, our system is able to segment a musical note and get its characteristics (e.g., filling and presence or absence of rods or flags) which can be employed in the recognition step. We perform experiments with 148 images of musical notes providing a success rate of 93.4% for note recognition and 100% for note position identification.**

## I. Introdução

Até hoje, não foi possível determinar com exatidão a origem da música. Ninguém sabe exatamente como, quando ou onde ela surgiu. Seus primeiros vestígios na humanidade aparecem em pinturas de homens das cavernas (há cerca de quarenta mil anos), as quais mostram cenas de dança e sugerem, com sua sofisticação, um provável acompanhamento musical. As primeiras manifestações musicais davam-se por meio de batidas de bastões em troncos de árvores, rochas, percussão corporal, gritos e imitações de sons de animais e da natureza. Desde então, sua utilização apenas se expandiu.

Notação musical é o nome dado para qualquer sistema de escrita de uma peça musical. Ela foi criada para representar visualmente os sons utilizados na música, sendo como um alfabeto para um músico, indicando tom, timbre, tempo de duração, entre outras características. A notação mais utilizada atualmente é a de partituras, as quais se utilizam de símbolos colocados sobre uma pauta de cinco linhas que também pode ser chamada de pentagrama.

Até recentemente, partituras eram encontradas apenas em formato impresso, ou em alguns casos, manuscritos. Com o avanço da informática muitos documentos são produzidos formatos digitais. A partitura está entre eles. Muitos sistemas voltados para a música foram desenvolvidos, como programas que auxiliam compositores, facilitam a edição de seus trabalhos, e até mesmo os reproduzem digitalmente para testes. Peças musicais criadas antes desses sistemas ainda precisam ser digitalizadas, mas existem inúmeras obras e o processo de digitalização seria muito demorado e custoso. Antes da existência de partituras impressas, cada peça era manualmente copiada e algumas delas possuem apenas seu formato manuscrito. Tais peças podem ser perdidas com o tempo. Para preservá-las, pode ser realizado o processo de digitalização, armazenando-as em um formato digital. Portanto, faz-se necessário o desenvolvimento de técnicas para a digitalização e o reconhecimento automática de partituras. Um exemplo de sistema para reconhecimento de partituras manuscritas foi desenvolvido por Rebelo [1].

Bainbridge e Bell[2] afirmam que sistemas de Reconhecimento Óptico de Música (*Optical Music Recognition* ou simplesmente OMR) possuem quatro estágios no processo de reconhecimento:

- **Identificação do pentagrama:** a posição do pentagrama é encontrada e, geralmente suas linhas são removidas;
- **Localização de objetos musicais:** os símbolos pertencentes ao pentagrama são localizados;
- **Identificação de símbolos:** o tipo de cada símbolo é determinado;
- **Semântica da notação musical:** a relação entre os símbolos é determinada e a informação é armazenada em um formato usado por editores de música.

Enquanto estas etapas são comuns nos sistemas, as regras utilizadas para reconhecimento diferem. Bainbridge e Bell [2] desenvolveram um sistema chamado de CANTOR (*Canterbury Optical Music Recognition*), o qual permite que usuários definam as regras de uma notação musical, tendo como objetivo deixar o sistema o mais flexível possível.

Bainbridge e Bell[3] afirmam ainda que OMR é, em princípio, uma extensão do Reconhecimento Óptico de Caracteres (Optical Character Recognition, ou simplesmente OCR). No entanto, ele oferece alguns desafios extras, por exemplo, o posicionamento das notas na vertical distingue seu o tom, enquanto que o posicionamento horizontar indica a ordem em que devem ser tocadas. Em OCR, cada caractere possui diferentes características que o distingue dos demais. Em OMR, cada nota musical possui formatos e características semelhantes, o que pode causar confusões; cada detalhe deve ser minuciosamente analisado e adquirido para que o reconhecimento seja feito com sucesso.

Adicionalmente, alguns requerimentos da notação musical podem levar à duas imagens distintas representarem um mesmo evento musical. Um dos tipos encontrados, é a rotação da nota. Normalmente, quando posicionadas abaixo da terceira linha, a nota tem uma orientação normal. Se, contudo, ela estiver localizada acima da terceira

linha, ela é rotacionada, ou seja, sua orientação é invertida. Se estiver na terceira linha, ela pode estar em qualquer uma das orientações.

Em síntese, este artigo tem como objetivo estudar e implementar técnicas para reconhecer o formato, orientação e posição de notas musicais em partituras.

Este artigo está organizado da seguinte maneira: a Seção II descreve a metodologia adotada; a Seção III demonstra alguns experimentos e resultados; e a Seção IV apresenta algumas conclusões do trabalho desenvolvido.

## II. Descrição da metodologia

Está seção apresenta os métodos propostos para o reconhecimento das notas musicais.

Primeiramente, o sistema isola as imagens das notas do pentagrama. Após carregar a imagem desejada, o próximo passo é binarizá-la. Partituras frequentemente possuem cores apenas em tons de cinza, mas é essencial trabalhar com imagens binárias. O limiar escolhido para as imagens de entrada é fixo: 225. Este alto valor foi escolhido levando-se em consideração que, quanto pior a qualidade de uma imagem, mais ruido ela possui. Desta forma, ela possuirá uma grande variação de valores. Visto que deseja-se obter os objetos da imagem (no caso, as figuras e linhas do pentagrama), não se pode correr o risco de perder informações devido à má qualidade de uma imagem. Em outras palavras, os pixels do objeto serão realçados, evitando a perda de dados.

O sistema desenvolvido trabalha com o pressuposto de que os objetos são brancos e o fundo é preto. Portanto, após a binarização, ainda é necessário inverter os valores da imagem, conforme pode ser observado na Figura 1.

(a) (b) (c)

Figura 1: Imagens de entrada após serem binarizadas e invertidas: (a) semibreve; (b) mínima; (c) semínima.

### A. Reconhecendo o pentagrama

Depois de realizar o pré-processamento da imagem, o primeiro passo da técnica desenvolvida para reconhecer as notas é encontrar a posição do pentagrama. Para isto, é necessário observar a projeção do eixo-y da imagem e os pontos de pico existentes.

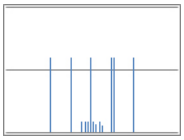


Figura 2: Projeção do eixo-y obtida a partir da Figura 1a

Pode-se observar na Figura 2 que as linhas do pentagrama possuem uma concentração de pixels muito mais alta do que o fundo e os outros objetos da imagem.

Para encontrar as linhas, é necessário percorrer a projeção, procurando os valores mais altos. Quando um pico é encontrado, incrementa-se um contador de picos e sua posição na projeção é armazenada em uma lista. Depois, continua-se a procura até que o contador seja igual ao número de linhas, no caso, cinco.

Ao armazenar as posições das linhas encontradas em uma lista, é possível calcular a distância média entre elas. Após a análise de algumas imagens, foi estimado que a largura da cabeça de uma nota musical é aproximadamente 1.5 vezes maior que sua altura de sua cabeça, a qual é a mesma que a distância entre duas linhas. Este tamanho estimado é armazenado em uma variável.

Ainda analisando a Figura 2, é possível observar que, nem sempre uma linha do pentagrama possui a largura de um pixel; quando dois picos são adjacentes, significa que eles pertencem à uma única linha do pentagrama. Encontrando valores adjacentes, é necessário armazenar o valor da largura da linha. Caso o pico encontrado não tenha outro adjacente, deve-se apenas incrementar um contador de picos

### B. Isolando a nota

Para isolar uma nota, é necessário obter algumas informações a partir da imagem:

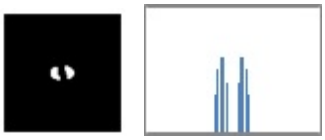
- largura máxima das linhas do pentagrama;
- posição de início e término da nota no eixo-x;
- posição de início e término da nota no eixo-y.

É necessário realizar uma operação morfológica de abertura na imagem [4], [5] para remover o pentagrama. O elemento estruturante usado é uma linha vertical com largura de 1 pixel e altura de $(largura + 2)$, onde $largura$ representa a largura da linha. A imagem resultante conterá apenas as notas.

(a) (b)

Figura 3: (a) Notas musicais mais uma divisão de um pentagrama, após a remoção deste; (b) resultado após uma abertura com elemento estruturante linha horizontal.

Há casos em que haverá uma divisão no pentagrama, representada por uma linha vertical. Para que esta não seja confundida com uma nota, é necessário realizar novamente uma abertura. O elemento estruturante será uma linha horizontal com altura de 1 pixel e largura de $(largura+2)$. Este caso é ilustrado na Figura 3.

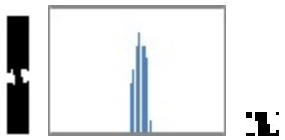


(a) (b)

Figura 4: (a) Figura 1a depois da remoção do pentagrama; (b) projeção do eixo-x da imagem da nota apresentada em (a).

Em seguida, a projeção do eixo-x da imagem é analisada. Pode-se observar na Figura 4 que os pontos onde a nota está localizada possuem pixels com valores diferentes de zero. Percorre-se então cada valor da projeção e, o primeiro valor diferente de zero encontrado indicará o início da nota. O próximo pixel com valor zero encontrado indicará o final da nota. Essa informação é utilizada para recortar a imagem, isolando a nota horizontalmente, conforme indicado na Figura 5a.

Com a imagem resultante, o processo anterior é repetido, percorrendo a projeção do eixo-y à procura do início e do término da nota e, depois de encontrá-los, recortando a imagem e isolando a nota por completo. O resultado é ilustrado na Figura 5c.



(a) (b) (c)

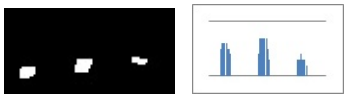
Figura 5: (a) Figura 4a isolada horizontalmente; (b) projeção do eixo-y da nota; (c) nota totalmente isolada.

*1) Uma divisão na projeção:* A Figura 4b mostra a única exceção existente nas notas que serão reconhecidas pelo sistema: há uma divisão na projeção. Com esta divisão, uma única nota pode ser considerada como duas distintas.

Para evitar este problema, é utilizado o tamanho estimado da nota, calculado anteriormente. Quando o início da nota for encontrado, um contador é iniciado. Se, quando um valor igual a zero for encontrado e o valor do contador ainda não se igualou ou ultrapassou o tamanho estimado, o algoritmo continuará percorrendo a projeção.

### C. A classe Nota

Tendo como base as informações já obtidas, é possível observar que será necessário um grande número de variáveis para guardar informações sobre as notas. Analisando a Figura 6a, nota-se que a projeção obtida (Figura 6b) possui três concentrações de pixels, correspondentes às três notas existentes. Com isso, pode-se inferir que é possível reconhecer mais de uma nota presente em uma imagem.



(a) (b)

Figura 6: (a) Três notas musicais com seu pentragrama removido; (b) histograma do eixo-x obtido.

Sabendo que o número de variáveis se multiplicará para cada nota existente, utilizar variáveis de classe para armazenar os valores necessários para isolar e reconhecer as notas não é viável.

A solução adotada foi a de criar uma classe Nota para armazenar os parâmetros mencionados. Na classe reconhecedora, uma lista de objetos Nota é mantida para guardar as notas encontradas.

Os parâmetros armazenados na classe Nota são:

- início e fim da nota no eixo-x, assim como sua largura;
- início e fim da nota no eixo-y, assim como sua altura;
- inicio e fim de sua cabeça no eixo-x, assim como sua largura;
- início e fim de sua cabeça no eixo-y, assim como sua altura;
- sua orientação (normal ou invertida);
- projeções dos eixos $x$ e $y$ da imagem isolada da nota;
- sua forma, depois de reconhecida;
- imagem da nota, assim como imagens auxiliares para uma verificação mais confiável.

Estas informações adquiridas ao longo do processamento são vitais para isolar e reconhecer as notas.

### D. Reconhecendo o formato

Para reconhecer o formato da nota, deve-se diferenciar as características de cada uma. Para facilitar o processo de reconhecimento, as notas foram divididas em dois grupos distintos: preenchidas e não preenchidas. Em cada grupo usou-se procedimentos distintos.

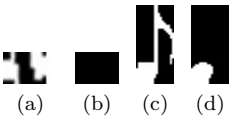
*1) Identificando Preenchimento:* Para identificar se uma nota está preenchida ou não, foi aplicada uma operação morfológica de abertura na imagem isolada anteriormente. O elemento estruturante será uma elipse, com largura e altura $estimado/2$, onde $estimado$ é o tamanho estimado da nota.

Há dois resultados possíveis para esta operação:

- **A imagem ficará vazia:** neste caso, afirma-se que a nota não possui preenchimento;

- **Restará apenas a cabeça da nota:** sabe-se, então, que a nota é preenchida.

Os resultados mencionados podem ser observados na Figura 7.



(a) (b) (c) (d)

Figura 7: (a) Semibreve isolada; (b) semibreve após abertura; (c) colcheia isolada; (d) colcheia após abertura.

*2) Obtendo a orientação da nota:* O próximo passo para o reconhecimento das notas é obter sua orientação. Depois, resta apenas distinguir sua forma. Se sua cabeça estiver na parte superior da imagem (haste do lado esquerdo), ela está invertida. Se sua cabeça estiver na parte inferior da imagem (haste do lado direito), ela está normal. Para cada grupo de notas, preenchidas e não preenchidas, serão utilizados procedimentos distintos para obter sua orientação.

*3) Notas Não Preenchidas:* Das notas estudadas, apenas duas não possuem preenchimento: semibreve e mínima.

O único fator que as diferencia é a presença ou ausência de uma haste.

Para verificar se a nota possui ou não uma haste, é necessário observar novamente sua projeção. O maior valor existente no eixo-x é procurado e, quando encontrado, seu valor deve ser verificado. Se este for maior do que a nota, concluímos que ela possui uma haste. Logo, ela é uma mínima. Verifica-se ainda se a haste foi encontrada no começo ou no final da nota para obter sua orientação.

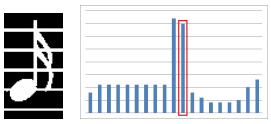
Se a imagem não possuir uma haste, então ela é identificada como uma semibreve e sua orientação é classificada como normal.

*4) Notas Preenchidas:* A característica que diferencia as notas preenchidas é a ausência ou presença de bandeiras e, quando presentes, a sua quantidade.

Primeiro, deve-se verificar a orientação da nota. Como visto na Figura 7d, pode-se identificar a posição da cabeça da nota dividindo a imagem em duas partes, superior e inferior, e somando seus pixels. Enquanto a soma dos pixels é realizada, também são verificados o início, o fim, a altura e a largura da cabeça para uso posterior. A parte com maior soma de pixels contém a cabeça. Pode-se então classificá-la como normal ou invertida.

Se a nota estiver normal, a largura de sua cabeça é comparada com a sua própria largura. Se ambas possuírem o mesmo valor, ou um valor próximo, significa que não há bandeiras. Neste caso, ela é identificada como semínima.

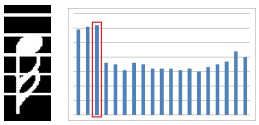
No caso de haver bandeira(s), deve-se encontrar o ponto em que a haste termina, como indicado na Figura 8b. É neste ponto que as bandeiras têm início. A coluna de pixels seguinte deve ser percorrida, fazendo uma contagem do número de bandeiras. Se for encontrada apenas uma bandeira, a nota é identificada como uma colcheia. Se forem encontradas duas bandeiras, ela é identificada como uma semicolcheia.



(a) (b)

Figura 8: (a) Semicolcheia com orientação normal; (b) projeção indicando a coluna em que a haste termina.

Para uma nota invertida, a largura de sua cabeça sempre será igual ao de sua largura. Neste caso, deve-se encontrar o final de sua haste (ilustrado na Figura 9b) e fazer uma varredura na coluna seguinte. Como a cabeça da nota está alinhada com as bandeiras, será encontrado 1 objeto se não houver bandeiras, 2 objetos se houver uma única bandeira e 3 objetos se houver duas bandeiras . Com base neste resultado, basta apenas classificar a nota em semínima, colcheia ou semicolcheia respectivamente.



(a) (b)

Figura 9: (a) Semicolcheia com orientação invertida; (b) projeção indicando a coluna em que a haste termina.

*E. Identificando a posição no pentagrama*

Após reconhecer o formato das notas, resta apenas identificar sua posição no pentagrama.

Para identificar se uma nota está contida em um espaço, é realizada uma contagem de pixels entre duas linhas. Esse procedimento é repetido para todas as linhas do pentagrama.

Figura 10: Exemplos de áreas (destacadas em vermelho) de pixels que o algoritmo percorre.

Para identificar se uma nota está contida em uma linha, o processo anterior é repetido, mas a contagem tem início no ponto médio entre duas linhas e termina quando a distância média entre duas linhas for percorrida.

Comparando os resultados obtidos, aquele que contém uma maior concentração de pixels é a localização da nota.

Figura 11: Exemplo de como a posição das notas serão indicadas.

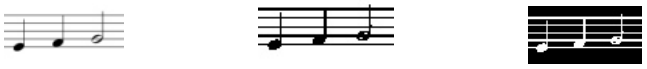
Visto que, a comparação é iniciada da primeira linha de cima para baixo, os resultados obtidos utilizarão o mesmo padrão. Um resultado indicando a 'linha 1' se refere à primeira linha de cima para baixo. Por outro lado, um resultado como 'espaço 4' diz respeito ao quarto espaço de baixo para cima, a partir da primeira linha.

Como indicado na Figura 11, o algoritmo reconhece notas posicionadas até uma hipotética sexta linha, e até um quinto espaço.

*F. Problemas e soluções*

As operações morfológicas nem sempre produzem resultados precisos. Quando se trata de notação musical, um mero pixel pode fazer a diferença. Uma forma de tentar evitar a influência de tais problemas na metodologia desenvolvida, é o uso de imagens auxiliares, mencionadas inicialmente na Seção II-C.

*1) Caso 1:* Há alguns casos em que uma nota não preenchida pode ser considerada como preenchida. Esta situação pode ocorrer quando uma linha com largura razoável atravessa a nota e, durante seu pré-processamento ou em seu isolamento ela é parcialmente preenchida.



(a) (b) (c)

Figura 12: (a) Pentagrama com notas musicais; (b) imagem binarizada; (c) imagem invertida. Nota-se que a última nota foi parcialmente preenchida pela linha do pentagrama.

A solução para este possível problema é utilizar uma imagem auxiliar. Antes que a operação de abertura (usando uma linha vertical como elemento estruturante) possa ser realizada para remover o pentagrama, o algoritmo faz uma abertura com uma elipse, idêntica à operação realizada na Seção II-D1: o tamanho do círculo é igual a metade do tamanho estimado da nota.

As notas não preenchidas irão desaparecer, deixando apenas as notas preenchidas. A imagem auxiliar então é verificada, comparando os pontos de início e término da nota contidos nas variáveis da classe Nota. Se ela estiver vazia (ou seja, se nenhum pixel na área verificada é um pixel de objeto), ela é considerada uma nota não preenchida.

(a) (b)

Figura 13: (a) Pentagrama com notas musicais; (b) Imagem auxiliar resultante da abertura com elemento estruturante elipse.

Pode-se observar que para o caso ilustrado na Figura 13 a técnica funciona.

*2) Caso 2:* O caso ilustrado na Figura 12 porém, mesmo após utilizar a técnica descrita, apresenta um resultado que ainda acusaria a última nota, uma não preenchida, como tendo preenchimento. O resultado é apresentado na Figura 14.

Figura 14: Exemplo de um caso em que a técnica descrita não obtém sucesso.

Este caso acontece devido ao uso de um limiar muito alto na binarização da imagem. Visto que, praticamente todo pixel que não seja de fundo é considerado um objeto, se a linha possuir um valor alto de largura, ela afetará as notas.

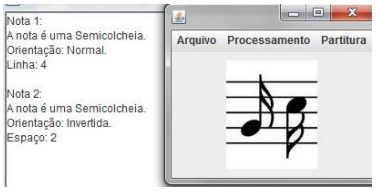
A solução para esta situação é a utilização de uma segunda imagem auxiliar. Esta será binarizada com um limiar menor. Desta forma, há uma chance menor de ela ser preenchida pela linha.

Diferente do caso anterior, nenhuma operação é realizada na imagem inteira. Primeiro, é feito o isolamento da nota. Depois, é realizada uma operação de abertura com uma elipse, como no caso anterior. Finalmente, ela é verificada para verificar os pixels da imagens estão vazios ou não. Se estiverem, ela é indicada como não preenchida.

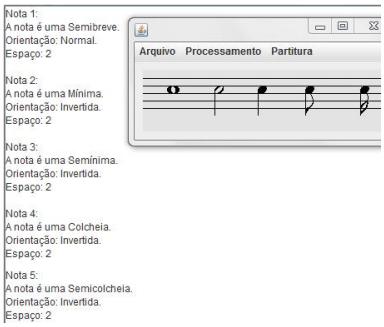
## III. Experimentos

Foram realizados diversos experimentos com notas musicais para testar a eficiência da metodologia. Primeiramente, foram utilizadas imagens isoladas de todas as cinco notas musicais estudadas. Posteriormente, foram testadas imagens com mais de uma mesma nota musical para verificar se o número de notas afetaria a técnica, conforme ilustra a Figura 15a. Finalmente, foram realizados experimentos com imagens contendo múltiplas notas musicais com formatos distintos. Pode-se observar no teste efetuado na Figura 15b que todas os formatos das notas foram reconhecidos, assim como sua posição no pentagrama (espaço 2).

O conjunto de notas utilizados durante os experimentos foi retirado de diversas imagens digitais de partituras tais como a apresentada na Figura 16a. Como pode ser observado neste exemplo, as notas mais comuns encontradas em



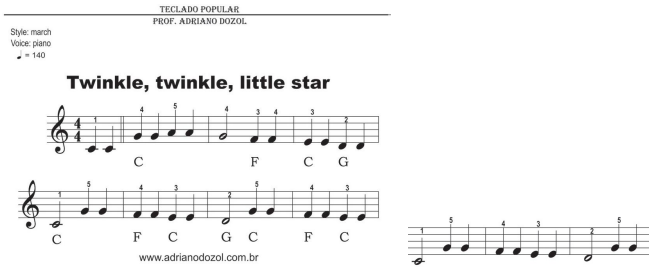

(a)




(b)

Figura 15: (a) Teste feito com duas semicolcheias, uma normal e uma invertida; (b) Teste feito várias notas em uma pauta, com formatos diferentes.

partituras básicas são as semínimas. Portanto, o número de testes realizados com esta nota é muito superior às demais. Convém salientar que as pautas foram retiradas da partitura e fornecidas para o método proposto, conforme ilustrado na Figura 16b.




(a) (b)

Figura 16: (a) Exemplo de uma partitura de onde notas musicais foram retiradas para teste; (b) partitura recortada. (Retirado de: Adriano Dozol[6])

Os resultados obtidos no reconhecimento de formato das notas podem ser observados na Tabela I. Com base nos dados apresentados, observa-se que a metodologia desenvolvida tem 93,24% de acerto.

Na Tabela II se encontram os resultados obtidos com relação à posição das notas musicais testadas. A taxa de acerto, neste caso, é de 100%.

## IV. Conclusão

Neste trabalho foi desenvolvida uma metodologia para segmentação e reconhecimento de notas musicais. As operações morfológicas e técnicas de segmentação merecem destaque nos estudos, pois foram o foco da implementação. Foram estudadas diversas formas de limiarização, entre elas a binarização de Otsu, e diversos testes foram feitos para poder identificar um limiar que não prejudicasse a imagem.

| | Imagens analisadas | Acertos | Erros |
|---|---|---|---|
| Semibreve | 14 | 14 | 0 |
| Mínima | 28 | 25 | 3 |
| Semínima | 80 | 76 | 4 |
| Colcheia | 15 | 13 | 2 |
| Semicolhciea | 11 | 10 | 1 |

Tabela I: Resultados do reconhecimento de formato.

| | Imagens analisadas | Acertos | Erros |
|---|---|---|---|
| Linha 1 | 6 | 6 | 0 |
| Linha 2 | 7 | 7 | 0 |
| Linha 3 | 13 | 13 | 0 |
| Linha 4 | 27 | 27 | 0 |
| Linha 5 | 20 | 20 | 0 |
| Linha 6 | 10 | 10 | 0 |
| Espaço 1 | 7 | 7 | 0 |
| Espaço 2 | 7 | 7 | 0 |
| Espaço 3 | 20 | 20 | 0 |
| Espaço 4 | 21 | 21 | 0 |
| Espaço 5 | 14 | 14 | 0 |

Tabela II: Resultados do reconhecimento de posição.

Nos experimentos, foram analisados os formatos de 148 notas musicais distintas e 138 foram reconhecidos com sucesso. A taxa de acerto é de 93,24%. Em relação à posição da nota no pentagrama, das 148 notas analisadas, 148 foram reconhecidas na posição correta. Portanto, os resultados obtidos com as técnicas implementadas foram satisfatórios, apesar de possuir algumas limitações que serão tratadas em trabalhos futuros. Algumas delas são: detecção de claves, reconhecimento de notas em linhas suplementares, detecção em vários pentagramas e notas com ligadura.

## Agradecimentos

Os autores agradecem ao apoio financeiro recebido da Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP).

## Referências

[1] A. M. Rebelo, "Robust optical recognition of handwritten musical scores based on domain knowledge," Ph.D. dissertation, University of Porto, 2012.

[2] D. Bainbridge and T. Bell, "An extensible optical music recognition system," in *Nineteenth Australasian Computer Science Conference*.

[3] ——, "The challenge of optical music recognition," pp. 95–121, May 2001.

[4] L. F. Costa and R. M. C. Jr, *Shape Analysis and Classification*. CRC Press, 2001.

[5] R. Gonzales and R. E. Woods, *Processamento Digital de Imagens*. Edgard Blucher, 2000.

[6] A. Dozol, "Twinkle, twinkle, little star - brilha, brilha estrelinha (partitura para teclado)," 2011.

# Real-Time Hand Gesture Recognition Based on Sparse Positional Data

Tatiana Schmidt, Felipe P. Araujo, Gisele L. Pappa, Erickson R. Nascimento
DCC - Departamento de Ciência da Computação
Universidade Federal de Minas Gerais, UFMG
{tati.sg, felipea, glpappa, erickson}@dcc.ufmg.br

***Abstract*—Low cost devices which provide depth data as well as images (RGB-D), such as the Kinect, have become extremely popular of late. A compact and accurate device launched recently draws attention: the Leap Motion. But despite its high acquisition rate and precision, the data it provides is extremely sparse. In this paper we present a methodology for gesture recognition capable of dealing with the sparse data provided by the sensor. The system extracts meaningful characteristics from the data, such as angles between fingers, to achieve a high-accuracy and noise-robust prediction system, which uses a classifier to decide which gesture is being performed. We conduct experiments to evaluate classifiers, comparing SVM against Random Forest. This paper also presents a dataset with $11$ different gestures performed by $6$ users. The experiments show that our methodology provides an accuracy of $94.26\%$ in the dataset by using Random Forest with real-time processing ($0.2$ms to predict a gesture).**

***Keywords*-Hand gesture recognition; Real-time; SVM; Random forest; Computer Vision;**

Fig. 1. An example of our gesture recognition methodology running with Leap Motion device. The name of the gesture identified by the system is shown in the monitor.

## I. Introduction

In the last few years, robust gesture recognition has been receiving a lot of attention in many researches in the computer vision field [1], [2], [3], [4]. This increasing attention is motivated by new depth and image capture devices that provide innovative styles of human-computer interaction, more intuitive and with a range of commands greater than traditional devices, such as keyboards, mice and controllers. The applications that can benefit from these innovative interactions are countless, and include improving accessibility, making more interactive and immersive games, allowing robot controlling and intelligent-rooms operation.

This work focuses on applications where the user interacts with his desktop from a sitting position using these innovative ways of interaction. Hence, it focuses on hand gestures, which demands less effort than body gestures and are more suitable to the sitting position. Several works have used Kinect ([4], [3]) to create gesture recognition systems. Although this choice brings many advantages, such as offering a depth image as well as a color image, it is not the most appropriate for the system described here. This is so because Kinect[1] is a tool originally designed to work with Xbox, a gaming console, and for this purpose it detects the whole body and requires considerable space to work properly. From a close-up perspective, such as users in front of computers, other devices are more appropriate. Among them is Leap Motion, which has the size of a flash drive, was created for personal computers and has already been embedded in some personal computers made by HP.

Leap Motion[2] (see Figure 1) is an accessible and portable device produced by Leap Motion, Inc. that uses 2 infrared cameras and 3 infrared LEDs to create a hand detection system that works with a standard deviation of less than 0.5mm on static scenarios [5]. It allows for very precise and minute movement detection and makes possible whole new ways of interacting with computers. Such fine-grain precision makes it the most suitable device for our objectives. On the other hand, the device does not provide point cloud data, as its API already processes and treats the data acquired by the sensor, giving us only keypoints and orientation vectors of the depth image. Hence, one of the main challenges of dealing with Leap Motion is data scarcity.

In this work we present a hand gesture recognition system for this novel device. Our system creates feature vectors from the sparse information from the sensor providing meaningful descriptors for each gesture. These descriptors are then used by a classifier to decide, in real-time, which gesture is being performed by the user.

The remainder of the paper is organized as follows: Section II reviews similar works. Section III describes our method. An experimental analysis is made in Section IV.Finally, Section

[1] www.xbox.com/en-US/KINECT

[2] www.leapmotion.com

V presents our conclusions and future work directions.

## II. Related work

In the last decades computer vision algorithms relied on the color or intensity of images. Applications such as the Camera-Mouse [6], which tracks a certain keypoint in a stream of images to control a computer's pointing device, including the mouse, were based on RGB data. With the popularization of RGB-D (color images and depth maps synchronized) capture devices, mainly due to the Microsoft Kinect, the gesture recognition field had a great push forward. Several works were published with different objectives [3], [4], [7]. Sign language translation, for example, a great challenge of Computer Vision, was tackled by [3]. They used a 3D trajectory description of one sign language word and matched it against a gallery of trajectories. [4], in contrast, used an RGB-D image from the Kinect to recognize the letters of the manual alphabet, or fingerspelling.

The problem of hand-gesture recognition has already been approached by [8], [9], [1], [2]. In the first three works, a time-of-flight camera is used, while the last one uses Kinect. A time-of-flight camera calculates the depth of an object by measuring the time that the light takes to travel from the camera to the object of interest and back. These works use data from a point cloud and require further processing for hand detection before actually detecting gestures. [8] and [9] use simple techniques to detect hands, assuming the hand will be inside a specific range of depth. We skip this step, as the Leap Motion already handles the detection by itself.

Unlike most methods, which rely on dense data such as 3D point clouds, our approach uses a small set of 3D points and normals. This data is obtained through the Leap Motion's API [10]. In spite of the lack of a rich data set, our system performs high quality static gesture recognition.

## III. Methodology

This section describes the methodology developed to recognize gestures from sparse positional data. Our approach is composed of three main steps, presented in the flowchart in Figure 2. The first step is collecting data to create a library of gestures. In this phase, we receive basic attributes from the Leap Motion device and use them to estimate more robust attributes for gesture recognition. The second step consists in inputting these features vectors to a classifier, which generates a model for each gesture. Here, different classifiers were tested and evaluated according to their accuracy and execution time. In the third step the trained classifier is used to recognize users' gestures in a real time application, which identifies and shows the gesture in the computer's screen, as shown in Figure 1.

In section III-A we describe the method used for collecting data and generating meaningful features. In section III-B we describe the methods used for creating a classifier model. Finally, in section III-C we describe the real-time gesture recognition application.

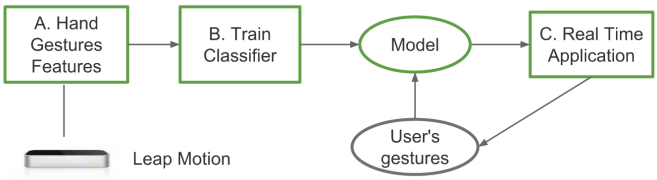



Fig. 2. The main steps of our system. After the collection phase, where data is read from the Leap Motion, the system creates features vectors that are used to train a classifier, which is used to recognize gestures in real-time.

### A. Hand Gestures Features

Data was collected using Leap Motion. This device provides six 3D points: the center of the hand and the location of each visible fingertip. Having only six points at most, data is extremely sparse. Apart from these points, we have only a normal of the palm (located at its center) and the radius of the sphere created by the hand's curvature. Thus, for each gesture in a frame we can read from the device up to eight three-dimensional points.

The absolute position of the hand and fingers are not relevant as gesture features (a gesture is always the same, no matter where it is made), but they can be used to obtain other meaningful features. We generate a feature vector based on the angle between two fingers located at $p_i$ and $p_j$, the angle between a finger and the hand's normal and the distance from the center of the hand to each fingertip. These 16 features are more robust for gesture recognition than using only positional information, since they do not vary with the hand's shift in global position, but with the hand's shift in shape, which distinguishes between hands gestures.

Let $i \in \{thumb, index, middle, ring, little\}$ be the index of a finger, with its fingertip located at $p_i$, $c \in R^3$ be the center of the hand and $n \in R^3$ its normal. The vector of finger $i$ is computed by

$$\vec{v}_i = p_i - c.$$

The feature vector $\mathcal{F}$ is created by concatenating 4 vectors. The first is built by computing the norm of the finger's vector:

$$\boldsymbol{f_1} = [\|\vec{\boldsymbol{v}}_{\boldsymbol{thumb}}\|, \|\vec{\boldsymbol{v}}_{\boldsymbol{index}}\|, \ldots, \|\vec{\boldsymbol{v}}_{\boldsymbol{little}}\|]^{\boldsymbol{T}}, \quad (1)$$

the second vector contains the angles between the vector of adjacent fingers:

$$\boldsymbol{f_2} = \begin{bmatrix} \langle \hat{\boldsymbol{v}}_{\boldsymbol{thumb}}, \hat{\boldsymbol{v}}_{\boldsymbol{index}} \rangle \\ \langle \hat{\boldsymbol{v}}_{\boldsymbol{index}}, \hat{\boldsymbol{v}}_{\boldsymbol{middle}} \rangle \\ \langle \hat{\boldsymbol{v}}_{\boldsymbol{middle}}, \hat{\boldsymbol{v}}_{\boldsymbol{ring}} \rangle \\ \langle \hat{\boldsymbol{v}}_{\boldsymbol{ring}}, \hat{\boldsymbol{v}}_{\boldsymbol{little}} \rangle \end{bmatrix}, \quad (2)$$

the third vector has the angles between the finger vector and the hand's normal:

$$\boldsymbol{f_3} = \begin{bmatrix} \langle \hat{\boldsymbol{v}}_{\boldsymbol{thumb}}, \hat{\boldsymbol{v}}_{\boldsymbol{normal}}, \rangle \\ \vdots \\ \langle \hat{\boldsymbol{v}}_{\boldsymbol{little}}, \hat{\boldsymbol{v}}_{\boldsymbol{normal}} \rangle \end{bmatrix}, \quad (3)$$

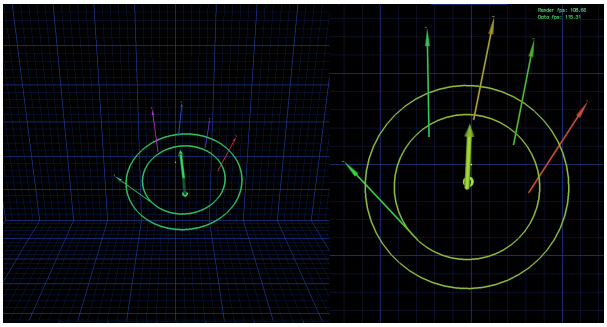


Fig. 3. An example of the data available in Leap Motion device. For each gesture we can read from the device 8 data: five 3D points with the location of the fingertips (represented by vectors in the figure), the center of the palm and hand's normal indicated by a green bold vector.

where $\|.\|$ gives the norm of a vector and $\langle . \rangle$ is the dot product (all vectors are normalized to compute the dot product, i.e. $\hat{v}_{thumb} = \frac{\vec{v}_{thumb}}{\|\vec{v}_{thumb}\|}$).

We also include a fourth vector, which is composed of two measures provided by the device that we considered relevant: the radius of the sphere created by the hand's curvature and the number of fingers detected:

$$\boldsymbol{f_4} = [\boldsymbol{radius_sphere}, \boldsymbol{number_fingers}]^{\boldsymbol{T}}. \quad (4)$$

The final feature vector is given by:

$$\boldsymbol{\mathcal{F}} = [\boldsymbol{f_1^T}, \boldsymbol{f_2^T}, \boldsymbol{f_3^T}, \boldsymbol{f_4^T}]^{\boldsymbol{T}}. \quad (5)$$

The image in Figure 3 shows the vectors available when presenting the gesture of an open hand to the device. One can see the fingers' vectors and hand normal. It also illustrates clear the limited amount of prior information available, which can be even further reduced if not all fingers are detected. This is one of the challenges that our system has to overcome.

### B. Train Classifiers

We tested two different classifiers: Support Vector Machines (SVM) and Random Forest (RF) [11]. These methods were chosen because they are considered the state-of-the art method for many different applications.

SVM models classification as an optimization problem. Each instance is a vector, where each feature represents a different dimension. SVM separates the space into two different subspaces with a hyperplane. It assumes that data can be linearly separable or that there is a kernel function able to transform non-linear spaces into linear ones. The method was conceived to work with two classes, but it can be extended to work with multiple classes by assuming one class as true and the remaining as false (one against all). In this way, it generates $n$ models, where $n$ is the number of classes and a new example is classified using the most confident of these $n$ models.

A Random Forest classifier [11] generates a number of decision trees from the training set. The forest is an ensemble of many randomly trained decision trees. Since the decision trees are built from random subsets of examples, a Random Forest classifier is in general robust and presents a high generalization power due to the decorrelation provided by the randomness of the trees. A test using a Random Forest is performed by simply walking in all trees of the forest until the corresponding leaf is reached, which represents the predicted class. The final classification can be computed by combining all tree predictions into a single forest prediction by a simple averaging operation among the posterior distribution of each decision tree. Therefore, the time it takes to make a prediction depends on the number and depth of the trees, and can be easily parallelized. Another benefit of using Random Forest is its capability of providing high accuracy rates even when using small feature vectors.

In order to compare the classifiers, in this step, we test both using a 10-fold cross-validation procedure. For the purpose of selecting the most appropriate model for real-time prediction, we evaluated performance measures such as accuracy, precision and recall, and also time consumption. The values of parameters and results evaluated will be detailed in section IV-B.

### C. Real-Time Application

The model created in the training phase is used to predict what gesture a user is currently presenting to the system. An interval of $f$ frames is considered to decide what is the user's gesture. This group of $f$ frames is classified and the class that is predicted more often is recognized by the application. On one hand, larger values of $f$ can be used to increase the confidence of the users' gesture identified by system. On the other hand, the values for $f$ should not be too large or the system will take too long to decide on a gesture and the user could even change his hand before a prediction is made.

We chose for this work an interval of $5$ frames. Since the device's average frame rate is $115$ frames per second we can expect that in most cases the user will not have changed the gesture in $\frac{1}{23}$ of a second.

## IV. Experiments

In order to evaluate the robustness of our methodology we performed several experiments. In this section we show the performance of our hand gesture recognition by using a dataset with different hand gestures and users. We built a dataset composed of eleven hand gestures with different individuals performing all gestures. We chose gestures commonly used in other works. Section IV-A explains how the dataset was created. Section IV-B details the parameters chosen to run ours tests and the results obtained.

### A. Hand Gesture Dataset

For our experiments we selected $11$ gestures based on other hand gesture recognition works, and gathered input from $6$ different users. Each user performed each gesture to the device for approximately $10$ seconds. Along those $10$

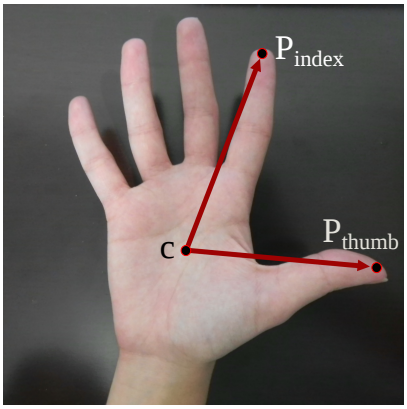



Fig. 5. Distance from center of the hand to index and thumb fingertips. These locations are used to created the feature vectors in training and test phase.

seconds the users rotated and moved their hands up, down and sideways. The final data is composed of $90,678$ instances, each instance being a descriptor of a frame captured by the device. Figure 4 shows one example for each gesture executed by a user.

*1) Users' Hands:* Our main goal is to create an application which is capable of recognizing gestures independently of the user. As different individuals usually have different shapes and sizes of hands, the same gesture will vary from user to user. Some might, for example, present a certain gestures with their hands slightly tilted to the right or left. To test the robustness of our approach we used users with different hand sizes to compose the dataset. Table I shows the distances from the center of the hand to the fingertip of each finger in $mm$ for each user, while Figure 5 illustrates these measurements.

In addition to the variability added to the database by using distinct hand sizes, different users also present large variance in the way they interact with the device. Some users preferred subtle movements, others tried to move their hands as quickly as possible. Some kept their gestures more orthogonal to the device, while others moved their hands in every direction. Thus, our dataset covers a large number of challenging situations which are added to the noise from the device's collection of the data.

| User | Thumb | Index | Middle | Ring | Little |
|---|---|---|---|---|---|
| $User1$ | 115.81 | 121.32 | 106.46 | 107.94 | 94.58 |
| $User2$ | 112.03 | 109.36 | 104.23 | 96.67 | 87.71 |
| $User3$ | 111.41 | 104.27 | 101.43 | 95.29 | 93.24 |
| $User4$ | 108.71 | 103.79 | 100.70 | 93.39 | 91.83 |
| $User5$ | 105.25 | 98.44 | 100.03 | 92.51 | 80.95 |
| $User6$ | 91.20 | 90.21 | 86.72 | 82.00 | 73.75 |
| $\sigma$ | 7.92 | 9.55 | 6.30 | 7.60 | 7.44 |

TABLE I
USERS' HAND (IN $mm$) SIZE USED TO CREATED THE DATASET. THE STANDARD DEVIATION FOR EACH FINGER IS SHOWN IN THE LAST ROW.

| Classifier | Accuracy | Precision | Recall | Time |
|---|---|---|---|---|
| $SVM$ | 89.64% | 89.9% | 89.6% | 592.19 s |
| $RF$ | **94.26%** | **94.3%** | **94.3%** | **99.88** s |

TABLE II
COMPARISON OF ACCURACY AND THE AVERAGE OF PRECISION, RECALL, F-MEASURE AND TRAINING TIME. RANDOM FOREST CLASSIFIER PROVIDED THE HIGHER ACCURACY WITH LESS PROCESSING TIME.

*2) Gestures:* The gestures presented in Figure 4 were chosen according to other works on hand gesture recognition [1], [9], [4], [12]. The gestures $One$, $Two$, $Three$, $Four$, $FiveOpen$, $FiveClosed$ and $ThumbUp$ appear in most works in the literature. Some gestures used in our dataset are similar to the ones presented in the American Sign Language (ASL). For example, $HangLoose$, $Three$, $Two$, $CrossedFingers$ and $Loser$ are similar to the ASL letters $Y$, $W$, $V$, $R$ and $L$, respectively. Additionally, we added the gesture $HeavyMetal$, because it is a popular gesture in society and it can be challenging to our system since there are two other gestures with 2 fingers: $HangLoose$ and $Two$.

### B. Results

Our evaluation was performed in the created dataset, which has 90,678 instances, using a 10-fold cross-validation. Figure 6 shows the confusion matrices for SVM and Random Forest. The diagonal of the matrix shows the correctly classified examples. The darker a cell is, the more examples it correctly classified for that class. Observing the confusion matrices, one can see that some gestures are more challenging than others. For example, $CrossedFingers$ and $One$ are often mistaken with one another. The same happens with $CrossedFingers$ and $FiveClosed$. This is due to a limitation of the sensor, which detects them as a single pointing, and there are not enough characteristics to differentiate between them. For SVM, $CrossedFingers$ gives lots of results in several False Positives and False Negatives. Random Forest also has some problems with fails for some instances of this gesture, but it is less problematic. Overall, Random Forest performs better than SVM.

Our methodology presented an accuracy of up to 94.26% when using Random Forest with 100 trees and depth of 25, and of 89.64% when using SVM. The best results we obtained using SVM and Random Forest are exhibited in Table II. The last column shows the time it took for each one to build its model. Random Forest outperforms SVM in all performance measures, and builds its model faster than SVM by a large margin.

The classifier validation method used was cross-validation with 10-folds, and we used Weka [13] to run it. The parameters used for SVM on our validation were the default ones, but in our system the parameters for SVM were determined by the OpenCV [14] function that finds the best values with a cross-validation of $k$-folds. As for our Random Forest implementation, which was implemented in our system using OpenCV

Fig. 4. Dataset used in our experiments. It is composed of eleven hand gestures performed by six users.

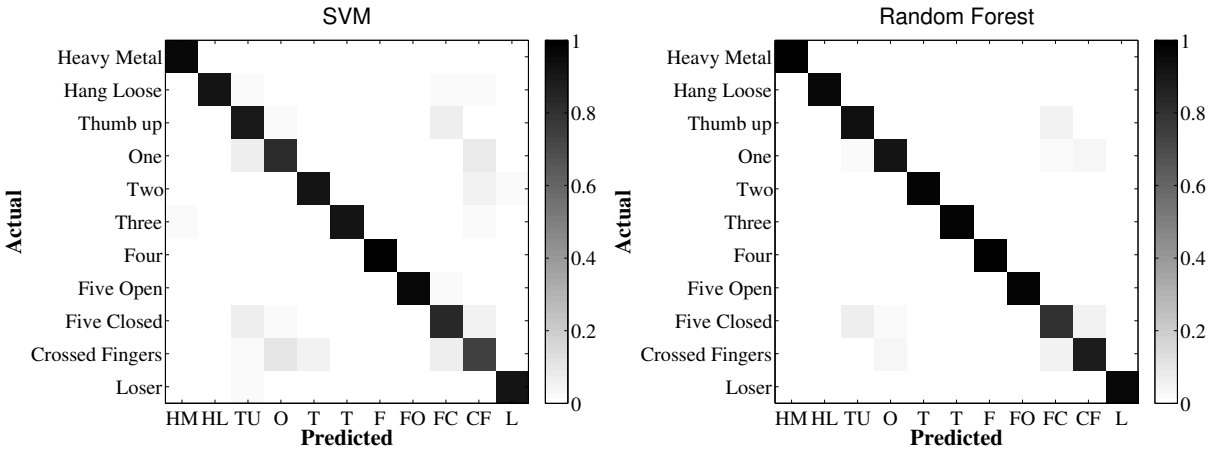



Fig. 6. Confusion Matrix using SVM and Random Forest, respectively. We can see a large spread for SVM confusion matrix. Indeed, Random Forest classifier presented the higher average accuracy (94.26%).

but was validated using Weka, we tested for different values of number of trees and maximum depth of trees. Keeping in mind that if these numbers were too big the prediction would be delayed and the Real Time application would be compromised. The values chosen for maximum depth of the trees were: 5, 15, 25, 35 and 45. And for number of trees in the forests we tested for: $10, 50, 100, 200$ and 300 trees. For the depth we found that there is no improvement beyond 25 for the tested values. As for the number of trees 100 was the best option, as raising the number of trees did not improve our results and increased the time considerably.

Finally, we compare the time of these classifiers to build a model and to predict a class. Table II shows that Random Forest runs faster and yields better results than SVM. Therefore, we chose Random Forest for our system. Furthermore, Random Forest is about 10 times faster than SVM when predicting a class, although both take a really small time to do it. In our tests, Random Forest took an average of $0.2ms$ to determine a gesture from 5 instances, using our voting system, while SVM took $2ms$ to perform the same task.

The time for Random Forest building a model is sensitive to the number of trees and their depths. We analyze the impact of these parameters to build a model for 90,678 instances. Random Forest, as expected, has the computational cost directly proportional to the number of trees (Figure 7). Since we do not have a significant performance gain by using

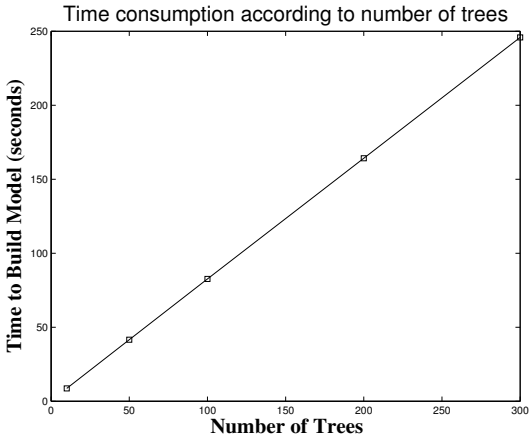



Fig. 7. Processing time for training of Random Forest according to number of trees. In experiments was used a forest composed of 100 trees.

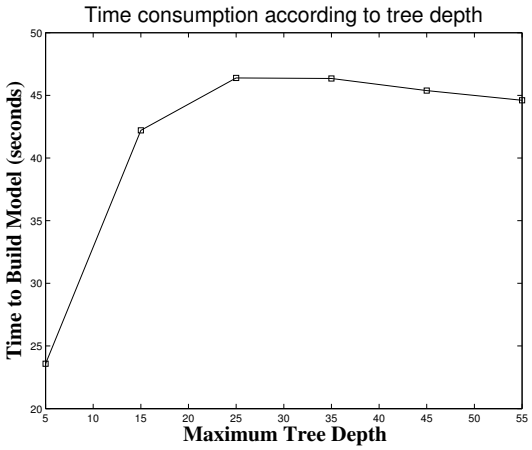



Fig. 8. Time consumption for Random Forest classifier according to the maximum depth of trees. In our experiments we use 25 as maximum depth due to the decrease of accuracy for larger depths.

a forest with more than 100 trees, we used 100 trees. The maximum depth of the trees also affects the computational cost, but its effect reaches a limit. Figure 8 shows that after 25 this parameter does not affect the time it takes to build a model, but the accuracy deteriorates. Therefore, we chose 25 as our maximum depth.

## V. Conclusion and Future Work

In this paper we presented a methodology for identifying hand gestures from sparse positional data. It uses the Leap Motion controller to obtain the number of fingers, hand sphere radius, fingertips, hand position and normal. From this data, we calculate the angles between fingers, angles between normal and fingers and distances from the hand center to each fingertip detected. All of these features are used as a discriminative feature vector for a hand gesture at a given frame. By using a classification algorithm, this descriptor is evaluated and the class with highest confidence is assigned as the current frame's hand gesture.

We perform several experiments using SVM and Random Forest classifiers. For our dataset, Random Forest, as we expected, outperforms SVM for processing time and accuracy.

The experiments have shown that our approach yields a high-accuracy and noise-resistant system. Even though the features are sparse and Leap Motion does not provide point cloud data, our methodology was able to extract meaningful characteristics from the data. Furthermore, the high precision obtained for a number of different gesture and users indicates that our approach is reliable for a real application.

As future works we plan to extend our work to recognize dynamic hand gestures as well. Two kinds of dynamic gestures will be considered: the combination and transition of static gestures, and patterns of movements of the hands combined with static gestures.

## VI. Acknowledgment

The authors would like to thank CAPES, CNPq and FAPEMIG for the financial support provided.

## References

[1] J. Molina, M. Escudero-Vinolo, and A. Signoriello, "Real-time user independent hand gesture recognition from time-of-flight camera video using static and dynamic models," *Machine Vision and Applications*, 2011.

[2] S. Oprisescu and E. Barth, "3D Hand Gesture Recognition using the Hough Transform," *Advances in Electrical and Computer Engineering*, 2013.

[3] X. Chai, G. Li, Y. Lin, Z. Xu, Y. Tang, X. Chen, and M. Zhou, "Sign Language Recognition and Translation with Kinect," *IEEE Intl. Conf. on Automatic Face and Gesture Recognition*, 2013.

[4] B. N. S. Estrela, G. Camara-Chavez, M. F. M. Campos, W. R. Schwartz, and E. R. Nascimento, "Sign Language Recognition using Partial Least Squares and RGB-D Information," *Workshop de Visão Computacional (WVC)*, 2013.

[5] J. Guna, G. Jakus, M. Pogačnik, S. Tomažič, and J. Sodnik, "An Analysis of the Precision and Reliability of the Leap Motion Sensor and Its Suitability for Static and Dynamic Tracking," *Sensors*, vol. 14, 2014.

[6] M. Betke, J. Gips, and P. Fleming, "The Camera Mouse: Visual Tracking of Body Features to Provide Computer Access for People With Severe Disabilities," *IEEE Transactions On Neural Systems and Rehabilitation Engineering*, 2002.

[7] A. W. Vieira, E. R. Nascimento, G. L. Oliveira, Z. Liu, and M. F. Campos, "On the improvement of human action recognition from depth map sequences using Space–Time Occupancy Patterns," *Pattern Recognition Letters*, vol. 36, pp. 221–227, 2014.

[8] P. Breuer, C. Eckes, , and S. Muuller, "Hand gesture recognition with a novel ir time-of-flight range camera – a pilot study," *Proceedings of the 3rd Intl. Conf. on Computer vision/computer graphics collaboration techniques*, 2007.

[9] X. Liu and K. Fujimura, "Hand gesture recognition using depth data," *IEEE Intl. Conf. on Automatic Face and Gesture Recognition*, 2004.

[10] F. Weichert, D. Bachmann, B. Rudak, and D. Fisseler, "Analysis of the Accuracy and Robustness of the Leap Motion Controller," *Sensors*, 2013.

[11] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[12] J. R. Pansare, S. H. Gawande, and M. Ingle, "Real-Time Static Hand Gesture Recognition for American Sign Language (ASL) in Complex Background," *Journal of Signal and Information Processing*, 2012.

[13] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The WEKA Data Mining Software: An Update," *SIGKDD Explorations*, vol. 11, 2009.

[14] D. G. R. Bradski and A. Kaehler, *Learning Opencv, 1st Edition*, 1st ed. O'Reilly Media, Inc., 2008.

# Improving texture classification with nonextensive statistical mechanics

Lucas Assirati
São Carlos Institute of Physics
University of São Paulo
São Paulo - Brazil
Email: assirati@usp.br

Núbia Rosa da Silva
Institute of Mathematics and Computer Science
University of São Paulo
São Paulo - Brazil
Email: nubiars@icmc.usp.br

Odemir Martinez Bruno
São Carlos Institute of Physics
University of São Paulo
São Paulo - Brazil
Email: bruno@ifsc.usp.br

***Abstract*—Texture features are important descriptors for images being useful for classifying a variety of materials, for example, plant species, rocks, soil among others. There are several methods in literature to perform feature extraction. However, if this task is made with higher quality, the information obtained will be richer and more relevant. This paper studies feature extraction by Difference of Gaussians (DoG) filters using Gaussian kernels and an alternative approach using $q$-Gaussian kernels. Results show the use of DoG filter with $q$-Gaussian kernels enable a better texture feature extraction with more information than traditional use of standard Gaussian kernels, which leads us to better rates classification in texture images.**

## I. Introduction

In image analysis, the aim is to obtain information from objects and scenes using mathematical models in a digital image. This allows identification of patterns in these images, useful in many applications and fields.

A very common approach in image analysis is the analysis of textures. Textures are images characterized by the presence of spatial or statistical patterns. These patterns are not necessarily periodic. Because these are complex images, textures present great challenges to the techniques of pattern recognition and image analysis [1].

To perform image analysis in textures, this study will use a mathematical approach: extract the edges of the textures through the convolution of a difference of Gaussians filter (DoG) with the original image. Then the standard deviation of the histogram of the gray levels of the filtered image will be used as an informational tool for subsequent classification.

However, if we have richer information at the end of the process, one is able to obtain higher accuracy of classifiers applied. In order to do this improvement, it is necessary a more informative feature extraction. The nonextensive statistical mechanics proposed by Tsallis Constantino [2], [3], gives us a new tool: the $q$-Gaussian function, a generalized version of the Gaussian distribution (or normal distribution). With this new tool, it is possible to create a new version of the DoG filter, now using $q$-Gaussian kernels.

The studies in this paper show that the DoG filters using $q$-Gaussian kernels can provide better results of the classifiers compared to classical Gaussian kernels.

## II. Edge detection and information in images

Edge detection is closely related to the amount of information we want to extract from an image. When we perform the edge detection, there is a reduction in the amount of information to be processed because the information considered redundant (and therefore less relevant) will be supressed.

There are several methods for edge detection, but in this study we limit ourselves to the difference of Gaussians (DoG) filter, which is a good alternative for one of the most used filters: the Laplacian of Gaussian (LoG).

To perform the edge detection using this filter, one needs to do the convolution between an image and the DoG filter. The result of this convolution is a new image with the edge detection of the original one.

The use of the DoG filter is justified since its morphology is very similar to the powerful filter Laplacian of Gaussian (LoG). Meanwhile the use of DoG filter demands less calculations and computational resources. While LoG involves derivatives, the DoG filter is built by subtracting two Gaussian kernels ($G(x,\sigma) = (1/\sqrt{2\pi\sigma^2})\exp(-x^2/2\sigma^2)$, $-\infty < x < \infty$ and $\sigma > 0$) with slightly different standard deviation $\sigma$.

The shape of the DoG function is very similar to the shape of the LoG function (Figure 1). Thus, in practice, the LoG function can be approximated with success by the DoG function.

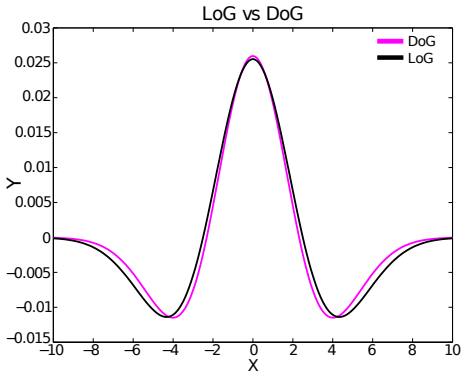



Figure 1. Comparison between LoG and DoG 1D functions.

## III. Gaussian kernels vs $q$-Gaussian kernels

Traditionally the DoG filter uses Gaussian kernels in its approach. However, if we have a more efficient image representation and richer in details of patterns, we will be able to perform better image analysis, especially in texture, since this kind of image present major challenges to these methods.

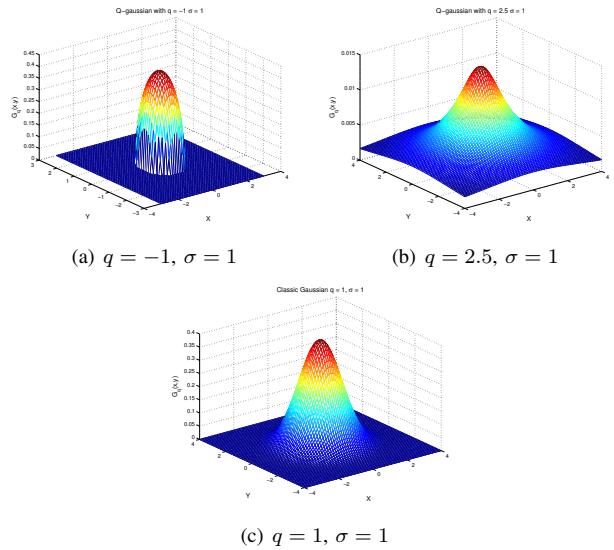
In this context, we refer to the generalized statistical mechanics proposed by Constantino Tsallis [2], [3]. Tsallis says different systems require different analysis tools, suitable to the peculiarities of the analyzed system. From this new formalism, Tsallis derives new information tools, including the $q$-Gaussian distribution. The $q$-Gaussian function differs from the standard one because, besides the adjustable parameter $\sigma$ (standard deviation) there is also the entropic index $q$. Variations in the $q$ index value promote changes in shape of the $q$-Gaussian distribution, adapting it to the problem that will be applied. The $q$-Gaussian function is defined as:

$$G_q(x) = \frac{1}{C_q\sqrt{2\sigma^2}} \exp_q \frac{-x^2}{2\sigma^2} \tag{1}$$

with $\exp_q(x) = [1 + (1-q)x]^{\frac{1}{1-q}}$ and

$$C_q = \begin{cases} \frac{2\sqrt{\pi}\Gamma(\frac{1}{1-q})}{(3-q)\sqrt{1-q}\Gamma(\frac{3-q}{2(1-q)})} & \text{if } -\infty < q < 1 \\ \sqrt{\pi} & \text{if } q = 1 \\ \frac{\sqrt{\pi}\Gamma(\frac{3-q}{2(q-1)})}{\sqrt{q-1}\Gamma(\frac{1}{q-1})} & \text{if } 1 < q < 3 \end{cases}$$

Figure 2 shows some elements of the family of 2D $q$-Gaussian. It is important to notice when $q = 1$ (Figure 2c), one can retrieves the standard gaussian distribution.



(a) $q = -1$, $\sigma = 1$ (b) $q = 2.5$, $\sigma = 1$

(c) $q = 1$, $\sigma = 1$

Figure 2. $q$-Gaussian 2D functions: shapes modified by different values of $q$.

So, from the innovations promoted by Tsallis' studies and the good results obtained in literature of performing edge detection [4] and describing option prices [5] using $q$-Gaussian instead of Gaussian, we have considered to use DoG filters with $q$-Gaussian kernels as an alternative tool to the traditional Gaussian kernels. Also, the flexibility to adjust the entropic index parameter $q$ allows extract intrinsic texture features by tuning the filter's shape better than Gaussian kernels.

## IV. Methodology

The methodology of this study consists in perform edge detection of texture images using the DoG filter. After the detection, from the filtered image, we build a histogram $H(x)$ of gray levels obtained by counting the number of pixels with a given intensity that can vary from 0 to 255. As the resulting image of a DoG filter has negative values, these values were re-scaled to be within this range. After this, the standard deviation of the histogram $S_H$ is calculated and used as informational tool. The standard deviation is defined by:

$$S_H = \sqrt{\frac{1}{N-1}\sum_{x=1}^{N}(H(x) - m)^2}, \tag{2}$$

$N$ is the number of bins of $H(x)$ and the mean value $m = \frac{1}{N}\sum_{x=1}^{N} H(x)$.

The DoG filter was used with Gaussian and $q$-Gaussian kernels for the purpose of comparison between them. Studies were performed in four texture gray-level converted databases: Brodatz [6] , Outex [7] , Usptex [8] and Vistex [9]. These, after having extracted the relevant information, were classified by $k$-Nearest Neighbors ($k$-NN) and the Naive-Bayes classifiers.

By using Gaussian kernels, 25 values of $\sigma$ equally spaced, in a range of 0.05 to 2.5, were used. Consequently, each image of the database was filtered 25 times, which led to the 25 values of standard deviation $S_H$. These 25 standard deviation values $S_H$ were used to create a feature vector $\vec{S}$ that was used to characterize the image in the classification process.

By using the $q$-Gaussian kernels, once it has two adjustable parameters ($\sigma$ and $q$), we have used just 5 values of $\sigma$ (that also varied in the range of 0.005 to 2.5), but in this case, 5 $q$ values varying in the range of -10 to 2.9 were used too. Both $\sigma$ and $q$ are equally spaced. Thus, by combining this two adjustable parameters, we reach a total of 25 different filters, which generated a feature vector $\vec{S}$ of the same size of the previously setting. Since the $q$-Gaussian can assume several shapes according with the $q$-value, we set a threshold value $\alpha = 4\sigma$ that is used as a limit for the filter size. From now on the filter size is $[-\alpha, \alpha]$.

The following steps summarize the method:

1) Read the images of databases;
2) For each element of the database, apply the DoG filters with Gaussian or $q$-Gaussian kernels for edge detection, respecting the intervals imposed by the method;
3) In possession of the filtered images, create the histogram of gray levels $H(x)$ of each one;
4) Calculate the standard deviation of the histograms $S_H$ and create a feature vector $\vec{S}$;
5) Apply the $k$-NN and Naive Bayes classifiers in the feature vectors $\vec{S}$ obtained.

Figure 3 shows a schematic drawing of the methodology:

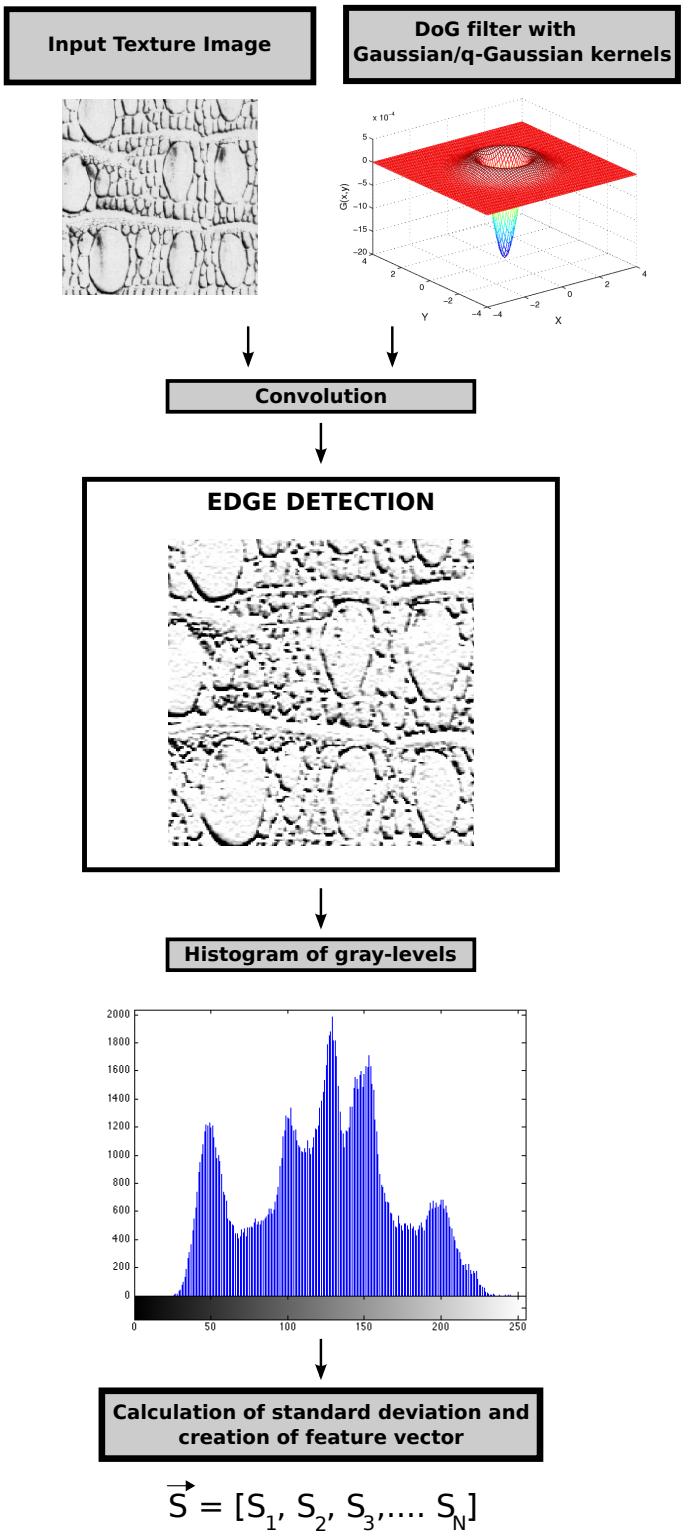



Figure 3. Schematic drawing of the methodology: Steps to perform the feature extraction.

## V. Results

Results of the methodology presented here were evaluated by $k$-NN (with $k$=1 and $k$=3 nearest neighbors) and Naive-Bayes classifiers. Table I shows the percentage of hit rate for each one of databases examined by the classifiers. The main purpose of these experiments is to promote a comparative analysis between the results obtained by using q-Gaussian and Gaussian. Thus, results obtained with state of the art approaches are not presented. Values into the brackets in front of the name of the database indicate the number of classes of the database.

| Database / Method \ Classifier | Naive-Bayes | $k$-NN ($k$=1) | $k$-NN ($k$=3) |
|---|---|---|---|
| Brodatz (111) [Gaussian kernels] | 57.65 | 65.31 | 61.17 |
| Brodatz (111) [$q$-Gaussian kernels] | 71.98 | 70.00 | 65.67 |
| Outex (68) [Gaussian kernels] | 58.38 | 74.19 | 72.86 |
| Outex (68) [$q$-Gaussian kernels] | 67.05 | 73.45 | 73.23 |
| Usptex (332) [Gaussian kernels] | 36.79 | 42.99 | 38.95 |
| Usptex (332) [$q$-Gaussian kernels] | 46.51 | 47.61 | 41.31 |
| Vistex (54) [Gaussian kernels] | 54.51 | 65.39 | 64.81 |
| Vistex (54) [$q$-Gaussian kernels] | 65.16 | 76.62 | 73.61 |

Table I. Results for 4 well known textures databases.

The best results were obtained with Naive-Bayes classifier. Figure 4 shows these results in a bar graph.

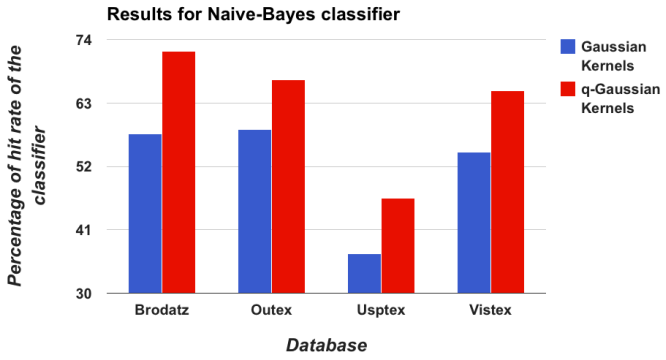



Figure 4. Results for the classifier that performs better: Naive-Bayes.

## VI. Conclusions

In most studied cases, the use of DoG filters with $q$-Gaussian kernels raises the hit rate of the classifiers, this because $q$-Gaussian kernels have a second adjustable parameter, the entropic index $q$, which allows a fine tuning of the filter's shape. Thus, certain $q$ indexes can reveals important details in the feature extraction process. Studies show through a more detailed filtered image, it is possible to extract richer information, enabling higher hit rates of the classifiers.

## ACKNOWLEDGMENTS

The authors gratefully thank the financial support of So Paulo Research Foundation (FAPESP), with grant Nos.: 2011/01523-1 and 2011/21467-9, National Council for Scientific and Technological Development (CNPq) with grant Nos.: 308449/2010-0 and 484312/2013-8 and Coordination for the Improvement of Higher Education Personnel (CAPES).

## REFERENCES

[1] J. B. Florindo, "Fractal descriptors applied to texture analysis." Ph.D. dissertation, São Carlos Institute of Physics, University of São Paulo, São Paulo - Brazil, 2013.

[2] C. Tsallis, "Possible generalization of Boltzmann-Gibbs Statistics," *Journal of Statistical Physics*, vol. 52, no. 1/2, pp. 479–487, 1988.

[3] C. Tsallis, F. Baldovin, R. Cerbino, and P. Pierobon, "Introduction to nonextensive statistical mechanics and thermodynamics," 2003.

[4] L. Assirati, N. R. Silva, L. Berton, A. A. Lopes, and O. M. Bruno, "Performing edge detection by difference of gaussians using q-gaussian kernels," *Journal of Physics: Conference Series*, vol. 490, no. 1, p. 012020, 2014. [Online]. Available: http://stacks.iop.org/1742-6596/490/i=1/a=012020

[5] L. Borland, "Option pricing formulas based on a non-gaussian stock price model," *Phys. Rev. Lett.*, vol. 89, p. 098701, Aug 2002. [Online]. Available: http://link.aps.org/doi/10.1103/PhysRevLett.89.098701

[6] P. Brodatz, *Textures; a photographic album for artists and designers*, D. P. N. York, Ed. Dover Publications New York,, 1966, available in: <http://www.ux.uis.no/ tranden/brodatz.html>.

[7] T. Ojala, T. Mäenpää, M. Pietikäinen, J. Viertola, J. Kyllönen, and S. Huovinen, "Outex - new framework for empirical evaluation of texture analysis algorithms," in *Proceedings of the 16 th International Conference on Pattern Recognition (ICPR'02) Volume 1 - Volume 1*, ser. ICPR '02. Washington, DC, USA: IEEE Computer Society, 2002, p. 10701, available in: <http://www.outex.oulu.fi>.

[8] A. R. Backes, D. Casanova, and O. M. Bruno, "Color texture analysis based on fractal descriptors," *Pattern Recognition*, vol. 45, no. 5, pp. 1984–1992, 2012, available in: <http://fractal.ifsc.usp.br/dataset/USPtex.php>.

[9] "Vision texture database. available in: <http://vismod.media.mit.edu/vismod/imagery/VisionTexture/vistex.html>. accessed in: May, 2014," 2009.

# Automatic Counting of Stomata in Epidermis Microscopic Images

Marcos William da Silva Oliveira*, Núbia Rosa da Silva*,
Dalcimar Casanova†, Luiz Felipe Souza Pinheiro‡, Rosana Marta Kolb‡ and Odemir Martinez Bruno†
*Institute of Mathematics and Computer Science
University of São Paulo (USP), São Carlos, São Paulo, Brazil
oliveimw@icmc.usp.br
†Scientific Computing Group, São Carlos Institute of Physics
University of São Paulo (USP), São Carlos, São Paulo, Brazil
www.scg.ifsc.usp.br
‡Department of Biological Sciences, Faculty of Sciences and Letters,
Univ Estadual Paulista Júlio de Mesquita Filho, UNESP, Assis, São Paulo, Brazil

***Abstract*—Stomata are small pores in epidermis of plants being their diversity an important feature for morphologists and physiologists. Through their diversity of characteristics (size, shape, density, responsiveness to desiccation, pore dimensions, orientation, etc.) the leaf epidermis may be studied. However, there is a great difficulty in counting stomata, especially when there are a large number of samples to be analysed, since this task is performed manually, depending on the expert experience. This paper proposes the automatic detection and counting of stomata through morphological operations in epidermis microscopic images. With this approach, an accuracy rate of 94.3% was achieved, detecting and counting the stomata in 24 images from 5 plant species, indicating that the results obtained are promising.**

## I. Introduction

Stomata are small openings located on the epidermis, especially in plant leaves, consisting in a channel for the exchange of gases and transpiration. Stomatal characteristics explain much of the intraspecific variation in the regulation of transpiration under water deficiency in plants [1]. The number of stomata per leaf varies according to environmental factors such as temperature and light, present in the places where the plants are growing [2], [3]. The proportion of stomata per area in high-light leaves can reach more than twice the number of stomata in low-light leaves. The stomatal density is an important feature because it can affect the stomatal diffusive resistance being the increase in stomatal density a primary factor for the decreasing of stomatal diffusive resistance.

Analysis of stomatal density is a very important task for plant analysis. However, tools available for identification of stomata are limited [4]. For that reason, the aim of this study is to develop a tool for automatic detection and counting of stomata from microscopic images of plant epidermis. To achieve this goal, the proposed method uses a sequence of processes in the image which includes Gaussian filter and morphological image processing operations. The results showed an average recognition rate of 94.3% of the stomata of 24 images from 5 plant species. This high rate of identification using a computational tool shows the potential of the proposed methodology for assistance in other areas, as studies of ecophysiology and ecological anatomy of plants.

This paper is organized as follows: a brief description of mathematical morphology is presented in the Section II, the proposed methodology is described in Section III, results and discussions are shown in Section IV and conclusions in Section V.

## II. Mathematical morphology

In order to detect the stomata in images (see Figure 1a), we have used the concept of mathematical morphology. The mathematical morphology is a theoretical model and technique for the analysis and processing of geometrical structures, based on set theory, lattice theory, topology, and random functions [5]. The basis of image processing by mathematical morphology is the use of some operators that are strongly related to Minkowski addition.

The mathematical morphology was originally developed for binary images, and was later extended to grayscale images. In both domains, the basic morphological operators are erosion, dilation, opening and closing [6]. In grayscale morphology the dilation of $f$ by $b$, denoted $f \oplus b$, is given by

$$(f \oplus b)(x) = \sup_{y \in E}[f(y) + b(x - y)], \tag{1}$$

where the image is given by $f(x)$, the structuring function by $b(x)$ and $E$ is the domain of $f$. Similarly, the erosion of $f$ by $b$, denoted $f \ominus b$, is given by

$$(f \ominus b)(x) = \inf_{y \in E}[f(y) - b(y - x)]. \tag{2}$$

Just like in binary morphology, the opening and closing are given respectively by

$$f \circ b = (f \ominus b) \oplus b \text{ and } f \bullet b = (f \oplus b) \ominus b. \tag{3}$$

### A. Opening-by-reconstruction

In a morphological opening, erosion typically removes small objects, and the subsequent dilation attempts to restore the shape of objects that remain. However, the accuracy of this restoration is highly dependent on the similarity of the shapes of the objects and the structuring element used. Opening-by-reconstruction restores exactly the shapes of the objects that remain after erosion [5].

The morphological reconstruction is a morphological transform involving two images and a structuring element [7]. One image, the marker, contains the starting points for the transformation. The other image, the mask, constrains the transformation. The structuring element is used to define connectivity. The process can be thought as repeated dilations of an image, called the marker image, until the contour of the marker image fits under a second image, called the mask image. In morphological reconstruction, the peaks in the marker image spread out, or dilate.

The opening-by-reconstruction of the image $f$ by an structuring element $b$, consist in an opening of $f$ by $b$, then the reconstruction from the opened image. This operation is denoted by [5]:

$$r_f(f \oplus b) \tag{4}$$

### B. Closing-by-reconstruction

The closing operation typically try filling in the small gaps in image while smoothing their outer edges. The closing-by-reconstruction however, try to preserve the original edges of the shapes [7].

This operation, dual of the opening-by-reconstruction, consists in a closing of $f$ by the structuring element $b$ followed by a morphological reconstruction. Can be denoted by [5]:

$$r_f(f \ominus b) \tag{5}$$

## III. Detection of stomata by morphological operations

At a first look at the original epidermis microscopic image (Figure 1a), we can observe that all stomata have a circular shape with two visible regions (center and border). These characteristics can be exploited when trying to identify the stomata between all other leaf structures. To use this idea, we are computing the foreground and background markers. These markers are connected blobs of pixels within each of the objects or the background. In order to do that we can use the above cited operators to find the stomata in leaf epidermis images. The following steps summarize the complete procedure.

First, we apply a Gaussian (low-pass) filter on the image in order to eliminate some noise. We employ a filter with size $20 \times 20$ pixels and parameter $\sigma = 0.5$. The results of this process can be viewed in Figure 1b.

After that, an opening-by-reconstruction morphological operation is applied over the image. As showed before, the opening is an erosion followed by a dilation, however the opening-by-reconstruction is an erosion followed by a morphological reconstruction. This process tends to preserve the original shapes of the objects that remain after erosion. The objective here is flatten the objects of interest, removing gray-scale variation inside them. The result of this process can be seen in Figure 1c. We can observe the stomata (and also the background) are now flattened with respect to their gray-scale value.

The next step is to perform a closing-by-reconstruction in order to fill the remain gaps in the flattening regions (a kind of noise removal). This morphological technique will clean up the result image. These operations will create a flat minima inside each object, the result of this step is showed in Figure 1d.

With the noiseless image, the next step is to determine what are the stomata and what is background. To accomplish that we compute the regional minima of $I$. Regional minima defined as connected components of pixels with a constant intensity value, and whose external boundary pixels all have a higher value. As we can see in the Figure 1d, after the morphological process, the stomata region present low values if compared with their neighbourhood. The result of this procedure can be seen in Figure 1e.

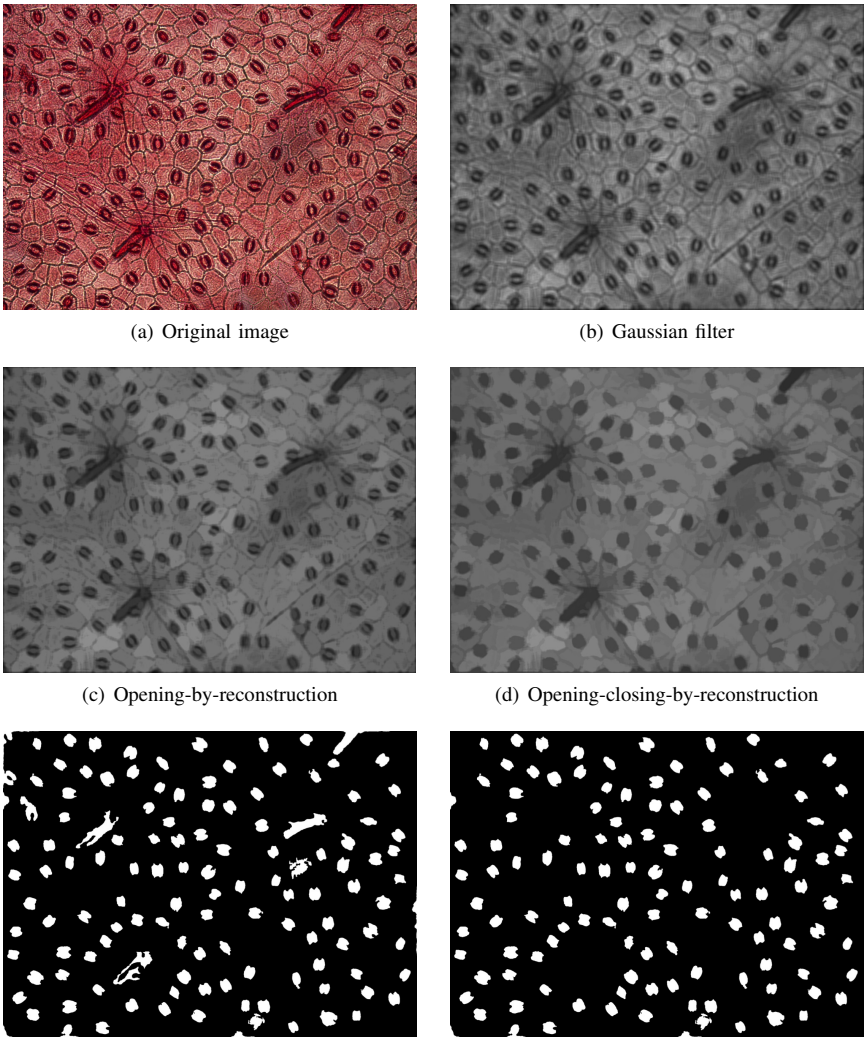
On the result image we can see that, both, stomata and some other leaf structures were identified. To eliminate the spurious identifications we have used a simple rule where the objects that have a perimeter that is twice larger than the mean perimeter of all objects of the image are eliminated of the selection. The final result is the image with all stomata identified in Figure 1f.

## IV. Results and Discussions

The images to perform the tests were captured at $20\times$ objective lenses, using an optical microscope *Zeiss Axio Scope A1* model, coupled to a digital camera *Zeiss AxioCam MRc* model. The abaxial surfaces of the epidermis were obtained from one leaf per individual of each species. Of each leaf collected, a sample of about $1cm^2$ was removed between the margin and the midrib of the leaf, in its middle portion, totalling 24 samples of 5 species. The dissociation of the leaf epidermis was performed with a 1:1 solution of glacial acetic acid and hydrogen peroxide at 60°C for 12 hours, or the time required for complete decoupling of epidermis (modified from [8]). After this process, the epidermis were washed with distilled water, placed on slides and stained with safranin. Glycerin was used as mounting medium.

Figure 2 shows an example of each species image side by side to the result of automatic identification of stomata by our proposed segmentation, described in Section III. In Figure 2, we see one original image of each species, on the left side, and the results of stomata identification with green squares that overlap the segmented objects, on right side. Some challenges of this segmentation can be seen in this figure. We observe strong contours of the cell walls, spots or lesions occurring in the skin as well as other structures that overlap the leaf epidermis. Despite this, the identification of stomata is quite reasonable, as indicated by small green squares on the result images.

Analysis of results for all samples in this experiment is organized in Table I. In this table we compare the traditional



(a) Original image (b) Gaussian filter

(c) Opening-by-reconstruction (d) Opening-closing-by-reconstruction

(e) Regional minima (f) Perimeter analysis

Fig. 1. Detection of stomata by morphological operations. (a) Original image. (b) Leaf epidermis after low-pass filter. (c) Opening-by-reconstruction morphological operation. (d) Opening-closing-by-reconstruction. (e) Regional minima of result image. (f) Final stomata identification after perimeter elimination rule.

method of manual counting of stomata with the proposed automatic counting. The columns of this table show the names of the specimens tested in *Test Image*, followed by the absolute amounts of stomata counted manually (*Manual Counting*) and the quantity obtained automatically (*Automatic Proposal*). The column *Not Identified* contains the quantities not identified in the automatic counting. *True Positive* presents the numbers of objects in the image identified as stomata that really are stomata, while *False Positive* indicates the number of objects erroneously identified as stomata. Evaluation measures *Recall* and *Precision* are calculated by

$$Recall = \frac{PT}{MC}, \quad Precision = \frac{PT}{PT + PF}, \tag{6}$$

where $MC$ is the *Manual Counting* quantity, and $PT$ and $PF$ is the positive true and false values, respectively. The absolute numbers of manual counting are considered as the actual amounts of stomata in the image to validate the proposal.

Thus, it is important to note in Table I the high precision rate of the proposal. The average precision about 94% and standard deviation of 0.044 show that the results are promising. Moreover, the absolute quantity of false positives is low, with a maximun of 7 and average proportion of stomata 5.5% compared to the manual count. As well, the proportion of unidentified stomata has an average of 11.8% in relation to the exact amount of stomata.

Two species analysed showing interesting results are *Callophyllum brasiliense* and *Xylopia sericea*. We obtained zero false positives for this first specie by applying our approach. As well, the accuracy rate is good for samples of *Xylopia sericea* even with the large number of stomata and the natural difficulties of the acquired images.

## V. Conclusion

Although the problem of identification of stomata is complex and very important for the analysis of plants, there is still

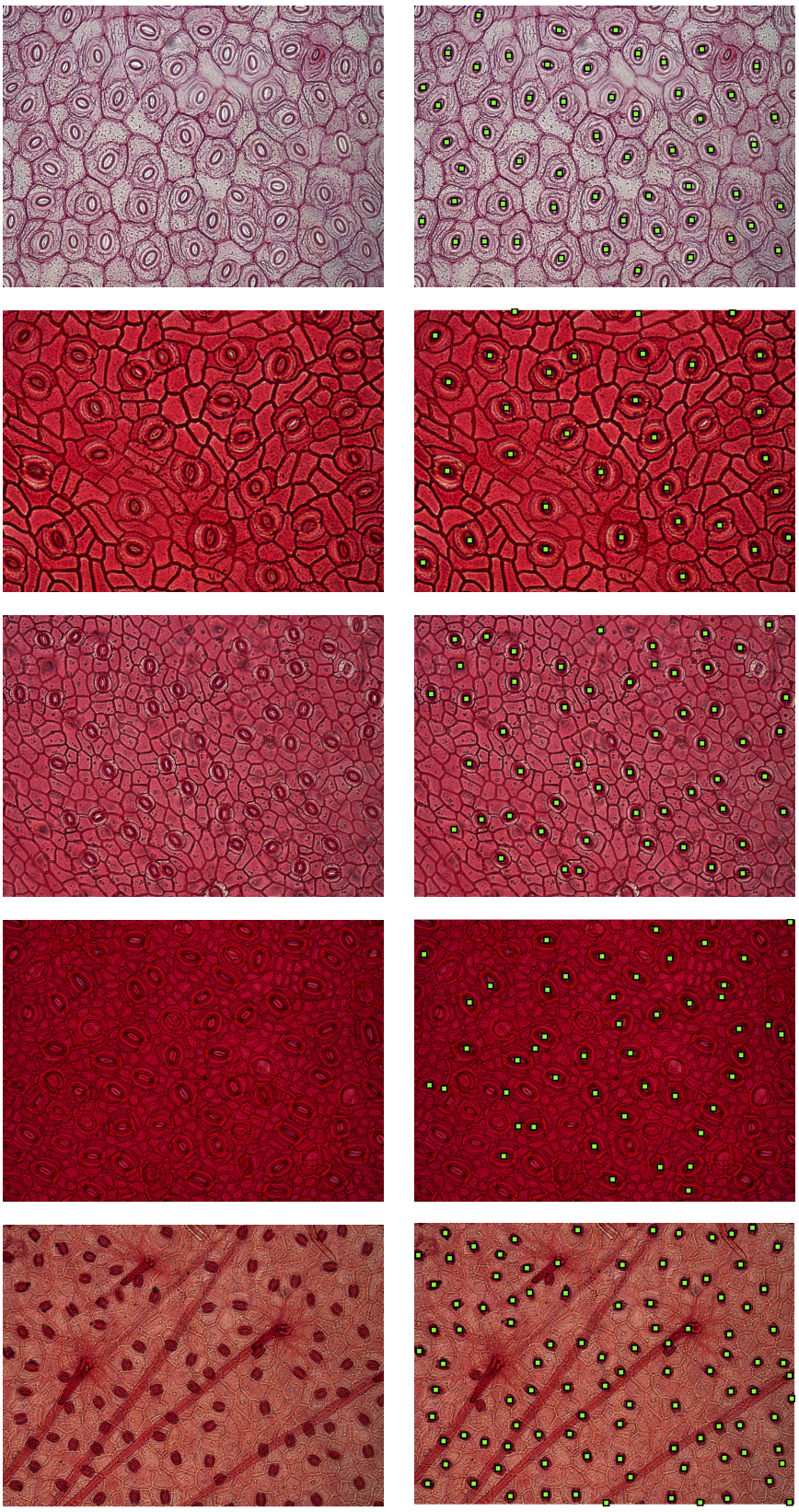


Fig. 2. Examples of stomata segmentation. On the left side, one original image of each species. On right side, the result of stomata segmentation using small green squares overlapping each stomata identified. The 5 different plant species tested are *Callophyllum brasiliense*, *Guapira areolata*, *Maytenus floribunda*, *Roupala montana*, and *Xylopia sericea*, displayed in this order.

TABLE I. COMPARISON BETWEEN OUR AUTOMATIC PROPOSAL AND THE TRADITIONAL METHOD OF COUNTING OF STOMATA APPLIED ON 24 MICROSCOPY IMAGES OF 5 DIFFERENT PLANT SPECIES. THE MEAN OF PRECISION (94.30%) SHOWS THAT THIS APPROACH IS SUITABLE TO STOMATA IDENTIFICATION.

| Test image | Manual Counting | Automatic Proposal | Not Identified | Positive True | Positive False | Recall (%) | Precision (%) |
|---|---|---|---|---|---|---|---|
| *Callophyllum brasiliense 1* | 58 | 56 | 2 | 56 | 0 | 96.55 | 100.0 |
| *Callophyllum brasiliense 2* | 60 | 56 | 4 | 56 | 0 | 93.33 | 100.0 |
| *Callophyllum brasiliense 3* | 62 | 57 | 5 | 57 | 0 | 91.94 | 100.0 |
| *Guapira areolata 1* | 34 | 33 | 4 | 30 | 3 | 88.24 | 90.91 |
| *Guapira areolata 2* | 36 | 39 | 4 | 32 | 7 | 88.89 | 82.05 |
| *Guapira areolata 3* | 31 | 33 | 1 | 30 | 3 | 96.77 | 90.91 |
| *Maytenus floribunda 1* | 53 | 54 | 3 | 50 | 4 | 94.34 | 92.59 |
| *Maytenus floribunda 2* | 47 | 48 | 4 | 43 | 5 | 91.49 | 89.58 |
| *Maytenus floribunda 3* | 44 | 46 | 1 | 43 | 3 | 97.73 | 93.48 |
| *Roupala montana 1* | 53 | 49 | 11 | 42 | 7 | 79.25 | 85.71 |
| *Roupala montana 2* | 73 | 50 | 24 | 49 | 1 | 67.12 | 98.00 |
| *Roupala montana 3* | 85 | 56 | 31 | 54 | 2 | 63.53 | 96.43 |
| *Xylopia sericea 1* | 82 | 78 | 10 | 72 | 6 | 87.80 | 92.31 |
| *Xylopia sericea 2* | 84 | 75 | 11 | 73 | 2 | 86.90 | 97.33 |
| *Xylopia sericea 3* | 85 | 79 | 7 | 78 | 1 | 91.76 | 98.73 |
| *Xylopia sericea 4* | 124 | 119 | 6 | 118 | 1 | 95.16 | 99.16 |
| *Xylopia sericea 5* | 123 | 120 | 9 | 114 | 6 | 92.68 | 95.00 |
| *Xylopia sericea 6* | 113 | 108 | 8 | 105 | 3 | 92.92 | 97.22 |
| *Xylopia sericea 7* | 123 | 98 | 31 | 92 | 6 | 74.80 | 93.88 |
| *Xylopia sericea 8* | 109 | 100 | 13 | 96 | 4 | 88.07 | 96.00 |
| *Xylopia sericea 9* | 124 | 89 | 41 | 83 | 6 | 66.94 | 93.26 |
| *Xylopia sericea 10* | 95 | 97 | 3 | 92 | 5 | 96.84 | 94.85 |
| *Xylopia sericea 11* | 92 | 96 | 3 | 89 | 7 | 96.74 | 92.71 |
| *Xylopia sericea 12* | 98 | 102 | 3 | 95 | 7 | 96.94 | 93.14 |

deficiency in material aiding this issue. However, the proposed approach showed that with simple computational tools is possible to obtain good results. This fact was demonstrated by the high rate of identification of stomata in epidermis microscopic images. These promising results leave open a range of possibilities for future works, among which the use of other computational tools for segmentation of stomata and their application in a larger number of images.

## ACKNOWLEDGMENT

The authors gratefully thank the financial support of São Paulo Research Foundation (FAPESP), with grant Nos.: 2011/01523-1, 2011/23112-3, 2011/21467-9 and 2013/14984-2, National Council for Scientific and Technological Development (CNPq) with grant Nos.: 308449/2010-0 and 484312/2013-8, PROPE/UNESP (14/2012/Renove) and Coordination for the Improvement of Higher Education Personnel (CAPES).

## REFERENCES

[1] H. Giday, K. H. Kjaer, D. Fanourakis, and C.-O. Ottosen, "Smaller stomata require less severe leaf drying to close: A case study in rosa hydrida," *Journal of Plant Physiology*, vol. 170, no. 15, pp. 1309 – 1316, 2013.

[2] R. Pandey, P. M. Chacko, M. Choudhary, K. Prasad, and M. Pal, "Higher than optimum temperature under {CO2} enrichment influences stomata anatomical characters in rose (rosa hybrida)," *Scientia Horticulturae*, vol. 113, no. 1, pp. 74 – 81, 2007.

[3] A. Wild and G. Wolf, "The effect of different light intensities on the frequency and size of stomata, the size of cells, the number, size and chlorophyll content of chloroplasts in the mesophyll and the guard cells during the ontogeny of primary leaves of sinapis alba," *Zeitschrift für Pflanzenphysiologie*, vol. 97, no. 4, pp. 325 – 342, 1980.

[4] C. A. Sweeney, "A key for the identification of stomata of the native conifers of scandinavia," *Review of Palaeobotany and Palynology*, vol. 128, no. 3–4, pp. 281 – 290, 2004.

[5] R. C. Gonzales and R. Woods, *Digital Image Processing*. Addison Wesley, 1992.

[6] I. Svalbe and R. Jones, "The design of morphological filters using multiple structuring elements, part II: Open (close) and close (open)," *Pattern Recognition Letters*, vol. 13, pp. 175–181, 1992.

[7] L. Vincent, "Morphological gray scale reconstruction in image analysis: Applications and efficient algorithms," *IEEE Transactions on Image Processing*, vol. 2, no. 2, pp. 176–201, 1993.

[8] G. L. Franklin, "Preparation of thin sections of synthetic resins and wood-resin composites, and a new macerating method for wood [10]," *Nature*, vol. 155, no. 3924, p. 51, 1945, cited By (since 1996):146. [Online]. Available: www.scopus.com

# Video-Based Rendering Architecture for Remote Education via Web

Maikon Cismoski dos Santos[†], Helio Pedrini[†]
[†]Instituto de Computação, Universidade Estadual de Campinas
Campinas, SP, Brasil, 13083-852

***Resumo*—In Web-based distance education, the dynamic real-world scene is traditionally captured by one or more cameras and the observer's viewpoint is usually fixed, the same as used by the video cameras, such that viewing consists of a passive playback of content. This work presents an archicteture for video-based rendering to support distance education via Web, where an observer can determine the viewpoint arbitrarily while viewing the scene. The views of the virtual camera are rendered by the proposed VBR method, which allows the rendering process is implemented to run on graphics processing units (GPUs). The acquisition of the scene is performed by means of a set of few cameras and using conventional low-cost equipments, such as video cameras and computers available in the market.**

## I. Introdução

Com o avanço da tecnologia, os canais de comunicação têm suportado diferentes tipos de mídias e o processo de aquisição de imagens tem se tornado cada vez mais poderoso, facilitando a transmissão, produção e visualização de vídeos. Neste contexto, os sistemas de ensino a distância via Web têm se tornando uma opção para educação e treinamento, permitindo a comunicação visual e remota entre os participantes, possibilitando a redução dos custos da educação em sua forma presencial, favorecendo a disseminação de informações para um grande número de usuários e evitando o deslocamento de pessoas residentes em regiões distantes. Tradicionalmente, em um sistema de ensino a distância, a cena dinâmica do mundo real é capturada por uma ou mais câmeras e os vídeos produzidos são transmitidos para os dispositivos de exibição, em que o ponto de vista do observador é fixo, o mesmo utilizado pelas câmeras de captura, tal que a exibição dos vídeos consiste na reprodução passiva dos quadros.

Na área de Computação Gráfica, o processo de renderização produz imagens relativas a uma cena sintética e a um ponto de vista arbitrário. Em contraste, na área de Visão Computacional, o processo é o inverso, em que um modelo tridimensional (3D) é extraído de uma cena do mundo real. A combinação dessas duas áreas de conhecimento proporcionou a criação de um novo campo de pesquisa denominado Renderização Baseada em Vídeo [1], conhecido pela sigla VBR (do inglês, *Video-Based Rendering*), o qual tem o propósito de estender a experiência que é oferecida por um vídeo tradicional, permitindo que o observador possa definir um ponto de vista arbitrário durante a visualização, não ficando restrito à reprodução passiva do conteúdo.

Um sistema de VBR utiliza, como entrada, múltiplos vídeos capturados de uma mesma cena do mundo real e de diferentes pontos de vista. A renderização de novas vistas da câmera virtual, a qual tem o ponto de vista definido arbitrariamente pelo observador, é realizada a partir das imagens dos vídeos de entrada, ao invés de utilizar as tradicionais primitivas geométricas, sintetizando vistas com qualidade fotorrealista.

Os métodos de VBR podem ser classificados em *on-line* e *off-line*. Usualmente, os sistemas *off-line* são focados na qualidade das vistas renderizadas, podendo utilizar uma grande quantidade de câmeras, dispositivos de alta definição e algoritmos robustos, adotando uma etapa de pré-processamento das imagens, em que as informações 3D são extraídas e armazenadas para que, posteriormente, apenas a renderização e a visualização das vistas da câmera virtual sejam realizadas em tempo real. Em oposição, métodos de VBR *on-line* devem ser capazes de executar as etapas de extração de informações dos vídeos de entrada, síntese e visualização das vistas, todas elas, várias vezes por segundo.

Este trabalho apresenta um método *on-line* de Renderização Baseada em Vídeo (VBR) para o ensino a distância, permitindo que o observador possa determinar o ponto de vista desejado durante a visualização da cena. Todo o processo, desde a captura da cena até a renderização das vistas da câmera virtual, é realizado *on-line*, utilizando equipamentos convencionais, de baixo custo, tais como câmeras e computadores encontrados no mercado. A captura da cena é realizada por meio de um conjunto formado por poucas câmeras, posicionadas em torno da região 3D de interesse no mundo real e o observador pode definir o ponto de vista da câmera virtual de uma forma intuitiva através de uma interpolação proposta.

## II. Trabalhos Relacionados

Esta seção apresenta os principais trabalhos relativos à VBR, destacando as vantagens e desvantagens de cada abordagem. Primeiramente, os métodos de VBR *off-line* são descritos na subseção II-A. Posteriormente, as abordagens de VBR *on-line* são apresentadas na subseção II-B.

### A. Métodos de VBR Off-Line

Um dos primeiros trabalhos de VBR propostos foi apresentado por Kanade et al. [2]. A estrutura 3D da cena é extraída com a estimação de *mapas de profundidade*[1] para cada vista capturada. A estimação é realizada com base na técnica descrita em [4], em que a correspondência de pontos é determinada com relação a múltiplos pares de imagens. Com a informação 3D da cena extraída, a cena é reconstruída e

[1]O *mapa de profundidade*, também conhecido como *imagem de profundidade*, armazena os valores de profundidade estimados para cada ponto de uma imagem 2D, representando a estrutura 3D da cena [3].

a síntese de novas vistas da câmera virtual é realizada com a renderização da malha triangular obtida na reconstrução, levando em consideração o ponto de vista desejado e aplicando as imagens das câmeras de entrada como texturas sobre os polígonos. A abordagem descrita por Kanade et al. [2], embora possibilite a captura e a renderização dinâmicas da cena e do mundo real, possui diversas desvantagens, as quais são citadas a seguir. Problemas na estimação do mapa de profundidade, decorrentes de descontinuidades de profundidade presentes nas bordas dos objetos e regiões sem textura, onde a cor é constante, são tratados manualmente. A computação não é realizada *on-line*. Um grande número de câmeras é empregado para a captura da cena, sendo 51 câmeras arranjadas em uma estrutura com forma de uma abóbada. A qualidade das vistas renderizadas é pobre, contendo buracos e artefatos.

Zitnick et al. [5] apresentaram um sistema de VBR baseado na reconstrução da cena 3D a partir de mapas de profundidade. Uma etapa de pré-processamento de dados é adotada, em que os mapas de profundidade são estimados e os dados (vídeos e informações de profundidade) são comprimidos e armazenados. Posteriormente, em tempo real, as vistas da câmera virtual são construídas a partir da renderização dos polígonos. As imagens das câmeras são usadas como texturas para o modelo reconstruído. A estimação dos mapas de profundidade é baseada na segmentação de cores das imagens de entrada. As principais vantagens dessa abordagem são o uso de poucas câmeras (8 câmeras são empregadas) e as vistas renderizadas são de alta qualidade e de alta resolução ($1024 \times 768$ pixels). A qualidade das vistas renderizadas está associada com a robustez do método de estimação de mapas de profundidade usado, que trata oclusões de forma explícita e descontinuidades. Em contraste, uma desvantagem a ser destacada é o tempo de computação do método proposto, em que somente a parte de renderização é executada *on-line*.

Li et al. [6] descreveram um sistema de VBR empregando câmeras não-calibradas. O algoritmo de renderização consiste em projetar pontos correspondentes entre duas imagens de referência para a câmera virtual. Inicialmente, a correspondência de pontos entre as imagens de referência é computada pela estimação da informação de disparidade, empregando um algoritmo baseado na segmentação de cores. A posição da câmera virtual é definida entre duas câmeras reais de referência e as vistas da câmera virtual são computadas pela transferência de pontos utilizando *matrizes fundamentais* [7]. A principal contribuição dessa abordagem é o método de extração dos parâmetros das câmeras, permitindo o posicionamento da câmera virtual e a renderização dos quadros utilizando câmeras não-calibradas. Entretanto, os parâmetros intrínsecos são obtidos por uma aproximação baseada nas dimensões das imagens, podendo gerar erros de projeção, afetando a qualidade das vistas renderizadas.

Oh et al. [8] e Mori et al. [9] propuseram uma abordagem para VBR usando mapas de profundidade para representar a informação 3D da cena e a técnica 3D *warping* [10] na renderização das vistas da câmera virtual. A técnica 3D *warping* consiste na projeção (*warping*) de uma imagem de referência para uma imagem destino, relativo ao ponto de vista desejado. Essa técnica requer, como dados de entrada, a imagem de referência, o mapa de profundidade associado com a imagem de referência e as matrizes de projeção de ambas as câmeras. O problema da renderização baseada na técnica 3D *warping* são os buracos e a desoclusão, denominados artefatos, que podem surgir na imagem destino. Buracos aparecem devido à diferença entre a resolução de amostragem entre a imagem de referência e a destino. Desoclusão corresponde a uma superfície que não é visível pela câmera de referência, mas faz parte do campo de visão da câmera virtual. A principal diferença entre os métodos apresentados por Oh et al. [8] e Mori et al. [9] é que, em [8], os artefatos são coloridos com as cores dos pixels que fazem fronteira com o artefato e, em [9], uma interpolação com as cores dos pixels vizinhos é empregada para minimizar a presença de artefatos.

### *B. Métodos de VBR On-Line*

Wang et al. [11] desenvolveram um sistema VBR *on-line*, estimando os mapas de profundidade e renderizando as vistas da câmera virtual sem usar uma etapa de pré-processamento de dados. A renderização dos quadros e a estimação da informação de profundidade foram implementadas com o modelo de programação e ambiente de desenvolvimento denominado *Compute Unified Device Architecture* (CUDA) [12], o qual permite que programas paralelos sejam escritos, usando uma extensão da linguagem C e executados em GPUs (*graphics processing units*) da NVIDIA [13]. Entretanto, o método funciona apenas em placas gráficas da NVIDIA.

Yang et al. [14] apresentaram um sistema de VBR em tempo real baseado na técnica *light field* [15] e *lumigraph* [16]. A captura da cena é realizada por $8 \times 8$ câmeras dispostas em um painel. A arquitetura do sistema é formada por 6 computadores responsáveis pelo gerenciamento das câmeras e o processamento distribuído da renderização das vistas, além de mais 1 computador empregado na composição das imagens finais. As técnicas *light field* e *lumigraph* permitem a síntese de novas vistas utilizando como dados de entrada somente múltiplas imagens, sem a necessidade da extração de informações geométricas da cena. O processo de renderização é baseado na parametrização dos raios que intersectam dois planos, o das câmeras $uv$ e o focal $st$. Embora a técnica de renderização apresentada seja em tempo real, o processamento das vistas é centralizado e pode comprometer o desempenho com a adição de múltiplos observadores. Além disso, um grande número de câmeras e computadores é empregado para a síntese das vistas, tornando a implementação de alto custo.

Liu et al. [17] estenderam o trabalho apresentado por Yang et al. [14], agregando a compressão dos múltiplos vídeos capturados, transmissão via Internet e suporte a múltiplos observadores. Ao invés de um computador central renderizar as vistas, computadores clientes realizam essa tarefa a partir de vídeos comprimidos recebidos pela rede. O sistema de VBR é em tempo real. As desvantagens dessa abordagem são o uso de diversos computadores empregados na captura e na transmissão dos dados e a grande quantidade de câmeras.

Em geral, as abordagens de VBR baseadas em *light field* e *lumigraph*, descritas na literatura que são em tempo real, empregam uma grande quantidade de câmeras e múltiplos computadores para o processamento distribuído das vistas. Outros métodos adotam algum tipo de informação geométrica [18], tal como a estimação dos mapas de profundidade das imagens capturadas, reduzindo o número de câmeras, mas

elevando o tempo de computação. Wei et al. [19] e Jeon e Park [20] definem múltiplos planos focais (ao invés de apenas um, como proposto originalmente) na renderização das imagens, produzindo vistas com qualidade e reduzindo o número de câmeras. Entretanto, nessas abordagens, a computação não é realizada *on-line* devido ao maior número de planos focais que devem ser processados.

Outras abordagens de VBR encontradas na literatura são baseadas na varredura de planos. Os métodos de VBR que empregam essa técnica podem ser implementados para executar em GPU, possibilitando que seu tempo de computação seja em tempo real. A técnica de varredura de planos foi inicialmente proposta por Collins [21] e posteriormente estendida por Yang et al. [22] para executar em GPU e *on-line*. O método apresentado por Yang et al. [22] utiliza 5 câmeras previamente calibradas e em tempo real, o método pode ser utilizado para a síntese de novas vistas, estimação de mapas de profundidade e reconstrução da geometria de objetos. Inicialmente, o espaço 3D é discretizado em planos paralelos à câmera virtual, de forma que os objetos da cena estejam localizados entre dois planos, o plano mais à frente e o mais afastado em relação às câmeras (*near* e *far*, respectivamente). A síntese de novas vistas é realizada pela projeção das imagens das câmeras reais para os planos no espaço 3D. As cores dos pontos projetados são comparadas e uma medida de similaridade entre as cores é computada. Por fim, os pontos projetados nos planos são reprojetados para a câmera virtual com base em uma votação, a qual é definida pela medida de similaridade de cores computada anteriormente. As principais desvantagens dessa abordagem são que a qualidade da vista sintetizada pode depender do número de planos, sendo que, em cenas complexas, um número maior de planos é normalmente requerido para discretizar o espaço 3D e a computação da média de similaridade de cor proposta por [22] restringe o posicionamento da câmera virtual, que deve estar próxima de uma câmera de referência e entre duas câmeras reais; caso contrário, a câmera virtual não pode ser representada. Além disso, artefatos podem surgir no vídeo renderizado quando há mudança na câmera de referência.

Nozick e Saito [23] estenderam o processo de renderização baseado na varredura de planos, de uma vista para múltiplas vistas. A partir de uma votação, a medida de similaridade das cores é computada e os pontos sobre os planos durante a varredura são projetados para múltiplas vistas, ao invés de uma única, como no método tradicional. Essa extensão foi proposta para obter pares de imagens capturadas da mesma cena e de diferentes pontos de vistas, utilizados por dispositivos *auto-estereoscópicos 3D*[2]. O método é executado em tempo real e utiliza quatro *webcams*.

## III. METODOLOGIA

Nesta seção, a metodologia do método de Renderização Baseada em Vídeo para o ensino a distância via Web é descrita, a qual tem apoio da revisão bibliográfica apresentada na Seção II. Entre as abordagens descritas na Seção II, os métodos de VBR que empregam mapas de profundidade tradicionalmente adotam uma etapa de pré-processamento dos dados, usando algoritmos robustos para a extração da informação 3D. As abordagens baseadas em *light field* e *lumigraph* permitem a criação de um sistema para VBR *on-line*, entretanto, requerem uma grande quantidade de câmeras e computadores para o processamento das informações, tornando o projeto custoso. Com base nessas premissas, a técnica de VBR escolhida para o desenvolvimento do protótipo foi a varredura de planos, pois permite que o processo de renderização seja implementado para executar em GPU.

### A. Visão Geral do Sistema

A visão geral do sistema de ensino a distância via Web proposto é ilustrada na Figura 1. A cena do mundo real é capturada por um conjunto de $n$ câmeras reais, sendo que uma câmera real $i$ é denotada por $cam_i$. O sistema é composto por um servidor de vídeo, o qual gerencia a captura da cena e é responsável pela codificação, compressão e transmissão dos vídeos para os dispositivos de visualização (observadores). Os observadores, ao receberem os vídeos capturados, definem a posição da câmera virtual no ambiente e renderizam os quadros da câmera virtual a partir das imagens das câmeras reais recebidas empregando a técnica de varredura de planos. Pode-se notar que a posição da câmera virtual, ilustrada na Figura 1, representa um ponto de vista diferente das câmeras de captura, estendendo a visualização da cena para pontos de vista arbitrários.

### B. Posicionamento da Câmera Virtual

O posicionamento de uma câmera consiste em definir um ponto de vista no qual pontos no espaço 3D são projetados para o plano da câmera. Assumindo que as câmeras são calibradas, o posicionamento é realizado utilizando-se os parâmetros intrínsecos e extrínsecos da câmera estimados e a projeção de pontos 3D sobre o plano da câmera é dada por uma matriz de projeção. O método de calibração usado neste trabalho é o proposto por Zhang [25].

Dado o modelo de câmera estenopeica [7], também conhecido como modelo *pinhole*, uma matriz de projeção $P_i$, de dimensão $3 \times 4$, relativa a uma câmera $i$ pode ser escrita como

$$P_i = K_i R_i \left[ I \;\middle|\; -\tilde{C}_i \right] = K_i \left[ R_i \;\middle|\; \mathbf{t_i} \right] \qquad (1)$$

em que $\mathbf{t}_i = -R_i \tilde{C}_i$, $K_i$ é uma matriz $3 \times 3$ que define os parâmetros intrínsecos, $R_i$ e $\tilde{C}_i$ representam os parâmetros extrínsecos, sendo $R_i$ uma matriz de rotação $3 \times 3$ e $\tilde{C}_i$ o centro de projeção da câmera em coordenadas não-homogêneas. Se os pontos no espaço 3D e os pontos no plano da imagem são representados por coordenadas homogêneas, a partir da Equação 1, um ponto $Q = (X, Y, Z, 1)^T$ no espaço 3D pode ser mapeado para o ponto $q_i = (x, y, 1)^T$ no plano da câmera $i$ por $q_i = P_i Q$.

O método que foi adotado para o posicionamento da câmera virtual consiste em definir a posição da câmera virtual entre duas câmeras reais de referência, levando em consideração uma transição relativa do ponto de vista que parte da câmera de referência 1 para 2. Na Figura 1, por exemplo, a posição da câmera virtual é estabelecida pela transição relativa que parte da câmera de referência $cam_1$ para $cam_2$. A transição do ponto de vista da câmera virtual é realizada por uma interpolação, controlada por um parâmetro $\alpha$, sendo

[2]Dispositivos auto-estereoscópicos 3D permitem a percepção 3D sem a necessidade de lentes especiais [24].

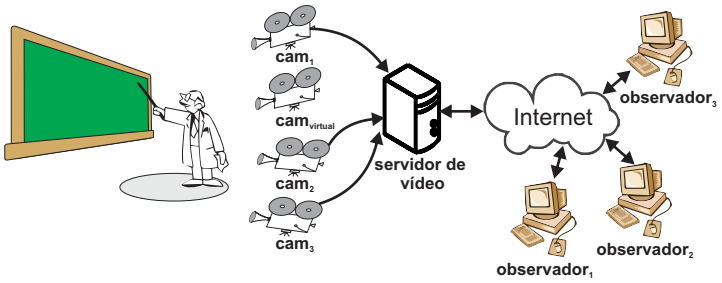



Figura 1. Visão geral do sistema de renderização baseada em vídeo para ensino a distância via Web.

$0 \leq \alpha \leq 1$. Quando $\alpha = 0$, o ponto de vista da câmera virtual é o mesmo da câmera de referência 1 e, conforme o valor de $\alpha$ é incrementado, o ponto de vista muda gradativamente para a segunda câmera.

Considerando as matrizes de projeção das câmeras de referência 1 e 2 como sendo $P_1$ e $P_2$ e a matriz de projeção da câmera virtual como

$$P_v = K_v R_v [I \mid -\tilde{C}_v] \tag{2}$$

a parametrização do caminho entre as duas câmeras de referência é dada por

$$P_v(\alpha) = K_v(\alpha) R_v(\alpha) [I \mid -\tilde{C}_v(\alpha)] \tag{3}$$

Para os parâmetros intrínsecos, a interpolação representa a mudança gradual do foco e do *eixo principal* da primeira para segunda câmera, que pode ser escrita como

$$K_v(\alpha) = (1-\alpha)K_1 + \alpha K_2 \tag{4}$$

Quando as câmeras são idênticas, nota-se que esta interpolação não terá o efeito desejado, pois se $K_1 = K_2$, durante a transição da primeira câmera para a segunda, $K_v(\alpha) = K_1 = K_2$.

Assumindo que $\mathbf{x}_i$, $\mathbf{y}_i$ e $\mathbf{z}_i$ são as colunas da matriz de rotação $R_i$, de forma que $R_i = [\mathbf{x}_i \ \mathbf{y}_i \ \mathbf{z}_i]$, a interpolação de $R_v(\alpha) = [\mathbf{x}_v(\alpha) \ \mathbf{y}_v(\alpha) \ \mathbf{z}_v(\alpha)]$ é estabelecida por

$$\begin{aligned} \mathbf{z}_v(\alpha) &= \frac{(1-\alpha)\mathbf{z}_1 + \alpha \mathbf{z}_2}{\|(1-\alpha)\mathbf{z}_1 + \alpha \mathbf{z}_2\|} \\ \mathbf{x}_v(\alpha) &= \frac{\mathbf{y}' \times \mathbf{z}_v(\alpha)}{\|\mathbf{y}' \times \mathbf{z}_v(\alpha)\|} \\ \mathbf{y}_v(\alpha) &= \mathbf{z}_v(\alpha) \times \mathbf{x}_v(\alpha) \end{aligned} \tag{5}$$

sendo

$$\mathbf{y}' = (1-\alpha)\mathbf{y}_1 + \alpha \mathbf{y}_2$$

Pode-se notar que $\mathbf{z}_v$ é calculado pela interpolação do eixo $Z$ das matrizes $R_1$ e $R_2$ e os demais vetores que correspondem aos eixos $X$ e $Y$ de $R_v$ são calculados pelos produtos vetoriais de eixos já computados, uma vez que $\mathbf{x_i}$, $\mathbf{y_i}$ e $\mathbf{z_i}$ são ortonormais.

Por fim, o centro de projeção da câmera virtual é computado pela interpolação

$$C_v(\alpha) = (1-\alpha)C_1 + \alpha C_2 \tag{6}$$

### C. Renderização por Varredura de Planos

O processo de síntese dos quadros da câmera virtual é realizado pela técnica de varredura de planos. Inicialmente, como mostra a Figura 2, o espaço 3D é discretizado em $k$ planos ao longo do eixo $Z$ e paralelos à câmera virtual, sendo que os objetos da cena estão localizados entre dois planos, o plano mais à frente ($\pi_1$) e o mais afastado ($\pi_k$), cobrindo todo o volume de interesse.

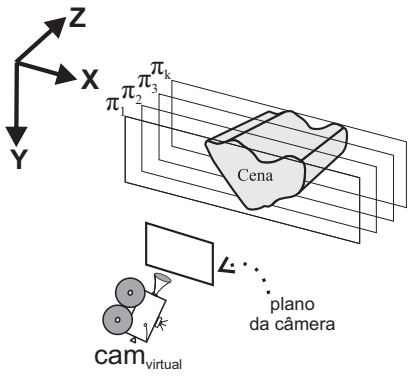



Figura 2. Definição de planos no espaço 3D.

A varredura é executada sobre $k$ planos e, ao final de cada varredura, uma vista da câmera virtual é renderizada. O processamento de um plano $\pi$ durante a execução de uma varredura consiste em quatro etapas principais:

i) projeção dos pontos sobre o plano da câmera virtual para o plano $\pi$.
ii) projeção dos pontos sobre $\pi$ para as câmeras reais.
iii) computação de uma medida denominada *consistência de cor* sobre os pontos projetados nas câmeras reais.
iv) síntese do quadro da câmera virtual a partir de uma votação baseada na consistência de cor computada na etapa anterior.

As etapas (i) e (ii) são ilustradas na Figura 3, considerando o ponto $q_v$ da câmera virtual e o plano $\pi_2$. Primeiramente, o ponto $q_v$ é projetado sobre o plano $\pi_2$. Posteriormente, $Q$ no espaço 3D é projetado para os planos das câmeras reais. Observa-se que os pontos $q_1$, $q_2$, $q_3$ e $q_v$ são correspondentes, uma vez que todos são projetados por $Q$.

A partir da Equação 1, o mapeamento da etapa (ii), $Q \rightarrow q$, em que um ponto $Q = (X, Y, Z, 1)^T$ no espaço 3D é projetado para um ponto $q = (x, y, 1)^T$ no plano da câmera, pode ser obtido por $q = PQ$, em que $P = K[R \mid \mathbf{t}]$. Na etapa (i),

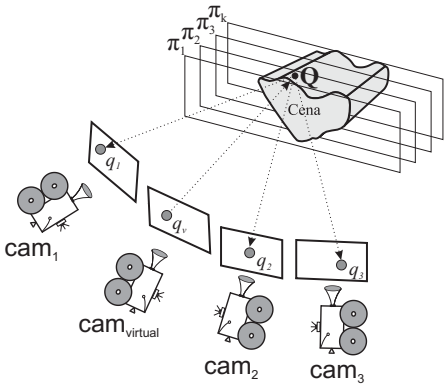



Figura 3. Projeção de pontos da câmera virtual para as demais câmeras via plano $\pi_2$.

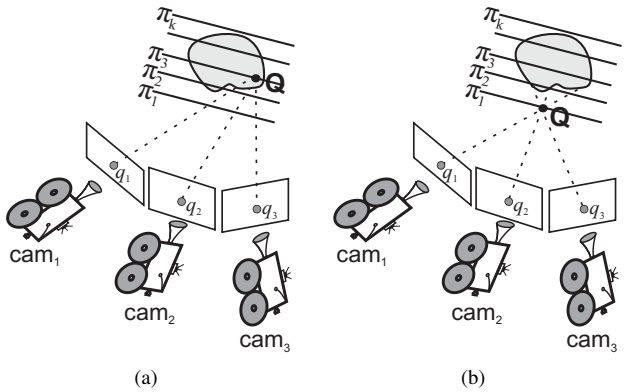



Figura 4. Projeção de um ponto $Q$ do espaço 3D sobre os planos das câmeras reais.

o procedimento é o inverso, o mapeamento $q \rightarrow Q$ deve ser estabelecido. Pode-se observar que $Q$ está sobre um plano $\pi$ e a coordenada $Z$ é conhecida, uma vez que os planos estão definidos ao longo do eixo $Z$. Com base em [3], o mapeamento $q \rightarrow Q$ pode ser obtido por

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \tilde{C} + \lambda R^{-1} K^{-1} q \tag{7}$$

em que

$$\lambda = \frac{Z - \tilde{C}_z}{z_3}$$

para $(z_1, z_2, z_3)^T = R^{-1}K^{-1}q$.

A terceira etapa do processamento de um plano $\pi$ consiste em computar a medida denominada *consistência de cor*, que mede a similaridade das cores dos pontos projetados nas câmeras reais durante a segunda etapa. Com base em [22], assumindo que as superfícies dos objetos são difusas e visíveis (não há oclusão), a consistência de cor é usada para determinar se um ponto $Q$ sobre $\pi$ está sobre uma superfície de um objeto na cena. Quanto maior for a consistência de cor, maior é a probabilidade de um ponto $Q$ estar sobre a superfície de um objeto na cena.

A Figura 4 (a) ilustra o ponto $Q$ sobre o plano $\pi_3$ e a superfície do objeto. Pode-se observar que os pontos $q_1$, $q_2$ e $q_3$ possuem a mesma cor e a medida de consistência de cor é alta, pois os raios de projeção das câmeras são referentes a um mesmo ponto na cena, o ponto $Q$. Em contraste, a Figura 4 (b) mostra o ponto $Q$ que não está sobre a superfície do objeto na cena e as cores dos pontos $q_1$, $q_2$ e $q_3$ podem não possuir a mesma cor, pois as cores são relativas aos raios de projeção sobre três pontos distintos no objeto na cena.

Assumindo-se que cada ponto $q_i$ tem uma cor $c_i$ associada, a consistência de cor $s$ relativa aos pontos $q_i$ é calculada pela soma das diferenças entre cada cor $c_i$ e a cor média $c_{\text{média}}$,

$$s = \sum_i |c_i - c_{\text{média}}| \tag{8}$$

em que $1 \leq i \leq n$, sendo $n$ o número de câmeras e

$$c_{\text{média}} = \frac{\sum_i c_i}{n} \tag{9}$$

Assim, quanto menor for o valor de $s$, maior será a medida de consistência de cor relativa ao conjunto de cores $c_i$.

Por fim, a quarta e última etapa do processamento de um plano $\pi$ consiste em renderizar a vista da câmera virtual a partir de uma votação. Cada ponto $x_v$ da câmera virtual é composto de um valor de cor $c_v$ e de consistência de cor $s_v$. Durante o processo de síntese, a cor $c_v$ é definida pela consistência de cor calculada na Equação 8. Se a cor de $c_{\text{média}}$ é mais consistente do que a atual cor $c_v$, o valor de cor $c_v$ e de consistência de cor $s_v$ são substituídos por $c_{\text{média}}$ e $s$, respectivamente. Com isso, durante a varredura de $k$ planos, a cor de cada ponto $x_v$ da câmera virtual é definida por uma votação, em que a cor mais consistente ao longo da varredura é atribuída.

### *D. Detalhes de Implementação*

A implementação do sistema proposto é dividida entre o desenvolvimento do servidor de vídeo e o método de renderização usado nos dispositivos de visualização. O servidor usa a CPU para a aquisição, compressão e transmissão dos vídeos. Os vídeos são comprimidos empregando-se a biblioteca OpenCV [26] e transmitidos usando *sockets*. Inicialmente, o sistema foi projetado para operar com 4 câmeras e 1 servidor de vídeo. Entretanto, a abordagem proposta é escalável, sendo que um maior número de câmeras pode ser agregado e o servidor de vídeo pode ser implementado de forma distribuída.

O Algoritmo 1 descreve os procedimentos executados para renderização e exibição dos quadros nos dispositivos de visualização. O processamento do algoritmo combina o uso de CPU e GPU. A CPU é usada no processamento da recepção dos vídeos e no posicionamento da câmera virtual. A tarefa de renderização é implementada em OpenGL para executar em GPU utilizando *fragment shader* [27]. Os planos são desenhados no espaço 3D (primitiva *GL_QUAD*) e as imagens das câmeras reais são usadas como texturas, as quais são mapeadas para os planos por um mapeamento de textura projetivo, em que as coordenadas de texturas são computadas a partir das matrizes de projeção das câmeras.

O uso de *multi-texturing* provê o acesso simultâneo de cada textura para cada fragmento, possibilitando a computação da medida de consistência de cor e da cor média (Equação 8). A etapa de votação pode ser obtida automaticamente pelo teste de profundidade do OpenGL por meio do armazenamento

**Algoritmo 1** Renderização dos quadros da câmera virtual empregando o sistema de VBR proposto.

**while** a cena está sendo capturada **do**
  dadas $n$ câmeras reais, receber $n$ vídeos
  definir o ponto de vista da câmera virtual com base em duas câmeras reais ajacentes (subseção III-B)
  **for** cada plano $\pi_k$ no espaço 3D **do**
    **for** cada ponto $q_v$ da câmera virtual **do**
      projetar $q_v$ sobre os pontos $q_i$ em cada câmera real $i$ (etapas (i) e (ii) detalhadas na subseção III-C)
      computar a cor média $c_{\text{média}}$ dos pontos $q_i$ com base nos vídeos capturados
      computar a consistência de cor $s$ relativa aos pontos $q_i$ (Equação 8)
      **if** $s$ é menor que a atual consistência de cor de $q_v$ **then**
        $c_{q_v} = c_{\text{média}}$ (definir a cor de $q_v$ como sendo $c_{\text{média}}$)
        $s_{q_v} = s$ (atualizar o valor de consistência de cor de $q_v$)
      **end if**
    **end for**
  **end for**
  exibir o quadro da câmera virtual renderizado
**end while**

da cor média em *gl_FragColor* e da consistência de cor em *gl_FragDepth*, fazendo com que, durante a varredura dos planos, a cor no *frame buffer* seja atualizada automaticamente para a cor mais consistente. A vista final renderizada pode ser obtida *offscreen* no *frame buffer*.

## IV. Conclusões

Este trabalho apresentou uma arquitetura para Renderização Baseada em Vídeo (VBR) para ensino a distância via Web, permitindo que o observador possa determinar o ponto de vista desejado durante a visualização da cena, agregando maior interação e realismo, não ficando restrito à visualização passiva do conteúdo como nos métodos convencionais.

A abordagem de posicionamento da câmera fornece um modo intuitivo para especificação do ponto de vista da câmera virtual, com base na transição entre duas câmeras reais de referência. O sistema foi projetado para executar todas as etapas *on-line*, desde a captura da cena até a renderização das vistas da câmera virtual. Os recursos computacionais necessários para a implementação do protótipo são de baixo custo, utilizando poucas câmeras e somente um servidor, embora o sistema seja escalável, permitindo a adição de novas câmeras e computadores.

## V. Agradecimentos

Os autores gostariam de agradecer à FAPESP (processos 2010/00605-1 e 2011/21260-5) e ao CNPq (processo 307113/2012-4) pelo apoio financeiro.

## Referências

[1] M. A. Magnor, *Video-Based Rendering*. AK Peters Ltd, 2005.

[2] T. Kanade, P. J. Narayanan, and P. W. R, "Virtualized Reality: Concepts and Early Results," in *IEEE Workshop on the Representation on Visual Scene*, 1995.

[3] Y. Morvan, "Acquisition, compression and rendering of depth and texture for multi-view video," Ph.D. thesis, Eindhoven University of Technology, The Netherlands, Apr. 2009.

[4] M. Okutomi and T. Kanade, "A Multiple-Baseline Stereo," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, pp. 353–363, Apr. 1993.

[5] C. L. Zitnick, S. B. Kang, M. Uyttendaele, S. Winder, and R. Szeliski, "High-Quality Video View Interpolation using a Layered Representation," *ACM Transactions on Graphics*, vol. 23, pp. 600–608, Aug. 2004.

[6] W. Li, J. Zhou, B. Li, and M. I. Sezan, "Virtual view specification and synthesis for free viewpoint television," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 19, no. 4, pp. 533–546, 2009.

[7] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, 2004.

[8] K.-J. Oh, S. Yea, and Y.-S. Ho, "Hole Filling Method Using Depth Based In-Painting for View Synthesis in Free Viewpoint Television and 3-D Video," in *Picture Coding Symposium*, May 2009, pp. 1–4.

[9] Y. Mori, N. Fukushima, T. Yendo, T. Fujii, and M. Tanimoto, "View Generation with 3D Warping using Depth Information for FTV," *Signal Processing: Image Communication*, vol. 24, no. 1-2, pp. 65–72, 2009.

[10] L. McMillan, "An Image-Based Approach to Three-Dimensional Computer Graphics," Ph.D. thesis, University of North Carolina, Apr. 1997.

[11] L.-H. Wang, J. Zhang, S.-J. Yao, D.-X. Li, and M. Zhang, "GPU Based Implementation of 3DTV System," in *Sixth International Conference on Image and Graphics*, Aug. 2011, pp. 847–851.

[12] M. Garland, S. Le Grand, J. Nickolls, J. Anderson, J. Hardwick, S. Morton, E. Phillips, Y. Zhang, and V. Volkov, "Parallel Computing Experiences with CUDA," *IEEE Micro*, vol. 28, no. 4, pp. 13–27, July-August 2008.

[13] NVIDIA Corporation, "CUDA GPUs," 2014, http://developer.nvidia.com/cuda-gpus.

[14] J. C. Yang, M. Everett, C. Buehler, and L. McMillan, "A real-time distributed light field camera," in *13th Eurographics Workshop on Rendering*, Aire-la-Ville, Suíça, 2002, pp. 77–86.

[15] M. Levoy and P. Hanrahan, "Light Field Rendering," in *23rd Annual Conference on Computer Graphics and Interactive Techniques*, New York, NY, USA, 1996, pp. 31–42.

[16] S. J. Gortler, R. Grzeszczuk, R. Szeliski, and M. F. Cohen, "The Lumigraph," in *23rd Annual Conference on Computer Graphics and Interactive Techniques*, New York, NY, USA, 1996, pp. 43–54.

[17] Y. Liu, Q. Dai, and W. Xu, "A Real Time Interactive Dynamic Light Field Transmission System," in *IEEE International Conference on Multimedia and Expo*, Jul. 2006, pp. 2173–2176.

[18] H. Schirmacher, M. Li, and H.-P. Seidel, "On-the-Fly Processing of Generalized Lumigraphs," in *Eurographics*, 2001, pp. 165–173.

[19] W. Wei, J. Z. Zhi, Y. S. Cong, and Z. Dan, "An efficient method for all-in-focused light field rendering," in *3rd IEEE International Conference on Computer Science and Information*, vol. 1, Jul. 2010, pp. 399–404.

[20] Y. Jeon and H. Park, "Fast All In-Focus Light Field Rendering Using Dynamic Block-Based Focusing Technique," in *3DTV Conference: The True Vision - Capture, Transmission and Display of 3D Video*, May 2011, pp. 1–4.

[21] R. T. Collins, "A Space-Sweep Approach to True Multi-Image Matching," in *Conference on Computer Vision and Pattern Recognition*, Washington, DC, USA, 1996, pp. 358–363.

[22] R. Yang, G. Welch, and G. Bishop, "Real-Time Consensus-Based Scene Reconstruction Using Commodity Graphics Hardware," in *10th Pacific Conference on Computer Graphics and Applications*, Washington, DC, USA, 2002.

[23] H. S. Vincent Nozick, "On-line Free-viewpoint Video: From Single to Multiple View Rendering," *International Journal of Automation and Computing*, vol. 5, no. 3, 2008.

[24] N. A. Dodgson, "Autostereoscopic 3D Displays," *Computer*, vol. 38, pp. 31–36, Aug. 2005.

[25] Z. Zhang, "A Flexible New Technique for Camera Calibration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1330 – 1334, Nov. 2000.

[26] G. Bradski and A. Kaehler, *Learning OpenCV: Computer Vision with the OpenCV Library*. Cambridge, MA: O'Reilly, 2008.

[27] R. J. Rost, B. Licea-Kane, D. Ginsburg, J. M. Kessenich, B. Lichtenbelt, H. Malan, and M. Weiblen, *OpenGL Shading Language*, 3rd ed. Addison-Wesley Professional, 2009.

# Lightness Constancy in Edge Detection: A Simple Approach using Luminance Ratios

Hugo Vieira Neto
Graduate Program in Electrical and Computer Engineering
Federal University of Technology – Paraná
E-mail: hvieir@utfpr.edu.br

***Abstract*—Reliable edge detection is of utmost importance to many sophisticated computer vision methods and has been historically computed using luminance differences, which are quite sensitive to changes in illumination. It is argued here that using luminance ratios for edge detection in computer vision, rather than luminance differences, results in more robust edge information with respect to changes in illumination, which is in line with theoretical models for the lightness constancy observed in human visual perception. In order to test this hypothesis, a simple technique that converts luminance differences into luminance ratios with minimal effort is proposed. Experiments regarding edge detection and image segmentation under simulated changes in illumination show promising results for the luminance ratio approach.**

## I. Introduction

Adaptation to different illumination conditions is a fundamental trait of the human visual system, which is also extremely desirable to computer vision systems that aim at extracting robust environmental descriptions from images regardless of illumination changes. This particular trait is known as perceptual constancy and consists in perceiving the properties of visual objects, which are usually constant over different viewing conditions, rather than the properties of their projected images, which are variable over different viewing conditions [1]. Figure 1 illustrates the concept of achromatic lightness constancy, which is the main subject of this paper.

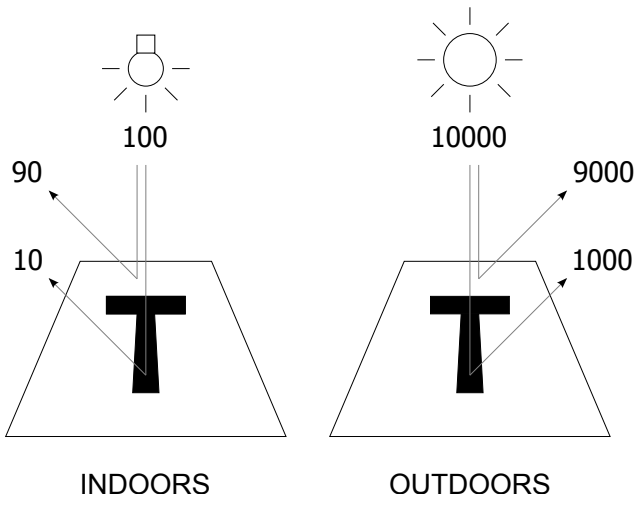



Fig. 1. Achromatic lightness constancy: human perception of black ink on white paper is the same under very different illumination levels (adapted from [1]).

Human perception of black ink on white paper is the same despite the amount of illumination in the environment. However, the example in Fig. 1 shows that the amount of reflected light outdoors is a hundred times larger than indoors – what remains constant in both illumination conditions is the relative amount of reflected light for dark and light regions of the image. According to the theory of lightness constancy of Hans Wallach [2], *luminance ratios* are what determine perceived lightness and contrast – one can notice that the luminance ratio is the same (9:1) in both illumination conditions depicted in Fig. 1.

In computer vision, a key concept regards edge detection – measurements of local contrast that allow the segmentation of image regions according to the reflected luminance. Edge detection in computer vision systems has been historically estimated by difference operators – for example, convolution with Prewitt or Sobel masks – which approximate luminance gradients. This technique, whose results are rather sensitive to changes in illumination, dates back to an age in which computing power was very limited in terms of mathematical functions available for practical implementations. With the increased computing power that became largely available and now allows practical implementations using more sophisticated mathematical functions and also more advanced numerical representations, a paradigm shift towards edge detection using luminance ratios may be in order. The hypothesis is that luminance ratios are able to provide edge information that is more robust to illumination changes than luminance differences. In this context, a quantitative performance comparison of edge detection using both luminance differences and luminance ratios is made here for simulated changes in illumination.

The remainder of this paper is structured as follows. In Section II, a simple mathematical property that conveniently converts luminance differences into luminance ratios is described. Section III presents the experimental setup used to test the ratio-based edge detection and compare it to the difference-based approach. The experimental results obtained are presented and discussed in Section IV, and finally, in Section V, conclusions are drawn and future work outlined.

## II. Luminance Ratios from Luminance Differences

A vast number of computer vision methods rely on information extracted from gray level images using luminance differences, which are quite simple to compute and provide effective results if illumination conditions are kept reasonably constant. The purpose here is to establish a method to convert luminance differences into luminance ratios without resorting

to radical changes in existing edge detection techniques, so that more sophisticated methods that rely on them can benefit almost immediately from edge information that is more robust to changes in illumination.

A very simple way of mapping ratios to differences consists in using the well-known property of logarithms expressed in Eq. 1:

$$\log_b \left( \frac{p}{q} \right) = \log_b(p) - \log_b(q). \quad (1)$$

Equation 2 shows a convenient way of mapping eight-bit gray levels to a normalised logarithmic scale, when $b = 256$ is used:

$$g(x, y) = \log_{256}[f(x, y) + 1], \quad (2)$$

where $g(x, y)$ is the logarithmic mapping in the range $[0, 1]$ of gray level $f(x, y)$. The inverse mapping, from normalised logarithmic values back to the linear scale in the range $[0, 255]$, can be done by using Eq. 3:

$$f(x, y) = 256^{g(x,y)} - 1. \quad (3)$$

Therefore, performing ratio-based edge detection merely consists in three steps: (1) mapping the original input image to the normalised logarithmic scale using Eq. 2; (2) performing conventional difference-based edge detection with no alterations; (3) optionally mapping the resulting edge image (magnitude only) back to the linear scale using Eq. 3. It is worth noting that ratio-based edge detection preserves the direction of local image gradients to great extent. Floating point numerical representations must be used in all steps in order to ensure accuracy of results.

## III. Experimental Setup

In order to test the proposed ratio-based edge detection method and make comparisons to the existing difference-based method, experiments were designed in which image intensities were artificially manipulated. A gray level version of the Lena image with $256 \times 256$ pixels in size was used as input to both techniques, with synthetic changes in illumination obtained by applying increasing attenuation factors (from 0% to 100% in steps of 5%) to the pixels of the image. Two scenarios were investigated: (1) uniform attenuation factors were applied globally to the input image; (2) varying attenuation factors were applied locally to the input image, following v-shaped profiles, as shown in Fig. 2.

The v-shaped attenuation profiles used in the experiments have maximum attenuation in the middle of the horizontal dimension of the input image. Figure 2 shows two examples of such attenuation profiles, one of them with maximum attenuation of 50% (blue line) and the other with maximum attenuation of 100% (red line). This approach is used to investigate the performance behaviour of both edge detection approaches when nonuniform illumination profiles affect the input image.

The expected outcome of these experiments is that ratio-based (logarithmic gradient) edges are less sensitive to changes in illumination than their difference-based (linear gradient) counterparts.

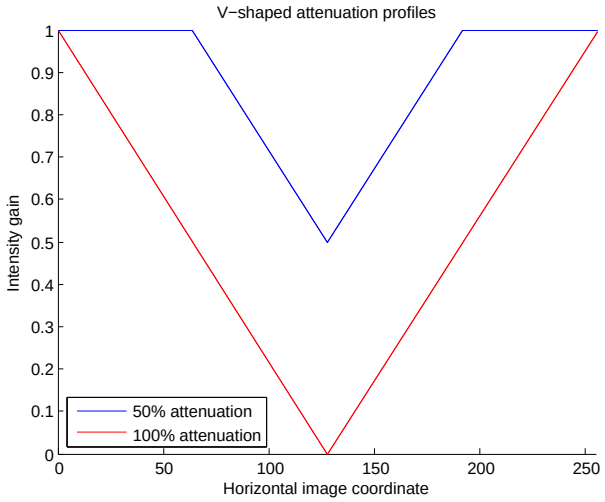
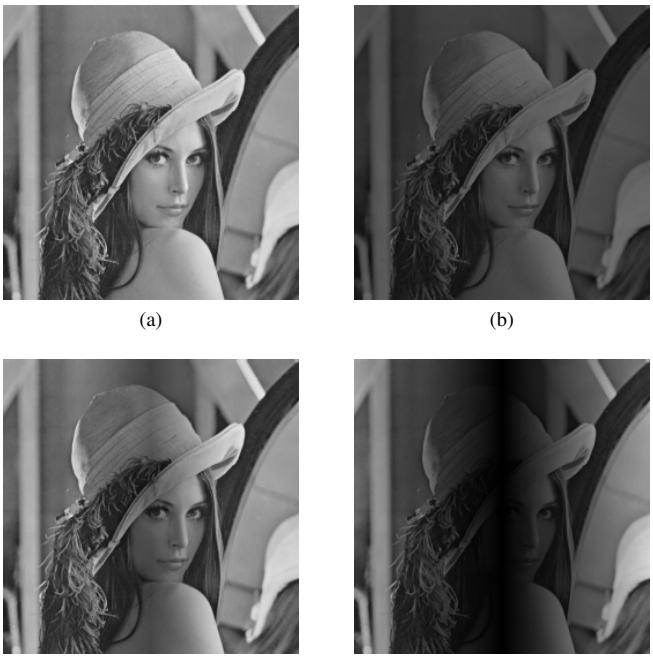



Fig. 2. V-shaped attenuation profiles: intensity attenuation is a function of the horizontal coordinate of pixels in the original input image. The blue line corresponds to a maximum attenuation of 50% while the red line corresponds to a maximum attenuation of 100% in the middle of the image.

Figure 3 shows examples of the resulting input images after the synthetic illumination changes described before. The original Lena image is shown in Fig. 3a, whereas Fig. 3b shows its uniformly attenuated version at 50%. Figures 3c and 3d show the resulting input images after the v-shaped attenuation profiles at 50% and 100% (see Fig. 2) were applied, respectively, where progressively dark vertical lines in the middle of the images can be noticed.



(a) (b) (c) (d)

Fig. 3. Input images: (a) original Lena image; (b) 50% uniform intensity attenuation; (c) 50% v-shaped intensity attenuation profile (blue line in Fig. 2); (d) 100% v-shaped intensity attenuation profile (red line in Fig. 2).

A first experiment was conducted, in which the original Lena image and its attenuated versions were used as inputs to the Sobel edge detection method [3], both using the conventional difference-based approach (linear gradient) and the proposed ratio-based approach (logarithmic gradient). In order to assess the influence of changes in luminance, the edge results for the original image were used as a baseline for computing the root mean square (RMS) error of edge results for each attenuated image.

The RMS error for edge magnitudes was computed using:

$$me_{RMS} = \sqrt{\frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}[m_a(x,y) - m_o(x,y)]^2}, \quad (4)$$

where $M$ is the horizontal image dimension, $N$ is the vertical image dimension, $m_a(x,y)$ are the resulting edge magnitudes for the attenuated image and $m_o(x,y)$ are the edge magnitudes for the original image. The RMS error for edge magnitudes was normalised using the extreme case in which no edges can be detected (100% uniform attenuation):

$$me_{RMS,max} = \sqrt{\frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} m_o(x,y)^2}. \quad (5)$$

The RMS error for edge directions was computed using a modified version of Eq. 4:

$$de_{RMS} = \sqrt{\frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}[\text{angdif}(o_a(x,y), o_o(x,y))]^2}, \quad (6)$$

where angdif$(\cdot)$ is a function that computes angular differences considering their periodicity, $o_a(x,y)$ are the resulting edge directions for the attenuated image and $o_o(x,y)$ are the edge directions for the original image, always having the difference-based edge directions as baseline. The RMS error for edge directions was normalised using the extreme case in which all directions are opposite to the baseline reference:

$$de_{RMS,max} = \sqrt{\frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}\pi^2}. \quad (7)$$

In order to minimise the effect of noise, directions corresponding to weak edge responses (less than 10% of the maximum edge magnitude in the entire image) are discarded when computing the RMS error for edge directions.

Finally, an additional experiment was conducted using the same set of images as inputs to the blob colouring segmentation algorithm [4], which is based in luminance differences between adjacent pixels and is therefore known to be sensitive to changes in illumination, which can cause flooding effects that merge adjacent regions. A preliminary quantitative assessment of how much changes in luminance affect blob colouring segmentation results was done by counting the number of segmented regions when using both concepts of linear and logarithmic gradients.

## IV. Experimental Results

### A. Sobel Edge Detection

The experiments regarding the impact of changes in illumination on Sobel edge detection [3] were initially conducted using uniform luminance attenuation (see Figs. 3a and 3b), in which resulting edge magnitudes and directions were assessed for increasing attenuation factors.

Fig. 4 shows the normalized RMS error for edge magnitudes as a function of uniform intensity attenuation for linear and logarithmic gradient approaches.

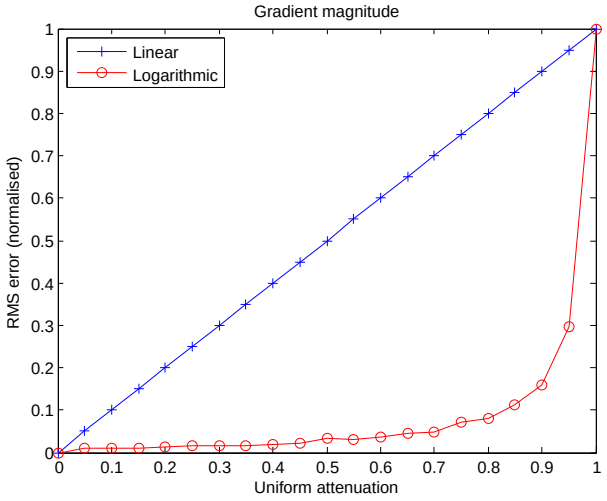



Fig. 4. Normalised RMS error for gradient magnitudes as a function of uniform intensity attenuation: linear gradient magnitudes (blue line) are heavily affected by changes in luminance, whereas logarithmic gradient magnitudes (red line) are much more robust – 80% intensity attenuation results in less than 10% of the maximum RMS error when using the logarithmic gradient.

As can be clearly noticed in Fig. 4, the normalised RMS error for linear gradient magnitudes increases linearly with uniform luminance attenuation, indicating that the conventional difference-based edge detection performance is heavily affected by changes in illumination. On the other hand, the normalised RMS error for logarithmic gradient magnitudes is less affected by changes in illumination, as predicted by Wallach's lightness constancy theory [2] – uniform attenuation factors as high as 80% result in less than 10% of the maximum expected RMS error for ratio-based edge magnitudes.

Figure 5 shows that the normalised RMS error for edge directions as a function of uniform changes in illumination is not critical for both linear and logarithmic gradient approaches, but the results indicate that the logarithmic gradient approach yields a small directional offset in comparison to the reference baseline, given by the conventional linear gradient directions of the original input image. However, Fig. 5 also indicates that logarithmic gradient directions are much more stable to changes in luminance than their linear gradient counterparts, a characteristic that is desirable to gradient-based descriptors often used in state-of-the-art object recognition methods such as SIFT [5], [6] and SURF [7].

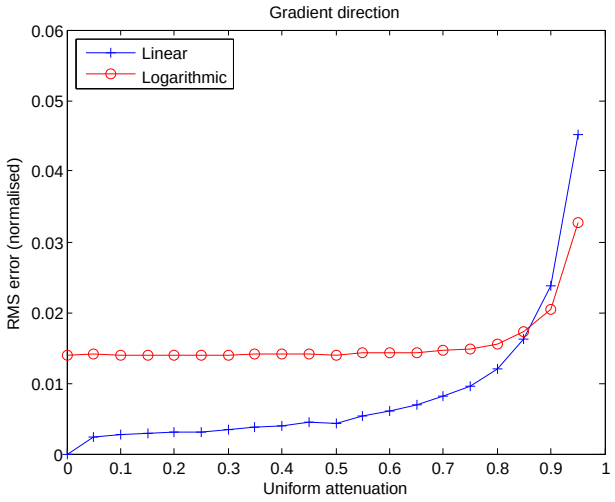



Fig. 5. Normalised RMS error for gradient directions as a function of uniform intensity attenuation: despite presenting a small average angular offset, logarithmic gradient directions (red line) are more stable to changes in luminance when compared to linear gradient directions (blue line).

Following the experiments regarding the impact of uniform changes in illumination on Sobel edge detection, additional experiments regarding nonuniform *gradual* changes in luminance were conducted. In these experiments, the resulting edge magnitudes and directions were assessed for increasing maximum attenuation factors in v-shaped attenuation profiles (see Figs. 2, 3c and 3d).

The normalised RMS error for edge magnitudes as a function of nonuniform intensity attenuation for linear (difference-based) and logarithmic (ratio-based) gradient approaches is shown in Fig. 6.

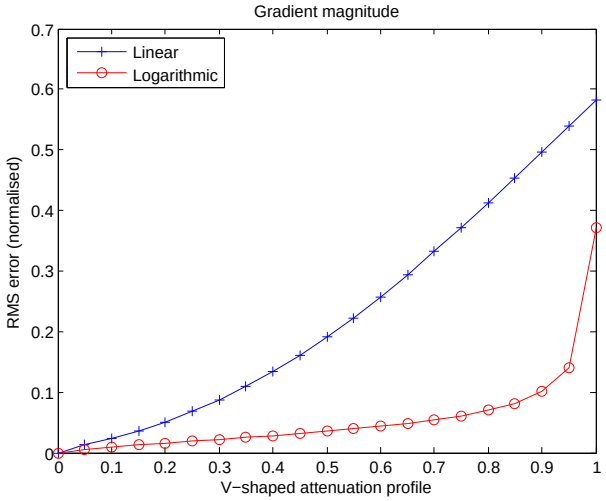



Fig. 6. Normalised RMS error for gradient magnitudes as a function of nonuniform intensity attenuation (v-shaped attenuation profile): as in the case of uniform intensity attenuation, linear gradient magnitudes (blue line) are much more affected by changes in luminance than logarithmic gradient magnitudes (red line).

It can be noticed in Fig. 6 that conventional linear gradient magnitudes are much more affected by nonuniform changes in illumination than logarithmic gradient magnitudes, as expected. Figure 7 visually illustrates the effects of the v-shaped attenuation profiles (100% maximum attenuation) on edge detection using both linear and logarithmic approaches.

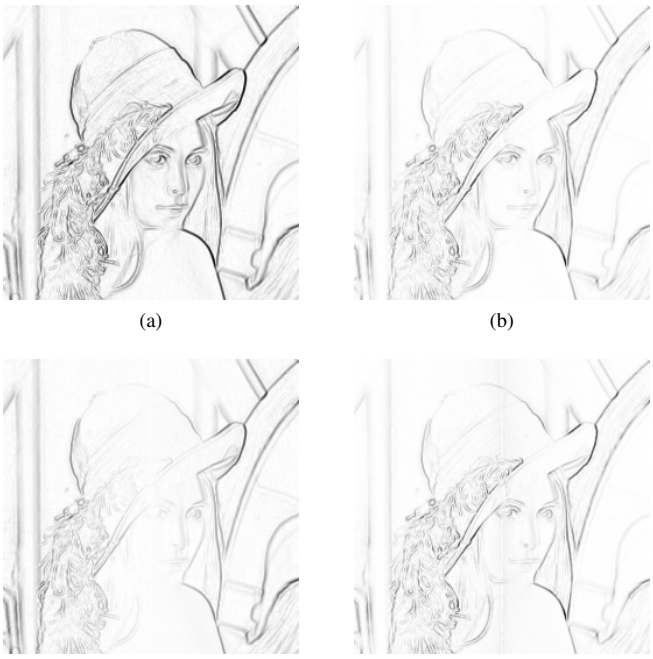



Fig. 7. Sobel edge detection results: (a) linear gradient magnitude (original input image); (b) logarithmic gradient magnitude (original input image); (c) linear gradient magnitude (100% v-shaped attenuation profile); (d) logarithmic gradient magnitude (100% v-shaped attenuation profile). The resulting linear gradient magnitude is much more affected than the resulting logarithmic gradient magnitude when comparing (c) and (d). All edge images were complemented for visualisation purposes, but absolute magnitudes were kept unchanged in order to allow fair visual comparisons between results.

Although stronger edge responses can be observed in the linear gradient results for the original input image (Fig. 7a), linear edge responses are somewhat noisy and clearly fade away when the image is affected by nonuniform attenuation, especially in the middle of the horizontal dimension (Fig. 7c). In the case of logarithmic gradient results for the original input image (Fig. 7b), edge responses are not as strong as the ones observed in the linear case. However, logarithmic edge responses are less noisy and do not fade away when the input image is affected (Fig. 7d) – except from some vertical artifacts that appear in the middle of the horizontal dimension as a result of extreme intensity attenuation in that region (100%), overall logarithmic edge responses are reasonably stable to nonuniform changes in luminance.

The impact of nonuniform changes in luminance on edge directions is shown in Fig. 8, where it can be noticed that difference-based edge directions are more severely affected by nonuniform changes in illumination – notice that the normalised RMS error for linear gradient directions rises at higher rates in comparison to Fig. 5. On the other hand, ratio-based edge directions show stable behaviour even in the presence of gradual changes in illumination.

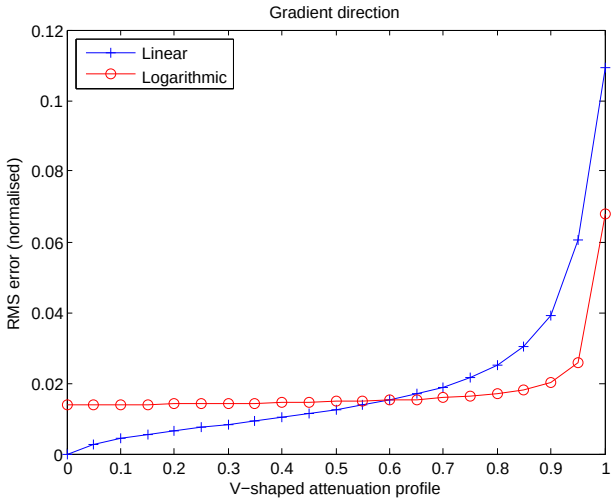



Fig. 8. Normalised RMS error for gradient directions as a function of nonuniform intensity attenuation (v-shaped attenuation profile): logarithmic gradient directions (red line) are more stable to changes in luminance than linear gradient directions (blue line), which are more severely affected by nonuniform than uniform intensity changes (compare to Fig. 5).

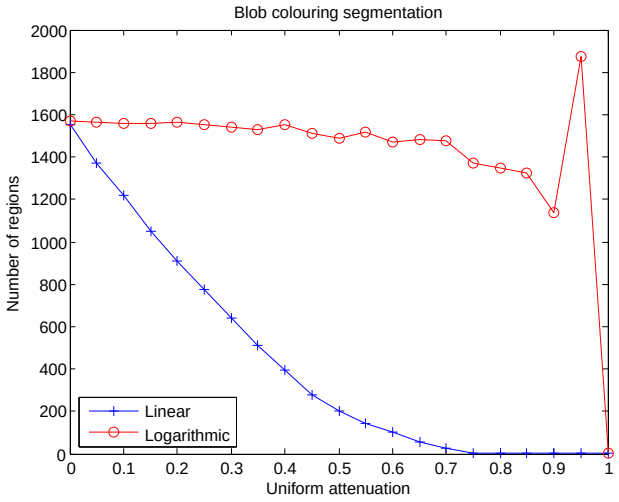



Fig. 9. Number of regions resulting from the blob colouring process as a function of uniform intensity attenuation: the number of segmented regions for the difference-based (linear) contrast decreases radically with increasing uniform attenuation factors, as a result of originally separated regions being merged, whereas the number of segmented regions for the ratio-based (logarithmic) contrast is reasonably stable (the unexpected peak at 95% attenuation is due to truncation errors in small intensity values).

### *B. Blob Colouring Segmentation*

The final experiment was conducted in the context of the blob colouring algorithm, which is a segmentation method that grows regions based on the contrast between adjacent pixels [4]. The objective of this last experiment was to assess how blob colouring segmentation results are affected by changes in illumination, when difference-based or ratio-based contrast estimation are used. For that, the number of resulting regions at the end of the segmentation process was used as a direct quantitative measurement of the impact of uniform and nonuniform changes in luminance. The threshold used for the difference-based approach was 7% of the maximum possible contrast magnitude and the threshold used for the ratio-based approach was 3.3% – these values were chosen to result in approximately the same number of regions for the original image.

Figure 9 shows the number of regions resulting from the blob colouring process as a function of uniform intensity attenuation for both difference-based and ratio-based contrast. It can be noticed that the number of regions for the linear approach are radically affected by luminance attenuation – originally separated regions are gradually merged because of the resulting difference-based contrast attenuation. On the other hand, the number of regions for the logarithmic approach is reasonably stable for a wide range of attenuation factors. The unexpected peak observed in the number of regions for the logarithmic approach is explained by truncation errors in small intensity values that happen as part of the simulation of changes in illumination.

The number of regions resulting from the blob colouring process when nonuniform luminance attenuation is applied to the original input image is shown in Fig. 10, where once more it can be noticed that the ratio-based approach is still much more resilient to changes in luminance when compared to the conventional difference-based approach.

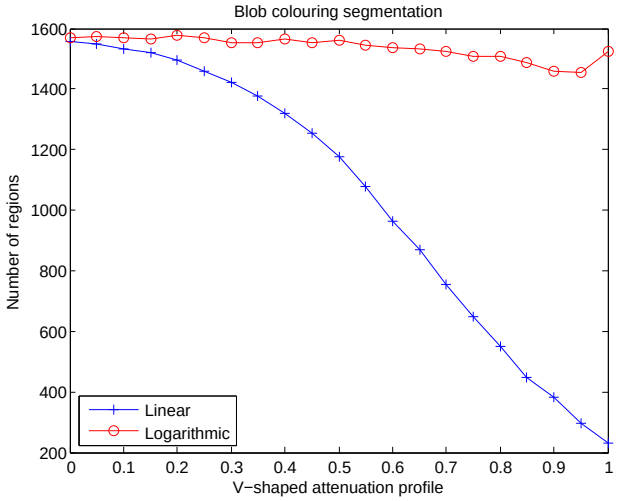



Fig. 10. Number of regions resulting from the blob colouring process as a function of nonuniform intensity attenuation (v-shaped attenuation profile): the number of segmented regions for the ratio-based (logarithmic) contrast is much more stable than the number of segmented regions for the difference-based (linear) contrast.

## V. Conclusion

Wallach's theory of lightness constancy states that human perception of contrast is based on luminance ratios [2], which provide a large degree of invariance to changes in illumination. However, contrast estimations in computer vision are traditionally computed by difference operators [3], which in their turn are quite sensitive to changes in illumination.

In this work, a simple mathematical property was used in order to convert luminance differences into luminance ratios, enabling changes with minimal effort in existing difference-based edge detection algorithms. Despite its simplicity, the proposed method provides edge detection results that are much more stable to changes in illumination and therefore can be of special interest to more versatile computer vision applications.

Preliminary experiments assessing the performance of difference-based and ratio-based gradients indicate that the ratio-based approach is indeed rather robust to simulated changes in luminance in the context of edge detection and image segmentation. Future work includes extending experiments to test the ratio-based edge detection approach using input images acquired in different physical lighting conditions, and also testing this concept in the context of object recognition algorithms such as SIFT [5], [6] and SURF [7].

## REFERENCES


[1] S. E. Palmer, *Vision Science: Photons to Phenomenology*. Cambridge: MIT Press, 1999.

[2] H. Wallach, "Brightness constancy and the nature of achromatic colors," *Journal of Experimental Psychology*, vol. 38, pp. 310–324, 1948.

[3] R. C. Gonzalez and P. A. Wintz, *Digital Image Processing*, 2nd ed. Reading: Addison-Wesley, 1987.

[4] D. H. Ballard and C. M. Brown, *Computer Vision*. Englewood Cliffs: Prentice-Hall, 1982.

[5] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of the 7th IEEE International Conference on Computer Vision*, 1999, pp. 1150–1157.

[6] ——, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[7] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (SURF)," *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346–359, 2008.

# Using image analysis and processing for morphological characterization of bovine spermatozoa

Rebecca Espírito Santo da Cruz
Faculdade de Medicina Veterinária
Universidade Federal de Uberlândia
Uberlândia, MG, Brasil
rebecca@vet.ufu.br

Marcelo Emílio Beletti
Instituto de Ciências Biomédicas
Universidade Federal de Uberlândia
Uberlândia, MG, Brasil
mebeletti@ufu.br

Bruno Augusto Nassif Travençolo
Faculdade de Computação
Universidade Federal de Uberlândia
Uberlândia, MG, Brasil
travencolo@gmail.com

***Resumo*—The semen analysis is the main method of diagnosis in male fertility used in veterinary medicine. However, due to the subjectivity intrinsic of an visual analysis, it has being proposed to use computer image analysis for the evaluation of sperm morphology. Thus, the present study aims to characterize the morphology of bull sperm head by using image processing and analysis techniques. For morphological analysis it was performed the preparation of smears, the capture and segmentation of the images, and the automatic evaluation of various characteristics obtained from the head of the sperm. In order to validate the efficiency of the automatic segmentation, an index of similarity from the comparison with manual segmentation was obtained. From this ratio it was found that 85% of the heads automatically detected had good quality segmentation. The results show that the method can be used by researchers in male fertility studies.**

## I. Introdução

Em termos mundiais, a pecuária bovina brasileira destaca-se pelo seu efetivo e inesgotável potencial de crescimento. Esse crescimento demonstra a consolidação da atividade, cada vez mais fortalecida pela maior disponibilidade de sêmen [1]. O aumento do número do rebanho está diretamente ligado à melhoria das taxas reprodutivas dos animais, que por sua vez depende diretamente da fertilidade. Nenhum teste isolado é capaz de predizer a fertilidade de uma amostra de sêmen, mas o exame de várias características físicas e morfológicas pode predizer uma maior fertilidade [2].

Várias técnicas têm sido empregadas para medir estes fatores, entre elas encontra-se o espermograma. O espermograma consiste na avaliação física e morfológica do sêmen, assumindo grande importância como diagnóstico, sendo que o médico veterinário é frequentemente requisitado para a avaliação do animal quanto à sua habilidade em produzir espermatozoides viáveis [3], seja em centrais de inseminação ou em propriedades que utilizam a monta natural para provar a aptidão do macho. É o principal método de diagnóstico de fertilidade do macho utilizado na rotina veterinária para a avaliação de várias características espermáticas, dentre elas a motilidade e a morfologia dos espermatozoides [4], [5].

Frequentemente, a morfologia e a motilidade dos espermatozoides são estimadas de forma subjetiva, por meio da avaliação visual das células espermáticas sob microscopia óptica convencional, representando a principal análise laboratorial utilizada pelas centrais de coleta e congelamento de sêmen [6]. Essa avaliação subjetiva, amplamente utilizada em laboratórios de andrologia, depende da habilidade e percepção do técnico [7]. Por isso, com o objetivo de reduzir a subjetividade, tendenciosidade, possíveis falhas e aumentar a reprodutibilidade entre diferentes examinadores, vem sendo proposto o uso de análise de imagem por computador para a avaliação da motilidade e morfologia dos espermatozoides [6], [7], [8], [9].

A análise de características estáticas do espermatozoide referentes à sua morfologia e a análise do movimento espermático por meio da computação é uma análise mais rápida e objetiva para avaliação do macho da espécie bovina. A análise de imagens por computador oferece uma maior precisão e confiabilidade nos resultados, pois reduz a subjetividade, e consequentemente falhas que possam surgir durante a avaliação, inerentes do processo feito por análise visual, além de fornecer novas metodologias e técnicas para a avaliação da fertilidade contribuindo desta forma com melhoria da reprodução animal [10], [11], [2].

No entanto os sistemas computacionais já disponíveis apresentam algumas desvantagens que tornam seu uso limitado como o elevado custo do equipamento, a necessidade de validação, o controle de qualidade e a padronização das avaliações realizadas [11].

Com o presente trabalho objetivou-se a caracterização morfológica da cabeça de espermatozoides de touros por meio de técnicas computacionais de processamento e análise de imagem. Foi realizada uma verificação da viabilidade de se usar essa metodologia, por meio da avaliação da precisão do algoritmo de segmentação. Essa avaliação foi feita comparando-se os resultados da segmentação automática com a segmentação manual. Após a segmentação, diversas medidas que permitem caracterizar

morfologicamente as cabeças foram realizadas.

## II. Materiais e Métodos

A análise da morfologia da cabeça de espermatozoides bovinos é composta de três principais etapas: (i) Preparação das amostras e captura das imagens; (ii) Segmentação automática das imagens; e (iii) Avaliação automática das características morfológicas das cabeças.

### *A. Preparação das amostras e captura das imagens dos espermatozoides*

Foram utilizadas 25 amostras de sêmen de touro fornecidas pelo Laboratório de Histologia da Universidade Federal de Uberlândia. As amostras estavam armazenadas em solução de citrato formalado, sendo 2,9% de citrato e 10% de formol até o momento do processamento. O sêmen foi utilizado para captura das imagens para avaliação da morfologia espermática. Para análise morfológica dos espermatozoides foram realizados esfregaços das doses de sêmen sendo que essas amostras já estavam fixadas pelo citrato formalado. Após secagem ao ar, os esfregaços foram hidrolisados por 25 min., em ácido clorídrico 4N, lavados em água destilada e secos à temperatura ambiente. Então, foi colocada uma gota de azul toluidina a 0,025% em tampão Mcllvaine (citrato de sódio-fosfato) pH 4 e coberto com lamínula [9].

Após confecção dos esfregaços, fez-se a leitura por microscopia óptica utilizando a objetiva de imersão (100X). Por meio de uma câmera conectada a um computador acoplado ao microscópio, as imagens digitais das cabeças espermáticas foram capturadas (Fig. 1).

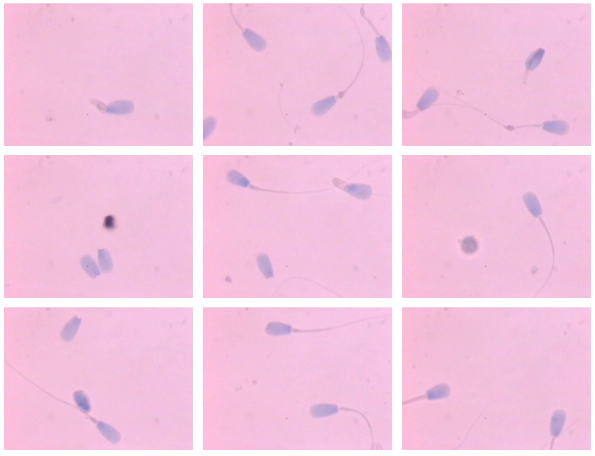


Figura 1. Imagens das cabeças espermáticas de touro obtidas após captura pela câmera acoplada ao microscópio óptico em aumento de 1000X.

### *B. Segmentação automática das imagens*

Posteriormente as imagens obtidas foram processadas realizando a segmentação [12], que neste caso significa identificar a cabeça dos espermatozoides na imagem. Essa identificação consiste em gerar uma nova imagem (Fig. 2). Nessa imagem, os pixels ou pontos em preto representam regiões que não pertencem aos espermatozoides, enquanto que esses são marcados pelos pixels em branco. O processo de segmentação é fundamental para que as medidas automáticas, descritas na próxima seção, possam ser calculadas. O método adotado para segmentação foi a limiarização, com limiar escolhido automaticamente baseando-se na análise dos picos da segunda derivada do histograma da imagem [13].

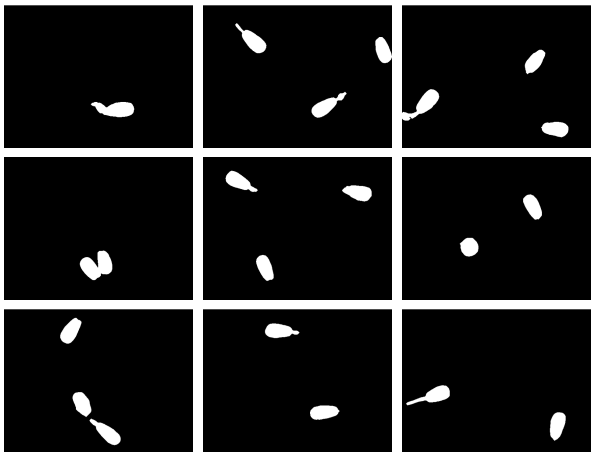


Figura 2. Resultado da segmentação automática das cabeças de espermatozoides de touro contidas na Fig. 1.

Para verificar se o processo automático de segmentação foi realizado corretamente, as cabeças dos espermatozoides de touro também foram segmentadas manualmente. Em seguida, foram comparados os dois métodos (Fig. 3). A imagem obtida pela segmentação automática (Fig. 3a) e pela segmentação manual (Fig. 3b) foram sobrepostas. O resultado dessa sobreposição gerou uma nova imagem com duas diferentes regiões (não considerando o fundo) (Fig. 3c): $R1$, que contém pixels que pertencem à cabeça do espermatozoide tanto na imagem segmentada automaticamente quanto na manual (pixels em branco na Fig. 3c); e R2, que contém os pixels da cabeça do espermatozoide somente na imagem obtida pela segmentação automática ou somente pela segmentação manual (pixels em cinza na Fig. 3c). Com isso, foi possível calcular um índice de similaridade entre a imagem original e a segmentada por meio da seguinte equação:

$$S = \frac{P(R1)}{P(R1) + P(R2)} \tag{1}$$

onde $S$ é o índice de similaridade e $P(R1)$ e $P(R2)$ indicam a quantidade de pixels que pertencem à região $R1$ e $R2$, respectivamente. Dessa forma, quando o resultado da segmentação automática for exatamente igual ao da manual, o valor de $S$ será 1. Não havendo nenhuma intersecção, o valor de $S$ será 0.

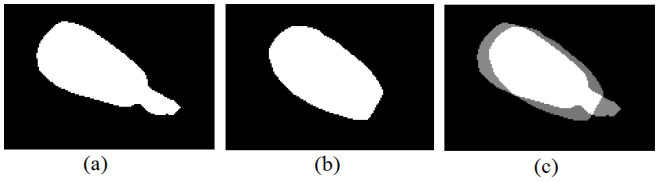



Figura 3. Segmentação das cabeças de espermatozoides de touro. (a) Resultado da segmentação automática. (b) Resultado da segmentação manual. (c) Sobreposição dos resultados para realização do cálculo do índice de similaridade $S$ (Eq. 1).

### *C. Avaliação automática das características morfológicas das cabeças*

Neste trabalho foram extraídas medidas morfológicas importantes das cabeças dos espermatozóides. As medidas foram calculadas após o processo de segmentação. Foram avaliadas área, perímetro, elongação, simetria, textura, entropia, largura e comprimento [14], [9], [5], [15], [16]. A partir desses valores é possível realizar análises que podem ajudar na determinação da existência ou não de anomalias na cabeça dos espermatozoides.

## III. Resultados e Discussão

### *A. Avaliação do índice de similaridade*

Analisando os resultados, verificamos empiricamente que uma boa segmentação foi obtida para cabeças com índice de similaridade $S$ maior que $0,80$. A Fig. 4 mostra alguns exemplos de cabeças com diferentes índices de similaridade. Para valores acima de $0,8$ a detecção funcionou bem – pode-se notar que as imagens segmentadas manualmente e as automáticas são bem semelhantes. No entanto, para valores menores que $0,8$ pode-se observar cabeças que possuem algum tipo de patologia ou então estavam sobrepostas a outras cabeças, ou apresentavam algum artefato de técnica que de alguma forma interferiu em sua morfologia, prejudicando desta forma a segmentação. Ou ainda algum artefato presente na lâmina que foi considerado como sendo uma cabeça e acabou sendo segmentado.

No total, 1228 cabeças foram consideradas, sendo que 953 (85%) obtiveram grau de similaridade maior que $0,8$ (Fig. 5). Com isso, podemos considerar que o algoritmo de segmentação automática funciona muito bem. As cabeças com similaridade menor que $0,8$ – o que indica que não foram corretamente segmentadas – poderiam ser descartadas por meio da análise de outras medidas (como, por exemplo, simetria ou área).

### *B. Análise da Morfologia da Cabeça*

Como o índice de similaridade mostrou que o algortimo de segmentação pode ser usado para detecção automática das cabeças de espermatozóides nas imagens, o próximo passo é obter medidas referentes a essas cabeças, de forma a caracterizar morfologicamente o espermatozoide. Foram calculadas, para cada cabeça, as seguintes medidas:

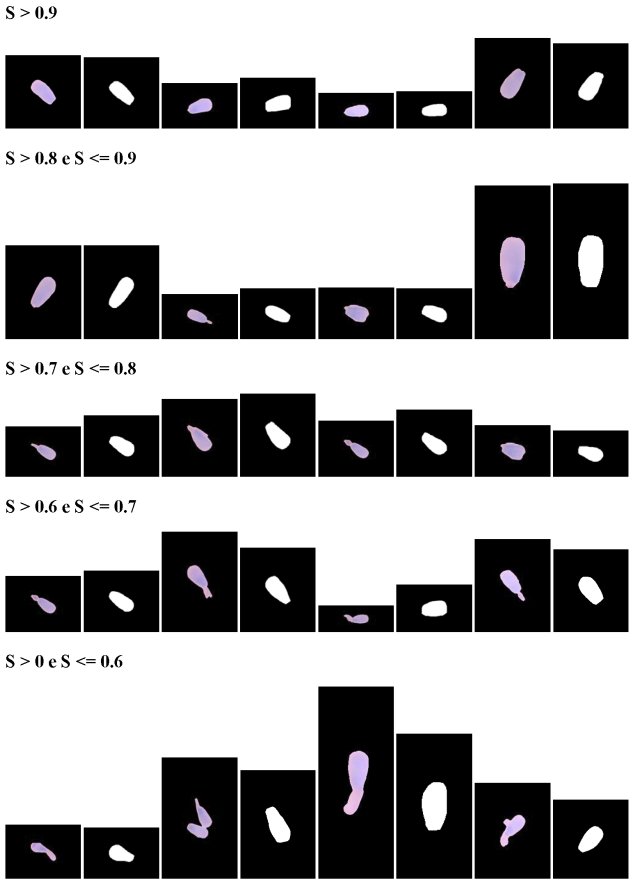



Figura 4. Exemplos de imagens com diferentes índices de similaridade. As images são apresentadas aos pares, onde a imagens à esquerda mostram o resultado da segmentação automática (mas com os pixels com intensidades da imagens originais) e as imagens à direita mostram a segmentação manual.

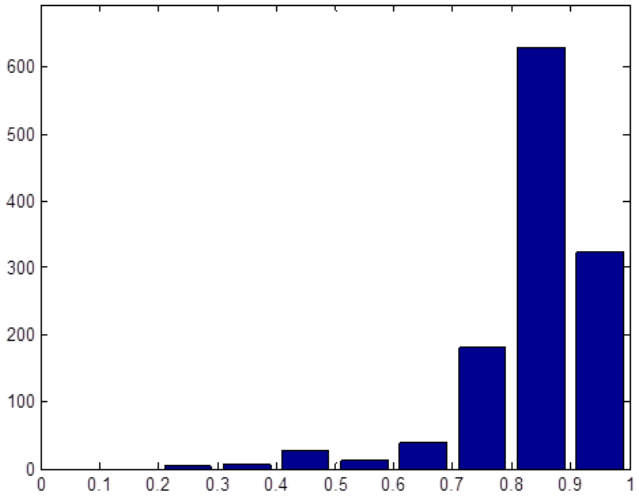



Figura 5. Grau de similaridade de todas as cabeças espermáticas analisadas.

Tabela I
Média e desvio padrão das 25 amostras analisadas, para cabeças com simetria maior que 0.9.

| Amostra | Área ($\mu$m) ($\mu$m) | Perímetro ($\mu$m) | Elongação | Simetria | Textura | Entropia | Compri–mento ($\mu$m) | Largura ($\mu$m) |
|---|---|---|---|---|---|---|---|---|
| 1 | 40,4 | 27,1 | 4,7 | 0,94 | 3,93 | 3,86 | 10,5 | 4,9 |
| 2 | 35,1 | 24,9 | 4,8 | 0,95 | 2,77 | 3,34 | 9,7 | 4,5 |
| 3 | 34,9 | 24,1 | 4,03 | 0,94 | 2,55 | 3,3 | 9,2 | 4,7 |
| 4 | 36 | 24,7 | 4,12 | 0,95 | 1,74 | 2,78 | 9,4 | 4,7 |
| 5 | 39,2 | 25,9 | 3,62 | 0,94 | 4,6 | 4,1 | 9,6 | 5,1 |
| 6 | 35,5 | 24,3 | 4,13 | 0,95 | 2,47 | 3,24 | 9,4 | 4,7 |
| 7 | 34,4 | 24,4 | 4,69 | 0,95 | 2,36 | 3,14 | 9,6 | 4,5 |
| 8 | 36,2 | 25 | 4,99 | 0,95 | 3,69 | 3,72 | 9,8 | 4,5 |
| 9 | 37,8 | 25,3 | 4,29 | 0,96 | 5,42 | 4,21 | 9,6 | 4,9 |
| 10 | 40,9 | 26,3 | 4,12 | 0,95 | 3,28 | 3,6 | 10 | 5 |
| 11 | 37 | 25,1 | 4,35 | 0,95 | 3,47 | 3,67 | 9,5 | 4,6 |
| 12 | 36,1 | 24,7 | 4,58 | 0,95 | 2,46 | 3,24 | 9,5 | 4,6 |
| 13 | 34,1 | 23,9 | 4,33 | 0,95 | 3,53 | 3,71 | 9,1 | 4,5 |
| 14 | 36,3 | 25 | 4,53 | 0,95 | 2,9 | 3,39 | 9,6 | 4,6 |
| 15 | 36,7 | 24,6 | 3,55 | 0,95 | 4,71 | 4,06 | 9,1 | 4,9 |
| 16 | 40,6 | 25,9 | 3,73 | 0,94 | 5,22 | 4,25 | 9,6 | 5,1 |
| 17 | 35,2 | 24,4 | 4,42 | 0,95 | 2,91 | 3,37 | 9,3 | 4,6 |
| 18 | 40,4 | 25,7 | 3,7 | 0,95 | 6,81 | 4,52 | 9,5 | 5,2 |
| 19 | 37,6 | 25,4 | 4,24 | 0,94 | 2,22 | 3 | 9,8 | 4,8 |
| 20 | 36 | 24,7 | 4,56 | 0,96 | 3,38 | 3,56 | 9,5 | 4,6 |
| 21 | 41 | 26,3 | 3,98 | 0,95 | 4,31 | 3,86 | 9,7 | 5,1 |
| 22 | 42,9 | 26,8 | 3,95 | 0,96 | 3,76 | 3,8 | 10 | 5,2 |
| 23 | 41,7 | 26,2 | 4,06 | 0,96 | 5,38 | 4,17 | 10 | 5,1 |
| 24 | 36 | 24,2 | 3,69 | 0,96 | 6,41 | 4,58 | 9,1 | 4,8 |
| 25 | 40 | 25,8 | 4,2 | 0,96 | 2,84 | 3,37 | 9,9 | 5 |
| **Média** | **37,68** | **25,21** | **4,21** | **0,95** | **3,72** | **3,67** | **9,6** | **4,8** |
| **Desvio** | 2,54 | 0,86 | 0,38 | 0,01 | 1,31 | 0,46 | 0,32 | 0,23 |

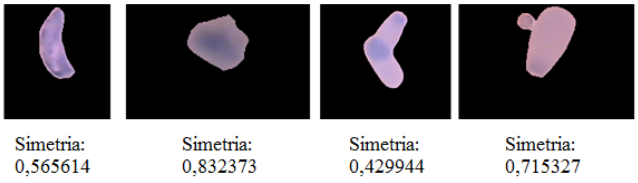



Figura 6. Exemplo de cabeças com patologias ou artefatos de técnica. Em muitos casos essas imagens podem ser eliminadas da análise morfológica por meio de uma consulta ao valor de simetria. Para imagens mostradas nesta figura, todas cabeças apresentam simetria abaixo de 90%.

área ($\mu^2$m), perímetro ($\mu$m), elongação, simetria, textura, entropia, comprimento ($\mu$m) e largura ($\mu$m). O cálculo dessas medidas serve também não só para caracterizar o espermatozoide, mas também como uma etapa de pós-processamento, pois é possível remover das análises cabeças com medidas discrepantes da média. No caso, a medida escolhida foi a simetria. A Tab. I apresenta o resultado das medidas para as 25 amostras. Foram selecionadas somente as cabeças com simetria acima de 90%, porque são as com maior chance de apresentarem morfologia normal.

Como pode ser visto na Fig. 6, cabeças com simetria abaixo de 90% possuem algum tipo de patologia, sobreposição com outras cabeças, ou presença de artefato de técnica que de alguma forma interferem na morfologia espermática, prejudicando dessa forma a segmentação.

Os valores obtidos para as medidas apresentadas na Tab. I foram semelhantes aos valores encontrados na literatura, demonstrando que a análise automática pode ser utilizada para análise da morfologia da cabeça. Vale notar que pode haver uma variação nas medidas devido à raça [15], espermatozoide normal ou defeituoso [17] e fatores extrínsecos como, por exemplo, má nutrição, traumas e variação da temperatura ambiental, que afetam direta ou indiretamente a espermatogênese. E essas variações são esperadas entre indivíduos mesmos que estes apresentem fertilidade comprovada [18].

Além disso, os métodos de fixação celular, calibração das imagens digitais, equipamentos e métodos de mensuração podem alterar a conformação ou os valores absolutos das medidas [18]. Alguns estudos relatam a possibilidade de existir uma relação entre a forma da cabeça do espermatozoide e a fertilidade de touros. Animais de maior fertilidade produziriam espermatozoides com cabeças alongadas e afiladas em comparação a touros de menor fertilidade [19].

## IV. Conclusão

Neste trabalho foi feita a caracterização morfológica das cabeças espermáticas de touros por meio da utilização de técnicas computacionais de processamento e análise de imagem. Com a utilização da análise de imagem computacional foi possível identificar diversas características da cabeça dos espermatozoides, como área, perímetro, elongação, comprimento, largura, simetria, textura e entropia. A análise da segmentação automática foi comparada com a segmentação manual, na qual foi possível avaliar a qualidade do método automático. Verificou-se que 85% das

cabeças detectadas automaticamente apresentam uma boa qualidade de segmentação.

As rotinas atuais já estão sendo utilizadas em vários projetos de pesquisa relacionados ao estudo da fertilidade bovina. Como trabalhos futuros pretende-se avaliar em detalhe a importância de cada medida usada (ou combinação de medidas) para definir novos parâmetros de fertilidade espermática.

## Agradecimentos

Os autores agradecem ao CNPq, CAPES, FAPEMIG e PROPP-UFU pelo suporte financeiro.

## Referências

[1] P. B. D. Gonçalves, J. R. FIGUEIREDO, and V. J. D. F. FREITAS, *Biotécnicas aplicadas à reprodução animal*. Editora Roca, 2008.

[2] R. Arruda, E. Celeghini, M. Alonso, H. Carvalho, L. Oliveira, J. Nascimento, D. Silva, F. Affonso, K. Lemes, and J. Jaimes, "Métodos de avaliação da morfologia e função espermática: momento atual e desafios futuros," *Revista Brasileira de Reprodução Animal*, vol. 35, no. 2, pp. 145–151, 2011.

[3] J. M. Soares and M. E. Beletti, "Avaliação da integridade cromatínica de espermatozóides de galos (gallus gallus, linnaeus, 1758) de linhagem pesada em duas idades," *Brazilian Journal of Veterinary Research and Animal Science*, vol. 43, no. 4, pp. 543–553, 2006.

[4] M. Unanian, *Integridade da cromatina: método complementar para avaliação da qualidade do sêmen bovino*. Embrapa Recursos Genéticos e Biotecnologia, 2000.

[5] M. E. Beletti, L. F. Costa, and M. Viana, "A computational approach to characterization of bovine sperm chromatin alterations," *Biotechnic & Histochemistry*, vol. 79, no. 1, pp. 17–23, 2004.

[6] C. Freitas-Dell'Aqua, A. Crespilho, F. Papa, and J. D. Junior, "Metodologia de avaliação laboratorial do sêmen congelado bovino," *Rev. Bras. Reprod. Anim*, vol. 33, no. 4, pp. 213–222, 2011.

[7] A. Contri, C. Valorz, M. Faustini, L. Wegher, and A. Carluccio, "Effect of semen preparation on casa motility results in cryopreserved bull spermatozoa," *Theriogenology*, vol. 74, no. 3, pp. 424–435, 2010.

[8] C. Garrett and H. Baker, "A new fully automated system for the morphometric analysis of human sperm heads.," *Fertility and sterility*, vol. 63, no. 6, pp. 1306–1317, 1995.

[9] M. E. Beletti and L. F. Costa, "A systematic approach to multispecies sperm morphometric characterization.," *Analytical and quantitative cytology and histology/the International Academy of Cytology [and] American Society of Cytology*, vol. 25, no. 2, pp. 97–107, 2003.

[10] J. Verstegen, M. Iguer-Ouada, and K. Onclin, "Computer assisted semen analyzers in andrology research and veterinary practice," *Theriogenology*, vol. 57, no. 1, pp. 149–179, 2002.

[11] D. Matos, A. Araújo, I. Roberto, and R. Toniolli, "Análise computarizada de espermatozóides: revisão de literatura," *Rev Bras Reprod Anim*, vol. 32, no. 4, pp. 225–232, 2008.

[12] R. Gonzalez and R. Woods, *Digital Image Processing*. Prentice Hall - USA, 3 ed., 2008.

[13] A. R. Backes, B. A. N. Travençolo, and M. C. Escarpinati, "A derivative based algorithm for image thresholding," *Submetido para publicação*, 2014.

[14] L. F. Costa and R. M. Cesar Jr, *Shape analysis and classification: theory and practice*. CRC Press, Inc., 2000.

[15] M. E. Beletti, L. F. Costa, and M. P. Viana, "A comparison of morphometric characteristics of sperm from fertile bos taurus and bos indicus bulls in brazil," *Animal reproduction science*, vol. 85, no. 1, pp. 105–116, 2005.

[16] M. E. Beletti, L. F. Costa, and M. P. Viana, "A spectral framework for sperm shape characterization," *Computers in Biology and Medicine*, vol. 35, no. 6, pp. 463–473, 2005.

[17] P. d. A. B. Miranda, C. E. dos Santos Fernandes, S. da Silva Moraes, A. R. de Oliveira, S. d. C. P. Lopes, and T. B. Amaral, "Morfometria da cabeça espermática em touros nelore submetidos a deficiência de zinco na dieta," *Embrapa Gado de Corte. Documentos*, vol. 147, 2004.

[18] D. C. Meldau, "Morfometria do núcleo espermático associada à avaliação histológica testicular em bovinos," Master's thesis, Universidade Federal de Mato Grosso do Sul, 2013.

[19] G. Freneau, "Aspectos da morfologia espermática em touros," *Rev. Bras. Reprod. Anim*, vol. 35, no. 2, pp. 160–170, 2011.

# Comparison of illumination normalization techniques for face recognition

Jonatan Patrick Margarido Oruê[1], Amaury A. Castro Junior and Wesley Nunes Gonçalves
Universidade Federal de Mato Grosso do Sul, UFMS
Ponta Porã, MS, Brasil
Email: jonatanorue@gmail.com, {amaury.junior, wesley.goncalves}@ufms.br

***Resumo*—Face recognition has become one of the most popular areas in computer vision due to its successful applications. In recent years, major advances have been reported in face recognition, and therefore the methods are becoming increasingly accurate and efficient. Despite recent advances, the performance of face recognition methods is still affected by different lighting conditions. This paper presents a comparison of three illumination normalization techniques for illumination invariant face recognition. Experimental results demonstrated that these techniques significantly improve the performance of the EigenFaces method. Particularly, the Difference of Gaussians filtering-based normalization technique provided satisfactory results even for feature vectors with low dimensionality (e.g. $n = 10$).**

## I. Introdução

Biometria é o estudo das características físicas ou comportamentais que tornam possível a identificação dos seres vivos [1]. A biometria tem sido utilizada, por exemplo, na identificação criminal e no controle de acesso, com o objetivo de identificar pessoas por meio de suas características únicas. As principais características que são exclusivas de cada ser humano incluem íris, impressão digital, face e voz. Apesar dos sistemas baseados no reconhecimento de íris, impressão digital e voz apresentarem resultados promissores nos últimos anos, eles impõem certas restrições para captura de informações. Por exemplo, para o reconhecimento de íris, o usuário deve permanecer parado em uma posição definida com os olhos abertos enquanto uma câmera captura a imagem. No reconhecimento por impressão digital, o usuário deve permitir que o sistema capture uma imagem da sua digital, inserindo o dedo no dispositivo de captura.

Por outro lado, o reconhecimento de faces possui como vantagem a característica não evasiva, isto é, a identificação ocorre mesmo sem a colaboração do usuário. Assim, este tipo de identificação é adequado para diversas situações, como a localização de pessoas desaparecidas por meio de câmeras localizadas em lugares estratégicos (aeroportos e cruzamento de avenidas) [2]. Os seres humanos possuem a incrível capacidade de identificar faces, mesmo com a constante mudança que pode haver nelas, como barba, rugas e acessórios, tais como óculos, entre outros [3]. Todavia, realizar a mesma tarefa por meio de um computador, ainda é considerado um problema em aberto devido à necessidade de construção de um modelo computacional que seja eficaz independente das mudanças citadas acima, além de outras variações, como iluminação, escala e ângulo de visão.

[1]Bolsista do Programa de Educação Tutorial PET/MEC/SESu (PET/Fronteira)

Em visão computacional, o reconhecimento de faces é um dos mais importantes campos de pesquisa devido às suas aplicações bem sucedidas em diferentes áreas, especialmente na indústria de segurança. Na literatura, existem basicamente duas abordagens para o reconhecimento de faces por imagens: holísticas [4] e locais [5]. A abordagem holística considera todos os pixels da imagem, portanto, a dimensionalidade dos dados é igual ao número de pixels das imagens. Para reduzir a dimensionalidade, são utilizados métodos para redução de características (e.g. Análise de Componentes Principais − PCA e Análise discriminante linear − LDA). Os dois principais métodos desta categoria são o *EigenFaces* [6], [7], [8] e o *FisherFaces* [4], [8], além de suas variantes [7], [9], [10]. A abordagem local extrai informações com base nas regiões estruturais da face (e.g. olhos, boca, nariz, etc), pois estas possuem informações essenciais para identificação de uma determinada face. Um dos primeiros trabalhos foi proposto por Kanade [11] para localizar e extrair automaticamente 16 parâmetros faciais tais como distâncias, áreas, ângulos entre os cantos dos olhos, boca e nariz. Métodos mais sofisticados aplicam modelos deformáveis [12], transformada de Hough [13] e características locais, como *Scale Invariant Feature Transform* [14] e Padrões Locais Binários [5].

Atualmente, uma das principais dificuldades para os métodos de reconhecimento de faces é a mudança da iluminação no momento em que a imagem da face é capturada [15], [16], [17], [18], [19]. A imagem da face pode ser captura em um ambiente controlado com iluminação adequada, mas também pode ser capturada em ambientes externos, onde a iluminação não pode ser controlada. Essa mudança afeta diretamente a qualidade da imagem, reduzindo consideravelmente o desempenho dos métodos de reconhecimento. Para contornar o problema de iluminação, alguns trabalhos propõem o uso de câmeras infravermelhas apesar do seu alto custo [20]. Uma segunda alternativa é normalizar a iluminação da imagem com técnicas de processamento de imagens [21]. Este processo auxilia a extração de características, gerando atributos mais precisos e, com isso, é obtida uma classificação mais precisa.

Neste trabalho, foram comparadas três técnicas de normalização de iluminação em imagens: *Single Scale Retinex* − SSR [22], Normalização no Domínio da Transformada Discreta de Cosseno − DCT [23] e Normalização baseada em Diferença de Gaussianas − DoG [24], [25]. Nos experimentos, as imagens foram normalizadas com cada uma das técnicas e, em seguida, a extração de características foi realizada por meio do método *EigenFace* [6], [7], [8]. Após a extração das características, estas foram classificadas com dois classi-

ficadores: *Naive Bayes* e *IBk - Instance Based k* [26], [27]. Os resultados obtidos da base *Yale Face Database B* [28], [29], mostraram que a utilização de técnicas de normalização de iluminação aumenta significativamente a porcentagem de acerto na classificação de faces, quando utilizado o método *EigenFaces* para extrair suas características.

O restante do artigo está organizado como segue. A Seção 2 descreve as técnicas de normalização de iluminação e o método de extração de características *EigenFaces*. Os experimentos e os resultados obtidos são apresentados na Seção 4. Por fim, a Seção 5 relata as conclusões e os trabalhos futuros.

## II. Teorias

Nesta seção, serão descritas as técnicas de normalização de iluminação utilizadas para realizar o reconhecimento de faces invariante a iluminação. Além disso, será descrito o método para extração de características *EigenFaces*.

### A. Normalização da Iluminação

A iluminação é um dos principais problemas no campo de reconhecimento de faces, pois dependendo da iluminação, geram-se imagens completamente diferentes para a mesma face. A iluminação afeta diretamente a extração de característica e, consequentemente, o desempenho no reconhecimento de faces. Para suavizar esses efeitos, foram propostas técnicas de normalização de iluminação para gerar um padrão de luminosidade nas imagens. As seções abaixo descrevem as técnicas utilizadas neste trabalho [21].

*1) Single Scale Retinex – SSR:* A técnica *Single Scale Retinex* foi proposta por Jobson et al. em [22]. Como a maioria das técnicas de normalização fotométricas, ela se baseia na chamada teoria *Retinex* [30]. A técnica basicamente convolui a imagem com um filtro Gaussiano em uma escala e, em seguida, subtrai o resultado da convolução com a imagem original. A técnica é baseada na Equação 1.

$$R(x,y) = \log(I(x,y)+1) - \log(F(x,y) \otimes I(x,y)) \quad (1)$$

onde $R(x,y)$ é a imagem com a iluminação normalizada, $I(x,y)$ é a imagem a ser normalizada, $\otimes$ é a operação de convolução e $F(x,y)$ é a função Gaussiana:

$$F(x,y) = ke^{-\frac{(x^2+y^2)}{c^2}} \quad (2)$$

onde $c$ é o tamanho espacial do filtro Gaussiano ($c = 15$ neste trabalho) e $k$ é o termo de normalização do filtro Gaussiano de tal forma que:

$$\int_x \int_y F(x,y)dxdy \quad (3)$$

*2) Normalização baseada em Diferença de Gaussianas – DoG:* A técnica de normalização baseada em diferença de Gaussianas baseia-se na diferença entre duas imagens convoluídas com filtros Gaussianos [24], [25]. Inicialmente, a transformada log é aplicada na imagem $I$:

$$I_l(x,y) = \log(I(x,y)+1) \quad (4)$$

Em seguida, a imagem no domínio log é convoluída com dois filtros Gaussianos $G_{\sigma_1}$ e $G_{\sigma_2}$ para $\sigma_1 < \sigma_2$ (neste trabalho $\sigma_1 = 1, \sigma_2 = 2$):

$$I_{\sigma_1} = I_l \otimes G_{\sigma_1} \quad (5)$$
$$I_{\sigma_2} = I_l \otimes G_{\sigma_2} \quad (6)$$

Por fim, a imagem com a iluminação normalizada $R$ é obtida pela diferença entre as imagens convoluídas:

$$R(x,y) = I_{\sigma_1}(x,y) - I_{\sigma_2}(x,y) \quad (7)$$

*3) Normalização no Domínio DCT:* A técnica conhecida como normalização no domínio da transformada discreta de cosseno – DCT foi proposta por Chen et al. [23]. Dado que as variações de iluminação residem principalmente na banda de faixa frequência, a remoção de componentes desta faixa reduz a influência da iluminação. A transformada DCT converte uma imagem no domínio espacial para o domínio da frequência. Portanto, os componentes de baixa frequência podem ser removidos fixando-os em zero. Inicialmente, a transformada DCT é aplicada na imagem no domínio logarítmico:

$$D = dct(\log(I(x,y)+1)) \quad (8)$$

A primeira posição da transformada DCT, que corresponde a iluminação geral das imagens, é ajustada seguindo:

$$D(0,0) = \log(\mu)\sqrt{wh} \quad (9)$$

Os autores sugerem que a constante $\mu$ seja o valor médio da intensidade dos pixels da imagem original. Em seguida, $n_b$ componentes de baixa frequência são removidos:

$$D(x,y) = 0, \text{ se } (x,y) \text{ é baixa frequência} \quad (10)$$

Após a remoção dos componentes de baixa frequência, a imagem normalizada é obtida através da transformada inversa DCT, convertendo do domínio da frequência para o domínio espacial:

$$R = dct^{-1}(D) \quad (11)$$

### B. EigenFaces

O *EigenFaces* [6], [7], [8] é um dos primeiros e mais populares métodos para extração de características faciais. Neste método, cada face é representada por um vetor com alta dimensionalidade composto pelas intensidades dos pixels da face. Dada uma imagem com $w \times h$ pixels, é obtido um vetor $\Gamma \in \Re^{w \times h}$ que armazena todos os pixels da imagem da face. Devido à alta dimensionalidade, existem muitas informações irrelevantes que não são úteis no processo de reconhecimento. Para contornar esse problema, o *EigenFaces* reduz a dimensionalidade dos vetores com base na análise de componentes principais – PCA. Este método pode ser sumarizado nos cinco passos abaixo:

1) Dadas as $M$ imagens com $w$ linhas e $h$ colunas, cada imagem é transformada em um vetor $\Gamma_i \in \Re^{w \times h}$

composto pelos $w \times h$ pixels. Assim, um conjunto $S = \Gamma_1, \ldots, \Gamma_M$ é obtido.

2) Cada vetor é subtraído do vetor médio $\Phi_i = \Gamma_i - \Psi$, onde $\Psi = \frac{1}{M}\sum_{i=1}^{M}\Gamma_i$.
3) A matriz de covariância é calculada por $C = \frac{1}{M}\sum_{i=1}^{M}\Phi_i\Phi_i^T$.
4) Os autovetores e autovalores são calculados a partir da matriz $C$ e, em seguida, os autovetores são ordenados com base nos autovalores.
5) Por fim, cada vetor é projetado $\Omega_i = U^T\Phi_i$ usando uma matriz $U$ composta pelos primeiros $n$ autovetores.

Após os passos descritos acima, as faces são representadas por um vetor $\Omega_i \in \Re^n$ com $n$ elementos. O valor $n$ é um parâmetro definido pelo usuário, e a dimensão projetada é muito menor que a dimensão inicial, isto é, $n \ll (w \times h)$.

## III. Experimentos e Resultados

Para avaliar os métodos de normalização de iluminação, foi utilizada a base de imagens *Extended Yale Face Database B* [28], [29]. Esta base disponibiliza 2.424 imagens que foram capturadas em um ambiente interno, onde a posição do ponto de iluminação foi alterada diversas vezes. São imagens de 38 pessoas com uma média de aproximadamente 64 imagens por pessoa. As imagens estão em níveis de cinza e com uma dimensão de $192 \times 168$ pixels. O grande desafio desta base são as diferentes condições de iluminação. A Figura 2 ilustra quatro amostras de oito pessoas que estão presentes nesta base.

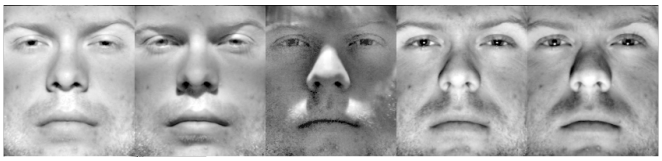
As imagens foram normalizadas com cada uma das três técnicas descritas na seção anterior. A Figura 1 apresenta o resultado da aplicação das técnicas em 5 imagens de uma mesma face com diferentes iluminações. Apesar das condições de iluminação serem desafiadoras, é possível perceber a melhora obtida nas imagens normalizadas. Após a normalização, as imagens tiveram suas características extraídas a partir da técnica *EigenFaces*, com a dimensão do vetor de características $n$ variando de 10 a 100 e com

Na etapa de classificação, os classificadores *IBk* e *Naive Bayes* foram utilizados por meio da validação cruzada com 10 dobras. Ambos são classificadores bastante utilizados, além de serem simples, evidenciando a importância das características extraídas. Os gráficos da Figura 3 apresentam a porcentagem de classificação correta obtida pelas técnicas de normalização para diferentes valores de $n$ (parâmetro do método *EigenFaces*). As Figuras 3 (a) e (b) apresentam os resultados obtidos pelos classificadores *IBk* e *Naive Bayes*, respectivamente. Além dos resultados das técnicas de normalização, os gráficos apresentam os resultados da aplicação do *EigenFaces* nas imagens originais, isto é, imagens sem a normalização. As três técnicas de normalização apresentaram resultados superiores comparado com a aplicação do *EigenFaces* nas imagens originais. Além disso, podemos notar que utilizando o classificador *Naive Bayes*, as três técnicas apresentaram resultados similares. Por outro lado, para o classificador *IBk*, a técnica *DoG* apresentou a maior taxa de acerto, seguida de *SSR* e *DCT*, nesta ordem.

Por meio dos dois gráficos, é possível notar que a porcentagem de classificação correta é incrementada à medida que $n$ é incrementado, estabilizando a partir de um determinado

(a) Imagens originais

(b) SSR



(c) DCT

(d) DoG

Figura 1. Exemplo de cinco imagens com diferentes iluminações (a) e as suas respectivas imagens normalizadas obtidas pelas técnicas SSR (b), DCT (c) e DoG (d).

valor. Entretanto, a técnica DoG se destaca mesmo para baixos valores de $n$. Para $n = 10$, a técnica DoG obteve porcentagens de $98.5974\%$ e $95.9158\%$ para os classificadores *IBk* e *Naive Bayes*, respectivamente. Como comparação, a técnica SSR obteve $81.4356\%$ e $63.8614\%$ e a técnica DCT obteve $79.1254\%$ e $65.0578\%$, para as mesmas configurações de $n$ e classificadores.

Para comparação, a Tabela I apresenta os melhores resultados obtidos por cada técnica usando o classificador *Naive Bayes* enquanto que a Tabela II mostra os melhores resultados para o classificador *IBk*. Além da porcentagem de classificação correta, a tabela também apresenta o desvio padrão, a medida-f e a área sob a curva ROC (*Receiver Operating Characteristic*), que são importantes na comparação. Esses resultados indicam que a utilização de técnicas de normalização melhora significativamente o desempenho no reconhecimento (e.g. de 84.7772% para 99.5875% para o classificador *Naive Bayes* e de 91.0066% para 99.5462% para o classificador *IBk*).

Tabela I. Resultados comparativos entre as técnicas SSR, DCT e DoG através do classificador *Naive Bayes*. A tabela apresenta o melhor resultado para cada técnica variando-se a dimensionalidade $n$.

| Método | % (Desvio Padrão) | Medida-F | Área ROC |
|---|---|---|---|
| Original | 84.7772 ($\pm$2.14) | 0.853 | 0.989 |
| SSR | 99.2162 ($\pm$0.50) | 0.992 | 0.994 |
| DCT | **99.5875** ($\pm$0.37) | 0.996 | 1 |
| DoG | 99.4637 ($\pm$0.42) | 0.995 | 1 |

Figura 2. Exemplo de oito pessoas com quatro amostras cada e em diferentes condições de iluminação. Imagens retiradas de [28], [29].

Tabela II. RESULTADOS COMPARATIVOS ENTRE AS TÉCNICAS SSR, DCT E DOG ATRAVÉS DO CLASSIFICADOR *IBk*. A TABELA APRESENTA O MELHOR RESULTADO PARA CADA TÉCNICA VARIANDO-SE A DIMENSIONALIDADE $n$.

| Método | % (Desvio Padrão) | Medida-F | Área ROC |
|---|---|---|---|
| Original | 91.0066 ($\pm$1.84) | 0.911 | 0.952 |
| SSR | 98.1848 ($\pm$0.87) | 0.982 | 0.990 |
| . DCT | 97.3597 ($\pm$1.06) | 0.974 | 0.986 |
| DoG | **99.5462** ($\pm$0.38) | 0.996 | 0.997 |

Por fim, a Figura 4 exibe as matrizes de confusão para as técnicas de normalização obtidas pelo classificador *IBk*. Nesta figura, uma técnica com bom desempenho é representada por uma matriz com a maioria dos seus valores na diagonal e poucos valores fora dela. Neste aspecto, podemos notar que a classificação nas imagens originais possuem várias faces classificadas incorretamente. Por outro lado, as classificações incorretas são drasticamente reduzidas com as técnicas de normalização, como pode ser visto nas matrizes de confusão das técnicas SSR, DCT e DoG, com destaque para a técnica DoG.

## IV. CONCLUSÃO E TRABALHOS FUTUROS

Este trabalho apresentou uma comparação entre técnicas de normalização de iluminação aplicadas no reconhecimento de faces. As técnicas SSR, DCT e DoG foram aplicadas nas imagens para normalização e, em seguida, as características foram extraídas pelo método *EigenFaces* devido a sua popularidade na comunidade. As características extraídas foram classificadas por dois classificadores populares *Naive Bayes* e *IBk*.

Os experimentos mostraram que o processo de normalização de iluminação melhora consideravelmente o reconhecimento de faces por meio do método *EigenFaces*. Aumentos de $14.8103\%$ e $8.5396\%$ foram, respectivamente, obtidos com os classificadores *Naive Bayes* e *IBk*. Entre as técnicas de normalização, a técnica DoG forneceu resultados promissores mesmo para vetores com baixa dimensão ($n = 10$).

Em trabalhos futuros, o objetivo é incluir novas técnicas de normalização de iluminação nas comparações realizadas. Além disso, pretende-se incluir outros métodos de extração de características para o reconhecimento facial, como os padrões locais binários e *FisherFaces*.

## AGRADECIMENTOS

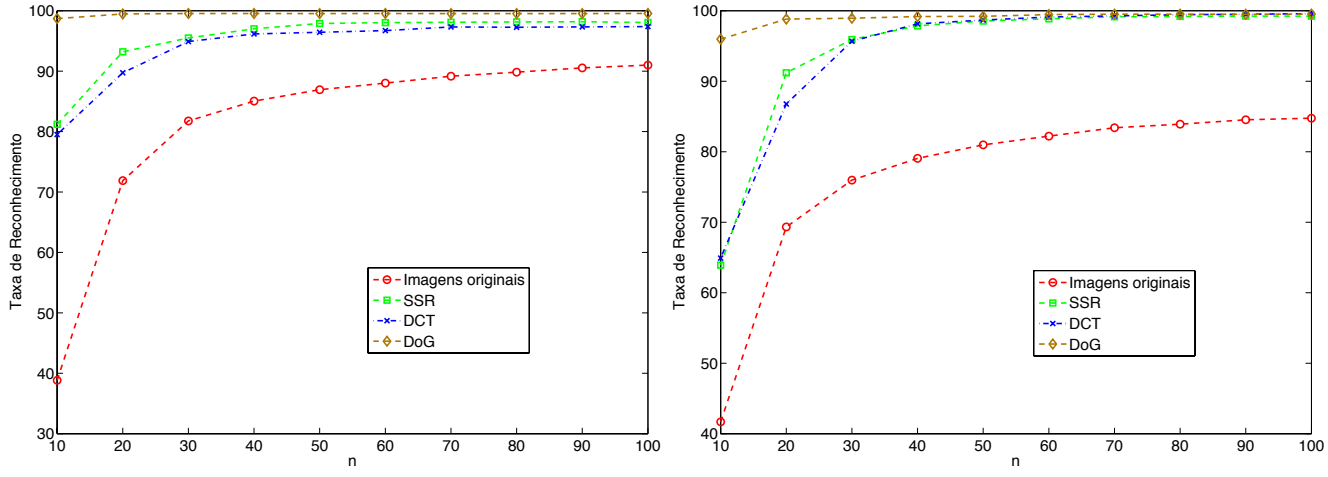
A Pró-Reitoria de Extensão, Cultura e Assuntos Estudantis (PREAE) da Universidade Federal de Mato Grosso do Sul (UFMS) e ao Programa NERDS da Fronteira pelo apoio financeiro na locomoção e estadia. Ao MEC/SESu, através do Programa de Educação Tutorial (PET/Fronteira), no qual o primeiro autor deste artigo é bolsista.

## REFERÊNCIAS

[1] A. Jain, L. Hong, and S. Pankanti, "Biometric identification," *Communications of the ACM*, vol. 43, no. 2, pp. 90–98, 2000.

[2] T. E. de Campos, "Técnicas de seleçao de caracterısticas com aplicaçoes em reconhecimento de faces," Ph.D. dissertation, Masters thesis, Universidade de Sa Paulo, 2001.

[3] R. Chellappa, C. L. Wilson, and S. Sirohey, "Human and machine recognition of faces: A survey," *Proceedings of the IEEE*, vol. 83, no. 5, pp. 705–741, 1995.




(a) IBk (b) Naive Bayes

Figura 3. Porcentagens de classificação correta obtidas pelas técnicas de normalização SSR, DCT e DoG para os classificadores *IBk* (a) e *Naive Bayes* (b). Além das técnicas de normalização, os gráficos apresentam os resultados obtidos nas imagens originais.


[4] P. N. Belhumeur, J. P. Hespanha, and D. Kriegman, "Eigenfaces vs. fisherfaces: Recognition using class specific linear projection," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 19, no. 7, pp. 711–720, 1997.

[5] T. Ahonen, A. Hadid, and M. Pietikäinen, "Face recognition with local binary patterns," in *Computer vision-eccv 2004*. Springer, 2004, pp. 469–481.

[6] M. Turk and A. Pentland, "Eigenfaces for recognition," *Journal of Cognitive Neuroscience*, vol. 3, no. 1, pp. 71–86, 1991.

[7] H.-C. Kim, D. Kim, and S. Yang Bang, "Face recognition using the mixture-of-eigenfaces method," *Pattern Recognition Letters*, vol. 23, no. 13, pp. 1549–1558, 2002.

[8] M. Smiatacz, "Eigenfaces, fisherfaces, laplacianfaces, marginfaces - how to face the face verification task," *Advances in Intelligent Systems and Computing*, vol. 226, pp. 187–196, 2013.

[9] M. Vasilescu and D. Terzopoulos, "Multilinear subspace analysis of image ensembles," in *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, vol. 2, 2003, pp. II–93–9 vol.2.

[10] D. Zhou and X. Yang, "Face recognition using direct-weighted lda," in *PRICAI 2004: Trends in Artificial Intelligence*, ser. Lecture Notes in Computer Science, C. Zhang, H. W. Guesgen, and W.-K. Yeap, Eds., vol. 3157. Springer Berlin Heidelberg, 2004, pp. 760–768.

[11] T. Kanade, *Picture Processing System by Computer Complex and Recognition of Human Faces*. Department of Information Science, Kyoto Univ., 1973.

[12] G. Edwards, T. Cootes, and C. Taylor, "Face recognition using active appearance models," in *ECCV'98*, ser. Lecture Notes in Computer Science, H. Burkhardt and B. Neumann, Eds., vol. 1407. Springer Berlin Heidelberg, 1998, pp. 581–595.

[13] A. PIETROWCEW, "Face detection in colour images using fuzzy hough transform," *OPTO-ELECTRONICS REVIEW*, vol. 11, no. 3, pp. 247–251, 2003.

[14] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[15] Y. Wu, Y. Jiang, Y. Zhou, W. Li, Z. Lu, and Q. Liao, "Generalized weber-face for illumination-robust face recognition," *Neurocomputing*, vol. 136, pp. 262–267, 2014.

[16] T. Wannakijmongkol, I. Khornrakhun, and T. H. Chalidabhongse, "An improved adaptive discriminant analysis for single sample face recognition," in *Computer Science and Software Engineering (JCSSE), 2014 11th International Joint Conference on*. IEEE, 2014, pp. 7–11.

[17] S. Shylaja, K. B. Murthy, and S. Natarajan, "Illumination invariant novel approaches for face recognition," 2014.

[18] H. Han, S. Shan, X. Chen, and W. Gao, "A comparative study on illumination preprocessing in face recognition," *Pattern Recognition*, vol. 46, no. 6, pp. 1691–1699, 2013.

[19] S. Li, R. Chu, S. Liao, and L. Zhang, "Illumination invariant face recognition using near-infrared images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 4, pp. 627–639, 2007.

[20] S. Serhan, M. Moussa, M. Naura, and A. Obeidat, "An infrared based facial recognition system using eigenfaces approach," *Advances in Modelling and Analysis B*, vol. 51, no. 1-2, pp. 34–48, 2008.

[21] V. Štruc and N. Pavešić, *Photometric normalization techniques for illumination invariance*. IGI-Global, 2011, pp. 279–300.

[22] D. Jobson, Z. Rahman, and G. Woodell, "Properties and performance of a center/surround retinex," *IEEE Transactions on Image Processing*, vol. 6, no. 3, pp. 451–462, 1997.

[23] W. Chen, M. J. Er, and S. Wu, "Illumination compensation and normalization for robust face recognition using discrete cosine transform in logarithm domain," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 36, no. 2, pp. 458–466, 2006.

[24] S. Ajagamelle, M. Pedersen, and G. Simone, "Analysis of the difference of gaussians model in image difference metrics," 2010, pp. 489–496.

[25] V. Vonikakis, D. Chrysostomou, R. Kouskouridas, and A. Gasteratos, "A biologically inspired scale-space for illumination invariant feature detection," *Measurement Science and Technology*, vol. 24, no. 7, 2013.

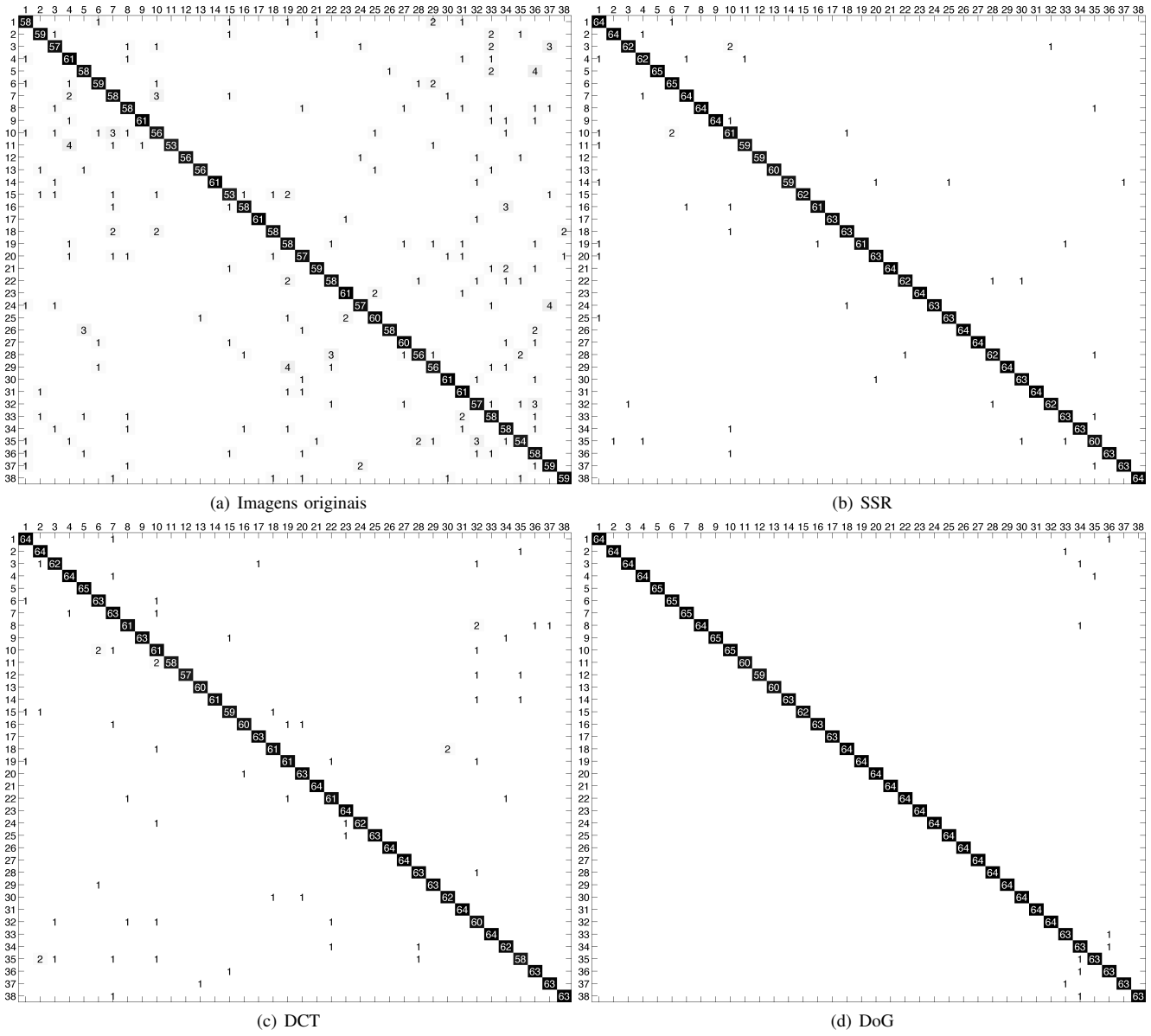
[26] D. W. Aha, D. Kibler, and M. K. Albert, "Instance-based learning algorithms," *Machine learning*, vol. 6, no. 1, pp. 37–66, 1991.

[27] G. H. John and P. Langley, "Estimating continuous distributions in bayesian classifiers," in *Proceedings of the Eleventh conference on Uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc., 1995, pp. 338–345.

[28] K.-C. Lee, J. Ho, and D. Kriegman, "Acquiring linear subspaces for face recognition under variable lighting," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 27, no. 5, pp. 684–698, 2005.

[29] A. S. Georghiades, P. N. Belhumeur, and D. Kriegman, "From few to many: Illumination cone models for face recognition under variable lighting and pose," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 23, no. 6, pp. 643–660, 2001.

[30] N. Wu and A. Caplier, "Illumination-robust face recognition using retina modeling," *Proceedings of ICIP*, pp. 3289–3292, 2009.



(a) Imagens originais

(b) SSR

(c) DCT

(d) DoG

Figura 4. Matrizes de confusão obtidas pelo classificador *IBk* nas imagens originais (a) e com a iluminação normalizada pelas técnicas SSR (b), DCT (c) e DoG (d).

# Bone age estimation

## Results obtained using *Anacarp* software versus *Artificial Neural Networks* methodology

André Henrique Silva, Celso Olivete Júnior, Ronaldo Celso Messias Correia, Rogério Eduardo Garcia
Departamento de Matemática e Computação – Faculdade de Ciências e Tecnologia, Universidade Estadual Paulista (UNESP) – CEP 19060-900 – Presidente Prudente – SP – Brasil
{olivete, ronaldo, rogerio}@fct.unesp.br

***Abstract*— INTRODUCTION: The degree of development of a child is often evaluated using indicators that reflect the physical changes that occur in individual maturation process. Several areas of the skeleton have been used to estimate skeletal maturation, with an x-ray of hand and wrist more frequently used due to the large amount of bones and epiphyses, which undergo changes at different times and speeds, located in an area not too extensive. AIM: Presents a confrontation between the results obtained in the process of bone age estimation using *Anacarp* software *versus* a methodology based on Artificial Neural Networks (ANN). METHODS: Carpal images of 685 children and adolescents aged 6-16 years for bone age estimation were used. In *Anacarp* software was used ER5 and ER3 methods. In the RNA methodology used to network multilayer perceptron with backpropagation learning algorithm. RESULTS: The results obtained in the methodologies were consistent with the average medical reports based on the three classical methods - Greulich & Pyle, Tanner & Whitehouse and Eklof & Ringertz. The verification was performed using the paired Student t test with the range of significance of 5% and, in most cases, there were no statistically significant differences (p <0.05). CONCLUSION: Analyzing the results, it is concluded that both methods obtained results consistent with the medical report.**

***Keywords*— bone age estimation, artificial neural network, image processing***

### I. INTRODUÇÃO

O grau de desenvolvimento de uma criança é frequentemente avaliado utilizando-se indicadores que refletem as mudanças físicas que se produzem no indivíduo em processo de maturação. Avalia-se, por exemplo, a ossificação das epífises e posterior fusão com as diáfises, o início do pico de velocidade de crescimento e a aparição de caracteres sexuais secundários. Estes indicadores obedecem a uma sequência razoavelmente constante de aparecimento, porem as idades cronológicas nas quais são atingidos variam consideravelmente entre os indivíduos, sendo também influenciados pelo gênero, raça, fatores genéticos, ambientais, socioeconômicos e hormonais.

A avaliação em conjunto destes indicadores fornece uma medida da idade biológica do indivíduo, a qual é de grande utilidade na odontologia, pois a idade cronológica nem sempre coincide com o estágio de maturação em que o paciente se encontra e por isso a idade cronológica não é um bom parâmetro para avaliar o estágio de maturação óssea de um indivíduo.

Na ortodontia, a estimativa do grau de maturação do indivíduo constitui um auxiliar valioso na avaliação do potencial de crescimento do paciente durante o tratamento, sendo importante na determinação do pico de velocidade de crescimento puberal, da velocidade de crescimento e da previsão de quando ocorrerá o chamado surto. Essas informações significam maior segurança durante o diagnóstico e planejamento mais adequado para o caso. Através da estimativa da maturação óssea e da velocidade de crescimento haverá uma maior possibilidade para que o início do tratamento coincida com o período de máximo crescimento facial e a influência desse crescimento se faça presente de maneira marcada na terapia.

Várias áreas do esqueleto têm sido utilizadas para estimar a maturação esquelética: o pé, o tornozelo, o quadril, o cotovelo, a mão e o punho e as vértebras cervicais. A radiografia de mão e punho tem sido a mais frequentemente utilizada, devido à grande quantidade de ossos e epífises, que sofrem mudanças em diferentes tempos e velocidades, localizados em uma área não muito extensa.

Entre os métodos desenvolvidos para estimar o grau de maturação do indivíduo em radiografias de mão e punho destacam-se: Greulich & Pyle (GP), que faz uma avaliação inspecional dos ossos da mão através de um Atlas [1;2]; o método de Tanner & Whitehouse (TW), que analisa 20 ossos da mão e punho atribuindo um escore específico para cada osso e sexo, e através da soma desses escores obtém-se a idade óssea [3;4;5]; e o método Eklof & Ringertz (ER), o qual se baseia em medidas de dimensão de 10 centros de ossificação, formados por ossos do carpo, largura da epífise distal do radio, comprimento e largura do capitato e do hamato, comprimento dos metacarpais II, III e IV, e comprimento das falanges proximais II e III [2;4]. Este método apresenta duas tabelas, uma para cada sexo, onde cada centro de ossificação possui um padrão mínimo e um máximo de medida. A idade óssea é calculada através de um valor médio das idades obtidas para cada centro de ossificação. Uma descrição mais detalhada destes métodos e dos processos de estimação pode ser obtida no portal de informações de idade óssea (www.fct.unesp.br/carpal).

Encontram-se alguns trabalhos nesta linha de pesquisa, dentre eles pode-se citar o de Haiter et al. [1], que desenvolveram uma metodologia de estimação da idade óssea através da automatização do método de TW. O trabalho de Haiter et al. [1] realiza um estudo com o objetivo de verificar

se os métodos de GP, TW e ER poderiam ser aplicados à população brasileira e qual destes métodos seria o mais confiável, quando comparados à idade cronológica do indivíduo.

O método de ER é um método baseado em procedimentos computacionais, nos quais realizam tarefas, como por exemplo, obtenção das dimensões dos centros de ossificação utilizados no processo de estimação. Todavia, antes de realizar esse procedimento, necessita-se segmentar a imagem, separando os ossos do tecido da mão e do fundo da imagem. Essa etapa é conhecida por segmentação, responsável por atuar no processo de extração dessas características [1]. Esse processo pode ser realizado utilizando os métodos de detecção de bordas, responsáveis por detectar e definir o contorno dos objetos presentes na imagem [3;4;5], onde neste estudo o objeto é uma imagem radiográfica da mão e as bordas definem as regiões de interesse (centros de ossificação). O método de contornos ativos *Snakes* foi a técnica selecionada para atuar na fase de extração de características dos centros de ossificação. Foi adotado este modelo por ser uma das abordagens mais vigorosas encontradas na literatura para a etapa de segmentação, oferecendo um poderoso método que mistura geometria, física e teoria de aproximação. Este modelo é caracterizado por tentar ajustar uma curva fechada sobre as bordas dos objetos de interesse pertencentes à imagem [3;5;6].

O presente trabalho apresenta resultados do processo de estimação utilizando a plataforma *Anacarp* [7,11] (presente em www.fct.unesp.br/carpal) que contém uma rotina, baseada em modelos de contornos ativos – *Snakes* [8;9], capaz de segmentar e extrair informações das regiões de interesses de imagens carpais. Essas informações, posteriormente, alimentam uma outra rotina responsável por estimar a idade óssea, utilizando simplificações do método de ER. Por fim, a plataforma informa qual a discrepância entre a idade óssea (IO) obtida e a média dos laudos médicos (obtido através dos métodos de ER, GP e TW). Na sequencia, traz os resultados alcançados com a metodologia RNA [10], que se utiliza da rede *perceptron* multicamadas com o algoritmo de aprendizagem *backpropagation* para atuar no processo de estimação da idade óssea. Por fim, é apresentado, em forma de gráficos, um estudo comparativo entre os resultados obtidos com as metodologias.

## II. MATERIAIS E MÉTODOS

### A. Banco de imagens

Para a realização desta pesquisa foram selecionadas 631 imagens radiográficas da mão esquerda obtidas junto à Faculdade de Odontologia de Piracicaba – UNICAMP. Os pacientes (303 do sexo masculino e 328 do sexo feminino), faixa etária entre 6 e 16 anos (72 e 203 meses, respectivamente), foram divididos em 11 grupos de acordo com a idade óssea e o sexo. A classificação foi realizada baseando-se nos laudos obtidos com os métodos de GP, TW e ER, e, com o auxílio da ferramenta *Anacarp* [7,11]. Essas imagens estão disponíveis no Portal de Informações da Idade Óssea – www.fct.unesp.br/carpal – e, na Fig.1 é ilustrada a distribuição geral das imagens de acordo com o sexo.

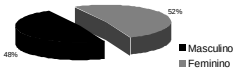



Fig. 1. Distribuição dos indivíduos da amostra de acordo com o sexo.

As análises deste estudo foram conduzidas baseando-se em um valor médio dos laudos, obtidos através da média aritmética dos métodos clássicos de GP, TW e ER. Na Fig. 2 é apresentada a distribuição do conjunto de imagens considerando os 11 grupos etários de acordo com esse valor médio e para sexo.

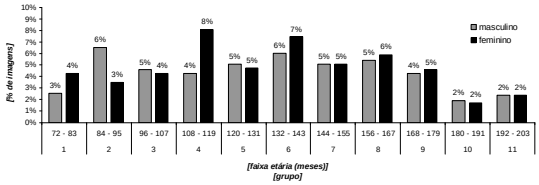



Fig. 2. Distribuição das imagens de acordo com a idade óssea.

### B. Segmentação das imagens carpais utilizando Snakes

O método tradicional de *Snakes* foi proposto por Kass, Witkin e Terzopoulos [5], onde uma *Snake* se inicia em uma configuração mais ou menos arbitrária, através de uma curva inicial que evolui até contornar o objeto de interesse. A evolução do contorno ocorre de forma a tentar minimizar a energia funcional [5], que basicamente é dado pela combinação das energias da curva ou energia interna e da imagem ou externa. Devido a estas forças é que a *Snake* pode se deformar até encontrar a borda do objeto, que neste caso são as regiões de interesse, formada por metacarpos e falanges [2]. A energia interna considera aspectos físicos como elasticidade, responsável por deformar a curva sob a ação de uma força específica e aspectos de rigidez, que é responsável por garantir a sua suavização. Por sua vez a energia externa considera as características intrínsecas da própria imagem, sendo responsável por fazer a *Snake* se guiar por regiões mais escuras na imagem, até a sua conformidade [6;8].

### C. O Modelo Discreto de Snakes

Uma definição em termos computacionais retrata uma *Snake* como um conjunto de N pontos de controle, ou *snaxels*, cujas posições, ilustradas na equação 1 são interligadas e variam no decorrer do tempo [6].

$$\{v_i(t) = (x_i(t), y_i(t)), i = 0, k, N-1\} \quad (1)$$

O modelo de energias utilizado no desenvolvimento desta ferramenta de pré-processamento é baseado na *Snake* tradicional e possui a característica de atuar sobre a curva de maneira simples. Desenvolveu-se a deformação da *Snake* tanto para processos de contração quanto para expansão, possuindo a característica de manter, da melhor forma possível, as *Snaxels* com critérios de espaçamento equivalentes [9].

Para a configuração inicial da *Snake* são considerados dois tipos de polígonos fechados: circular e definido pelo usuário. Assim, a criação da *Snake* por intermédio de um polígono circular é realizada por uma equação que descreve um círculo,

dado um ponto e seu raio. Para esta configuração também é necessário especificar a quantidade de *Snaxels*, que serão dispostos pela curva de maneira eqüidistante. A configuração por um polígono definido pelo usuário descreve a curva inicial pela interligação de *Snaxels* escolhidas manualmente. No entanto, a curva inicial depende de uma combinação de energias para proporcionar a sua evolução até as bordas de interesse. A deformação da Snake é feita através da ação da energia interna normal, de tensão, rigidez e da energia externa, iterativamente sobre todos os *Snaxels* [6;8;9].

A energia interna normal e de tensão agem de forma conjunta para movimentar e manter um espaçamento uniforme entre os *Snaxels*. O processo de contração da curva é exemplificado na Fig. 3.

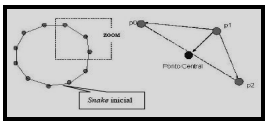



Fig. 3. Processo de atuação da Snakes: no ponto p1: a deformação é iterativa e ocorre deslocando o ponto analisado para um novo ponto central, que é o produto vetorial entre os Snaxels de p0p1 e p1p2.[9].

A força interna de rigidez relaciona-se à suavidade da *Snake*, sendo descrita através de *Splines* sobre as *Snaxels*. Por fim, a energia externa é definida em função das características de interesse na imagem, neste caso as bordas [9].

Fig. 4. Processo de atuação do modelo de *Snakes* na segmentação de centros de ossificação.

Utilizou-se este modelo para atuar no processo de extração das informações presentes nos centros de ossificação formados pelos metacarpais II, III e IV e pelas falanges proximais II e III. Esses ossos foram escolhidos por fazerem parte do método clássico de ER [4;11]. Um exemplo da atuação desse modelo é representado pela Fig. 4.

Como pode ser observado (Fig. 4) o método computacional *Snakes* foi capaz de extrair as regiões de interesse – falanges – integrantes do processo de estimação da IO. A seguir é apresentado o processo de estimação via *Anacarp*.

### *D. Processo de estimação da IO via plataforma Anacarp*

A plataforma *Anacarp* [7,11] baseia-se no método de ER [4;11] para estimar a IO. Uma das dificuldades encontradas neste processo está relacionada com a sobreposição dos ossos do carpo e punho, causando dificuldades no processo de obtenção de suas medidas, o que prejudica a obtenção precisa das medidas dos centros de interesse e pode, por consequência, produzir resultados pouco confiáveis. Visando sanar este problema buscou-se excluir estes ossos do processo de estimação, propondo-se dessa forma uma simplificação do método ER, denominada ER5.

### *D.1. Simplificação ER5*

Nesta simplificação, a estimação da idade óssea é realizada baseando-se na análise de apenas 5 centros de ossificação (ER5), compostos por duas falanges proximais e três ossos metacarpais, descartando os ossos do carpo e punho (ver Fig.5).

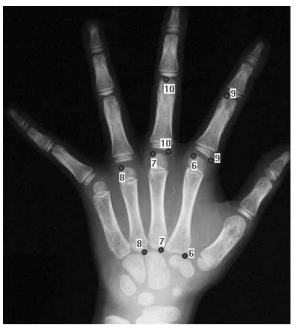



Fig. 5. Centros de ossificação utilizados no método ER5 - (6), (7) e (8): comprimento dos metacarpais II, III e IV, (9) e (10): comprimento da falange proximal II e III.

Posteriormente, realizou-se um estudo estatístico objetivando uma nova simplificação para o método ER, sendo assim elaborado o método ER3.

### *D. 2. Simplificação ER3*

Buscou-se verificar o limite mínimo do número de centros ósseos que apresentassem resultados estatisticamente significantes, ou seja, próximos ao valor da média dos laudos médicos. Foram feitas todas as combinações possíveis entre os 5 ossos (tomados 2 a 2, 3 a 3 e média dos 5 ossos) utilizados no ER5, comparando-as com a média dos laudos médicos. A denominação para os ossos utilizada nas combinações, foi: P1, P2 e P3 – comprimento dos metacarpais II, III e IV; P4 e P5 – comprimento das falanges proximais II e III. Em seguida, foram gerados gráficos para cada imagem do conjunto (Fig. 6), analisando quais combinações mais se aproximaram do valor do laudo médico.

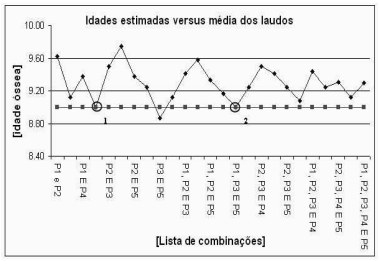



Fig. 6. Gráfico das combinações em relação à media dos laudos médicos

A linha traçada no gráfico refere-se ao valor médio dos laudos médicos (ER, GP e TW). Os pontos selecionados no gráfico acima foram as combinações: P1 P5 e P1 P3 P5, representados pelos círculos. A média da combinação (P1, P3 e P5 – formados pelos ossos em destaques na Fig. 7) obteve excelente aproximação em relação ao valor médio dos laudos médicos, indicando uma simplificação para o método ER.

Fig. 7. Os três centros de ossificação utilizados no método ER3 - (6) e (8): comprimento dos metacarpais II e IV, (10): comprimento da falange proximal III.

A seguir são apresentados os passos necessários para a realização do processo de estimação da IO.

### *D. 3. Passos da metodologia de estimação da idade óssea via Anacarp*

Na Fig. 8 são apresentados os passos para estimação da idade óssea. A seguir são explicadas, de forma sintetizada, cada uma das etapas. A Fig. 6 representa o resultado após a realização de todas as etapas deste processo.

Fig. 8. Sequência de processamento utilizado na metodologia

1) Pré-processamento

Para eliminar as variações da não uniformidade encontrada no fundo da imagem, inerente ao processo de obtenção das radiografias, foi aplicado um algoritmo desenvolvido por Nascimento et al. [12] e apresentado no trabalho de Olivete e Rodrigues [2] para correção do efeito *Heel* e um filtro passa-baixa para suavizar os ruídos.

2) Segmentação dos ossos

Uma forma simples de se isolar os ossos da mão pode ser conseguida com a utilização de um método eficiente de thresholding [12]. Os métodos implementados foram: Otsu, Pun, Niblack e Rosenfeld [13;14;15;16].

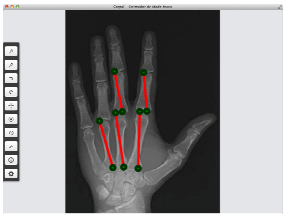


Fig. 8. Os três centros de ossificação utilizados no método ER3 - comprimento dos metacarpais II e IV e comprimento da falange proximal III.

3) Isolamento dos centros de ossificação

Para selecionar os dedos da mão, construiu-se um procedimento que realiza varreduras horizontais na imagem, guiando-se pelo pixel com intensidade 255 (cor do fundo). Posteriormente, inseriram-se marcadores para isolar os ossos metacarpais II, III e IV e as falanges proximais II e III.

4) Medidas para os centros de ossificação

Após a inserção dos marcadores, é calculado o comprimento, em pixels, de cada osso utilizando a distância Euclidiana. Para ser utilizado pelo método ER esse valor é convertido para milímetros.

5) Estimação da idade óssea

Para cada medida encontrada, consulta-se a tabela de medidas para identificação do valor referente à idade [4;7]. A idade final é obtida através da média da idade obtida para todos os centros de ossificação.

### *E. Processo de estimação da IO via RNA*

A primeira etapa desta metodologia é realizar a padronização do contraste e aplicação da filtragem mediana, de modo que os ossos e o tecido fiquem visíveis e os ruídos amenizados. Posteriormente é feita a extração de características. Para tanto, foi criado um sistema semi-automático para calcular a distância Euclidiana dos centros de ossificação, onde o usuário determina os pontos onde a distância será calculada [10].

A etapa de estimação da IO baseia-se em um método formado por uma rede neural *perceptron* multicamadas com o algoritmo de aprendizagem *backpropagation* [17]. Foram utilizadas 10 entradas na rede e acrescentado o *bias* para uma melhor estabilidade. A rede é composta por 4 camadas, sendo a primeira constituída de 11 neurônios, a segunda e a terceira com 13 neurônios e a camada de saída com 1[7]. Para o treinamento da rede foi utilizado o termo *momentum* com fator de 0.9, taxa de aprendizagem de 0.2 e precisão de $10^{-6}$ . O número (em média) de épocas de treinamento foi de 2000 para que a rede convergisse.

Para a composição da matriz de treinamento foram utilizados os dados dimensionais do Atlas de ER [4], da faixa etária entre 5 e 15 anos, compondo um total de 100 padrões de treinamento [10]. A Tabela 1 traz parte desta matriz.

TABELA 1. EXEMPLO DOS DADOS UTILIZADOS NA MATRIZ DE TREINAMENTO

| X1 | X2 | X3 | X4 | X5 | X6 | X7 | X8 | X9 | X10 | Saida |
|---|---|---|---|---|---|---|---|---|---|---|
| 22,65 | 17,7 | 11,5 | 14,15 | 12,05 | 53,15 | 50,05 | 44,05 | 31,55 | 35 | **9** |
| 22,7 | 17,8 | 11,6 | 14,2 | 12,1 | 53,4 | 50,3 | 44,4 | 31,6 | 35,2 | **9,1** |
| 22,75 | 18,9 | 11,65 | 14,35 | 12,2 | 53,6 | 50,65 | 44,5 | 31,75 | 35,3 | **9,2** |
| 23,1 | 18,1 | 11,72 | 14,5 | 12,3 | 53,85 | 50,75 | 44,6 | 31,95 | 35,4 | **9,3** |
| 23,38 | 18,2 | 11,8 | 14,6 | 12,4 | 54,2 | 51 | 44,9 | 32,2 | 35,7 | **9,4** |

Para o processo de treinamento foi necessária a normalização dos dados, pois como a função de ativação utilizada foi a sigmoide, os dados devem estar entre zero e um para que o treinamento possa ser executado.

Após o treinamento da rede, foi realizado um teste com as características das imagens do banco de imagens, sendo que todas apresentavam um laudo do radiologista. A distribuição dos dados na matriz foi de acordo com a dimensão do osso específico, e maneira que as medidas obtidas de imagens diferentes ficassem na mesma coluna, ou seja, cada comprimento é um dado de entrada, assim como na matriz de treinamento, porém os dados são diferentes, pois são obtidos

de imagens radiográficas de pacientes. A saída da rede corresponde a idade óssea do paciente (ver Fig. 9).

Outras topologias também foram testadas, porém não apresentaram resultados satisfatórios, a convergência não era realizada com sucesso [10].

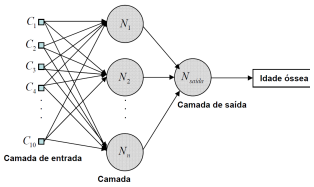



Fig. 9. Rede neural proposta para estimar a IO

### *F. Análise dos Resultados*

Após a obtenção dos resultados das idades ósseas, utilizando a plataforma *Anacarp* e a metodologia *RNA*, os mesmos foram tabulados e analisados estatisticamente utilizando o programa Microsoft Excel e o software *BioEstat* 5.0 [18]. Analisou-se a confiabilidade das metodologias (ER5, ER3 e RNA) a partir de uma confrontação com os valores médios dos laudos, através de gráficos de diagramas de dispersão (que representam a correlação entre idades estimadas e o valor médio dos laudos) e coeficientes de determinação ($R^2$). Posteriormente, geraram-se fatores de correção (equações de regressão linear). Considerou-se o nível de significância de 5%. A seguir são apresentados esses resultados.

## III. Resultados e Discussões

As metodologias para estimação da IO – via *Anacarp* e *RNA* - foram aplicadas em todos os grupos de indivíduos e para ambos os sexos. Na Tabela 2, são apresentados os dados referentes à média, desvio-padrão e intervalo de confiança (5% de significância) para as imagens do sexo masculino.

TABELA 2. MÉDIA, DESVIO-PADRÃO E INTERVALO DE CONFIANÇA PARA A MÉDIA DOS LAUDOS *VERSUS* METODOLOGIA *ER5*, *ER3* E *RNA* - SEXO MASCULINO.

| *Grupo* | *Faixa etária (meses)* | *Média dos laudos* | | | *ER5* | | | *ER3* | | | *RNA* | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | *Média* | *DP* | *IC (LI - LS)* | *Média* | *DP* | *IC (LI - LS)* | *Média* | *DP* | *IC (LI - LS)* | *Média* | *DP* | *IC (LI - LS)* |
| 1 | 72 - 83 | 76,86 | 5,41 | 74,21 - 79,51 | 78,38 | 5,88 | 74,30 - 82,45 | 73,50 | 2,78 | 71,58 - 75,42 | 77,25 | 5,01 | 73,78 - 80,72 |
| 2 | 84 - 95 | 89,71 | 5,71 | 88,80 - 90,62 | 93,28 | 5,25 | 89,74 - 96,81 | 88,83 | 8,73 | 83,84 - 93,83 | 97,61 | 7,12 | 95,71 - 99,51 |
| 3 | 96 - 107 | 99,95 | 3,33 | 98,73 - 101,16 | 101,05 | 6,62 | 98,95 - 103,16 | 89,84 | 7,67 | 87,40 - 92,28 | 103,42 | 5,96 | 101,53 - 105,32 |
| 4 | 108 - 119 | 111,74 | 2,91 | 110,64 - 112,83 | 113,68 | 4,11 | 112,16 - 115,20 | 102,32 | 4,92 | 100,50 - 104,14 | 114,32 | 6,17 | 112,04 - 116,61 |
| 5 | 120 - 131 | 127,08 | 3,41 | 125,90 - 128,26 | 126,70 | 5,47 | 125,06 - 128,33 | 116,72 | 6,80 | 114,69 - 118,75 | 121,47 | 4,08 | 120,25 - 122,68 |
| 6 | 132 - 143 | 136,25 | 2,15 | 135,57 - 136,94 | 138,60 | 4,64 | 137,24 - 139,96 | 128,13 | 5,99 | 126,38 - 129,88 | 137,00 | 3,75 | 135,90 - 138,10 |
| 7 | 144 - 155 | 147,90 | 2,43 | 147,06 - 148,74 | 148,89 | 5,05 | 147,21 - 150,56 | 143,74 | 8,21 | 141,02 - 146,46 | 152,74 | 4,81 | 151,15 - 154,34 |
| 8 | 156 - 167 | 160,27 | 2,73 | 159,35 - 161,19 | 162,15 | 4,04 | 159,00 - 165,31 | 156,69 | 8,44 | 154,04 - 159,34 | 159,38 | 5,71 | 157,59 - 161,18 |
| 9 | 168 - 179 | 170,06 | 1,82 | 169,38 - 170,75 | 172,14 | 8,19 | 169,16 - 175,12 | 169,03 | 3,95 | 167,60 - 170,47 | 171,93 | 6,28 | 169,65 - 174,22 |
| 10 | 180 - 191 | 184,26 | 2,07 | 183,09 - 185,43 | 185,81 | 6,52 | 182,62 - 189,01 | 177,75 | 5,42 | 175,09 - 180,41 | 185,06 | 3,28 | 183,46 - 186,67 |
| 11 | 192 - 203 | 193,86 | 1,81 | 192,94 - 194,78 | 191,75 | 0,87 | 191,26 - 192,24 | 168,00 | 6,42 | 160,41 - 175,59 | 171,00 | 4,40 | 168,51 - 173,49 |
| | Total | 136,18 | 3,07 | 135,06 - 137,29 | 137,49 | 5,15 | 135,15 - 139,83 | 128,60 | 6,30 | 125,69 - 131,51 | 135,56 | 5,14 | 133,60 - 137,53 |

*DP: Desvio-padrão; IC: intervalo de confiança (95%); LI: limite inferior; LS: limite superior*

Os dados apresentados na Tabela 2 mostram que para todos os grupos as idades estimadas pelo *Anacarp* – métodos ER5 e ER3 – e pela metodologia *RNA* quando comparados com os valores das médias dos laudos são semelhantes entre si. Isso pode ser constatado observando o valor médio obtido para o desvio-padrão – *DP* (3.07 na média dos laudos, 5.15 na ER5, 6.30 com a ER3 e 5.14 na metodologia *RNA*) onde essa variação foi de 2.08 meses, 3.23 meses e 2.07 meses, respectivamente (ER5, ER3 e *RNA*). Considerando o valor médio das idades ocorreu uma superestimativa de 1.32 meses para ER5 em relação à média dos laudos e subestimativas de 7.58 meses para ER3 e 0.61 para a metodologia RNA. Os intervalos de confiança (*IC*), que expressam o relacionamento entre as idades estimadas por ER5, ER3 e RNA e a média dos laudos médicos, foram gerados com base na significância estatística de 5%. Como se pode observar, os valores não apresentaram variações muito significantes (intervalos de confiança apresentaram tamanho reduzido) para todos os grupos. Para ilustrar esses comportamentos, gerou-se um gráfico (Fig. 10) que demonstra a correlação entre as médias dos laudos médicos (obtidos por GP, TW e ER) e as idades estimadas pela plataforma *Anacarp* (métodos ER5 e ER3) e a metodologia *RNA*, com distinção entre os grupos (faixa etária) - sexo masculino.

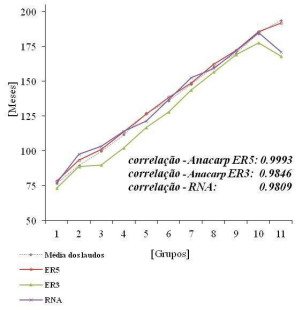



Fig. 10. Correlação entre as médias dos laudos e as idades estimadas utilizando ER5, ER3 e *RNA* - de indivíduos do sexo masculino.

A partir do gráfico de correlação (Fig. 10), nota-se que as idades estimadas apresentaram valores semelhantes e que se aproximaram das médias dos laudos médicos. Notam-se pequenas diferenças (subestimativa) apenas no grupo 11. Os gráficos presentes na Fig. 11 traz o diagrama de dispersão e a reta ajustada pela equação de regressão entre as idades estimadas pelo *Anacarp* (métodos ER5 e ER3) e metodologia *RNA* – eixo horizontal – e as médias dos laudos (vertical) para o sexo masculino. Pode-se observar também o coeficiente $R^2$, que representa o coeficiente de determinação e a equação de regressão.

Analisando o $R^2$ dos gráficos da Fig. 11, comprova-se que mais de 96% das idades ósseas estimadas pelos métodos ER%, ER3 e RNA estão cobertos pela reta de regressão (coincidem com a média dos laudos médicos), ou seja, ao estimarmos a idade óssea utilizando o *Anacarp* ou a metodologia RNA, teremos uma concordância com os valores das médias dos laudos médicos de mais de 96% para sexo masculino.

A mesma análise foi realizada para o grupo de imagens do sexo feminino, obtendo resultados semelhantes ao do grupo masculino.

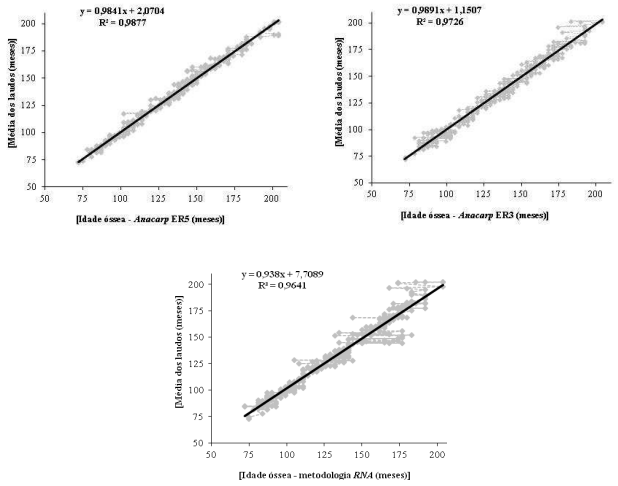



Fig. 11. Diagramas de dispersão e reta ajustada pela equação entre média dos laudos (X) e IO estimada por ER5, ER3 e RNA – sexo masculino

## IV. Conclusões

Tanto para o sexo masculino quanto para o feminino, ocorreu um alto índice de correlação entre os valores estimados (acima de 98%) quando comparados com GP, TW e ER [4;11]. Não houve diferenças estatisticamente significativa em todos os grupos e sexos.

Através dos intervalos de confiança pode-se constatar que os laudos estimados com os métodos ER5, ER3 e RNA não apresentaram grandes variações em relação à média dos laudos, sendo explicado através da pequena amplitude de valores expressos pelos desvios-padrão (ver Tabela 2).

Foram estabelecidos fatores de correção para a metodologia de área óssea, por meio da equação da reta de regressão, fazendo com que as idades estimadas a partir da metodologia área óssea coincidam em 100% com os valores médios dos laudos (apresentados na Fig.9).

## Agradecimentos

Os autores agradecem à Fundação de Amparo a Pesquisa do Estado de São Paulo (FAPESP) pelo suporte financeiro referente a esse trabalho.

## Referências

[1] Haiter, F.; Kurita, L. M. e Casanova, M. S. (2006) "Skeletal age assessment: A comparison of 3 methods", American Journal of Orthodontics and Dentofacial Orthopedics.

[2] Olivete, C.J.; Nascimento, M.Z. e Rodrigues, E.L.L.. (2004), "Metodologia de Pré-Processamento Baseada na Correção do "Efeito Heel" Aplicada em Imagens Radiográficas da Mão", In: *Anais do III Congresso Latino-Americano de Engenharia Biomédica*, João Pessoa, 2004, Brasil, pp. 106.

[3] Xu, C. e Prince, J. L. (1997). "Gradient Vector Flow: A New External Force for Snakes", *IEEE Proc. Conf. on Comp. Vis. Patt. Recog*., p. 66-71.

[4] Olivete. C.J. e Rodrigues ELL. (2010) Bone maturity: estimation by means of Eklof and Ringertz method simplifications. Radiol Bras vol.43 no.1 São Paulo Jan./Feb. 2010;2:15–9. [acessado em 25 de março de 2014]. Disponível em: http://dx.doi.org/10.1590/S0100-39842010000100006.

[5] Kass, M.; Witkin, A. e Terzopoulos, D. (1998) "Snakes: Active Contour Models". In: *International Journal of Computer Vision*, 1988.

[6] Young, D., (1995) "Active Contour Models Snakes". Disponível em: http://www.cogs.svsx.ac.uk/users/davidy/teachvision/vision7.html

[7] Olivete, C. J. e Rodrigues, E. L. L. (2007). "Idade Óssea: Uma Nova Metodologia de Estimação", Congresso Latino-Americano de Engenharia Biomédica, Venezuela, 2007.

[8] Sonka, M.; Hlavac, V. e Boyle, R (1998). "*Image Processing, Analysis and Machine Vision*", Chapamn & Hall Computing, London, 1998. p. 33

[9] Mackiewich, B. (2014) "*Active Contour Models Snakes*". Disponível em: http://www.cs.sfu.ca/people/Faculty/Atkins/papers/blairthesis/main/node28.html. Acesso em 05/04/2014.

[10] Queiroz, A. C. (2006) *Metodologia de Extração Automática de Características da Mão para Estimação da Idade Óssea Utilizando Redes Neurais Artificiais no Processo de Decisão.* Dissertação (mestrado). Escola de Engenharia de São Carlos, Universidade de São Paulo, 2006.

[11] Olivete, C. J. e Rodrigues, E.L.L. (2009) Software automático para determinação da idade óssea baseado no método de Eklof & Ringertz. Rev Bras Fís Méd. 2009;2:15–9. [acessado em 25 de março de 2014]. Disponível em: http://www.abfm.org.br/rbfm/publicado/au_53_tra_72_p15-19.pdf

[12] Nascimento, M. Z.; Frere, A.F; e Marques, M.A, et al. (2003) Subtraction of images for automatic determination of center of the radiation field. Engineeringin Medicine and Biology Society. *Proceedings of the 25th Annual International Conference of the IEEE*. 2003;1:937–40.

[13] Gonzalez, R.C.; Woods RE. (1993). *Digital image processing techniques*. New York: Addison Wesley.

[14] Otsu, N. (1997) A threshold selection method from graylevel histograms. IEEE Trans Systems Man Cybernet. 1997;9:62–6.

[15] Abutaleb, A. S. (1989) Automatic thresholding of graylevel pictures using two dimensional entropy. Comput Vision Graph Image Process. 1989;47:22–32.

[16] Kapur, J.N.; Sahoo, P.K. e Wong, A.K.C. (1985) A new method for gray-level picture thresholding using the entropy of the histogram. Comput Vision Graph Image Process. 1985;29:273–85.

[17] Haykin, S. (2001), Redes Neurais: princípios e prática, Editora Bookmam.

[18] Ayres, M. (2014). *BioEstat.* Version 5.3. Sociedade Civil Mamirauá, MCT – CNPQ, Belém, Pará, Brasil, Recuperado em maio de 2014 de, www.mamiraua.org.br.

# Reconhecimento de Caminhos Táteis com Visão Computacional em Ambientes Controlados

Rodrigo Lima, Maurício Marengoni, *Universidade Presbiteriana Mackenzie*

**Abstract**—This article presents the initial steps to build an application that for tactile flooring recognition in controlled environments providing help to visually impaired humans. The solution presented describes the use of a set of efficient algorithms for image processing such as: spatial and morphological filters, adaptive threshold and depth estimation. The system does depth estimation using Monocular Vision and the results obtained shows that it is possible to design a mobile system to help the impaired to move along this type of flooring.

## 1 Introdução

COM a legislação e normas em vigor, locais públicos devem seguir determinados padrões para aumentar e facilitar a mobilidade de pessoas de baixa ou nenhuma visão. A ABNT NBR 9050, descreve padrões que os pisos táteis deverão seguir para se enquadrar à lei de acessibilidade [1]. Sendo isto uma norma, juntamente com a preocupação de grandes empresas e empresas públicas em melhorar e facilitar a acessibilidade de pessoas de baixa ou nenhuma visão, busca-se com este artigo, demonstrar que é possível, com a ajuda da visão computacional, reconhecer pisos táteis e contribuir, em trabalhos futuros com a criação de hardware específico, para melhorar a forma como estas pessoas se locomovem em ambientes controlados. Demonstra-se também neste trabalho, que é possível estimar a profundidade de campo com o uso de visão monocular e com isto evitar sistemas complexos e caros do ponto de vista computacional. Este aplicativo poderá auxiliar deficientes visuais no uso de dispositivos móveis integrado com uma Bengala de Hoover.

Infelizmente, mesmo com normas e leis, muito ainda deve ser feito para a adaptação destes pisos, conforme pode ser observado na Figura 1.

Figura 1: Descaso com pisos táteis[2]

A Figura 1, ilustra o que ocorre nas cidades, onde o caminho direcional, muitas vezes não leva a lugar algum, sendo interrompido repentinamente, ou pode levar o deficiente visual a um obstáculo. Dado este tipo de problema, este trabalho teve como foco o desenvolvimento de uma aplicação para reconhecimento de pisos táteis em ambientes controlados para sistemas móveis.

Em Sousa e Marengoni (2013), é demonstrada a possibilidade da utilização de aparelhos móveis para detecção de faixa de pedestres. Este trabalho serviu como motivação para melhorar, e até mesmo ampliar o uso de dispositivos móveis por deficientes visuais afim de melhorar sua capacidade de locomoção [3]. Este trabalho, baseia-se na utilização do algoritmo de Taylor et al. (2004, p. 567-571), *Monocular Vision as a Range Sensor*, que descreve os problemas na utilização de visão estéreo em dispositivos móveis: consumo maior de ener-

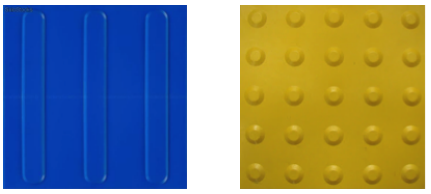
gia, uso de memória e espaço para montagem das câmeras; o artigo menciona ainda uma forma de solucionar o problema da medição da profundidade para a extração da informação em 3D usando visão monocular.

Embora o contexto do projeto como um todo seja desenvolver um protótipo do aplicativo que ajudará na acessibilidade, este artigo limita-se a demonstrar o processo de identificação do piso tátil tendo como entrada um stream de vídeo ou uma única imagem. As informações obtidas sobre o piso, devem gerar uma resposta com avisos, que poderão ser sonoros, ou em caso de implementação em celulares, vibração. Neste projeto foi utilizado o framework OpenCV 2.4.6, que é uma plataforma de visão computacional e fornece ao desenvolvedor diversos métodos já implementados que facilitam o desenvolvimento.

O trabalho esta separado em 5 seções. Na seção 2 são apresentados conceitos básicos sobre a definição da norma, enquanto na seção 3 é descrito o algoritmo proposto. As seções 4 e 5 descrevem os resultados parciais e os trabalhos futuros que serão necessários para alcançar o resultado ideal do sistema.

## 2 Conceitos básicos

Os pisos táteis, com o uso, apresentam imperfeições como sujeiras e o desgaste natural do piso, portanto, o método desenvolvido utiliza filtros para primeiramente identificar e remover estas imperfeições. Na Figura 2 pode-se observar os dois tipos de pisos descritos pela ABNT. É importante ressaltar que o piso pode variar suas medidas de acordo com os limites estabelecidos pela norma.



(a) Piso Direcional (b) Pisto de Alerta

Figura 2: Piso tátil

Na Figura 2a ilustra-se um piso direcional indicado para seguir, já na Figura 2b um piso de alerta, utilizado para informar desníveis, escadas e mudança de direção. Como o padrão ABNT limita-se a dizer apenas que os pisos devem ser cromo diferenciados, a segmentação por contorno foi o método escolhido para identificação.

A Figura 3 ilustra um cenário típico onde o sistema deverá funcionar. Visualmente distingue-se contornos que servirão para determinar a assinatura do piso, ou seja, os padrões a serem extraídos do piso e comparados com a norma ABNT NBR 9050.

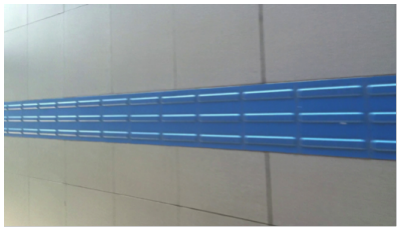


Figura 3: Piso tátil em ambiente controlado

Ainda na Figura 3, é possível observar a deformação causada pelo distanciamento da posição da câmera em relação ao piso, somando-se a isto o ângulo de captura do frame, este é um dos problemas para a estimativa de profundidade em visão monocular. O ângulo de posicionamento da câmera, é tratado como parâmetro de entrada, porém sabe-se que os smartphones atuais possuem outros artifícios tecnológicos, como giroscópios, capazes de identificar ângulos na horizontal e vertical.

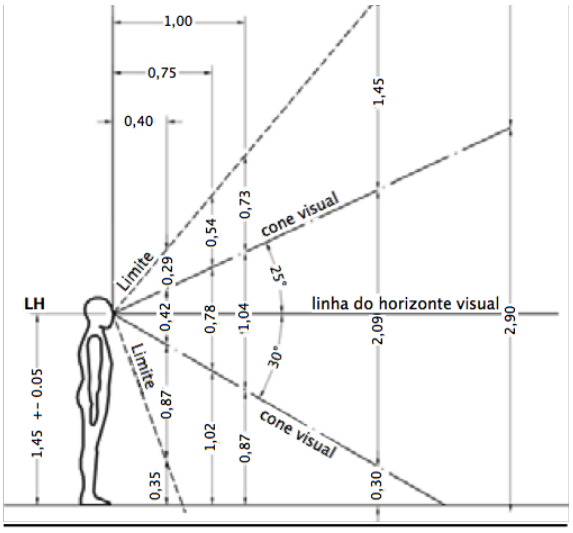



Figura 4: Cones visuais de pessoas em pé [1]

Também será adicionado como parâmetro, o ângulo de abertura da câmera obtido obtido de acordo com as especificações do dispositivo.

Os ângulos de alcance visual são importantes pois fornecem ao algoritmo parâmetros que ajudam a simular a visão humana.

Conforme indicado na Figura 4 a norma ABNT NBR 9050 ilustra a posição dos olhos, ou linha do horizonte (LH), a uma altura entre 1,40 m e 1,50 m do piso. O valor LH será uma entrada parametrizada, de acordo com a altura em que o dispositivo será utilizado.

## 3 ALGORITMO PROPOSTO

Mesmo sendo uma tarefa fácil para o sistema de visão humano, a identificação de pisos táteis requer um algoritmo onde serão necessários vários passos até a extração da informação final. Um passo importante para este trabalho são as operações de filtragem. Filtros são ferramentas básicas utilizadas para remoção de ruídos [6], neste caso, é considerado ruído, qualquer parte da imagem que possa ser confundido com a informação desejada, como por exemplo manchas no piso. Ruídos como pequenas manchas no solo podem contribuir na detecção de falsos-positivos e ruídos próximos ao eixo do piso podem gerar falsos-negativos, sendo assim, o resultado final estaria comprometido. Porém, mesmo sendo uma etapa fundamental para o processo, há de se tomar certos cuidados em relação a desempenho, pois algumas operações de filtragem podem ter alto custo computacional, dificultando a obtenção do resultado em um tempo aceitável. Para minimizar este problema o processamento foi feito em camadas, onde a imagem de entrada com resolução de 1280x720 foi recortada para trabalhar apenas com os primeiros 400x720 pixels, ou seja, a parte mais próxima do usuário. Caso neste primeiro corte não fosse identificado o piso tátil, o algoritmo recorreria aos demais cortes reduzindo assim o tempo de filtragem. O filtro gaussiano, definido pela equação,

$$G(x, y; \delta) = \frac{1}{2\pi\delta^2} e^{-} \frac{x^2 + y^2}{2\delta^2} \quad (1)$$

é usado para suavizar a imagem, de forma ponderada e simétrica. Ele é o resultado de uma convolução de cada ponto da matriz de entrada com o kernel gaussiano, que depois é somado para devolver a imagem de saída. Ele não é o mais rápido, porém, Bradski e Kaehler (2008) o classifica como o mais útil dos filtros de suavização [8]. O filtro gaussiano é utilizado antes de um filtro passa-alta, que tem como característica principal ressaltar bordas, regiões de interesse e mudança de intensidade, porém acaba por ressaltar alguns ruídos, que, com a utilização da gaussiana, serão diminuídos antes deste processo [6]. O filtro passa-alta foi aplicado com a Máscara ilustrada na Tabela 1.

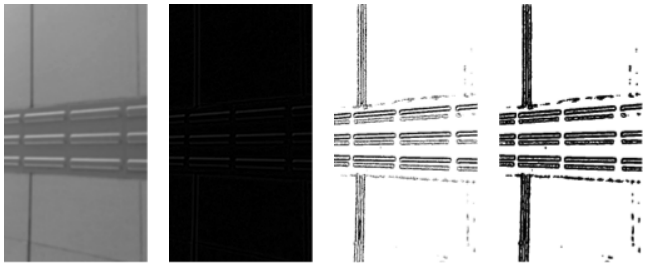
Tabela 1: Máscara para o filtro passa-alta

| | | |
|---|---|---|
| -3 | -3 | -3 |
| -3 | 24 | -3 |
| -3 | -3 | -3 |

Após estas duas operações a imagem de saída ainda pode apresentar pequenos ruídos, que podem ser removidos, completa ou parcialmente, por operações morfológicas utilizando uma operação de fechamento (uma dilatação após uma erosão) por um elemento estruturante do tipo retangular de tamanho 5x5.

Um threshold adaptativo, com função de média, é realizado após as operações de filtragem. Bradski e Kaehler (2008) descrevem que o filtro threshold é uma decisão final sobre quais pixels o algoritmo irá trabalhar, isso por que, ao utilizarmos esta função determina-se um limiar para binarização obtendo um resultado final para os pixels que estão abaixo do limiar e outro para os que estão acima. O threshold adaptativo é na verdade uma variação do threshold, que se adapta melhor as condições de iluminação [8].

As etapas destacadas na Figura 5, ilustram passo a passo o processamento necessário para obter a imagem a ser utilizada na segmentação e na extração da informação.



(a) (b) (c) (d)

Figura 5: Processamento da imagem

Utilizando a busca por contornos para extrair da imagem a informação de bordas, e com isso, retirar delas, usando o algoritmo Moments, o centro de massa podendo assim identificar se existe um piso tátil e se este é um piso direcional ou de alerta. A função findContours, trabalha com imagens binárias, neste caso imagens processadas com o threshold, porém também poderia ser utilizado o algoritmo de Canny, que retorna apenas os pixels de borda. O algoritmo de moments é definido pela seguinte equação,

$$m_{p,q} = \sum I(x,y)x^p y^q \qquad (2)$$

Bradski e Kaehler (2008) descrevem que o algoritmo de moments soma todos os pixels do contorno, e é normalmente utilizado para comparar dois contornos [8]. No caso desta proposta, o algoritmo de moments é utilizado para achar o centro de massa dos contornos e comparar com contornos vizinhos, buscando identificar se a distância entre eles corresponde a alguma das marcações de um piso tátil. Esta distância é uma distância euclidiana definida em milímetros. A profundidade da imagem, pode ser extraída usando duas câmeras, visão estéreo, mas, como já foi mencionado, esta técnica apresenta alguns problemas em dispositivos móveis como tempo de processamento, consumo de energia e mais espaço para montagem das câmeras.

A Figura 6 ilustra o problema de extrair informações em fotos. Percebe-se que, embora os pisos tenham a mesma dimensão, ao afastar-se da origem a foto sofre uma distorção simétrica. No trabalho de Taylor et al. (2004, p. 567-571), é descrito como resolver o problema de estimativa de profundidade de imagem com a utilização de uma única câmera. Said et al. (2012, p. 946), descrevem em seu trabalho como aumentar a precisão deste método, porém para isto seria necessário uma calibragem o que pode ser complicada, visto o tipo de público ao qual a aplicação é destinada [4], [5]. Sendo assim, optou-se por adotar um modelo sem calibragem.

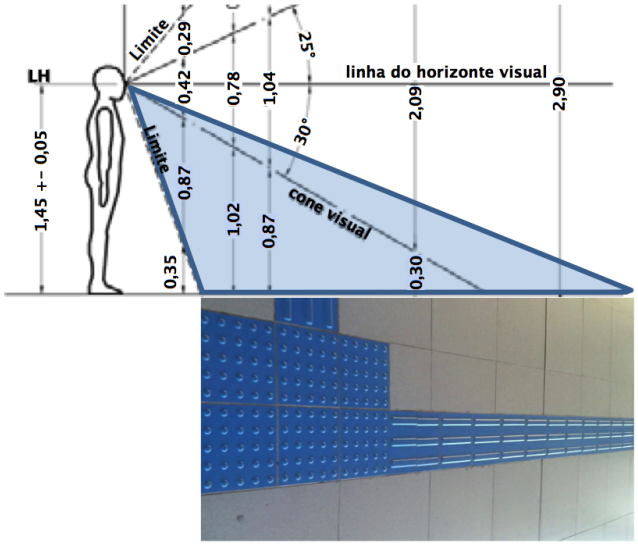



Figura 6: Problema de profundidade de campo

Como pode ser observado na Figura 6, o ângulo proposto, dependendo da câmera utilizada, é semelhante ao cone visual descrito pela ABNT, e neste cone é possível observar uma área mais próxima do usuário,onde nem os olhos nem a câmera alcançarão. A área mais próxima será o ponto inicial, que é importante pois precisa-se dele para calcular a profundidade total. O ângulo demonstrado na imagem deve levar em consideração, tanto o campo de visão (FOV) quanto o ângulo de posicionamento da câmera. Deve-se tomar o cuidado para que o ponto inicial seja o mais próximo possível do cone visual, tentando aproximar a visão da câmera, à mesma visão obtida pelos olhos humanos. Além disto, o aplicativo não necessitará que seja parametrizada a altura do usuário. Mais informações sobre os ângulos e parâmetros utilizados no sistema, podem ser observados na Figura 7, onde Taylor et al. (2004, p. 567-571) demonstram o FOV, vertical e horizontal.

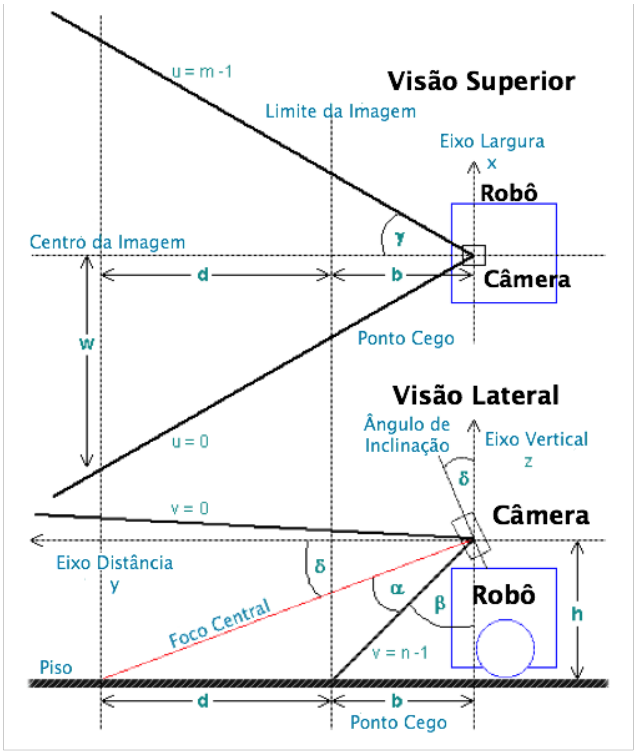



Figura 7: Modelo de visão monocular [3]

Na Figura 7 é demonstrado uma visão do Robô utilizado no experimento, Taylor et al. (2004, p. 567-571), especificando os ângulos de abertura e inclinação e ilustra as variáveis que são utilizadas nas equações 3-6.

$$\alpha + \beta + \delta = 90\,^{\circ} \tag{3}$$

$$\tan(\beta) = \frac{b}{h} \tag{4}$$

$$\tan(\alpha + \beta) = \frac{b+d}{h} \tag{5}$$

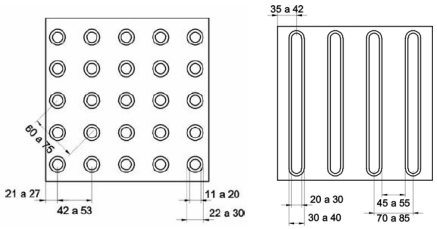
$$\tan(\gamma) = \frac{w}{b+d} \tag{6}$$

As equações de 3-6 expõem um modelo trigonométrico, porém, como visto na Figura 5, a imagem sofre uma distorção, fica menor conforme se afasta da câmera, nos dois eixos, vertical e horizontal. Para solucionar este problema, relacionam-se as distancias reais com coordenadas (u,v), que são na verdade o posicionamento dos pixels aos quais quer-se obter a medida. Dado uma imagem de m por n pixels, neste caso 1280x720, tem-se :

$$y = h\tan(\frac{\beta + 2\alpha(n-1-v)}{n-1}) \tag{7}$$

$$x = y\tan(\frac{\gamma(2\mu - m + 1)}{m-1}) \tag{8}$$

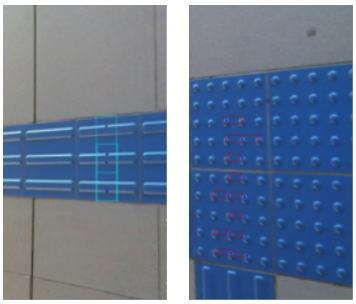
Sendo assim, para cada coordenada (u,v) determina-se um distanciamento em (x,y). Com estes valores, pode se comparar para cada contorno, a partir de seu centro de massa, o distanciamento entre seus centros [4].




(a) (b)

Figura 8: Piso de alerta e direcional [1]

Sabendo o distanciamento entre os eixos, definida pela norma ABNT, pode-se confrontar a distância encontrada entre os dois centros de massa. Para definir de que se trata de um piso direcional, além de respeitar o distanciamento da norma, deve-se achar a distância em x, com uma margem de 5mm de variação em y, do outro centro de massa. Uma regra semelhante é aplicada ao piso de alerta, sendo que a distância para este é calculada no y, com uma margem de 5mm em x. Esta margem é aplicada para corrigir possíveis desvios e nos certificarmos de que o cálculo esta sendo feito com o eixo correto.



(a) (b)

Figura 9: Piso direcional e de alerta após identificação

A Figura 9a e a Figura 9b, ilustra o resultado parcial obtido, onde pode-se observar os diversos pontos encontrados. Nos casos apresentados, o piso estava limpo, o que não gerou resultados indevidos, porém, este é um problema real e deve ser tratado em trabalhos futuros.

## 4 Resultados Parciais

Foram feitas algumas filmagens com aparelho celular, para realização dos testes, a uma altura de aproximadamente 1,20m. Os dados de abertura da câmera foram considerados como parâmetros. A altura foi escolhida aleatoriamente e desde que inserida como parâmetro de entrada no aplicativo não causará prejuízo aos cálculos. As filmagens foram feitas em horários diferentes, para garantir que a iluminação não fosse um impeditivo para a realização do projeto.

O algoritmo apresentou resultados satisfatórios, porém para aplicação em um ambiente real, necessitará de mais testes e desenvolvimento. Foram extraídos e analisados um total de 120 frames, retirados de 3 vídeos em horários diferentes em um mesmo local. Foram 93 as imagens de pisos direcionais, 9 apresentaram falsos positivos, 4 falsos negativos. E foram analisadas 27 de alerta, sendo que ocorreu apenas um falso negativo.

A análise dos pisos de alerta resultaram em 20 falsos positivos em imagens de pisos direcionais. Os pisos de alertas se mostraram mais suscetíveis a iluminação e sujeira enquanto os pisos direcionais a velocidade do usuário. A velocidade influenciou em alguns casos a capacidade de foco da câmera e como os pisos direcionais são compostos, nesta análise, de apenas 3 barras paralelas, são poucos os centros de massa a serem analisados. Os pisos de alerta são dispostos em 5x5, ou seja, mesmo que não reconheça 3 ou 4, ainda será possível distinguir vários centros de massa no piso de alerta, porém, por serem menores são mais suscetíveis a iluminação e sujeira.

Sujeira e, até mesmo desgaste dos pisos podem gerar resultados inesperados e para isto, deve-se considerar outros fatores, como por exemplo, o histórico dos frames já observados. Parte importante deste trabalho foi solucionar o problema de estimativa de distância com as equações desenvolvidas por Taylor et al. (2004, p. 567-571).

## 5 Trabalhos Futuros

É importante para este trabalho que tanto a minimização de ruídos quanto regras de validação do distanciamento dos centros de massa sejam aprimoradas, pois, como dito anteriormente, basear o resultado em apenas dois eixos pode gerar resultados falsos, e, até mesmo, em caso de aplicação real, trazer problemas ao usuário. Também, é importante, que na sequência, seja desenvolvido um protótipo capaz de gravar o stream, e a informação do ângulo de captura, frame a frame, para tornar os próximos testes mais aproximados do ambiente real ao qual este algoritmo se destina. Uma pesquisa de campo pode ser necessária afim de descobrir uma velocidade média do público pesquisado, para descobrir qual a velocidade de captura necessária para rastrear e identificar o piso, sem comprometer a segurança e a velocidade que eles já tem usando a bengala de rastreamento. Para versões futuras, estuda-se uma forma de utilizar uma outra câmera, com processamento em paralelo, para dar outras funções a aplicação, como leitura de placas de alerta, e detecção de pessoas.

## Referências


[1] ABNT NBR 9050, *Acessibilidade a edificações, mobiliário, espaços e equipamentos urbanos*, ¡http://www.pessoacomdeficiencia.gov.br/app/normas-abnt¿. Acesso em Maio de 2014.

[2] Uol Noticias, Piso para cegos acaba em muro em avenida da zona oeste de São Paulo, ¡http://noticias.uol.com.br/cotidiano/ultimas-noticias/2012/12/13/piso-para-cegos-acaba-em-muro-em-avenida-da-zona-oeste-de-sp.htm¿ , 13 de dezembro de 2012. Acesso em maio de 2014.

[3] Kelly Sousa, Maurício Marengoni , Uso de Visão Computacional em Dispositivos Móveis para o Reconhecimento de Faixa de Pedestres, 2013

[4] Monocular Vision as a Range Sensor , T.Taylor, S.Geva and W.W.Boles, 2004

[5] Depth Estimation for a Mobile Platform Using Monocular Vision, Z. Said, K. Sundaraj e M. N. A. Wahab, 2012

[6] Maurício Marengoni e Denise Stringhini, Introdução a Processamento de Imagens e Visão Computacional usando OpenCV. SIBGRAPI, 2008.

[7] Richard Szeliski, Computer Vision, Algorithms and Applications, 2011

[8] Gary Bradski, Adrian Kaehler, Learning OpenCV – Computer Vision with the OpenCV Library, O'Reilly, 2008

# Grammatical Inference and SIFT for Scene Recognition

Lucas C. Ribas*[1], Marcelo Borth‡, Amaury A. Castro Jr.*, Wesley N. Gonçalves* e Hemerson Pistori†
*Universidade Federal de Mato Grosso do Sul, UFMS - CPPP
LaRPP Laboratório de Robótica de Ponta Porã, Ponta Porã, MS 79907-414
Email: lucascorreiaribas@gmail.com, {amaury.junior, wesley.goncalves}@ufms.br
†Universidade Católica Dom Bosco, UCDB
INOVISAO Laboratório de Visão Computacional, Campo Grande, MS
Email: pistori@ucdb.br
‡Instituto Federal de Mato Grosso do Sul, IFMS
Ponta Porã, MS 79909-000
Email: marcelo.borth@ifms.edu.br

***Resumo*—Grammatical inference in computer vision regained attention in recent years due to the emergence of new local feature techniques, such as Scale Invariant Feature Transform (SIFT) e Speeded Up Robust Features (SURF). This paper presents a methodology that converts an image into a string based on the SIFT and bag-of-visual-words (BOW). Given the strings, grammar induction techniques are used for scene recognition. In the BOW, the vocabulary is usually learned using an unsupervised approach. To improve image description, this paper proposes a supervised vocabulary learning. This approach improves the string obtained from the image and the experimental results have demonstrated its robustness in a challenging scene database.**

## I. Introdução

Nos últimos anos, o reconhecimento de cenas e ambientes (e.g. corredor, escritório, sala, entre outras) é um problema que tem recebido muita atenção na comunidade de visão computacional. Entre as principais aplicações, o reconhecimento de cenas auxilia a localização de robôs móveis, aumentando a compreensão semântica do ambiente e a complexidade das tarefas que eles podem realizar.

Para a descrição das imagens, a extração de características locais tem sido extensivamente empregada no reconhecimento de cenas. Os extratores *Scale Invariant Feature Transform* (SIFT) [1] e *Speeded Up Robust Features* (SURF) [2] são dois dos principais extratores desta categoria. Em conjunto com a metodologia de histograma de palavras visuais (*Bag-of-visual-words − BOW*) [3], estes extratores têm apresentado resultados promissores na tarefa descrita acima. O BOW utiliza as características extraídas pelo SIFT ou SURF para construção de um vocabulário. Para isso, são utilizadas as características locais extraídas de imagens de todas as categorias/classes. Embora grandes avanços foram reportados nos últimos anos, os histogramas (ocorrência de palavras do vocabulário) desconsideram a relação espacial das características locais.

Para superar esse problema, trabalhos foram propostos para incluir informação espacial na metodologia BOW [4], [5], [6], [7], [8]. Entre os trabalhos atuais, Pistori et al. [5] e Tu et al. [9] propõem o uso de inferência gramatical. Estes trabalhos resgatam as técnicas de reconhecimento sintático de padrões visuais muito usadas nas décadas de 60 e 70 [10]. De acordo com [5], a principal questão no uso de gramáticas em imagens está na representação da imagem por uma cadeia que represente as características relevantes da mesma e não ignorem relações estruturais entre palavras visuais.

Este artigo tem como proposta utilizar inferência gramatical para o reconhecimento de cenas. No método, características locais são extraídas das imagens e utilizadas para construção do vocabulário. Em seguida, as características são rotuladas e uma cadeia é obtida para cada imagem. A cadeia impõe a ordem e, consequentemente, a informação espacial entre as características locais. Por fim, uma gramática é construída para representar cada categoria. Dada uma gramática e uma cadeia obtida de uma imagem, é possível responder se a cadeia pertence à gramática, implicitamente, respondendo se a imagem pertence à categoria que a gramática representa. Além do método acima, este artigo apresenta uma comparação entre a construção do vocabulário de forma supervisionada [11] e não-supervisionada.

Os resultados obtidos em uma base de imagens complexa mostraram que o uso da inferência gramatical e características locais é adequado para o reconhecimento de cenas. Além disso, a construção do vocabulário supervisionado forneceu resultados superiores em comparação com a construção do vocabulário não-supervisionado. Esse resultado demonstra que a obtenção de palavras individuais para cada categoria é importante para o processo de descrição de imagens.

O restante do artigo é descrito como segue. A Seção 2 apresenta os conceitos e métodos necessários para a compreensão do artigo. O método para utilizar inferência gramatical no reconhecimento de cenas é descrito na Seção 3. A Seção 4 e 5 apresentam a configuração dos experimentos e os resultados obtidos, respectivamente. Por fim, as conclusões e os trabalhos futuros são discutidos na Seção 6.

[1]Bolsista do Programa de Educação Tutorial PET/MEC/SESu (PET/Fronteira)

## II. TEORIAS

### A. Scale Invariant Feature Transform (SIFT)

O *Scale Invariant Feature Transform* (SIFT) [1] é um dos mais importantes métodos para extração de características de uma imagem. O SIFT extrai da imagem uma coleção de vetores de características locais, chamados de *keypoints* ou pontos-chave [1]. Cada um dos pontos-chave é invariante a escala, rotação, e parcialmente invariante a mudança de iluminação. Essas propriedades são de suma importância para tarefas como reconhecimento de cenas e objetos.

O método para extração de características pode ser descrito como segue. O principal passo do método é a identificação de pontos-chave $\rho_i$ que sejam invariantes a mudança de escala da imagem. Este passo é implementado usando a função de diferença de Gaussianas [12]. Em seguida, cada ponto-chave $\rho_i$ é representado por quatro elementos:

1) $(x_i, y_i)$, localização espacial na imagem.
2) $\sigma_i$, escala em que ele foi detectado.
3) $\theta_i$, orientação predominante do gradiente.
4) $\varphi_i \in \Re^{128}$, vetor de características contendo 128 valores que descrevem a região ao redor do ponto.

Segundo Lowe [1] um aspecto importante deste algoritmo é a capacidade de gerar um grande número de características, que pode cobrir toda imagem e em diferentes variações e escalas. A Figura 1 mostra um exemplo de uma imagem com os pontos-chave detectados pelo SIFT. A localização $(x_i, y_i)$ do ponto-chave na imagem é representada pelos pontos brancos. Enquanto que os círculos brancos envolta dos pontos-chave estão relacionados à escala e os traços a orientação.

### B. Histograma de Palavras Visuais

O histograma de palavras visuais (*Bag-of-Visual-Words* – BOW) [3] é uma das técnicas mais populares usadas para reconhecimento de cenas nos últimos anos. De acordo com Csurka et al. [3], as principais etapas dessa técnica, visualizadas na Figura 2, são:

a) Detecção e descrição da imagem (SIFT).
b) Agrupamento dos pontos-chave em grupos de acordo com seus descritores, ou seja, criar um vocabulário (K-Means).
c) Rotulação de cada ponto-chave com uma palavra do vocabulário.
d) Construção de um histograma de ocorrência das palavras na imagem.

Uma etapa importante nessa técnica é a criação do vocabulário, nesta, o tamanho do vocabulário $k$ é um parâmetro importante na descrição das imagens. Apesar dos avanços recentes e dos resultados promissores, o poder descritivo dessa técnica acaba sendo limitado, pois esta descarta informações espaciais das palavras na imagem. Quando apenas contamos a ocorrência de uma palavra em uma imagem, não estamos considerando onde esta palavra está localizada na imagem e nem a relação de uma palavra com as demais. Essas informações podem ser características importantes para a classificação da imagem.

### C. K-Testable

O K-Testable é uma técnica que dado um tamanho de memória $k_t$, tenta-se encontrar um autômato para reconhecer uma linguagem representada por um número de cadeias passadas como parâmetro. Uma linguagem k-testável é uma subclasse de uma linguagem regular que encontra prefixos, sufixos e sub-cadeias nos dados de treinamento [5]. A principal característica é que cada caractere é dependente apenas dos $k_t$-1 caracteres anteriores e segundo Pistori et al. [5] a análise de uma cadeia de caracteres pode ser feita usando uma memória de tamanho fixo $k_t$.

## III. MÉTODO PROPOSTO

Este trabalho propõe uma melhoria no método que utiliza inferência gramatical no problema de classificação de imagens proposto por Pistori et al. [5]. A primeira é em relação ao algoritmo de extração de características locais, neste trabalho foi usado o SIFT enquanto que o trabalho anterior utilizou o SURF. Outro ponto relevante é quanto ao tamanho $k$ do vocabulário, que neste trabalho foi experimentado em uma ampla faixa de valores. Os valores de $k$ baixos acabam limitando a descrição das características de cada categoria/classe, consequentemente, o desempenho acaba sendo limitado. Por fim, os vocabulários são construídos de duas formas: não-supervisionada (trabalho anterior) e supervisionada.

O método proposto pode ser descrito em 5 etapas: a) extração de pontos-chave, b) construção do vocabulário, c) rotulação dos pontos-chave, d) construção da cadeia, e) aprendizagem da gramática. As seções abaixo descrevem cada etapa em detalhes.

### A. Extração de Pontos-chave

Nesta etapa, os pontos-chaves de cada imagem são encontrados por meio do SIFT. Considere $\rho_i^j$ como sendo o ponto-chave $i$ extraído da imagem $j$. Cada ponto é representado por 5 elementos descritos nas seções anteriores: $\rho_i^j = \{(x_i^j, y_i^j), \sigma_i^j, \theta_i^j, \varphi_i^j\},\ 1 \leq i \leq M$. Em geral, o número de pontos-chave $M$ variam de imagem para imagem, pois estes dependem diretamente do gradiente.

### B. Construção do Vocabulário

Dados os pontos-chave extraídos das imagens, esta etapa constrói um vocabulário com base no conjunto de descritores $D$ (Equação 1). O conjunto de descritores é composto por todos os descritores de todas as imagens. Devido às restrições de memória, em alguns casos, utiliza-se um subconjunto de $D$ escolhido de forma aleatória.

$$D = [\varphi_i^j],\ 1 \leq i \leq M_j, 1 \leq j \leq N \tag{1}$$

onde $N$ é o número de imagens de treinamento e $M_j$ é o número de pontos extraídos da imagem $j$.

Dado o conjunto de descritores, estes são agrupados por meio do algoritmo k-means e o conjunto de $k$ centróides $C$ é obtido:

$$C = \text{k-means}(D) \tag{2}$$

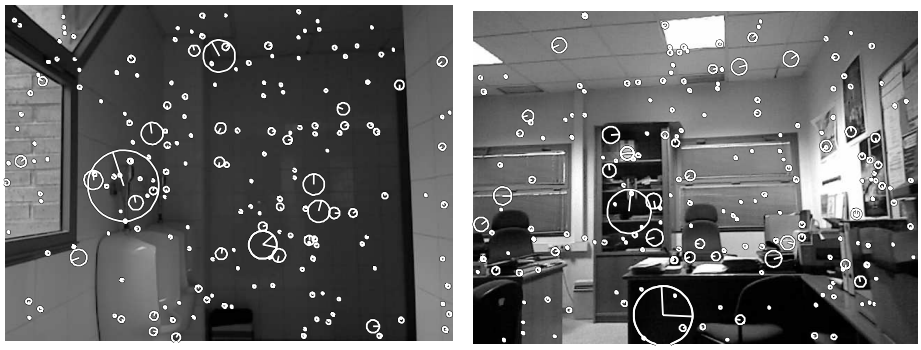


Figura 1. Imagens de cenas da base de imagens com pontos-chave detectados pelo SIFT. Cada ponto-chave é representado por uma posição espacial na imagem $(x_i, y_i)$, uma escala $\sigma_i$, uma orientação $\theta_i$ e um vetor de características que descreve a região ao redor.

O conjunto $C$ é descrito como o vocabulário aprendido para as imagens de treinamento. Cada centróide $C_l \in C,\ 1 \leq l \leq k$ possui a mesma dimensão dos descritores, isto é, $C_l \in \Re^{128}$.

O processo descrito acima é dito ser não-supervisionado, pois descritores de todas as classes são utilizados para construir o vocabulário. Para tornar esse processo supervisionado, um vocabulário é construído para cada classe das imagens de treinamento. Para isso, os centróides $C_p$ são obtidos por meio do k-means aplicado em um conjunto $D_p$ contendo somente os descritores das imagens pertencentes à classe $p$:

$$D_p = [\varphi_i^j],\ \text{se } j \text{ pertence à classe } p \quad (3)$$
$$C_p = \text{k-means}(D_p) \quad (4)$$

Considerando $n_c$ como o número de classes, $n_c$ conjuntos de centróides serão obtidos. A vantagem desta construção é que palavras serão obtidas para cada classe, o que não é garantido na construção não-supervisionada. Por fim, para a construção do vocabulário supervisionado, os $n_c$ conjuntos de centróides são concatenados.

$$C = [C_p],\ 1 \leq p \leq n_c \quad (5)$$

### C. Rotulação dos Pontos-chave

A próxima etapa é a rotulação de cada ponto-chave das imagens de treinamento. Dados os pontos-chave $\rho_i^j$ para a imagem $j$, cada ponto-chave é rotulado com o índice da palavra mais próxima do vocabulário, isto é, o centróide mais próximo conforme apresentado na Equação 6.

$$r_i^j = \arg\min_{l=1}^{k} |\varphi_i^j, C_l| \quad (6)$$

onde $r_i^j$ corresponde ao rótulo do ponto-chave $i$ da imagem $j$ e $|.|$ é a distância Euclidiana.

Após essa etapa, cada ponto-chave possui um rótulo de forma que pontos-chaves com descritores parecidos possuem os mesmos rótulos. Um exemplo pode ser visto na Figura 3, onde os rótulos são indicados por letras do alfabeto.

### D. Construção da Cadeia

Nesta etapa os pontos-chave na imagem são percorridos gerando uma cadeia com as palavras visuais correspondentes. Para percorrer os pontos-chave na imagem, a ordem radial foi definida por ser invariante à rotação. Nesta ordem, as palavras são extraídas da imagem de acordo com a distância Euclidiana da sua posição espacial para o centro da imagem, onde as palavras mais próximas do centro da imagem serão extraídas primeiro. Um exemplo é ilustrado na Figura 3 onde podem ser visto os círculos concêntricos e a cadeia formada é CACFEBCACBFCBEFDEFA.

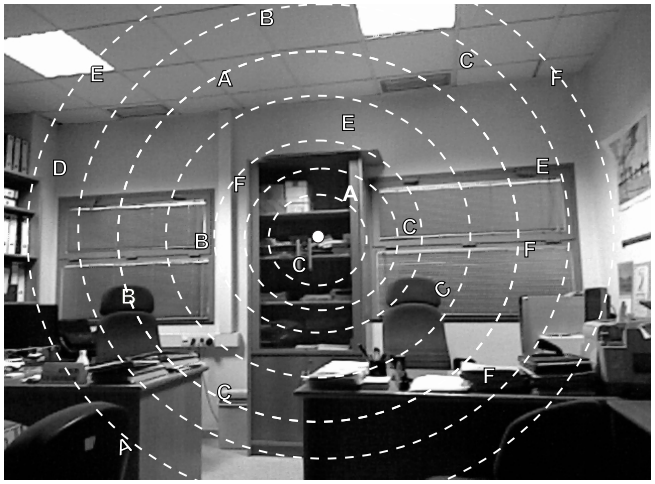



Figura 3. Extração das palavras da imagem e formação das cadeias. Na imagem acima a cadeia gerada é CACFEBCACBFCBEFDEFA.

### E. Aprendizagem da Gramática

A última etapa do método é inferir uma gramática para cada classe através das cadeias de palavras. Para a inferência, o algoritmo K-Testable é utilizado.

Após as etapas acima, as cadeias que representam as imagens de teste são validadas nas gramáticas geradas. Para validar uma cadeia, é contado o número de erros da cadeia para cada gramática que representa uma classe. Erros ocorrem quando há um caractere na cadeia que não pertence a linguagem ou não existe a transição na linguagem. A gramática que retornar o menor número de erros é a qual pertence à cadeia.

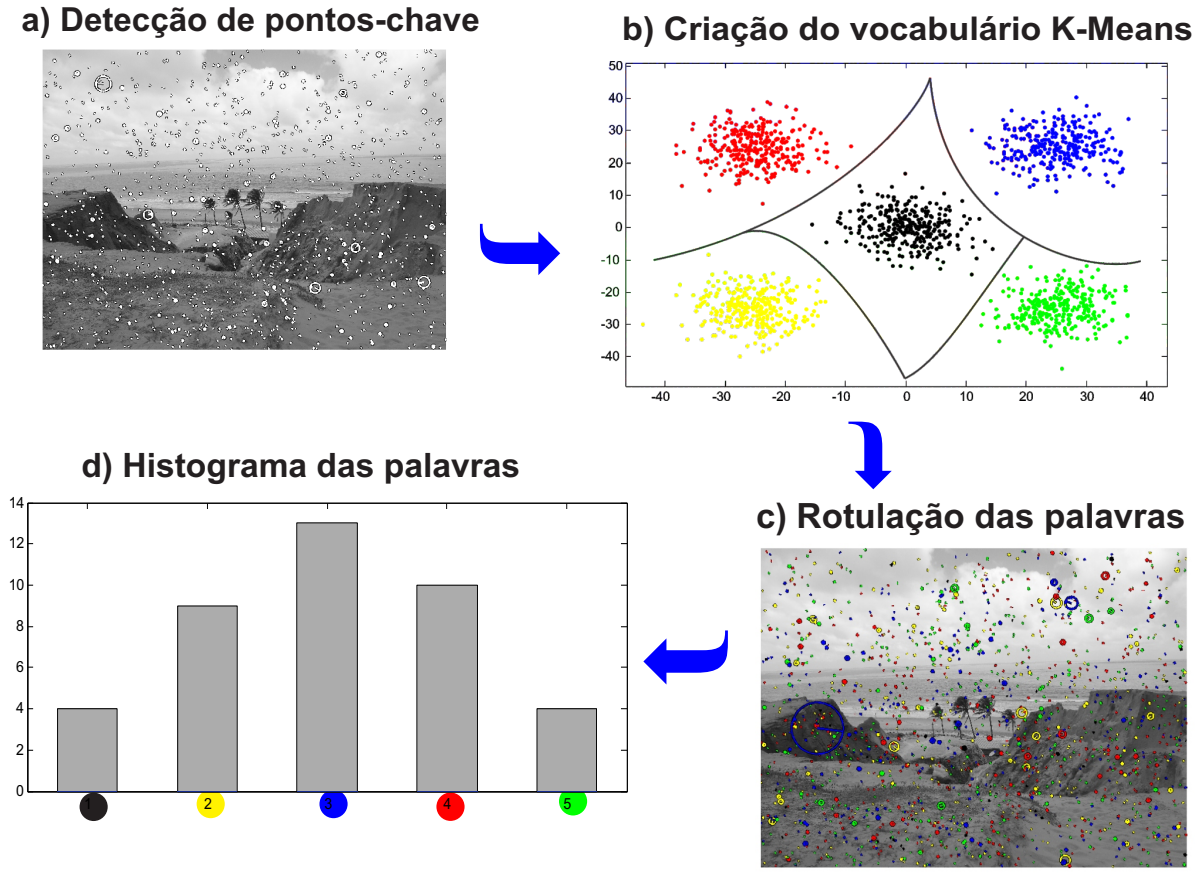



Figura 2. Etapas para extração de características utilizando o BOW. a) Detecção dos pontos-chaves utilizando o algoritmo SIFT. b) Criação do vocabulário utilizando o algoritmo K-Means. c) Rotulação das palavras de acordo com o vocabulário gerado pelo K-Means. d) Contagem de ocorrência das palavras em uma imagem.

## IV. Experimentos

Os experimentos foram realizados utilizando imagens de cenas, a Figura 4 mostra exemplos de imagens da base. A base de imagens usada é composta por um total de 1105 imagens coloridas, divididas em 6 classes de cenas classificadas por humanos [13]. A Tabela I resume o número de imagens de cada classe. O tamanho de cada imagem é $640 \times 480$.

A construção do vocabulário foi realizada de duas formas, supervisionada e não-supervisionada com $k$ variando entre 60 e 2700. O parâmetro $k_t$ do K-Testable utilizado nos experimentos foi de $k_t$ = 2. Para a divisão do conjunto de dados de treinamento e teste foi usado o modelo de validação cruzada, com um número de pastas igual a 10.

Tabela I. Número de imagens de cada classe utilizado nos experimentos.

| Classe | Quantidade de imagens | Porcentagem do Total |
|---|---|---|
| Elevador | 40 | 3.61 |
| Banheiro | 188 | 17.01 |
| Sala de professor | 200 | 18.09 |
| Secretaria | 181 | 16.38 |
| Laboratório | 282 | 25.52 |
| Sala de TI | 214 | 19.36 |

## V. Resultados e Discussões

A Tabela II apresenta os resultados dos experimentos na forma supervisionada e não-supervisionada e com diferentes valores de $k$. A maior taxa de classificação correta (TCC) foi de 77.33 % usando construção de vocabulário supervisionada e um tamanho de vocabulário igual 2.400. Na Figura 5 temos um gráfico comparando os dois processos de construção de vocabulário para os diferentes tamanhos de $k$. Nesta, pode-se observar que os vocabulários construídos de forma supervisionada e com valores de $k$ altos levam vantagem em relação ao método comparado. Os vocabulários construídos de forma supervisionada garantem palavras para cada classe, enquanto, na construção não-supervisionada não há essa garantia, isso pode explicar o desempenho superior da construção supervisionada. Outro fator relevante é quanto ao desempenho superior dos experimentos usando tamanhos de vocabulário maiores, o que pode ser explicado pela maior descrição das características das imagens ao utilizar valores altos de vocabulário.

A Figura 6 compara as matrizes de confusão para a construção supervisionada e não-supervisionada do vocabulário. Na construção supervisionada do vocabulário a maior TCC foi de 77.33 % com $k$ = 2400 e na não-supervisionada a TCC foi de 72.70 % com $k$ = 1440. Nas duas matrizes, pode-se

(a) Área de Elevador (b) Sala de Professor (c) Secretaria

(d) Sala de Estudante (e) Sala de TI (f) Banheiro

Figura 4. A base é composta por 6 classes com imagens de $640 \times 480$.

Tabela II. RESULTADOS DOS EXPERIMENTOS (TCC E DESVIO PADRÃO) PARA DIFERENTES TAMANHOS DE VOCABULÁRIO. AS MELHORES TCC PARA CADA TAMANHO DE VOCABULÁRIO ESTÁ EM NEGRITO.

| Tamanho de Vocabulário | TCC (%) | |
|---|---|---|
| | Não-Supervisionado | Supervisionado |
| 60 | 28.10 ($\pm$0.01) | **29.15** ($\pm$0.02) |
| 120 | **33.80** ($\pm$0.02) | 33.70 ($\pm$0.03) |
| 240 | 52.30 ($\pm$0.02) | **52.43** ($\pm$0.02) |
| 480 | 63.80 ($\pm$0.02) | **70.05** ($\pm$0.03) |
| 960 | 69.70($\pm$0.02) | **71.22** ($\pm$0.03) |
| 1200 | 69.80 ($\pm$0.04) | **73.79** ($\pm$0.02) |
| 1440 | 72.70 ($\pm$0.03) | **74.32** ($\pm$0.02) |
| 1800 | 71.15 ($\pm$0.05) | **76.03** ($\pm$0.02) |
| 2100 | 72.00 ($\pm$0.02) | **77.18** ($\pm$0.03) |
| 2400 | 69.42 ($\pm$0.02) | **77.33** ($\pm$0.02) |
| 2700 | 68.63 ($\pm$0.04) | **76.77** ($\pm$0.02) |

observar que em algumas classes possuem uma considerável taxa de classificação incorreta. Como exemplo, a classe Sala de Professor é classificada incorretamente como Laboratório em 39.50 % na construção não-supervisionada e em 36.50 % na supervisionada. Um dos prováveis motivos dessa confusão na classificação está na complexidade da base de imagens. Nesta, há imagens de diferentes classes, mas com bastantes características similares, como computadores, cadeiras, mesas e etc.

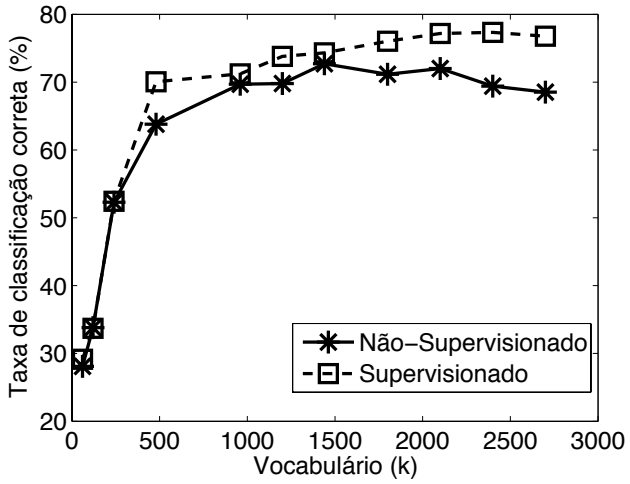
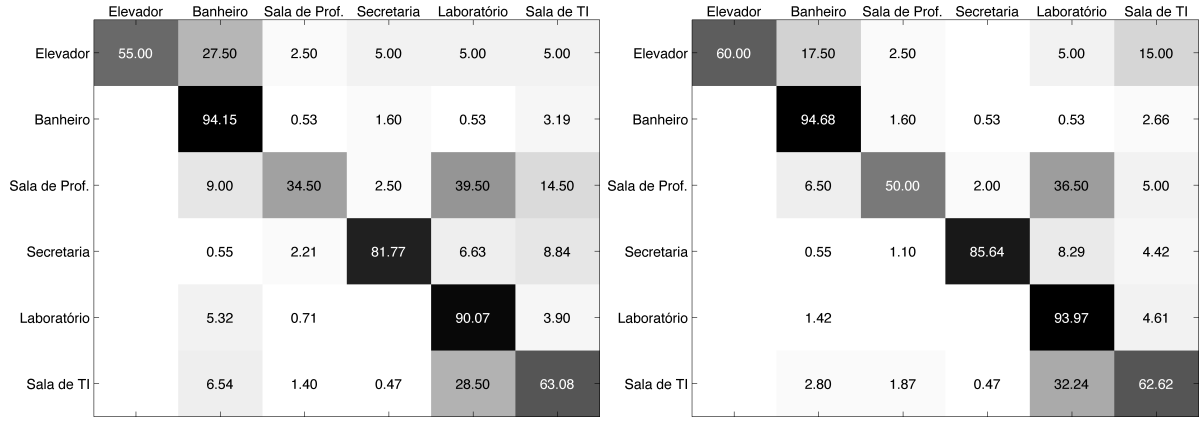



Figura 5. Gráfico com a TCC (eixo y) para diferentes tamanhos de vocabulário $k$ (eixo x) e com modo de treinamento supervisionado e não-supervisionado.

## VI. CONCLUSÃO E TRABALHOS FUTUROS

O problema de classificação de imagens (e.g. reconhecimento de cenas) em visão computacional tem sido um campo de estudo que vem recebendo grande atenção na área nos últimos anos. Uma das linhas de pesquisa neste campo é a conversão de imagens em cadeias e o uso de gramáticas para a classificação de imagens. Esta ganhou maior atenção nos últimos anos devido ao surgimento de novas técnicas para extração de características de imagem (e.g. SIFT). Neste trabalho a proposta é a conversão de imagens em cadeias com base na metodologia de BOW, como já foi proposto em [5]. No entanto, a principal contribuição deste artigo está na construção supervisionado do vocabulário e no uso de uma ampla faixa de valores de tamanho de vocabulário. Como pode se visto nos resultados há um considerável aumento de desempenho ao usar valores altos de $k$, combinado com vocabulários construídos de forma supervisionada.

Para trabalhos futuros, sugere-se o uso de novas estratégias para percorrer as palavras na imagem e gramáticas mais poderosas. A estratégia de percorrer as palavras na imagem é determinante para adicionar informação espacial e características da imagem nas cadeias. Portanto, se as características da imagem estiverem implícita na cadeia, o desempenho na classificação será melhor. O uso de gramáticas mais poderosas, também é outro fator a ser levado em consideração em trabalhos futuros. Gramáticas que aceitam exemplos positivos e negativos ou gramáticas não-regulares, podem ser alternativas para melho-




(a) Não-Supervisionado (b) Supervisionado

Figura 6. Matriz de confusão com a TCC para cada classe de cena referente a maior TCC total do modo não-supervisionado 72.70 % e supervisionado 77.33 %.

rar o desempenho. Para outros trabalhos pretende-se avaliar imagens com ruídos e diferentes condições de iluminação.

## AGRADECIMENTOS

A Pró-Reitoria de Extensão, Cultura e Assuntos Estudantis (PREAE) da Universidade Federal de Mato Grosso do Sul (UFMS) e ao Programa NERDS da Fronteira pelo apoio financeiro na locomoção e estadia. Ao MEC/SESu, através do Programa de Educação Tutorial (PET/Fronteira), no qual o primeiro autor deste artigo é bolsista.

## REFERÊNCIAS

[1] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[2] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (surf)," *Comput. Vis. Image Underst.*, vol. 110, no. 3, pp. 346–359, Jun. 2008. [Online]. Available: http://dx.doi.org/10.1016/j.cviu.2007.09.014

[3] G. C. e Christopher R. Dança e Lixin Fan e Jutta Willamowski e Cédric Bray, "categorização visual com sacos de pontos-chave," in *Em Workshop sobre Estatística Aprendizagem em Visão Computacional, ECCV*, 2004, pp. 1–22.

[4] E. Zhang and M. Mayo, "Improving bag-of-words model with spatial information," in *Image and Vision Computing New Zealand (IVCNZ), 2010 25th International Conference of*, Nov 2010, pp. 1–8.

[5] H. Pistori, A. Calway, and P. Flach, "A new strategy for applying grammatical inference to image classification problems," in *Industrial Technology (ICIT), 2013 IEEE International Conference on*, 2013, pp. 1032–1037.

[6] C. Zhang, S. Wang, Q. Huang, J. Liu, C. Liang, and Q. Tian, "Image classification using spatial pyramid robust sparse coding," *Pattern Recognition Letters*, vol. 34, no. 9, pp. 1046 – 1052, 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167865513000573

[7] G. Pedrosa and A. Traina, "From bag-of-visual-words to bag-of-visual-phrases using n-grams," in *Graphics, Patterns and Images (SIBGRAPI), 2013 26th SIBGRAPI - Conference on*, Aug 2013, pp. 304–311.

[8] J. Yuan, Y. Wu, and M. Yang, "Discovery of collocation patterns: from visual words to visual phrases," in *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, June 2007, pp. 1–8.

[9] Z. Tu, X. Chen, A. L. Yuille, and S.-C. Zhu, "Image parsing: Unifying segmentation, detection, and recognition," *International Journal of Computer Vision*, vol. 63, no. 2, pp. 113–140, 2005.

[10] K. S. Fu and K. S. Fu, *Syntactic methods in pattern recognition*. Academic press New York, 1974, vol. 197, no. 1.

[11] M. Jiu, C. Wolf, C. Garcia, and A. Baskurt, "Supervised learning and codebook optimization for bag-of-words models," *Cognitive Computation*, vol. 4, no. 4, pp. 409–419, 2012. [Online]. Available: http://dx.doi.org/10.1007/s12559-012-9137-4

[12] G. L. G. Gonzales, "Aplicação da técnica sift para determinação de campos de deformações de materiais usando visão computacional," Mestrado, PONTIFíCIA UNIVERSIDADE CATóLICA DO RIO DE JANEIRO - PUC-RIO, 2011.

[13] J. Martinez-Gomez, I. Garcia-Varea, M. Cazorla, and B. Caputo, "Overview of the imageclef 2013 robot vision task," in *Working Notes, CLEF 2013*, 2013.

# An automated method for quantifying cytoplasmic gene expression level on early *Drosophila* embryos

Daniela Justiniano de Sousa
Faculdade de Computação
Universidade Federal de Uberlândia
daniela@mestrado.ufu.br

Maira Arruda Cardoso
Instituto de Biofísica Carlos Chagas Filho
Universidade Federal do Rio de Janeiro
mairaarrudacardoso@gmail.com

Paulo Mascarello Bisch
Instituto de Biofísica Carlos Chagas Filho
Universidade Federal do Rio de Janeiro
pmbisch@biof.ufrj.br

Francisco José Pereira Lopes
Instituto de Biofísica Carlos Chagas Filho
Universidade Federal do Rio de Janeiro, Polo de Xerém and Inmetro - National Institute of Metrology, Quality and Technology
flopes@ufrj.br

Bruno Augusto Nassif Travençolo
Faculdade de Computação
Universidade Federal de Uberlândia
travencolo@gmail.com

***Resumo*—In this paper we present an automatic method for identification and quantification of the RNA level on the cytoplasm of early *Drosophila melanogaster* embryos. The method is part of a set of algorithms assembled for analysis and evaluation of gene patterns of this biological model. The central point of the proposed method is the ability to measure the RNA average intensity, individually, for each nucleus present in obtained images from the sagittal plane. The results demonstrate high performance for quantification on images with cellular resolution, and without loss of spatial information.**

## I. Introdução

Todas as células de um ser vivo compartilham o mesmo DNA. Sendo assim, a princípio, possuem a informação gênica utilizada para codificar todas as proteínas necessárias para a definição e funcionamento do organismo. No entanto, cada célula de um organismo, ao longo de seu desenvolvimento embrionário, expressa apenas um subconjunto específico de genes, dependendo da sua função [1]. Isto é, à medida que um embrião em desenvolvimento sofre sucessivas divisões, genes específicos são ativados em diferentes células durante diferentes períodos de tempo.

Esse processo é definido como diferenciação celular, e explica a grande diversidade de fenótipos observada, ou seja, existência de células com divergências funcionais e estruturais, apesar de serem originadas a partir de uma única célula e apresentarem o mesmo código genético [2]. Em cada célula, apenas uma parte do seu genoma é expresso. Nele, sequências específicas de DNA são utilizadas para sintetizar e acumular diferentes conjuntos de RNA e proteínas [3].

Cada estágio da diferenciação celular pode, então, ser caracterizado por um conjunto específico de genes transcritos. Esta expressão seletiva de genes formam padrões de expressão distintos para cada célula.

A caracterização dos níveis e domínios espaço-temporais dos padrões de expressão gênica compreende um dos principais objetivos da genômica funcional. Este processo fornece informações importantes sobre o funcionamento de uma célula. Dados sobre onde e quando um RNA está presente numa dada célula são essenciais para compreender a relação entre a sequência de DNA de um gene e sua função fisiológica. Além disso, a descrição e análise da expressão gênica permitem compreender a rede de interações gênicas que fundamenta o processo de desenvolvimento normal do embrião [4].

Sendo assim, neste trabalho é apresentado um método automático para quantificar o nível médio de RNA expresso no citoplasma basal dos embriões de Drosophila. O método de quantificação produz perfis de expressão gênica para diferentes embriões, em fases distintas do ciclo de desenvolvimento embrionário.

## II. Drosophila melanogaster

A *Drosophila melanogaster* é um inseto díptero, com aproximadamente $3mm$ de comprimento, popularmente conhecido como a "mosca da fruta" ou "mosca do vinagre". Esta espécie é atualmente um dos organismos mais estudados na biologia moderna, por facilitar a investigação de muitos processos celulares e de desenvolvimento, comuns aos eucariotos superiores, incluindo humanos [5].

Atualmente, vem ganhando atenção da comunidade o estudo do plano sagital (Fig. 1) da *Drosophila*, sendo notadamente importante para a realização de variadas investigações biológicas [6]. Este tipo de imagem expressa

o plano focal dentro da amostra, possibilitando a visualização de células e tecidos, localização de genes e proteínas, e visualização da dinâmica de expressão gênica em embriões fixados ou *in vivo* [7], [8].

Imagens sagitais apresentam algumas vantagens quando comparadas à imagens da superfície dos embriões, como a possibilidade de capturar sinais de expressão de RNAs e proteínas em diferentes planos focais. Outra vantagem é a preservação física dos embriões, uma vez que se torna possível maximizar as regiões dos núcleos e sua vizinhança sem precisar achar as amostras de embriões [6]. Também é importante mencionar que, o estudo do eixo dorso-ventral requer necessariamente o uso deste tipo de imagem. Este trabalho visa abordar o plano sagital dos embriões de *Drosophila melanogaster*.

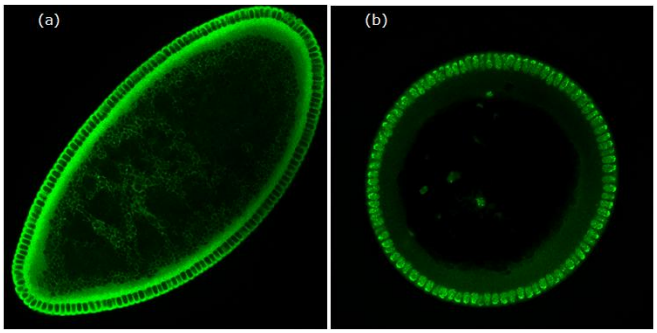



Figura 1. Exemplo de imagem referente ao plano sagital do embrião de Drosophila. Padrão de expressão do gene *Hunchback*. (a) Imagem exibindo o eixo Anterior-Posterior (A↔P). (b) Imagem do eixo Dorso-Ventral (D↕V).

## III. Método de quantificação proposto

A análise de imagens relativas ao plano sagital de embriões da *Drosophila*, para fins de avaliação e quantificação das proteínas e RNAs, não é uma tarefa trivial, já que envolve a identificação e estudo simultâneo de um grande número de genes [9]. Tal procedimento é normalmente realizado por ferramentas computadorizadas, as quais incluem normalmente, etapas de (1) pré-processamento, (2) segmentação e (3) análise.

### *A. Pré-processamento*

A etapa (1) pré-processamento visa preparar as imagens para a extração dos níveis de intensidade de fluorescência de RNAs e proteínas em cada embrião.

Aqui, as imagens são tratadas por meio de técnicas tradicionais de Processamento Digital de Imagens, a fim de compensar a não-uniformidade da iluminação e tratar o ruído existente. Para a remoção do ruído localizado nas extremidades dos núcleos celulares aplicou-se o filtro de suavização Gaussiano [10]. Tais ruídos são caracterizados pela ocorrência aleatória de valores de luminosidade significativamente diferentes dos valores de pixel que constituem os núcleos. Para tratar a iluminação irregular dos núcleos celulares e do fundo, duas técnicas de realce foram utilizadas, sendo o Realce Linear e Filtro morfológico "Top-hat" [11], [10], [6].

Ainda nesta etapa, busca-se padronizar a base de dados, ou seja, preparar as imagens coletadas de modo que cada uma contenha somente o embrião de interesse. Para realização dessa tarefa utilizou-se uma abordagem simples, com base em operações morfológicas e nas técnicas Transformada Distância (TD) e limiarização, tal como realizado por Janssens et al. [12]. A partir de operações de dilatação, erosão e preenchimento é possível remover os objetos indesejados em função de seu tamanho. O processo pode ser melhor compreendido por meio da Fig. 2.

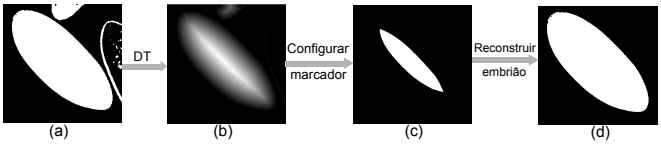



Figura 2. Etapas para isolamento do embrião principal. Em (a) é exibida a imagem binarizada, após remoção de detalhes espúrios e preenchimento do embrião. Em (b) apresenta-se o resultado da transformada distância, o qual é processado para definir o marcador em (c). Em (d) é ilustrado o resultado obtido após a reconstrução morfológica.

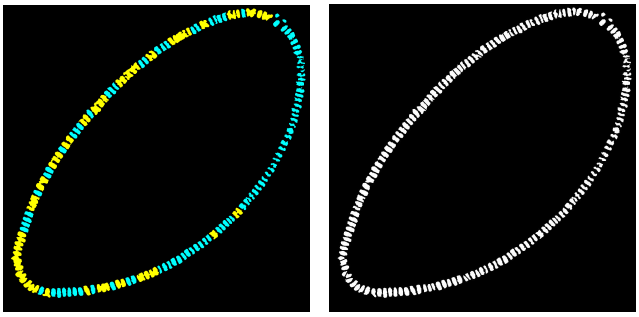
### *B. Segmentação*

Após o pré-processamento (passo 1), as imagens estão prontas para o processo de extração dos níveis de intensidade da fluorescência do citoplasma basal. Isto é conseguido por meio de máscara nuclear, ou seja, imagem binária em que os pixels com valor 1 (um) representam os núcleos celulares, e os pixels com valor 0 (zero) as regiões fora destes objetos e o fundo da imagem (como ilustrado na Figura 3(b)). A criação desta máscara é feita utilizando métodos de segmentação, e é fundamental para mostrar onde cada núcleo do embrião está localizado.

Além disso, a precisão da máscara nuclear representa um fator crítico para operações subsequentes, tais como, quantificação e visualização dos padrões de expressão gênica. Por exemplo, a máscara nuclear pode ser usada para quantificar os dados sobre a expressão gênica em cada núcleo do embrião ou para caracterizar a quantidade de produtos gênicos no interior dos núcleos [13].

No entanto, essa tarefa não é trivial devido a algumas dificuldades intrínsecas do processo de segmentação, e também pelo fato de imagens confocais apresentarem algumas características que dificultam obter uma segmentação eficiente, tais como, a iluminação irregular, ruídos de aquisição, contornos nucleares mal definidos e sinais de fluorescência indesejados na vizinhança das regiões de interesse. Estes fatores afetam os resultados das tradicionais técnicas de segmentação e não proporcionam a separação desejada dos núcleos.

Portanto, o uso de métodos específicos torna-se necessário. O presente trabalho aborda um método de segmentação publicado em [6], que aproveita o conhecimento *a*



(a) Otsu Mask (b) Nuclear Mask

Figura 3. Máscara Otsu e máscara de segmentação nuclear. (a) Imagem binária após o *Thresholding Otsu*. A cor amarelo destaca os componentes formados por núcleos agregados. A cor ciano mostra os núcleos corretamente identificados. Em (b) a respectiva máscara de segmentação nuclear, após aplicação do método de segmentação proposto em [6].

*priori* sobre os objetos de interesse da imagem. A estratégia é realizada em dois passos principais, (a) binarização e (b) pós-processamento.

Num primeiro momento (a), o método aborda a técnica *Otsu Thresholding* [14] a fim de realizar uma segmentação inicial nas imagens da base. Esta técnica mostrou-se rápida e regular, entretanto insuficiente para obter a correta segmentação dos núcleos celulares, tendo em vista a grande parcela de núcleos agregados (grudados), como ilustrado na Figura 3(a)). Desta forma, visando melhorar os resultados da binarização propõem-se a etapa (b) pós-processamento.

O pós-processamento baseia-se na análise de curvatura dos objetos binarizados e visa encontrar os pontos extremos de mínimos locais no contorno desses objetos, os quais caracterizam posições candidatas para divisão dos núcleos agregados. Esta abordagem é normalmente aplicada sobre o contorno paramétrico dos objetos de interesse [6]. O contorno $C(t)$ foi estimado a partir do algoritmo Seguimento de Contorno [15], o qual define o contorno do objeto em termos de (x(t),y(t)).

A curvatura $k$ para cada elemento do contorno discreto $C(t)$ pode ser obtida de acordo com Eq. (1), onde $x'(t)$ e $y'(t)$ são as derivadas de primeira ordem com respeito à $x$ e $y$, e $x''(t)$ e $y''(t)$ são as derivadas de segunda ordem em relação a $x$ e $y$, respectivamente [15].

$$k(t) = \frac{x'(t)y''(t) - y'(t)x''(t)}{(x'(t)^2 + y'(t)^2)^{\frac{2}{3}}}. \quad (1)$$

A natureza discreta deste cálculo normalmente gera sinais ruidosos. Para contornar este problema, o método aborda um filtro Gaussiano (pela inclusão do termo $G(f,\sigma) = exp(-2(\sigma\pi f)^2)$) na equação de curvatura (Eq. (1)). A adição deste termo é capaz de reduzir (ou até mesmo eliminar) a influência de ruídos e sinais indesejados. Ao utilizar as propriedades da Transformada de Fourier para cálculo das derivadas, aliado ao filtro de suavização Gaussiano, a curvatura pode ser estimada de forma precisa, de modo que os picos de curvatura correspondentes aos mínimos locais do contorno de agregados nucleares são identificados de forma eficaz. A estimativa de curvatura suavizada foi então calculada para todo componente conexo com dois ou mais núcleos agregados. Na abordagem utilizada, tais componentes foram identificados em função de seus tamanhos [6].

A partir disso, a ideia consiste em traçar retas entre os pontos extremos de mínimos locais, formando assim fronteiras de divisão entre os objetos agregados. A mesma ideia foi utilizada no trabalho de Beletti et al. [16], o qual usou a estimativa de curvatura na análise de imagens referentes a espermas, a fim de separar a cabeça da cauda.

Para traçar as retas, inicialmente é necessário identificar todos os pontos de mínimos relativos (ou mínimos locais) nos valores de curvatura $k(t)$ calculados. Entre dois pontos de máximo relativo, há sempre um mínimo relativo, e vice-versa. A partir desta análise é possível encontrar os pontos extremos desejados [6].

Após encontrá-los, é necessário estabelecer a correspondência entre eles. Isto é, para cada ponto de mínimo local no contorno da forma analisada, deve-se encontrar um ponto no lado oposto que esteja na mesma região de unificação de seus núcleos agregados. Este processo consiste em estabelecer os pontos inicial e final, a partir dos quais deve-se traçar fronteiras de divisão para separar núcleos vizinhos.

O modelo de segmentação permitiu gerar contornos fechados e regulares, para cada núcleo contido nas imagens, fator essencial para a construção precisa da máscara nuclear. Em geral, o problema de núcleos agregados e supersegmentados foram eliminados, ou amplamente minimizados. Maiores detalhes referentes aos parâmetros e técnicas utilizadas podem ser encontrados em [6].

### C. Quantificação

A partir das máscaras nucleares tem-se a etapa (3). Neste momento é possível determinar o perfil de expressão gênica associado às regiões intranucleares ou citoplasmáticas dos embriões de Drosophila.

A maioria dos trabalhos existentes na literatura retratam métodos de quantificação relativos à região intranuclear, os quais visam mensurar os níveis de proteína por núcleo contidos na blastoderme de embriões da *Drosophila melanogaster* [12]. Por outro lado, essa pesquisa aborda um método de alto desempenho que permite quantificar a expressão gênica da região do citoplasma basal no nível de resolução celular e sem perda de informação espacial.

A principal diferença entre os padrões de expressão de proteínas e RNAs de fatores transcritos é a localização do RNA, que não se restringe apenas à região dos núcleos, mas inclui também a região do citoplasma.

Para analisar a região basal do citoplasma foi inicialmente necessário definir o entorno da região de cada núcleo. Para isso, aplicou-se na máscara nuclear rotulada

(Fig. 4(c)) a operação geométrica transformada distância (TD), com base na métrica euclidiana. O resultado dessa técnica permitiu estender a região dos núcleos celulares nas direções basal e apical. O tamanho da região estendida foi pré-estabelecido com base na análise visual do citoplasma expresso em várias amostras de imagens. Quanto maior o tamanho pré-definido, maior será a região quantificada do citoplasma.

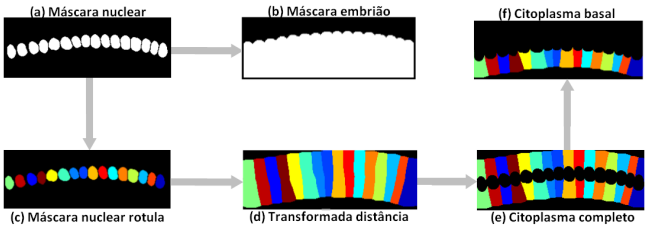



Figura 4. Visão geral do método utilizado para captura do citoplasma basal, ilustrado em um pequeno trecho de imagem sagital. Em (a) é possível visualizar a máscara nuclear. Em (d) é exibido o resultado da operação geométrica TD, aplicada sobre a imagem rotulada (c). Uma operação de multiplicação entre a máscara embrião (b) e a região filtrada da TD (e) resulta no citoplasma basal (f).

É importante mencionar que a região é estendida de modo a propagar o rótulo de cada núcleo [15], operação importante para estabelecer a região citoplasmática individual para cada núcleo. O processo final permite gerar a máscara *Transformada distância (TD)* (Fig. 4(d)). A partir desta imagem, o objetivo é isolar a região de interesse, isto é, citoplasma basal (fundamental para o processo de quantificação proposto) e desconsiderar as regiões dos núcleos celulares e citoplasma apical.

As regiões dos núcleos celulares são eliminadas ao ignorar os pixels da TD localizados a uma distância inferior a 2. A solução encontrada para eliminar a região apical foi combinar a *Máscara embrião* (Fig. 4(b)) com *Citoplasma completo* (Fig. 4(e)), por meio de multiplicação entre as matrizes das respectivas imagens. A *Máscara embrião* utilizada trata-se uma imagem binária em que os pixels do fundo da imagem (região externa do embrião) são assinalados para 0 (zero), e todo o embrião (regiões dos núcleos juntamente com o interior do embrião) são rotulado como 1 (um). Ela pode ser obtida a partir de operações morfológicas realizadas sobre a *Máscara nuclear* (Fig. 4(a)) .

A combinação entre as Figs.4(b) e 4(e) permitiu anular os pixels da região apical, tendo em vista sua multiplicação pelos zeros da matriz *Máscara embrião*. O resultado desse processo permite obter a máscara desejada, isto é, *Citoplasma basal* (Fig. 4(f)). Por fim, ela é utilizada para quantificar a média dos valores de pixels em cada região do citoplasma basal, de cada imagem calculada. Isto foi realizado multiplicando, ponto a ponto, a máscara *Citoplasma basal* pela imagem original do embrião, a qual contém os dados de expressão do RNA.

A Fig. 5 ilustra o resultado final do processo descrito em uma imagem arbitrária da base de dados. O contorno do citoplasma basal detectado é sobreposto à imagem original do embrião, a qual contém os sinais de fluorescência do RNA transcrito. Dessa forma, fica claro perceber a captura isolada do citoplasma basal para cada núcleo contido no embrião da Drosophila.

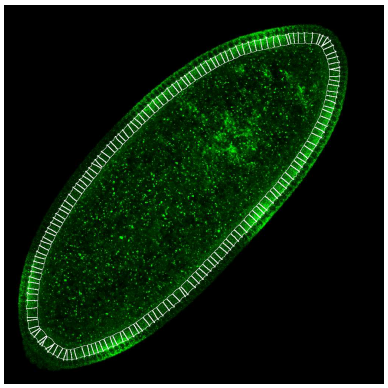


Figura 5. Resultado do processo de marcação da região do citoplasma basal, sobreposto à imagem original do embrião de Drosophila. Fica notório a identificação isolada da região do citoplasma para cada núcleo celular.

## IV. Experimentos

As imagens utilizadas neste trabalho foram adquiridas no *Instituto Nacional de Metrologia, Qualidade e Tecnologia (INMETRO)*, por meio da técnica Microscopia confocal, em que foi utilizado o microscópio confocal Leica TCS SP5. As células de interesse foram capturadas nos canais do DAPI, "proteína azul florescente", tal que, quando introduzida na célula, se liga aos núcleos, tornando-os fluorescentes quando iluminados por um laser. Cada imagem foi capturada no formato *TIFF* com dimensão $1024 \times 1024$ pixels e representadas numa escala de 256 níveis de cinza. Todas as imagens referem-se a padrões de expressão gênica referentes ao plano sagital e 14º período do ciclo mitótico de embriões de Drosophila. Os sinais de expressão coletados referem-se a proteínas e/ou RNAs transcritos do gene *Hunchback*.

## V. Resultados

As máscaras nucleares resultantes da etapa de Segmentação permitem determinar a intensidade média de proteína ou RNA nas regiões intranuclear e nas regiões do citoplasma dos padrões de expressão gênica.

Nesta seção, apresentam-se os resultados da análise quantitativa dos perfis de expressão do RNA *Hunchback (Hb)*. Esses perfis são extraídos de citoplasma basal, ao longo do eixo AP (1a), em imagens sagitais de embriões de Drosophila.

A Fig. 6 mostra um padrão de expressão arbitrário contido na base de imagens, assim como, o respectivo perfil de intensidades média do RNA *Hunchback*. O processamento é tal que as intensidades de fluorescência apresentadas no

gráfico correspondem às intensidades médias emitidas por cada região do citoplasma associada ao núcleo.

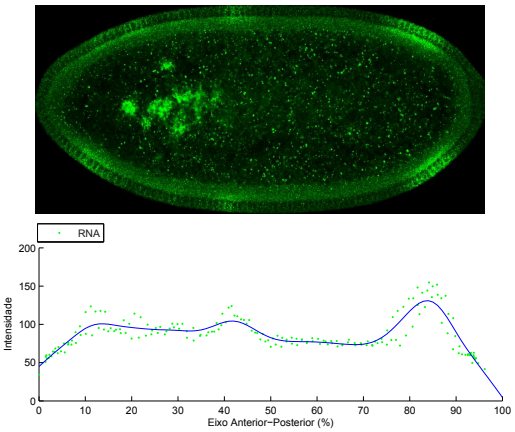



Figura 6. Perfil de intensidade média do RNA *Hunchback* referente a uma imagem aleatória da base.

O gráfico da Fig. 7 mostra os perfis de expressão do RNA de Drosophila para o conjunto total de imagens processadas. Através dele é possível perceber o perfil de distribuição do RNA de *Hunchback* assumido no citoplasma, ao longo do eixo AP das imagens sagitais.

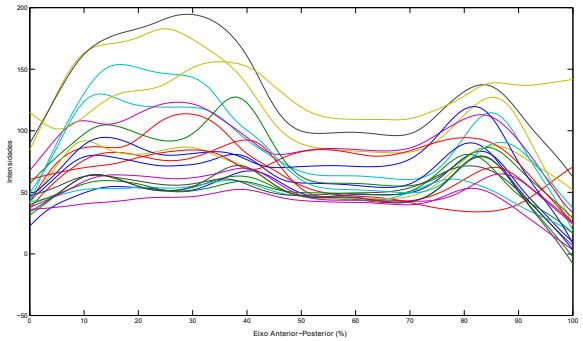



Figura 7. Perfil de expressão do RNA de Drosophila assumido no citoplasma, ao longo do eixo AP, para o conjunto total de imagens processadas. Cada curva representa o perfil de uma imagem.

A Fig. 8 demonstra o Gráfico de Médias referente ao conjunto de imagens com sinal do RNA de *Hunchback*. Esse gráfico exibe as intensidades médias para cada ponto das curvas exibidas na Fig. 7. Por meio dele é possível analisar o padrão de expressão do RNA de *Hunchback* localizado no citoplasma, para o conjunto imagens analisadas. Cada barra na cor preta evidencia um intervalo de desvio padrão (mais/menos) em torno da respectiva média.

## VI. Conclusão

Neste artigo foi descrito um método automático para quantificação dos níveis médios de RNA (por citoplasma basal) em embriões de *Drosophila melanogaster*. Os resultados mostram um grande potencial para sistemas de

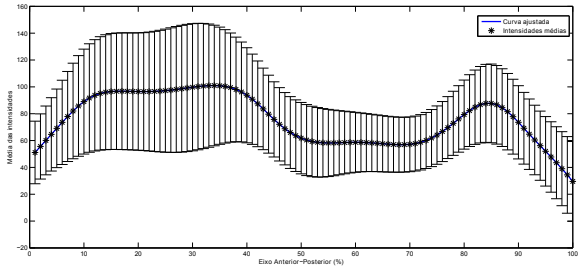



Figura 8. Gráfico de Médias para o conjunto de imagens com sinal do RNA de *Hunchback*. Nele são mostrados as intensidades médias para cada ponto das curvas, com intervalos (mais/menos) dos respectivos desvios-padrões.

quantificação e visualização de imagens de padrões de expressão, e demonstram o grau de precisão e confiabilidade do método apresentado.

A metodologia proposta é capaz de gerar perfis de intensidade relativos às expressões gênicas localizadas na região do citoplasma basal, ao longo dos eixo AP, em imagens sagitais de *Drosophila*. A principal contribuição em relação à propostas similares é a quantificação de RNA não restrita aos núcleos, mas também extensa ao citoplasma basal e apical. Para superar essa limitação abordaram-se técnicas capazes de propagar o rótulo de cada núcleo à região desejada do citoplasma.

Os resultados obtidos sugerem que novos trabalhos podem ser gerados, sendo importante destacar a fácil adaptabilidade do método para quantificar os níveis de expressão de RNA em imagens obtidas do plano focal Dorso-ventral (DV).

Em suma, a abordagem contribui amplamente para a realização de investigações biológicas importantes. Além disso, permite eliminar a subjetividade do processo, quando comparado à simples análise visual.

## Agradecimentos

Os autores agradecem a PROPP-UFU, CNPq, FAPERJ and FAPEMIG pelo suporte financeiro.

## Referências

[1] B. Lewin, *Genes VII*. Porto Alegre: Artmed, 2001, tradução H. B. Ferreira.

[2] A. F. Fumia, "Um modelo para regulação gênica e a diferenciação celular nos estágios iniciais da embriogênese," Ph.D. dissertation, Pós-Graduação em Física Aplicada, Universidade Federal de Viçosa (UFV), 2010.

[3] D. C. Sabiston and C. M. Townsend, *Tratado de Cirurgia: a base biológica da moderna prática cirúrgica*. Rio de Janeiro: Elsevier, 2005, tradução Ione Ferreira Araújo.

[4] N. Pollet, H. A. Schmidt, V. Gawantka, C. Niehrs, and M. Vingron, "In silico analysis of gene expression patterns during early development of xenopus laevis," *Pacific Symposium on Biocomputing*, vol. 5, pp. 440–451, 2000, pac Symp Biocomput.

[5] M. D. Adams, S. E. Celniker, R. A. Holt, C. A. Evans, J. D. Gocayne, and et al., "The genome sequence of drosophila melanogaster," *Science*, vol. 287, no. 5461, pp. 2185–2195, Mar 2000.

[6] D. J. de Sousa, M. A. Cardoso, P. M. Bisch, F. J. P. Lopes, and B. A. N. Travençolo, "A segmentation method for nuclei identification from sagittal images of Drosophila melanogaster embryos," in *International Conferences in Central Europe on Computer Graphics, Visualization and Computer Vision - WSCG*, 2013, pp. 133–142.

[7] O. Grimm, M. Coppey, and E. Wieschaus, "Modelling the bicoid gradient," *Development*, vol. 137, no. 0950-1991 (Linking), pp. 2253–2264, 2010.

[8] S. Wheeler, S. Stagg, and S. Crews, "Multiple notch signaling events control Drosophila cns midline neurogenesis, gliogenesis and neuronal identity," *Development*, vol. 135, no. 18, pp. 3071–3079, 2008.

[9] T. Calsa Junior, V. A. Benedito, and A. V. O. Figueira, "Análise serial da expressão gênica (sage) na genômica de plantas," *Revista Biotecnologia Ciência e Desenvolvimento*, vol. Ano 7, no. 33, pp. 88–100, 2004.

[10] R. Gonzalez and R. Woods, *Processamento de imagens digitais*, 3rd ed. Prentice Hall, 2010.

[11] C. Solomon and T. Breckon, *Fundamentals of Digital Image Processing: A Practical Approach with Examples in Matlab*. Wiley-Blackwell, 2010.

[12] H. Janssens, D. Kosman, C. Vanario-Alonso, J. Jaeger, M. Samsonova, and J. Reinitz, "A high-throughput method for quantifying gene expression data from early drosophila embryos," *Development genes and evolution*, vol. 215, no. 7, pp. 374–381, 2005.

[13] A. Pisarev, E. Poustelnikova, M. Samsonova, and J. Reinitz, "Flyex, the quantitative atlas on segmentation gene expression at cellular resolution," *Nucleic Acids Research*, vol. 37, no. suppl 1, pp. D560–D566, 2009.

[14] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 9, no. 1, pp. 62–66, Jan. 1979.

[15] L. F. Costa and R. Cesar, *Shape classification and analysis: theory and practice*, 2nd ed. CRC Press, 2009.

[16] M. E. Beletti, L. F. Costa, and M. P. Viana, "A spectral framework for sperm shape characterization," *Computers in Biology and Medicine*, vol. 35, no. 6, pp. 463–473, 2005.

# Local Method for Lens Distortion Correction Applied to Stereo Vision

Osmando Pereira Junior
Instituto Federal do Triângulo Mineiro
Paracatu - MG - Brasil
Email: osmandoj@gmail.com

Joceli Mayer
Digital Signal Processing Laboratory
Universidade Federal de Santa Catarina
Florianópolis - SC - Brasil Email: joceli.mayer@lpds.ufsc.br

***Abstract*—Lens distortion is one of the main factors that limits the accuracy of stereo vision system reconstruction. We propose a new method for correction of the lens distortion by applying compensation to each region of an image. Our method splits the image into smaller regions and compensates for each region for a fixed lenses model order. When compared to the conventional method, which models the entire image with only one model, our approach provides better compensation and reduce the depth error as show in the experiments with synthetic data.**

## I. INTRODUÇÃO

Visão estereoscópica é o processo de estimação da informação de profundidade de uma cena, ou de um objeto em particular, por meio da análise de duas imagens capturadas em diferentes pontos de vista, usando um modelo apropriado de câmera. A aquisição das imagens pode ser realizada simultaneamente ou em diferentes instantes de tempo e pela mesma câmera ou por diferentes dispositivos de captura de imagem [1], [2].

As câmeras permitem uma rica representação da cena quando comparadas a outros sistemas, como *laser*, *radar* e *sonar*, sendo cada vez mais usadas em aplicações de robótica móvel e do setor automotivo, tais como navegação autônoma e detecção de objetos e obstáculos [3], [4]. Sistemas de visão estereoscópica são também utilizados em aplicações de sensoriamento remoto e de metrologia [5], e são caracterizados por três etapas principais: calibração das câmeras, registro de imagens (também conhecido por correspondência entre os *pixels*) e reconstrução tridimensional (3D) [1], [2], [6], [7].

A calibração consiste em inferir sobre os parâmetros característicos da câmera, os quais são classificados em intrínsecos e extrínsecos [1]. É uma etapa fundamental na determinação do desempenho de sistemas estereoscópicos, uma vez que o erro residual de calibração tem influência sobre o erro de reconstrução da informação tridimensional da cena [8], [9].

O processo de calibração é comprometido por diferentes distorções, dentre as quais destaca-se a distorção das lentes [10], [11]. Em aplicações que requerem maior precisão, é necessário utilizar técnicas de compensação da distorção da lente, pois quando a distorção da lente não é negligenciável, a não compensação desta pode resultar em erro de calibração relevante [12], [13].

*Brown* [12], [14] propôs, na década de 70, um modelo de distorção da lente em que são consideradas componentes radial, deslocamento do ponto principal e prisma fino. Este modelo está consolidado na literatura, sendo o mais conhecido e utilizado. É recorrente a utilização de métodos iterativos, como *bundle adjustment* [12], para melhorar a estimação dos parâmetros de distorção da lente.

*Tsai* [15] afirma que, para aplicações industriais de visão computacional, é suficiente considerar apenas a componente radial e de $2^a$ ordem, sendo que modelos mais elaborados, além de não melhorarem a resposta do sistema, podem causar instabilidade numérica.

Alguns autores [9], [16] encontram suporte no trabalho de *Tsai* e limitam-se a compensar a distorção da lente usando modelos de baixa ordem e considerando apenas a distorção radial. No entanto, para lentes de baixa qualidade ou de alta distorção, ou ainda para aplicações que requerem maior precisão, é necessário considerar componente radial de $4^a$ ou até mesmo $6^a$ ordem [6], [14].

*Wang et al.* [8] afirmam que considerar apenas a componente radial não é suficiente para uma boa correção da distorção da lente, haja vista que as distorções radial e tangencial são dependentes uma da outra, e mostraram, empiricamente, que a consideração da componente tangencial implica significativamente na redução do erro residual de calibração.

*Hartley & Kang* [11] afirmam que o modelo convencional não é adequado para corrigir lentes com distorção muito elevada, como a lente *fish-eye*, ademais, afirmam que métodos iterativos possuem algumas desvantagens: tempo de processamento, possibilidade de não convergência, inicialização dos parâmetros e definição do critério de parada (convergência).

Os métodos conhecidos e supra-citados corrigem a distorção da lente homogeneamente sobre toda a imagem. Visando aumentar o desempenho de sistemas de visão estereoscópica de baixo custo, que utilizam lentes de baixa qualidade, consequentemente alta distorção, propõe-se, neste trabalho, um novo método de compensação da distorção da lente, denominado Método Local de Correção da Distorção da Lente, o qual considera correção heterogênea da distorção sobre a imagem. Fundamenta-se na estimação de um conjunto de coeficientes de correção da distorção para cada região da imagem.

A avaliação do sistema, feita por simulação com imagens sintéticas, indica que a aplicação do método proposto possibilita obter erro de reconstrução menor que o obtido pela aplicação do método convencional.

Este artigo está organizado da seguinte forma: introduz-se, na Seção II, os modelos matemáticos utilizados para a formação da imagem e para a reconstrução 3D. Na Seção III, apresentam-se o modelo de distorção da lente e o modelo convencional de correção. O método proposto é apresentado na Seção IV. Apresenta-se, na Seção V, a metodologia para simulação do sistema e comparação do método proposto com o método convencional. Reserva-se à Seção VI a apresentação dos resultados e a discussão acerca deles. As conclusões do artigo são descritas na Seção VII.

## II. Modelo Ideal do Sistema de Visão Estereoscópica

Em sistemas de visão estereoscópica, utiliza-se um conjunto binocular de câmeras, em que cada uma delas captura uma imagem da mesma cena, ou de um objeto em particular da cena, de diferentes pontos de vista, resultando em um par de imagens estereoscópicas.

Uma vez capturadas as imagens e conhecidas as coordenadas dos *pixels* homólogos nas imagens da esquerda e da direita (par de imagens estéreo), é possível estimar a posição do ponto tridimensional de origem. Para tal, faz-se necessário determinar um modelo de câmera.

Neste artigo, adotou-se o modelo de câmera *pinhole* com inclusão das dimensões dos *pixels* nas direções $u$ e $v$ do plano do sensor (em [6], [7], esse modelo é referenciado como modelo de câmera *CCD*).

Sendo um ponto no espaço tridimensional representado por $\mathbf{M} = [X\,,\,Y\,,\,Z]^T$, a projeção de $\mathbf{M}$ no plano da imagem é representada por $\mathbf{m}$. O índice '$w$' ($\mathbf{v}_w$) é usado para denotar um ponto representado no sistema de coordenadas 3D do mundo, $R_w$, cuja unidade de medida é *mm*, enquanto o índice '$i$' ($\mathbf{v}_i$), para denotar um ponto 2D no sistema de coordenadas da imagem, $R_i$, medido em *pixel*. Utiliza-se $\tilde{\mathbf{v}}$ para representar um ponto em coordenadas homogêneas ($\tilde{\mathbf{v}} = [\mathbf{v}^T\ 1]^T$) e $\hat{\mathbf{v}}$ para representar um múltiplo real de $\tilde{\mathbf{v}}$ ($\hat{\mathbf{v}} = a\,\tilde{\mathbf{v}}$), em que $a$ corresponde a um fator de escala. Utiliza-se os índices '*e*' e '*d*' para representar pontos nas imagens da esquerda $I_e$ e da direita $I_d$, respectivamente.

A projeção do ponto tridimensional $\mathbf{M}_w$ no plano de imagem é dada pelas Equações (1) e (2) [1], [6], [7].

$$\hat{\mathbf{m}}_{i_{[pixel]}} = \begin{bmatrix} \eta_u & 0 & 0 \\ 0 & \eta_v & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f & 0 & u_0 \\ 0 & -f & v_0 \\ 0 & 0 & 1 \end{bmatrix} [\mathbf{R}|\mathbf{t}]\ \tilde{\mathbf{M}}_w \qquad (1)$$

$$\tilde{\mathbf{m}}_i = \frac{\hat{\mathbf{m}}_i}{\hat{m}_{i_3}} \qquad (2)$$

em que ($\eta_u\,,\,\eta_v$) são os fatores de escala do *pixel* nas direções $u$ e $v$ do sensor de imagem; $f$ é a distância focal da câmera; ($u_0\,,\,v_0$) são as coordenadas do ponto principal da imagem; $\mathbf{R}$ é a matriz de rotação do sistema de coordenadas do mundo $R_w$ para o sistema de coordenadas da câmera $R_c$ [1], [17]; $\mathbf{t}$ é a translação do centro de projeção da câmera em relação a $\mathbf{O}_w$, origem de $R_w$, representada em $R_c$ ($\mathbf{t} = -\mathbf{R}\,\mathbf{C}_w$).

Uma vez conhecidos os parâmetros intrínsecos e extrínsecos das câmeras e o par de *pixels* homólogos ($\mathbf{m}_{e_w}\,,\,\mathbf{m}_{d_w}$), projeções do ponto $\mathbf{M}_w$ nas imagens $I_e$ e $I_d$, respectivamente, pode-se estimar as coordenadas do ponto tridimensional $\mathbf{M}_w$ por meio da Equação (3).

$$\begin{aligned} \mathbf{M}_w &= \lambda_1\ (\mathbf{m}_{e_w} - \mathbf{C}_{e_w}) + \mathbf{C}_{e_w} \\ \mathbf{M}_w &= \lambda_2\ (\mathbf{m}_{d_w} - \mathbf{C}_{d_w}) + \mathbf{C}_{d_w} \end{aligned} \qquad (3)$$

sendo $\lambda_1$ e $\lambda_2$ fatores de escala calculados pela Equação (4); e $\mathbf{m}_w$, projeção de $\mathbf{M}_w$ no plano de imagem, representado no sistema de coordenadas do mundo $R_w$, é dado pela Equação (5).

$$\begin{aligned} &[\lambda_1 \quad \lambda_2] \left( \begin{bmatrix} \mathbf{m}_{e_w}^T \\ \\ -\mathbf{m}_{d_w}^T \end{bmatrix} - \begin{bmatrix} \mathbf{C}_{e_w}^T \\ \\ -\mathbf{C}_{d_w}^T \end{bmatrix} \right) + \\ &+ [1 \quad 1] \begin{bmatrix} \mathbf{C}_{e_w}^T \\ \\ -\mathbf{C}_{d_w}^T \end{bmatrix} = \mathbf{0}^T \end{aligned} \qquad (4)$$

$$\mathbf{m}_w = \mathbf{R}^T \begin{bmatrix} \mu_x & 0 & 0 \\ 0 & \mu_y & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & -u_0 \\ 0 & -1 & v_0 \\ 0 & 0 & f \end{bmatrix} \tilde{\mathbf{m}}_i + \mathbf{C}_w \qquad (5)$$

em que $\mu = \dfrac{1}{\eta}$; as coordenadas do ponto principal, $u_0$ e $v_0$, e do ponto $\mathbf{m}_i$ são medidas em unidades de *pixel*; e a distância focal $f$ é medida em *mm*.

## III. Distorção da Lente

As lentes desempenham um papel importante na formação de imagens, permitindo que a captura das imagens seja feita em menor tempo e com maior definição, no entanto, apresentam imperfeições, o que faz com que tenham um comportamento diferente do previsto para modelos ideais. Tais imperfeições decorrem de fatores inerentes à fabricação da lente e ao processo de montagem da câmera [6], [18].

As imperfeições das lentes reais podem ser caracterizadas pela descentração (não coincidência entre os eixos óptico e mecânico de um elemento de lente), assim como pelo desalinhamento e pela inclinação entre os diferentes elementos de lente e o sensor de imagem [19]. Wang *et al.* [8] citam que são também caracterizadas, conquanto em menor intensidade, pela não-uniformidade do material que constitui a lente.

Em decorrência destas imperfeições, a utilização de lentes implica em distorções geométricas sobre a posição dos *pixels* na imagem, tanto na direção radial quanto tangencial, sendo um dos principais fatores que comprometem o processo de calibração [8], [9].

$$\begin{aligned} \breve{x} &= x + \delta_x(\mathbf{m}) \\ \breve{y} &= y + \delta_y(\mathbf{m}) \end{aligned} \qquad (6)$$

As coordenadas dos *pixels* das imagens reais, ditas coordenadas distorcidas, diferem das ideais por um fator $\delta_x$ e

$\delta_y$, como pode-se notar na Equação (6), em que $x$ e $y$ são as coordenadas ideais de $\mathbf{m}$ (correspondentes à imagem sem distorção do modelo de câmera *pinhole*), representadas em $R_c$; $\check{x}$ e $\check{y}$, as coordenadas distorcidas, também representadas em $R_c$; e $\delta_x(\mathbf{m})$ e $\delta_y(\mathbf{m})$, as componentes da distorção sobre as coordenadas $x$ e $y$, respectivamente.

São vários os modelos de distorção da lente propostos na literatura [8], [20]–[22], sendo que o mais conhecido e utilizado, o qual pode ser denominado modelo convencional [8], considera a distorção da lente $\delta(\mathbf{m}) = [\delta_x(\mathbf{m}) \quad \delta_y(\mathbf{m})]^T$ como uma combinação linear de três componentes: radial ($\delta_r(\mathbf{m})$), deslocamento do ponto principal ($\delta_d(\mathbf{m})$) e prisma fino ($\delta_p(\mathbf{m})$), conforme Equação 7.

$$\delta(\mathbf{m}) = \delta_r(\mathbf{m}) + \delta_d(\mathbf{m}) + \delta_p(\mathbf{m}) \tag{7}$$

### A. Tipos de Distorção

*1) Distorção Radial:* A distorção radial ocorre devido a aberrações esféricas da lente [23], [24]. Ela é proporcional à distância do *pixel* ao centro da imagem (quando se considera que o centro da imagem coincide com o centro da distorção).

De acordo com [10], [14], [25], a distorção radial pode ser modelada como apresentado na Equação (8), que advém da expansão em série de Taylor [7], [14], [26].

$$\begin{aligned}\delta_{r_x}(\mathbf{m}) &= x\,(\check{k}_1||\mathbf{m}||^2 + \check{k}_2||\mathbf{m}||^4 + \ldots + \check{k}_n||\mathbf{m}||^{2\,n}) \\ \delta_{r_y}(\mathbf{m}) &= y\,(\check{k}_1||\mathbf{m}||^2 + \check{k}_2||\mathbf{m}||^4 + ... + \check{k}_n||\mathbf{m}||^{2\,n})\end{aligned} \tag{8}$$

em que $\mathbf{m} = [x \quad y]^T$, $||\mathbf{m}||$ é a norma euclidiana do vetor $\mathbf{m}$, determinada por $||\mathbf{m}|| = \sqrt{x^2 + y^2}$, e $\check{k}_i$ é o i-ésimo coeficiente da distorção radial.

Geralmente, o coeficiente de distorção radial de segunda ordem, $\check{k}_1$, é predominante, tanto que diversos autores consideram desnecessária, em grande parte das aplicações, a utilização dos demais coeficientes, fazendo $\check{k}_i = 0$ para $i = 2, 3, ..., n$, entretanto, quando utiliza-se lentes de alto nível de distorção, como lentes comerciais de baixo custo (desenvolvidas para aplicações outras que não fotogrametria), ou lentes de grande abertura angular (*e.g.*, lentes do tipo *fish-eye*) ou de pequena distância focal, recomenda-se considerar também o coeficiente de $4^a$ ordem ($\check{k}_2$), ou até mesmo o de $6^a$ ordem ($\check{k}_3$) [8]–[10], [14]–[16], [25], [27].

*2) Deslocamento do Ponto Principal:* O deslocamento do ponto principal, também classificado como distorção de descentralização [2], ocorre devido à não colinearidade entre os centros ópticos dos elementos de lente que compõem o conjunto óptico de uma câmera. Contribui tanto com componente radial como com tangencial [25], [28].

$$\begin{aligned}\delta_{d_x}(\mathbf{m}) &= \check{p}_1(2x^2 + ||\mathbf{m}||^2) + 2\check{p}_2 xy \\ \delta_{d_y}(\mathbf{m}) &= 2\check{p}_1 xy + \check{p}_2(2y^2 + ||\mathbf{m}||^2)\end{aligned} \tag{9}$$

A Equação (9) representa as componentes da distorção deslocamento do ponto principal, em que $\check{p}_1$ e $\check{p}_2$ são os coeficientes da distorção de descentralização.

*3) Distorção Prisma Fino:* A distorção prisma fino resulta, principalmente, da inclinação entre os elementos de lente e o sensor de imagem, mas também de imperfeições no *design* e na produção da lente. Assim como o deslocamento do ponto principal, contribui com deformação nas direções radial e tangencial [25], [28].

Pode ser modelada pela Equação (10)

$$\begin{aligned}\delta_{p_x}(\mathbf{m}) &= \check{s}_1||\mathbf{m}||^2 \\ \delta_{p_y}(\mathbf{m}) &= \check{s}_2||\mathbf{m}||^2\end{aligned} \tag{10}$$

em que $\check{s}_1$ e $\check{s}_2$ são os coeficientes da distorção prisma fino.

*4) Modelo de Distorção Total da Lente:* Como apresentado na Equação (7), a distorção da lente é dada pela combinação linear das componentes radial, deslocamento do ponto principal e prisma fino. Substituindo as Equações (8), (9) e (10) em (7), realizando a seguinte mudança de variáveis: $\check{q}_1 = \check{p}_1 + \check{s}_1$ e $\check{q}_2 = \check{p}_2 + \check{s}_2$, e redefinindo $\check{p}_1$ e $\check{p}_2$, tal que $\check{p}_1 = 2\check{p}_1$ e $\check{p}_2 = 2\check{p}_2$, chega-se a (11) [25], [28], expressão que determina as coordenadas distorcidas, $\check{\mathbf{m}}$, a partir das coordenadas ideais, $\mathbf{m}$, e dos coeficientes de distorção.

$$\begin{aligned}\check{x} =& x + x\,\check{k}_1\,||\mathbf{m}||^2 + \ldots + x\,\check{k}_n\,||\mathbf{m}||^{2\,n} + \\ &+ \check{q}_1\,||\mathbf{m}||^2 + x\,(\check{p}_1\,x + \check{p}_2\,y) \\ \check{y} =& y + y\,\check{k}_1\,||\mathbf{m}||^2 + \ldots + y\,\check{k}_n\,||\mathbf{m}||^{2\,n} + \\ &+ \check{q}_2\,||\mathbf{m}||^2 + y\,(\check{p}_1\,x + \check{p}_2\,y)\end{aligned} \tag{11}$$

### B. Correção da Distorção da Lente

Pela Equação (11), é possível estimar as coordenadas distorcidas dos *pixels* da imagem conhecidas as coordenadas ideais e os coeficientes de distorção. No entanto, em casos práticos, é desejável estimar as coordenadas ideais dos *pixels* de uma imagem a partir das coordenadas distorcidas, $\check{\mathbf{m}}$, o que é possível por meio da Equação (12) [6], [9], [10], [14].

$$\begin{aligned}x =& \check{x} + \check{x}\,k_1\,||\check{\mathbf{m}}||^2 + \ldots + \check{x}\,k_n\,||\check{\mathbf{m}}||^{2\,n} + \\ &+ q_1\,||\check{\mathbf{m}}||^2 + \check{x}\,(p_1\,\check{x} + p_2\,\check{y}) \\ y =& \check{y} + \check{y}\,k_1\,||\check{\mathbf{m}}||^2 + \ldots + \check{y}\,k_n\,||\check{\mathbf{m}}||^{2\,n} + \\ &+ q_2\,||\check{\mathbf{m}}||^2 + \check{y}\,(p_1\,\check{x} + p_2\,\check{y})\end{aligned} \tag{12}$$

em que $k_i$ representa o i-ésimo coeficiente de correção da distorção radial e $p_1$, $p_2$, $q_1$ e $q_2$ são os coeficientes de correção das distorções prisma fino e descentralização.

### C. Estimação dos Coeficientes de Correção da Distorção da Lente

Os coeficientes de correção da distorção da lente podem ser estimados pela otimização por mínimos quadrados [16]. Reescrevendo a Equação (12) tal como apresentado na Equação (13), obtém-se um sistema de duas equações para cada ponto característico de uma imagem.

$$\begin{bmatrix} \check{x}\,||\check{\mathbf{m}}||^2 & \dots & \check{x}\,||\check{\mathbf{m}}||^{2\,n} & ||\check{\mathbf{m}}||^2 & 0 & \check{x}^2 & \check{x}\check{y} \\ \check{y}\,||\check{\mathbf{m}}||^2 & \dots & \check{y}\,||\check{\mathbf{m}}||^{2\,n} & 0 & ||\check{\mathbf{m}}||^2 & \check{x}\check{y} & \check{y}^2 \end{bmatrix} \begin{bmatrix} k_1 \\ \vdots \\ k_n \\ q_1 \\ q_2 \\ p_1 \\ p_2 \end{bmatrix} =$$

$$= \begin{bmatrix} x - \check{x} \\ y - \check{y} \end{bmatrix} \quad (13)$$

Conhecidos $M$ pontos de $N$ imagens diferentes e concatenando as equações de todos os $M\,N$ pontos, obtém-se um total de $2\,M\,N$ equações, representadas matricialmente por $\mathbf{D}\,\mathbf{l} = \mathbf{d}$ [16], em que $\mathbf{l} = [k_1 \dots k_n \quad q_1 \quad q_2 \quad p_1 \quad p_2]^T$, $\mathbf{D}$ é a matriz de monômios das coordenadas dos pontos distorcidos e $\mathbf{d}$ é o vetor de diferenças entre as coordenadas ideais e distorcidas dos diferentes pontos.

A estimação do conjunto de coeficientes de correção da distorção da lente, $\mathbf{l}$, é dada pela Equação (14) [16].

$$\mathbf{l} = \left(\mathbf{D}^T\mathbf{D}\right)^{-1}\mathbf{D}^T\mathbf{d} \quad (14)$$

## IV. Método Local de Correção da Distorção da Lente

O método convencional de correção da distorção da lente considera-a homogênea sobre toda a imagem, ou seja, o conjunto de coeficientes de correção da distorção $\mathbf{l} = [k_1 \quad \dots \quad k_n \quad q_1 \quad q_2 \quad p_1 \quad p_2]^T$ é o mesmo para todos os pontos de uma imagem.

No entanto, de acordo com Wang *et al.* [8], a distorção da lente é consequência, dentre outros fatores, de imperfeições na forma da lente e da não-uniformidade do material que a constitui, sendo assim, a consideração de homogeneidade das características da lente é questionável, havendo a possibilidade de uma ou mais regiões da lente possuírem características de distorção particulares, diferentes do restante da lente, ou seja, o conjunto de coeficientes de distorção da lente, consequentemente o de correção da distorção, pode não ser o mesmo para toda a imagem.

Propõe-se, então, um novo método de correção da distorção da lente, denominado Método Local de Correção da Distorção da Lente, baseado em uma correção heterogênea sobre a imagem, a partir da qual estima-se um conjunto de coeficientes de correção da distorção da lente para cada região específica da imagem.

O Método Local consiste em dividir a imagem em $N$ seções e estimar um conjunto de correção da distorção da lente de mesmo modelo, mas de coeficientes diferentes, para cada seção da imagem, gerando a matriz de correção $\mathbf{L} = [\mathbf{l}_1 \quad \mathbf{l}_2 \quad \dots \quad \mathbf{l}_N]$, em que $\mathbf{l}_N$ corresponde ao conjunto de coefientes de correção da distorção da lente da *N*-ésima seção da imagem.

Apresenta-se, na Figura 1, uma máscara de seccionamento da imagem em que utiliza-se $N = 9$ para garantir um número suficiente de pontos de controle em cada seção.

Fig. 1: *Representação das 9 seções em que a imagem é dividida para aplicar o método local de correção da distorção da lente.*

## V. Simulação com imagens sintéticas

O desempenho do método proposto foi comparado ao do método convencional por meio de simulação computacional de um sistema estereoscópico usando imagens sintéticas.

Os parâmetros do sistema estereoscópico foram baseados em um protótipo para medição de superfície de corpos sólidos [5]. Considerou-se que as duas câmeras são idênticas, ou seja, possuem os mesmos parâmetros intrínsecos.

Um total de $34$ imagens foram geradas e divididas em dois conjuntos: conjunto de treinamento, composto por 22 imagens (11 referentes a vistas da câmera da esquerda e as demais, a vistas da câmera da direita); e conjunto de teste, composto por 12 imagens (6 referentes à câmera da esquerda e as demais, à câmera da direita). De acordo com Zhang [16], $11$ imagens para a calibração da câmera é um número adequado para se obter um baixo erro residual de calibração.

O conjunto de treinamento é utilizado para estimar os coeficientes de correção da distorção da lente para as câmeras da direita e da esquerda, ou seja, trata-se do conjunto de imagens de calibração. Enquanto que o conjunto de teste é utilizado para validar os coeficientes estimados a partir do conjunto de treinamento.

São apresentadas, na Figura 2, as imagens da câmera da esquerda que compõem o conjunto de treinamento.

Para simular a distorção da lente, aplicou-se sobre as imagens ideais um conjunto de distorção $\check{\mathbf{l}} = \left[\check{k}_1 \quad \check{q}_1 \quad \check{q}_2 \quad \check{p}_1 \quad \check{p}_2\right]$, baseado no modelo de distorção estimado por *Wang et al.* [8] para uma lente padrão $CCTV$ com distância focal $f$ igual a $8$ *mm*. Os coeficientes de distorção são: $k_1 = -4,41 \cdot 10^{-7}$, $q_1 = -3,5 \cdot 10^{-9}$, $q_2 = 3,4 \cdot 10^{-8}$, $p_1 = 3,1 \cdot 10^{-5}$ e $p_2 = 1,4 \cdot 10^{-7}$.

Na Figura 3, mostra-se um exemplo de uma imagem ideal, em azul, e a respectiva imagem distorcida, em vermelho, resultante da aplicação do modelo de distorção $\check{\mathbf{l}}$.

Dispondo das coordenadas ideais, $\mathbf{m}$, e corrigidas, $\hat{\mathbf{m}}$, dos *pixels* das imagens, determinou-se o erro residual de calibração, $mse$, para os conjuntos de treinamento e de teste. A determinação de $mse$ é expressa na Equação (15).

$$mse = \frac{1}{N_1}\sum_{p=0}^{N_1-1} ||\hat{\mathbf{m}}_p - \mathbf{m}_p|| \quad (15)$$

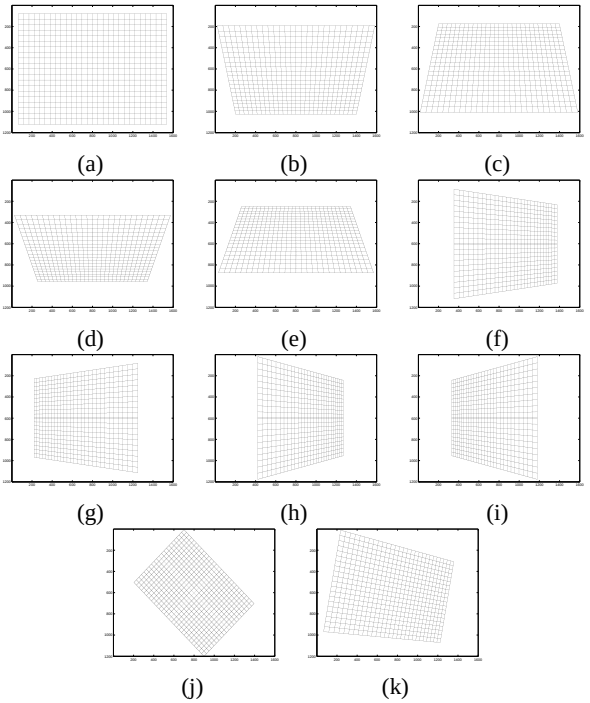



Fig. 2: *Imagens da câmera da esquerda que compõem o conjunto de treinamento.*

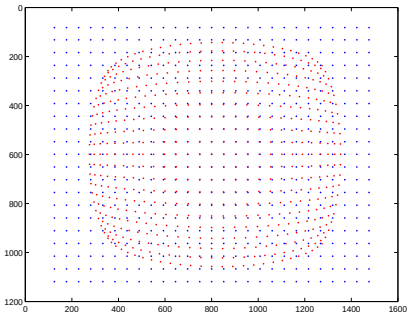


Fig. 3: *Pontos de controle de uma imagem distorcida pelo modelo de distorção* ĭ, *em vermelho, sobrepostos aos pontos de controle da imagem ideal, em azul.*

em que $N_1$ representa o número de pontos de controle do conjunto de imagens.

Com o intuito de determinar o impacto dos métodos de correção da distorção da lente na recuperação da informação 3D, estimou-se, por meio das Equações(3), (4) e (5), a profundidade para os pares de imagens corrigidas.

Dispondo das informações de profundidade ideal, $Z$, e estimadas, $\hat{Z}$, calculou-se o erro de reconstrução, $e_z$, como expresso na Equação (16), para os pares de imagens dos conjuntos de treinamento e de teste.

$$e_z = \frac{1}{N_2} \sum_{p=0}^{N_2-1} |\hat{Z}_p - Z| \qquad (16)$$

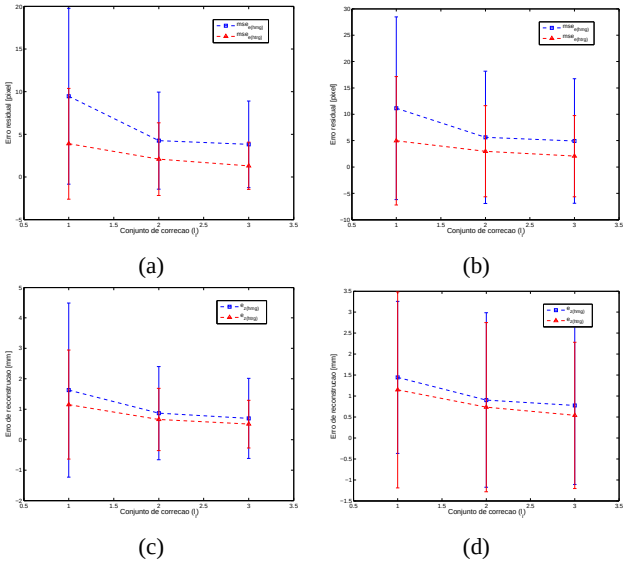



Fig. 4: *Evolução do erro residual e do erro de reconstrução em função da ordem do modelo de correção da distorção da lente, aplicado aos conjuntos (a) e (c) de treinamento e (b) e (d) de teste, corrigidos usando o método convencional e o método local proposto.*

A simulação foi realizada utilizando-se o software *MATLAB*® em um computador com as seguintes especificações: sistema operacional *Linux Ubuntu 11.10*, processador *Intel*® *Core 2 Duo T5850* de $2,16$ GHz (x 2) e memória RAM de 3GB.

TABLE I: Erro residual, $mse_e$ e $mse_d$, em *pixel*, e erro de reconstrução, $e_Z$, em *mm*, para as imagens dos conjuntos de treinamento e de teste distorcidas pelo modelo ĭ.

| **Erro** | **Conjunto** | |
|---|---|---|
| | *Treinamento* | *Teste* |
| $mse_e$ | 44,35 | 51,12 |
| $mse_d$ | 44,16 | 51,12 |
| $e_Z$ | 3,43 | 3,47 |

Apresenta-se na Tabela I, os valores do erro residual, $mse$, das câmeras da esquerda e da direita, e do erro de reconstrução, $e_Z$, calculados para as imagens dos conjuntos de treinamento e de teste, distorcidas pelo modelo de distorção da lente ĭ.

Os resultados obtidos com a aplicação do método convencional e do método proposto de correção da distorção da lente sobre as imagens do conjunto de treinamento e de teste são apresentados na Seção seguinte.

## VI. Resultados

Na Figura 4, apresenta-se a evolução do erro residual e do erro de reconstrução em função do aumento da ordem do modelo de correção da distorção da lente, tanto para o método convencional (correção homogênea) quanto para o método local proposto (correção heterogênea). Em (a) e (b), são apresentados o erro residual de calibração das imagens dos conjuntos de treinamento e de teste, respectivamente, para o método de correção convencional, em azul, e para o método

de correção local, em vermelho; em (c) e (d), apresentam-se os respectivos erros de reconstrução.

Nota-se que o erro residual de calibração e o erro de reconstrução obtidos por meio da aplicação do método local, para ambos conjuntos de imagens, são menores que os obtidos pela aplicação da correção homogênea, tanto em valor médio quanto em desvio padrão (salvo para a correção das imagens de teste usando modelo de correção de ordem 2, $\mathbf{l}_1$ ). Observa-se, ainda, que eles decrescem à medida que se aumenta a ordem do modelo de correção e que o erro residual obtido para correção heterogênea (método local) usando modelo de correção de ordem 2, $\mathbf{l}_1$, é menor que aqueles obtidos pela correção homogênea (método convencional) usando modelo de correção de maior ordem (ordem 6, $\mathbf{l}_3$), para ambos conjuntos de imagens.

Ademais, para o conjunto de treinamento, ao comparar a utilização da correção heterogênea, modelo $\mathbf{l}_3$, à utilização da correção homogênea, também modelo $\mathbf{l}_3$, nota-se uma redução do erro residual de $3,82$ *pixels* para $1,30$ *pixels* e do erro de reconstrução de $0,7$ *mm* para $0,51$ *mm*, sendo que os desvios padrões reduzem de $5,09$ *pixels* para $2,77$ *pixels* e de $1,3$ *mm* para $0,78$ *mm*.

Os erros obtidos para o conjunto de teste são maiores que os obtidos para o conjunto de treinamento. Ao comparar os resultados obtidos da aplicaç do método local com os do método convencional, nota-se uma redução do erro residual de $4,95$ *pixels* para $2,06$ *pixels* e do erro de reconstrução de $0,77$ *mm* para $0,54$ *mm*. Os desvios padrões reduzem de $11,81$ *pixels* para $7,70$ *pixels* e de $1,88$ *mm* para $1,74$ *mm*.

Para aplicações que requerem maior precisão, os resultados obtidos pela aplicação do método local proposto aportam significativa relevância.

## VII. Conclusão

Apresentou-se, neste artigo, um estudo acerca da correção da distorção da lente visando reduzir o erro de reconstrução da informação de profundidade em sistemas de visão estereoscópica. Um novo método de correção da distorção da lente, denominado Método Local de Correção da Distorção da Lente, foi proposto e comparado ao método convencional. A aplicação do método proposto possibilitou obter erro residual e erro de reconstrução menores que os obtidos pela aplicação do método convencional, apresentando significativa relevância principalmente para aplicações que requerem maior precisão.

## Agradecimentos

Os autores agradecem ao Conselho Nacional de Desenvolvimento Científico e Tecnológico - CNPq pelo suporte financeiro.

## References

[1] E. Trucco and A. Verri, *Introductory Techniques for 3-D Computer Vision*. Prentice Hall, 1998.

[2] H. Pedrini and W. R. Schwartz, *Análise de imagens digitais: princípios, algoritmos e aplicações*. Thomson Learning, 2008.

[3] F. A. Moreno, J.-L. Blanco, and J. Gonzalez, "A probabilistic observation model for stereo vision systems: Application to particle filter-based mapping and localization." in *IbPRIA*, 2007.

[4] D. F. Llorca, M. A. Sotelo, I. Parra, M. Ocaña, and L. M. Bergasa, "Error analysis in a stereo vision-based pedestrian detection sensor for collision avoidance applications," *Sensors*, vol. 10, no. 4, pp. 3741–3758, April 2010.

[5] T. Pinto, C. Kohler, and A. Albertazzi, "Regular mesh measurement of large free form surfaces using stereo vision and fringe projection," *Optics and Lasers in Engineering*, vol. 50, pp. 910 – 916, July 2012.

[6] G. Bradski and A. Kaehler, *Learning OpenCV: Computer Vision with the OpenCV Library*, 1st ed. O'Reilly, 2008.

[7] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, 2003.

[8] J. Wang, F. Shi, J. Zhang, and Y. Liu, "A new calibration model of camera lens distortion," *Pattern Recognition*, vol. 41, no. 2, pp. 607–615, February 2008.

[9] D. Xu, Y. F. Li, and M. Tan, "Method for calibrating cameras with large lens distortion," *Optical Engineering*, vol. 45, no. 4, April 2006.

[10] D. G. Aguilera, J. G. Lahoz, and P. R. GonzÁlvez, "An automatic approach for radial lens distortion correction from a single image." *IEEE Sensors Journal*, vol. 11, no. 4, pp. 956–965, April 2011.

[11] R. Hartley and S. B. Kang, "Parameter-free radial distortion correction with center of distortion estimation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 8, pp. 1309–1321, August 2007.

[12] C. Ricolfe Viala and A. Sánchez Salmerón, "Robust metric calibration of non-linear camera lens distortion," *Pattern Recognition*, vol. 43, no. 4, pp. 1688–1699, April 2010.

[13] S.-W. Shih, Y.-P. Hung, and W.-S. Lin, "When should we consider lens distortion in camera calibration," *Pattern Recognition*, vol. 28, no. 3, pp. 447–461, March 1995.

[14] D. C. Brown, "Close-range camera calibration," *PHOTOGRAMMETRIC ENGINEERING*, vol. 37, no. 8, pp. 855–866, 1971.

[15] R. Tsai, "A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses," *RA*, vol. 3, no. 4, pp. 323–344, 1987.

[16] Z. Zhang, "Flexible camera calibration by viewing a plane from unknown orientations," in *The Proceedings of the Seventh IEEE International Conference on Computer Vision*, vol. 1, 1999, pp. 666–673.

[17] C. D. Meyer, *Matrix Analysis and Applied Linear Algebra*. SIAM, 2000.

[18] T. A. Clarke, J. G. Fryer, and X. Wang, "The principal point and CCD cameras," *The Photogrammetric Record*, vol. 16, pp. 293–312, October 1998.

[19] R. G. Willson and S. A. Shafer, "What is the center of the image?" *J. Opt. Soc. Am. A*, vol. 11, no. 11, pp. 2946–2955, November 1994.

[20] G. Garibotto, "A new optical distortion model for multi-camera calibration," in *IEEE International Conference on Advanced Video and Signal Based Surveillance - AVSS*, sept. 2009, pp. 508 –513.

[21] L. Alvarez, L. Gomez, and J. Sendra, "An algebraic approach to lens distortion by line rectification," *Journal of Mathematical Imaging and Vision*, vol. 35, no. 1, pp. 36–50, 2009.

[22] Z. Kukelova and T. Pajdla, "A minimal solution to the autocalibration of radial distortion," in *CVPR07*, 2007, pp. 1–7.

[23] J. R. Meyer-Arendt, *Introduction to classical and modern optics*, 4th ed. Prentice Hall, 1995.

[24] M. V. Klein and T. E. Furtak, *Optics*, 2nd ed. John Wiley & Sons, 1986.

[25] J. Weng, P. Cohen, and M. Herniou, "Camera calibration with distortion models and accuracy evaluation," *PAMI*, vol. 14, no. 10, pp. 965–980, October 1992.

[26] J. Fryer and D. Brown, "Lens distortion for close-range photogrammetry," *PhEngRS*, vol. 52, no. 1, pp. 51–58, 1986.

[27] K. Josephson and M. Byrod, "Pose estimation with radial distortion and unknown focal length," in *IEEE Conference on Computer Vision and Pattern Recognition - CVPR*, Junho 2009, pp. 2419–2426.

[28] J. Wang, F. Shi, J. Zhang, and Y. Liu, "A new calibration model and method of camera lens distortion," in *IEEE International Conference on Intelligent Robots and Systems*, 2006.

# Image Retrieval: Importance and Applications

João Augusto da Silva Júnior
Universidade Federal de Goiás - CAC
Computer Science Department
Catalão - GO, Brazil
jhonaugustjunior@gmail.com

Rodiney Elias Marçal
Universidade Federal de Goiás - CAC
Computer Science Department
Catalão - GO, Brazil
rod@wgo.com.br

Marcos Aurélio Batista
Universidade Federal de Goiás - CAC
Computer Science Department
Catalão - GO, Brazil
marcos@catalao.ufg.br

***Abstract*—This paper presents a brief analysis of the main techniques used for image retrieval, while pointing out the importance of this emerging technology. Due to the alarming growth of the Internet and the high volume of data, we emphasize the technique of CBIR - Content-Based Image Retrieval. For the case study, we developed one application of this technique over an image database using the Euclidean Distance metric.**

**Keywords - content-based image retrieval; euclidean distance, indexing, feature vector;**

## I. Introduction

Advances in data storage and image acquisition technologies have enabled the creation of large image datasets. In this scenario, it is necessary to develop appropriate information systems to efficiently manage these collections. The commonest approaches use the so-called CBIR - Content-Based Image Retrieval systems.

Basically, CBIR systems try to retrieve images similar to a user-defined specification or pattern (e.g., shape sketch, image example). Their goal is to support image retrieval based on content properties (e.g., shape, color, texture), usually encoded into feature vectors. One of the main advantages of the CBIR approach is the possibility of an automatic retrieval process, instead of the traditional keyword-based approach, which usually requires very laborious and time-consuming previous annotation of database images. The CBIR technology has been used in several applications such as fingerprint identification, biodiversity information systems, digital libraries, crime prevention, medicine, historical research, among others [6].

This paper aims to introduce the importance and applications in the field of image retrieval. In particular, we focus in the technique known as Content-Based Image Retrieval. A simple implementation of this technique is then presented so that we can analyse the results.

This article introduces the motivation for image retrieval systems and its importance in Section II. Section III presents some techniques used in image retrieval, while Section IV discusses distances metrics. Some applications that take advantage of the CBIR technologies are discussed in Section V. Section VI explains the algorithm used in the implementation of a basic CBIR system. Next, we discuss experiments and results involving our CBIR system. Section VIII states our conclusions.

## II. Motivation

Due to the increase of online users on the Internet, the amount of collections of digital images have grown continuously during this period, for example, in web applications that allows adding images and digital albums [1]. Also is important to note that the images are globally used. The influence of television, old photographs and games has contributed to this growth as well. Images are increasingly used to convey information, whether one local information, weather, advertising, etc. In this context, it is necessary the development of appropriate systems to manage effectively these collections [6].

Another problem was the complexity of image data, and these data can be interpreted in various ways, thus raising the question of how to work in order to manipulate these data and represent or establish policies to its content. This motivated the birth of the image retrieval area whose goal is try to solve those problems.

## III. Techniques for Image Retrieval

In this section we present two common methods for image retrieval: TBIR - Text-Based Image Retrieval and CBIR - Content-Based Image Retrieval.

### A. Text-Based Image Retrieval

It is an old method, starting in 1970s [8]. This technique requires a text as input to search for image. Example of queries would be "search results for flowers or even "search results for flowers added on 2014-10-05.". So, the keyword may be by image name, date of adding, deleting, modifying and others.

Main problems of the query by text:

- Unexpressed feelings, emotions;
- Many ways of saying the same thing;
- Synonyms and homonyms;
- Misspellings.

### B. Content-Based Image Retrieval

Different from the previous one, *Content-Based Image Retrieval* take as input a query image and the goal is search similar images by color, texture or form, as a comparison. Example of queries of this techniques would be something like "search for results similar to that image containing flowers ".

So, the user owns an image of a flower and the search will return similar images to that query image.

A CBIR system is composed of a query interface for the acquisition of the query image, databases for storing indexing data and metrics, similarity and retrieval system. Figure 1 illustrates these schema:

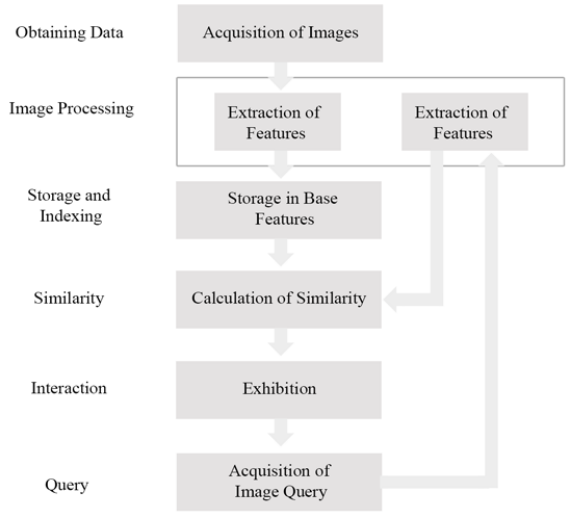



Fig. 1. Basic Steps of CBIR system.

It may be noted that CBIR is a process that may require long processing time due to the amount of images to be analyzed in a database, so comparison between images is made using a set of features extracted from the images [6].

The implementation of such a system requires the extraction and storing of the image features to be compared with the features of the query image. With this flow, the implementation process is more dynamic, since all features have already been stored somewhere.

## IV. Distance Metrics

The calculation of similarity is given by the shortest distance between the feature vectors.

There are several functions responsible for calculating this distance, such as Minkowsky function and Euclidean distance. In practice, the latter is the most widely used measure of dissimilarity.

$$D(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

In this equation $x_i$ and $y_i$ are elements of features vectors $x$ and $y$. Therefore, the vectors must have the same dimensions. The shortest distance vector gives the result of this calculation, that is, the smaller the distance between a vector and another, higher is the similarity for the analyzed image.

The Minkowsky function can be considered a generalization of the Euclidean distance. Its mathematical equation is given by:

$$D(x,y) = (\sum_{i=1}^{n}|x_i - y_i|^P)^{\frac{1}{P}}$$

.

In this equation, the value of P can vary between 1 or 2. We have Euclidean distance in the case when P assumes the value of 2.

## V. Applications

Content-Based Image Retrieval has been used in several applications, such as medicine, fingerprint identification, biodiversity information systems, digital libraries, crime prevention, historical research, among others.

### A. Medical Applications

The number of medical images produced by digital devices has increased more and more. For instance, a medium-sized hospital usually performs procedures that generate medical images that require hundreds or even thousands of gigabytes within a small space of time. The task of taking care of such huge amount of data is hard and time-consuming. That's one of the reasons that has motivated research in the field of Content-Based Image Retrieval. In fact, the medical domain is frequently mentioned as one of the main areas where Content-Based Image Retrieval finds its application. [7].

### B. Biodiversity Information Systems

Biologists gather many kinds of data for biodiversity studies, including spatial data, and images of living beings. Ideally, Biodiversity Information Systems (BIS) should help researchers to enhance or complete their knowledge and understanding about species and their habitats by combining textual, image content-based, and geographical queries. An example of such a query might start by providing an image as input (e.g., a photo of a fish) and then asking the system to *"Retrieve all database images containing fish whose fins are shaped like those of the fish in this photo"*. [6].

### C. Digital Libraries

There are several digital libraries that support services based on image content. One example is the digital museum of butterflies, aimed at building a digital collection of Taiwanese butterflies. This digital library includes a module responsible for content-based image retrieval based on color, texture, and patterns. [6].

In a different image context, [5] present a content-based image retrieval digital library that supports geographical image retrieval. The system manages air photos which can be retrieved through texture descriptors. Place names associated with retrieved images can be displayed by cross-referencing with a Geographical Name Information System (GNIS) gazetter.

## VI. Implementation

The Python programming language was chosen for developing the Content-Based Image Retrieval System that we mention in this work.

Python is a high-level programming language which supports multiple programming paradigms, including object-oriented, imperative and functional programming or procedural styles.

Python was conceived in the late 1980s by Guido van Rossum and has a large standard library, providing tools suited to many tasks, such as scientific computing, text and image processing.

The procedure used in the implementation can be summarized in the following topics:

1) Convert all images in a collection $C$ to grayscale;
2) For each image $I$, apply a function that does pixels intensity counting. This function returns an feature vector where each of its positions contains the amount of intensity referring to each pixel in grayscale.
3) For each $i$ in the feature vector produced in the previous step, apply a normalization function. There are a number of ways for computing such normalization. The one chosen here divide each value of the vector by the highest value of the same vector. This way, the resulting vector only contains values between 0 and 1.
4) Let $Q$ be an query image. Apply to $Q$ all the very same previous functions. Then, apply the Euclidean Distance between the vector of $Q$ and each vector of the images $I$. The images with shorter distances mean that they are the most similar to the query image. We use the Euclidean Distance because it is easy to calculate and it's the most common technique.

Taking in consideration that the dataset for the images may be large and still contains high-definition images, the process for obtaining the feature vectors ends up being infeasible every time a query is requested. Ideally, such a process should be performed only once every time the dataset suffers an update. So, the feature vectors should be stored in a data structure in order to facilitate and further optimize performance and time for the retrieving process. Having this ideia in mind, we decided to divide the implementation in two phases. Each phase correspond to the execution of a python script:

- *generate.py*: this script is responsible for creating and storing in the file *db.txt* all the feature vectors from the images in our dataset. This file *db.txt* is just a regular text file containing two cólon separated columns: the first column has the name of the image being characterized; the second has the normalized feature vector itself.
- *query.py*: Using the Euclidean distance metric, the task of this script is to find the images from the database that are closer (i.e., images that have the lowest distances) to the query image. To accomplish this task, this script uses the data file *db.txt*. The script output is a report in HTML *(HyperText Markup Language)* that is used for easy exhibition of the results. The results consist in 16 images that are closer to the input image (or query image).

## VII. Experiments and Results

In our Content-Based Image Retrieval system experiments, we used the *Corel* [4] image database. This database has 10,800 images belonging into 80 concept groups, e.g., autumn, aviation, bonsai, castle, cloud, dog, elephant, iceberg, primates, ship, stalactite, steam-engine, tiger, train, and waterfall. All the images have an approximated size of 5KB and dimensions 120 x 80 pixels or 80 x 120 pixels.

The first step was to use the script *generate.py* to generate the feature vectors of all the images of our *Corel* database, thus producing the textual file *db.txt*. The time taken for this task using a Intel Corel i5 2.5GHz 16GB RAM machine was approximately 1 minute. Soon after this task, we used the script *query.py* to generate a report in *HTML* format containing the 16 closest images of the image input by using Euclidean distance metric.

Figure 2 shows the query image and figure 3 shows, left to right, all the results in descending order of similarity.

Fig. 2. Query or input image.

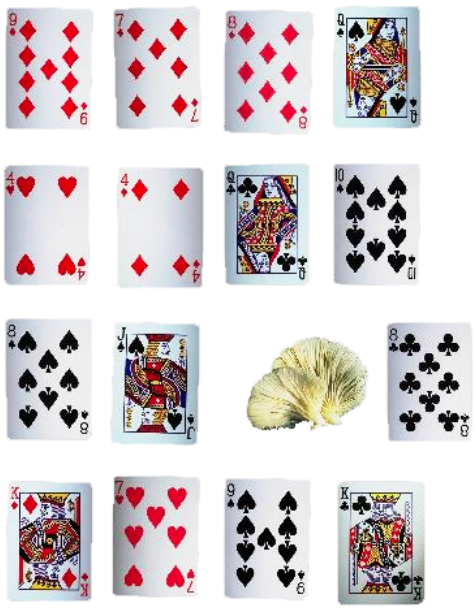


Fig. 3. From left to right: images most similar to the query image by using the Euclidean distance.

The second experiment used the query image shown in figure 4. The results can be seen in figure 5.

Fig. 4. Query image from the second experiment.

Fig. 5. From left to right: images most similar to the query image used in the second experiment.

Third experiment used the query image shown in figure 6 and its results are show in figure 7.

Fig. 6. Query image of the third experiment.

In all experiments, the response time for the retrieval was about 2 seconds.

The results demonstrate that the ranking of the returning images really take in consideration the lowest distances computed based in the intensity counting of the pixels in grayscale. Also, the first image in the result set is the same as the query image. This happens because we have included the query image in our image collection just for the checking if there would be an exact match. This means, then, that the distance computed for identical images is null.

The experiments revealed that, although the technique we used was simple, our Content-Based Image Retrieval system still may return good results when the query image and the imagens in the dataset have a homogeneous aspect.

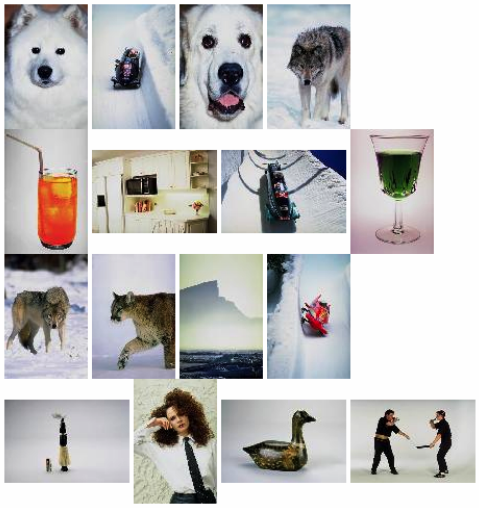


Fig. 7. From left to right: images most similar to the query image of the third experiment.

## VIII. Conclusions

Advances in data storage and image acquisition technologies have enabled the creation of large image collections. Thus, we need appropriate information systems able to efficiently manage such collections. Systems that address this task are commonly known as *Content-Based Image Retrieval Systems* whose operation is basically trying to retrieve images similar to an image sample. For this purpose, parameters such as shape, color, texture, etc. are used, usually encoded in a feature vector.

This paper presented an overview of the field of image retrieval. In special, we emphasized the *Content-Based Image Retrieval Systems*. Concepts, applications and experiments using the method of *Euclidean Distance* for calculating similarity between two images served to elucidate aspects of this area that is gaining more and more importance each day.

As already mentioned in this paper, to optimize the response time of a search system and make it truly scalable on a collection of large images, it is necessary to use a data structure for efficient representation and indexing of the feature vector. In addition, there are several ways to calculate the similarity distances between two images. As a future work, we propose a comparative study of the various measures of distances, as well as the techniques of extraction and indexing of the feature vector.

## Acknowledgments

The authors would like to thank CNPq and FAPEG for the financial support.

## REFERENCES


[1] Santos, Ana Paula de Oliveira - Recuperação de Imagens Mamográficas Baseada em Conteúdo. Trabalho de Conclusão de Curso. Fundação de Ensino Eurípides Soares da Rocha, 2006.

[2] H. Kopka and P. W. Daly, *A Guide to LaTeX*, 3rd ed. Harlow, England: Addison-Wesley, 1999.

[3] John Eakins and Margaret Graham. Content-based Image Retrieval - University of Northumbria at Newcastle

[4] Wang, James Z. and Li, Jia and Wiederhold, Gio. SIMPLIcity: Semantics-Sensitive Integrated Matching for Picture Libraries, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 23, no. 9, pp. 947-963, 2001.

[5] B. Zhu, M. Ramsey, and H. Chen. Creating a Large-Scale Content-Based Airphoto Image Digital Library. IEEE Transactions on Image Processing, 9(1):163-167, January 2000.

[6] R.S. Torres and A.X. Falcão. Content-Based Image Retrieval: Theory and Applications. Revista de Informática Teórica e Aplicada, nro 2, volume 13, 2006, pags 161–185.

[7] Castanon, Cesar Armando Beltran. Recuperação de imagens por conteúdo através de análise multiresolução por Wavelets. São Carlos : Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, 2003. Dissertação de Mestrado em Ciências de Computação e Matemática Computacional.

[8] Dr. Fuhui Long, Dr. Hongjiang Zhang and Prof. David Dagan Feng - Fundamentals of Content-Based Image Retrieval

# Method of Inserting Poisson Noise in Digital Mammography Images to Simulate Radiation Dose Reduction

Lucas R. Borges[1], Helder C. R. de Oliveira[1], Polyana F. Nunes[1], Marcelo A. da C. Vieira[1]
[1]Departamento de Engenharia Elétrica, Universidade de São Paulo, São Carlos, Brasil
{ lucas.rodrigues.borges , heldercro, polyananunes}@usp.br
{mvieira}@sc.usp.br

*Abstract*— **It is well known that a reduction on the radiation dose in digital mammography increases image noise. The urge to optimize image quality vs. radiation dose caused the necessity of obtaining clinical images at different doses. However, acquiring several images from the same patient into different doses results in undesirable additional exposures and serious risk of cancer induction and other health problems. Thus, methods for simulating dose reduction are essential tools for any research in this area. This work investigates the feasibility of simulating dose reduction in digital mammography by calculating the noise present in a lower-dose image and inserting the additional noise in previously acquired clinical images, based on a method originally proposed for conventional radiographic images. Simulated images were compared to real mammography images of a breast phantom acquired at different doses. Results obtained by calculating several image-similarity parameters showed good agreement between simulated and original images for different doses.**

Keywords — Radiation dose reduction; image noise; Poisson noise; noise power spectrum, digital mammography.

## I. INTRODUÇÃO

De acordo com dados da Organização Mundial de Saúde [1], o câncer de mama é considerado uma das principais causas de morte entre mulheres de todo o mundo. Sua detecção no estágio inicial aumenta as chances de cura em até 30% e o método mais indicado para o rastreamento da doença é a mamografia digital por raios-X [2]. É sabido que a exposição excessiva à radiação pode trazer sérios riscos à saúde do paciente, inclusive com a possibilidade de induzir o aparecimento do câncer de mama em algumas das mulheres radiografadas [3,4]. Esse fato tem levado a uma série de pesquisas no sentido de avaliar a possibilidade de se reduzir as doses de radiação utilizadas nos exames mamográficos, principalmente por causa do elevado número de mulheres que participam dos programas de rastreamento do câncer de mama em todo o mundo [5,6].

A principal dificuldade envolvida em uma possível diminuição da dose de radiação é que isso influencia diretamente na qualidade da imagem. A mamografia obtida será mais ruidosa à medida que a exposição do paciente à radiação é reduzida [7]. Nesse sentido, diferentes abordagens têm sido estudadas para a mamografia digital no sentido de reduzir os níveis de exposição dos pacientes à radiação, sem comprometer a qualidade das imagens e do diagnóstico médico [8,9]. Para o desenvolvimento e avaliação desses métodos faz-se necessária a utilização de imagens clínicas com diferentes níveis de ruído, obtidas com diferentes doses de radiação. O problema é que para a aquisição de um conjunto de imagens do mesmo paciente utilizando diferentes doses de radiação torna-se necessário expor esse paciente à doses extras de radiação, o que normalmente não é aceitável na prática clínica devido aos sérios riscos envolvidos nesse processo.

Nesse sentido, esse trabalho propõe o uso de um algoritmo para a geração de imagens mamográficas com diferentes níveis de ruído quântico, simulando uma redução proporcional nas doses de radiação utilizadas na sua aquisição. Isso é feito a partir do cálculo do ruído extra gerado pelo sistema de aquisição quando a dose de radiação é reduzida, utilizando a metodologia proposta por Båth *et al*. [10]. A partir disso, esse ruído extra é inserido na imagem original para produzir uma imagem com as mesmas características de uma imagem adquirida com a dose reduzida, sem a necessidade de exposição extra do paciente à radiação.

## II. MATERIAIS E MÉTODOS

### *A. Metodologia utilizada*

Segundo Båth *et al*. [10], o cálculo do ruído quântico que é incorporado na imagem radiográfica, quando a dose de radiação utilizada na sua aquisição é reduzida, pode ser feita a partir do cálculo da densidade espectral do ruído do sistema de imagem (*NPS - Noise Power Spectrum*) [11] em diferentes doses de radiação.

Considerando que a imagem original é uniforme (todos os pixels possuem a mesma intensidade) e o NPS do equipamento seja conhecido para qualquer dose, o ruído extra que deve ser incorporado na imagem para simular uma redução na dose de radiação pode ser calculado da seguinte forma:

Seja $\mathrm{Im}(x,y)_{\mathrm{orig}}$ uma imagem nas coordenadas x e y obtida na dose original $D_{\mathrm{orig}}$ e $p_{\mathrm{médio,orig}}$ o valor médio dos pixels nesta imagem, pode-se calcular o valor médio dos pixels, $p_{\mathrm{médio,sim}}$, da imagem que seria obtida (imagem simulada) se a dose de radiação utilizada fosse $D_{\mathrm{sim}}$ utilizando a eq. (1):

$$p_{médio,sim} = p_{mean,orig}\left(\frac{D_{sim}}{D_{orig}}\right). \quad (1)$$

Considerando-se a linearidade do sistema de aquisição de imagens, pode-se definir uma imagem normalizada $Im(x,y)_{orig,norm}$ como sendo:

$$Im(x,y)_{orig,norm} = Im(x,y)_{orig}\left(\frac{D_{sim}}{D_{orig}}\right). \quad (2)$$

O NPS varia com o quadrado do valor do sinal. Assim, pode-se encontrar o valor do NPS da imagem normalizada ($NPS(u,v)_{Im_{orig,norm}}$) pela equação:

$$NPS(u,v)_{Im_{orig,norm}} = NPS(u,v)_{Im_{orig}} \cdot \left(\frac{D_{sim}}{D_{orig}}\right)^2, \quad (3)$$

sendo, $NPS(u,v)_{Im_{orig}}$, o NPS da imagem original.

A imagem simulada, $Im_{sim}$, é dada por: $Im_{sim} = Im(x,y)_{orig,norm} + Im(x,y)_{ruído}$ onde $Im_{ruído}$ é a máscara de ruído que deverá ser adicionada à imagem original, resultando na imagem desejada. Como estamos considerando que a imagem é homogênea, a imagem original normalizada ($Im_{orig,norm}$) não está correlacionada com a máscara de ruído $Im_{ruído}$. Logo, o NPS do ruído ($NPS(u,v)_{Im_{ruído}}$) pode ser facilmente encontrado através da seguinte equação:

$$NPS(u,v)_{Im_{ruído}} = NPS(u,v)_{D_{sim}} - NPS(u,v)_{D_{orig}}\left(\frac{D_{sim}}{D_{orig}}\right)^2 \quad (4)$$

sendo, $NPS(u,v)_{D_{sim}}$, o NPS do sistema na dose que se deseja simular e, $NPS(u,v)_{D_{orig}}$, o NPS na dose original.

Agora, vamos considerar que o NPS do sistema, na dose exata da imagem original e da simulada, não sejam conhecidos; mas que se tenha o NPS em outros valores de dose para o mesmo equipamento. Nesse caso, o procedimento para a estimativa do NPS nas doses desejadas (imagem original e simulada) é o seguinte:

Baseando-se na expressão do DQE (*Detective Quantum Efficiency*) de um sistema de imagem e considerando que este seja linear, é possível calcular o NPS em uma dose $D_b$ a partir da medida do NPS em uma dose $D_a$. A expressão a seguir apresenta esta relação.

$$NPS(u,v)_{D_a} = NPS(u,v)_{D_b}\left(\frac{D_a}{D_b}\right)\left(\frac{DQE(u,v)_{D_b}}{DQE(u,v)_{D_a}}\right) \quad (5)$$

Combinando as equações (4) e (5), temos:

$$NPS(u,v)_{Im_{ruído}} = \left(\frac{D_{sim}}{D_a}\right)\left(\frac{DQE(u,v)_{D_a}}{DQE(u,v)_{D_{sim}}}\right)NPS(u,v)_{D_a} - \left(\frac{D_{sim}^2}{D_{orig}D_b}\right)\left(\frac{DQE(u.v)_{D_b}}{DQE(u,v)_{D_{orig}}}\right)NPS(u,v)_{D_b} \quad (6)$$

Considerando-se que o DQE de imagens adquiridas pelo mesmo equipamento em diferentes doses é aproximadamente constante. Considere $D_1$ uma dose próxima de $D_{sim}$ cujo NPS é conhecido e $D_2$ uma dose próxima de $D_{orig}$ cujo NPS também é conhecido. Com as novas suposições, a equação (6) pode ser reescrita da seguinte forma:

$$NPS(u,v)_{Im_{ruído}} = \left(\frac{D_{sim}}{D_1}\right)NPS(u,v)_{D_1} - \left(\frac{D_{sim}^2}{D_{orig}D_2}\right)NPS(u.v)_{D_2} \quad (7)$$

Finalmente, para que seja possível a utilização dessa metodologia em imagens mamográficas reais, a suposição de homogeneidade da imagem deve ser desconsiderada. Uma das características que diferencia o ruído quântico dos demais ruídos é que este é dependente do sinal, isto é, a variância do ruído depende da intensidade da imagem (taxa de contagem da distribuição de Poisson). Assim, o valor absoluto do ruído em regiões de baixa dose é inferior ao de regiões com altas doses. Para que o ruído seja correlacionado com o sinal da imagem é necessária uma correção. Considerando que o DQE é invariante às mudanças de dose, a variância do ruído em cada pixel da imagem será proporcional ao seu valor de intensidade, que por sua vez é proporcional à dose. Esta variância é proporcional ao quadrado do valor de intensidade do pixel [10]. Assim, a imagem final, com o ruído correlacionado com a imagem original, mas proporcional à taxa da redução da dose, pode ser obtida a partir da seguinte expressão:

$$Im(x,y)_{ruído,corr} = Im(x,y)_{ruído}\sqrt{\left(\frac{Im(x,y)_{orig,norm}}{p_{medio,sim}}\right)} \quad (8)$$

*B. Banco de imagens*

Para os testes com o método apresentado foi criado um banco de imagens adquiridas em um equipamento mamográfico digital na marca Hologic, modelo *Selenia Digital Mammography*. As imagens foram adquiridas em diferentes doses utilizando uma placa de acrílico de 4 cm, para a aquisição de imagens homogêneas. Para a aquisição de imagens mamográficas heterogêneas, foi utilizando um simulador antropomórfico da anatomia mamária (*phantom "Rachel"*) [12], que apresenta todas as características e detalhes que são encontrados em uma mamografia clínica real (Figura 1).

Ambas as séries de imagens, homogêneas e do *phantom "Rachel"* foram adquiridas com valor fixo de tensão no tubo de 31kVp variando-se a exposição (corrente x tempo) desde 160 mAs (considerada aqui como 100% da dose) até 20 mAs (correspondente à 12.5% da dose original). Para fins de praticidade, o método foi testado em 2 amostras menores retiradas de cada imagem mamográfica.

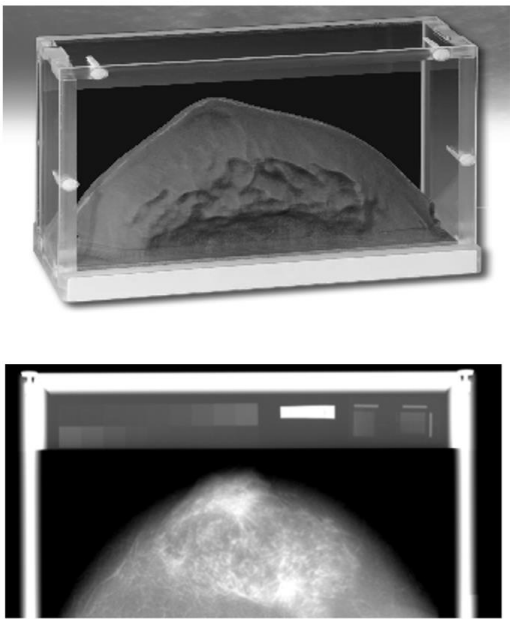


*Figura 1 – Acima: phantom "Rachel" utilizado nos testes para aquisição de imagens heterogêneas reais. Abaixo: imagem mamográfica real obtida com o uso desse phantom.*

### *C. Validação do método*

Para que o método de simulação pudesse ser validado foram utilizadas duas abordagens. Uma delas foi a comparação entre índices de qualidade e índices de similaridade calculados para a imagem simulada e a imagem real. A segunda abordagem se baseou na similaridade entre os histogramas das imagens reais e simuladas.

Os índices de qualidade e similaridade utilizados nesse trabalho foram o Pico de Relação Sinal-Ruído (*PSNR – Peak Signal-to-Noise Ratio*) [13] e o Índice de Similaridade Estrutural (*SSIM – Structural Similarity Index*) [14]. Essas métricas se baseiam na comparação entre uma imagem real e uma imagem de referência livre de ruídos (*Ground Truth*). Entretanto, neste trabalho não foi possível a obtenção da imagem de referência livre de ruídos, uma vez que as imagens de teste foram adquiridas através de um processo real em um mamógrafo. Portanto, nesse trabalho consideramos como referência a imagem adquirida com 100% da dose (que é a que possui menos ruído) suavizada pelo filtro da média 3 x 3.

Assim, os resultados para a validação do método proposto foi feito efetuando-se as medidas de PSNR e SSIM das imagens reais adquiridas em diferentes doses (menores que 100%), criando-se um conjunto de índices. Após a aplicação do método para a aquisição de imagens simuladas nas mesmas doses das imagens reais, os mesmos índices foram calculados, para verificar se estão próximos das imagens reais adquiridas em doses reduzidas. Assim, a similaridade entre as imagens simuladas e reais será maior quando o desvio entre os índices de qualidade medidos forem menores. A Figura 2 exemplifica o método de validação utilizado nesse trabalho.

A segunda abordagem para validação do método leva em conta a similaridade entre os histogramas da imagem simulada e da imagem real com dose reduzida. Para que fosse possível quantificar a similaridade entre os histogramas, utilizou-se a distância correlacional entre os vetores contendo os valores de ocorrência de cada nível de cinza. A distância correlacional assume valores próximos de 1 para vetores independentes e se aproxima de zero para vetores correlacionados.

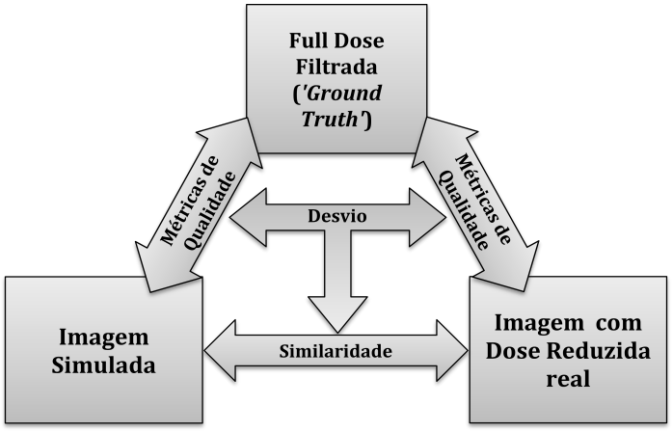



*Figura 2 - Método de comparação entre as imagens reais e simuladas em diferentes doses menores que 100%.*

## III. RESULTADOS

Na Figura 3 é apresentado um exemplo de resultado obtido com o método proposto: A imagem acima é um recorte extraído da imagem original (obtida com 100% da dose - 160 mAs); abaixo à esquerda: imagem real adquirida utilizando 12,5% da dose original (20 mAs) e; abaixo à direita: imagem simulada obtida com o método proposto considerando uma redução de 160 mAs para 20 mAs (12,5%).

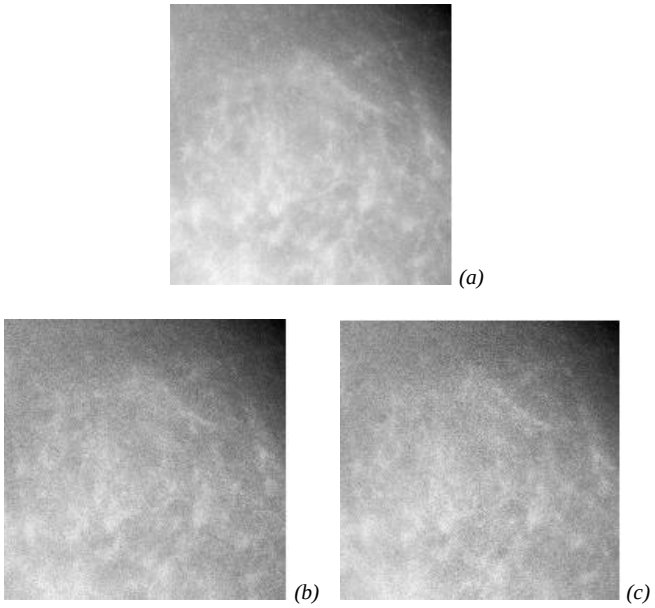



*Figura 3 - Amostras extraídas das imagens do phantom "Rachel". (a) imagem original adquirida com 100% da dose. (b) imagem real adquirida com 12,5% da dose. (c)imagem simulada a partir de 100% da dose para 12,5% da dose.*

Estas amostras foram retiradas de uma região da mamografia onde é fácil perceber a intensidade do ruído quântico. É evidente que a característica do ruído na imagem simulada se aproxima da imagem adquirida com aquela dose.

A Tabela 1 apresenta os valores de PSNR e SSIM calculados na comparação entre a imagem de referência e a imagem adquirida com 100% da dose, a partir da qual irá se originar todas as imagens simuladas. É interessante perceber que o valor de PSNR é relativamente alto (34 dB) e que a similaridade estrutural (SSIM) possui um valor bem próximo de 1 (0,944). Este comportamento era esperado, uma vez que a imagem de referência foi obtida após a aplicação do filtro da média na imagem original. Os valores desta primeira tabela serão importantes para fins de comparação do índice.

Tabela 1: Métricas de qualidade para a imagem original (100% da dose).

| **Antes da Simulação** | | |
|---|---|---|
| Referência / 100% da dose | | |
| Amostra | PSNR (dB) | SSIM |
| 1 | 33,86 | 0,937 |
| 2 | 34,90 | 0,952 |
| Média | 34,38 | 0,944 |

Os dados obtidos após a geração das imagens simuladas para 75% da dose são apresentados na Tabela 2. Na média, o valor de PSNR obtido com a imagem real obtida com 75% da dose foi de 21,90 dB. Com a simulação, o valor médio do PSNR calculado foi 21,19 dB, o que representa um desvio de 7,8% do esperado. Vale destacar que o valor de PSNR da imagem antes da simulação era de 34,38 dB, o que significa que a qualidade da imagem original foi reduzida em aproximadamente 14,19 dB quando a dose é reduzida para 75% do valor original. Já o valor médio do SSIM calculado para a imagem real e para a imagem simulada ficaram iguais.

Tabela 2: Métricas para 75% da dose.

| **Após Simulação** | | | **Objetivo** | | |
|---|---|---|---|---|---|
| Referência/Simulada | | | Referência/Real | | |
| Amostra | PSNR (dB) | SSIM | Amostra | PSNR (dB) | SSIM |
| 1 | 19,55 | 0,920 | 1 | 21,28 | 0,915 |
| 2 | 20,83 | 0,930 | 2 | 22,52 | 0,935 |
| Média | 20,19 | 0,925 | Média | 21,90 | 0,925 |

Os resultados para a redução de 50% da dose é apresentada na Tabela 3. O primeiro comportamento que se pode notar é a queda nos valores de qualidade. Este comportamento era esperado, uma vez que a redução da dose causa o aumento do ruído, o que degrada a imagem. Os valores esperados e obtidos para o PSNR são muito próximos, apresentando apenas 2,9% de desvio. Já o desvio para o MSSIM é de 1,7%. Nesta situação a simulação fez com que o PSNR da imagem original fosse reduzido em 58% do seu valor inicial, partindo de 34,38 dB na imagem original e chegando a 14,18 dB na imagem simulada.

Tabela 3: Métricas para 50% da dose.

| **Após Simulação** | | | **Objetivo** | | |
|---|---|---|---|---|---|
| Referência/Simulada | | | Referência/Real | | |
| Amostra | PSNR (dB) | SSIM | Amostra | PSNR (dB) | SSIM |
| 1 | 13,54 | 0,782 | 1 | 13,96 | 0,795 |
| 2 | 14,83 | 0,786 | 2 | 15,24 | 0,802 |
| Média | 14,18 | 0,784 | Média | 14,60 | 0,798 |

A próxima simulação, com redução para 25% da dose, apresentou resultados semelhantes aos anteriores, com o desvio do PSNR de 2,9% do esperado. Já a diferença entre o MSSIM esperado e obtido foi de apenas 1,7%. Os dados são apresentados na Tabela 4.

Tabela 4: Métricas para 25% da dose.

| **Após Simulação** | | | **Objetivo** | | |
|---|---|---|---|---|---|
| Referência/Simulada | | | Referência/Real | | |
| Amostra | PSNR (dB) | SSIM | Amostra | PSNR (dB) | SSIM |
| 1 | 10,00 | 0,464 | 1 | 10,07 | 0,470 |
| 2 | 11,29 | 0,465 | 2 | 11,35 | 0,473 |
| Média | 10,65 | 0,465 | Média | 10,71 | 0,471 |

A última simulação, apresentada na Tabela 5, foi criada apenas para fins ilustrativos, pois uma redução para 12,5% da dose convencional não possui nenhuma aplicação imediata na área médica. Mesmo com esta redução o algoritmo ainda se comporta de maneira satisfatória e apresenta uma imagem com índices que são iguais aos esperados.

Tabela 5: Métricas para 12,5% da dose.

| **Após Simulação** | | | **Objetivo** | | |
|---|---|---|---|---|---|
| Referência/Simulada | | | Referência/Real | | |
| Amostra | PSNR (dB) | SSIM | Amostra | PSNR (dB) | SSIM |
| 1 | 8,66 | 0,243 | 1 | 8,66 | 0,242 |
| 2 | 9,94 | 0,244 | 2 | 9,94 | 0,243 |
| Média | 9,30 | 0,243 | Média | 9,30 | 0,243 |

Também foi realizada uma comparação entre os histogramas das simuladas e as imagens reais adquiridas na mesma dose. Um exemplo de resultado obtido pode ser visto na Figura 4. Nota-se que os histogramas são muito próximos.

Para uma análise quantitativa entre os histogramas, foi calculada a distância entre os histogramas utilizando uma distância correlacional entre os dois vetores, considerando a ocorrência dos níveis de cinza. Tais valores são normalizados e variam entre 0 e 1, com resultados próximos de 1 indicando

baixa similaridade entre os histogramas e valores próximos de 0 indicando alta similaridade entre eles.

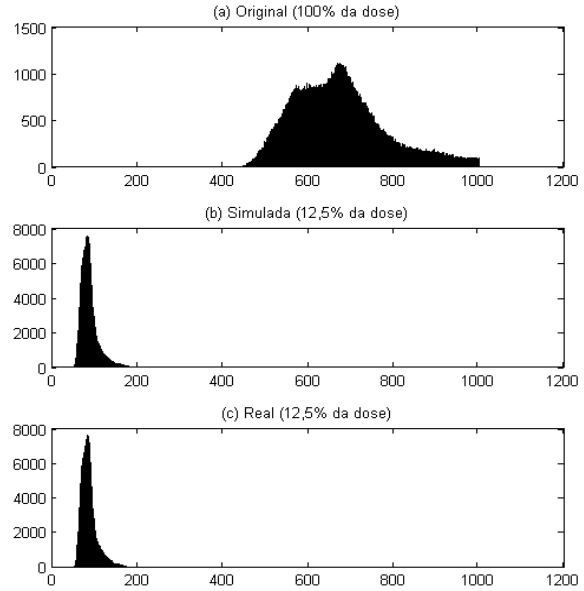



*Figura 4 - Comparação de histogramas. (a) Imagem original com 100% da dose. (b) Imagem Simulada com 12,5% da dose. (c) Imagem Real adquirida com 12,5% da dose.*

A Tabela 6 comprova que as imagens simuladas apresentaram os histogramas muito similares aos das imagens reais, adquiridas com as mesmas doses de radiação. Apenas para fins de comparação, a Tabela 6 também apresenta a similaridade entre os histogramas da imagem real original (100% da dose) com as imagens reais adquiridas com dose reduzida.

Tabela 6: Similaridade dos histogramas antes e após a simulação.

| **Antes da Simulação** | | **Após a Simulação** | |
|---|---|---|---|
| Real/Original | | Real/Simulada | |
| Dose | Similaridade | Dose | Similaridade |
| 75% | 0,79 | 75% | 0,09 |
| 50% | 0,96 | 50% | 0,07 |
| 25% | 1,00 | 25% | 0,05 |
| 12,5% | 1,00 | 12,5% | 0,06 |

## IV. Conclusões

Neste trabalho foi investigado a aplicação de um método de simulação de redução de dose de radiação para imagens de mamografia digital.

As avaliações sobre a similaridade entre as imagens antes da simulação, após a simulação e quando adquiridas nas doses simuladas mostraram, através de medidas quantitativas objetivas – PSNR, MSSIM e comparação de histogramas – que a simulação resulta em imagens muito similares às esperadas. O método foi testado para reduções de até 12,5% da dose inicial, mantendo um desempenho praticamente constante para todas as reduções.

Algumas das limitações deste trabalho foram a suposição de que o DQE se mantém constante para qualquer variação da dose, o que não é verdade para a maioria dos equipamentos mamográficos e a utilização de imagens com doses muito discrepantes, como a redução de 100% para 12,5% da dose. Em trabalhos futuros poderemos estudar o comportamento do DQE para variações de dose, generalizando o método para qualquer equipamento e adquirir imagens com intervalo de amostragem menor, com reduções menores.

## Agradecimentos

Os autores gostariam de agradecer à FAPESP (Proc. 2013/18915-5) e à CAPES (Proc. 88881.030443/2013-01) pelo apoio financeiro.

Agradecemos também à *University of Pennsylvania* por disponibilizar o equipamento Hologic Selenia Dimensions e o *phantom* antropomórfico para geração das imagens mamográficas utilizadas neste trabalho.

## Referências

[1] WHO - World Health Organization. Disponível em: http://www.who.int/en.

[2] Glynn CG, Farria DM, Monsees BS, Salcman JT, Wiele KN Hildebolt CF: Effect of transition to digital mammography on clinical outcomes, Radiology 260(3):664-670, 2011.

[3] Hendrick RE: Radiation doses and cancer risks from breast imaging studies, Radiology 257(1):246-253, 2010.

[4] Yaffe MJ, Mainprize, JG: Risk of radiation-induced breast cancer from mammographic screening, Radiology 258(1):98-105, 2011.

[5] Ruschini M, *et al*.: Dose dependence of mass and microcalcification detection in digital mammography: Free response human observer studies, Med Phys 34(2):400-407, 2007.

[6] Samei E, Saunders RS, Baker JA, Delong DM: Digital mammography: effects of reduced radiation dose on diagnostic performance, Radiology 243(2):396-404, 2007.

[7] Yaffe MJ: "Digital Mammography". In: Beutel J, Kundel HL and Van Metter RL (Eds). Handbook of Medical Imaging Vol 1. Physics and Psychophysics, Bellingham, WA: SPIE Press, pp 329-372, 2000.

[8] Obenauer S, Hermann KP, Grabbe E. Dose Reduction in Full-Field Digital Mammography: An Anthropomorphic Breast Phantom Study. Br J Radiol 76(907): 478-482, 2003.

[9] Saunders RS, Baker JA, Delong DM, Johnson JP, Samei E: Does image quality matter? Impact of resolution and noise on mammographic task performance. Med Phys 34(10):3971-3981, 2007.

[10] Båth M, Håkansson M, Tingberg A, Månsson LG: Method of simulating dose reduction for digital radiographic systems. Radiat. Prot. Dosim 114:253-259, 2005.

[11] Wu G, Mainprize J G, Yaffe M J: Spectral analysis of mammographic images using a multitaper method, Med Phys 39(2): 801-810, 2012.

[12] Caldwell CB, Yaffe MJ: Development of an anthropomorphic breast phantom, Med Phys 17(2): 273-280, 1990.

[13] Wang Z, Bovik AC: Mean squared error: Love it or leave it? A new look at signal fidelity measures. IEEE Signal Proc Mag 26(1):98-117, 2009.

[14] Wang Z, Bovik AC, Sheikh HR, Simoncelli, EP: Image quality assessment: from error visibility to structural similarity. IEEE T Image Process 13(4):600-612, 2004.

# Face Recognition Systems Based on Wavelet Transform and SVM Ensembles

Daniel M. M. da Costa, Felipe G. S. Teodoro, Sarajane M. Peres, Clodoaldo A. M. Lima
Programa de Pós-Graduação em Sistemas de Informação, Escola de Artes, Ciências e Humanidades, Universidade de São Paulo
Av. Arlindo Bettio, 1000, 03828-000 São Paulo, SP, Brasil
Email:{daniel.moura.costa, fteodoro, sarajane, c.lima}@usp.br

***Abstract*—The improvement of information protection techniques is a demand in the modern society. Biometric Systems have been embedded in applications ranging from buildings security to locking systems on smartphones. Therefore, scientists and practitioners have worked to make more accurate such systems, especially the face recognition systems. Biometric Systems are decomposed on sensors, features extraction and classification modules. In this paper, the two last modules are discussed. We propose to use Wavelet Transform as feature extraction technique, and two approaches as options for implementing classification modules. The first approach is based on Support Vector Machine (SVM) and the second is based on Ensemble built with SVMs. From experimental results, we detected good conditions to build face recognition systems. Both approaches achieved the best results using coefficients of the DB2 Wavelet Function; for the latter approach, such results were obtained through Ensembles composed of five SVMs. The Ensembles overcame the individual SVMs.**

Keyword: Wavelet Transform, Support Vector Machines, Ensembles, Face Recognition, Biometric Systems.

## I. Introdução

Devido à necessidade de garantir a segurança da informação em diversos contextos, como por exemplo, na execução de uma transação monetária através da internet ou na realização de restrição de acesso a um determinado local, geralmente, são utilizadas senhas, tokens, números PIN ou documentos pessoais para realizar a identificação do indivíduo e/ou sua permissão de acesso. Esse método apesar de muito prático e suficiente em muitos casos é suscetível à fraude por se basear em uma chave que pode ser roubada, descoberta ou falsificada por outro indivíduo [1]. Dessa forma torna-se necessário o desenvolvimento de tecnologias que possam oferecer uma camada maior de segurança caso seja necessário. Uma dessas tecnologias é a biometria, o campo de estudo que busca realizar a identificação baseada em alguma característica física ou comportamental do indivíduo como a impressão digital [2], [3], a forma da íris dos olhos [4], os aspectos faciais [5], [6], padrões dos sinais produzidas pelo cérebro [7], [8] ou pelos batimentos cardíacos [9], [10].

Para que uma característica fisiológica ou comportamental possa ser utilizada como biometria é necessário que as seguintes propriedades sejam satisfeitas [11]: i)*Universalidade*, todas as pessoas devem possuir essa característica; ii)*Distinção*, duas pessoas diferentes devem ser diferenciadas em relação a característica em questão; *Permanência*, a característica não deve sofrer alterações significativas ao longo dos anos; *Mensurável*, é necessário que a característica possa ser medida quantitativamente. Além disso essa característica deve permitir a construção um sistema de reconhecimento rápido e com baixa taxa de erro, não deve ser intrusiva de modo que o usuário não se sinta desconfortável ao fornecê-la e não deve permitir fraudes [12].

Dependendo do contexto, um sistema biométrico pode atuar como um sistema de verificação (autenticação) ou identificação (reconhecimento). Na verificação é realizada uma comparação um para um, cujo objetivo é confirmar ou negar a identidade reivindicada pelo indivíduo. Neste caso o sistema visa responder a seguinte questão: "Esta pessoa é quem ela diz ser?". Já na identificação é realizada uma comparação de um para muitos, o qual visa encontrar o cadastro pertencente ao indivíduo em um banco de dados, aqui o sistema é projetado para responder a questão: "Quem é esta pessoa?". Para tornar possível o uso do sistema é necessário que o indivíduo primeiramente realize o cadastramento para que em um segundo momento possa ser realizado a requisição de autenticação ou identificação. Este artigo irá focar na identificação de usuários.

O sistema biométrico pode ser dividido em quatro módulos principais ([11]): o *módulo sensorial*, o qual é responsável pela captação dos dados; *módulo de extração de característica* que extraí do dado biométrico capturado alguma informação útil; *módulo de comparação* realiza a comparação da informação extraída com a armazenada no banco de dados visando realizar a autenticação ou identificação do indivíduo; e por fim um *módulo de decisão*, de acordo com o valor de similaridade, o sistema aceita ou rejeita o usuário.

Os sistemas biométricos baseados em face utilizam uma câmera para realizar a captura da imagem do usuário, em seguida é aplicado algoritmos de detecção de face. Com a imagem contendo apenas a face do usuário, podemos usar os pixels da imagem como a representação do usuário na base de dados. Entretanto, esta abordagem irá gerar vetores de caraterísticas de alta dimensionalidade degradando o desempenho dos métodos de aprendizado empregados para classificação. Uma abordagem bastante utilizada para atacar esse problema consiste em empregar métodos de análise de subespaço, como análise de componentes principais (PCA) [13], análise de discriminante linear (LDA) [14] e análise de componentes independentes (ICA)[15], para reduzir a dimensionalidade dos dados e fornecer um conjunto de características com grande variação inter-classe e pequena variação intra-classe. Entretanto, esses métodos podem não ser adequados para representar variações de expressão, iluminação e pose.

Outra forma de atacar a alta dimensionalidade em problemas de reconhecimento de face é empregar técnicas que realizam algum tipo de transformação do domínio. Em [16], os autores argumentam que, quando a PCA é aplicada a imagens sem nenhum processamento, a correlação das características faciais não é refletida adequadamente no espaço de autovetores. Assim, eles sugerem que inicialmente filtros de Gabor sejam utilizados para a extração de características faciais e em seguida seja aplicada a PCA para extrair as características e classificá-las otimamente. Atualmente, a análise multiresolução é frequentemente realizada como um passo de pré-processamento para redução da dimensionalidade. Alguns trabalhos considerando informação de multiresolução tem sido propostos para reconhecimento facial [15], [17]. A técnica de análise de multiresolução mais popular é a transformada de wavelet. Na análise baseada em wavelet uma imagem é normalmente decomposta em diferentes escalas ou resoluções usando uma função mãe. Em [15] foi proposto um método de reconhecimento que utiliza transformada wavelet para extração de características. Para redução da dimensionalidade das características foi utilizado PCA e ICA.

Para a construção do módulo de comparação podemos destacar o uso dos classificadores, tais como: Redes Neurais Artificiais [18], [19], Máquinas de Vetores Suporte (SVM) [13], [20] e K-Vizinhos Próximos [21], [22]. Nesse trabalho será utilizado SVM para compor o módulo de comparação, as SVMs foram propostas em [23] como sendo um método de reconhecimento de padrões para problemas de classificação linear e não-linear. No entanto, existem aspectos estruturais e paramétricos de projeto que podem conduzir a uma degradação de desempenho das SVMs. Isto implica que as garantias teóricas de maximização da capacidade de generalização podem ser drasticamente violadas, pela simples atribuição equivocada de algum parâmetro (ou um subconjunto de parâmetros) de projeto. Dentre os parâmetros de projeto, destaca-se: (i) o tipo de função kernel; (ii) parâmetros da função kernel adotada; (iii) parâmetros do problema de otimização quadrática com restrições.

Análise de multiresolução podem extrair várias características das imagens usando diferentes escalas e diferentes funções mãe. Se usarmos um método de fusão de característica simples e efetivo para combiná-las, pode-se melhorar a eficiência e o desempenho do sistema de reconhecimento facial. Por outro lado a fusão no nível de decisão é uma prática popular para combinar múltiplos classificadores (também conhecido como comitê de maquinas) em biometria. Estudos teóricos e empíricos vêm evidenciando ganhos por parte de comitê de máquinas frente a técnicas tradicionais de aprendizado de máquina em diferentes contextos, inclusive no reconhecimento facial. Neste artigo, analisamos o desempenho do sistema de reconhecimento facial obtido através da fusão de característica e da fusão de decisão. A fusão no nível de decisão será realizada via voto majoritário.

O artigo é organizado nas seguintes seções: revisão sobre a transformada wavelet é apresentada na Seção II. Uma descrição suscinta sobre Máquinas de Vetores Suporte e Comitê de Máquinas é apresentado nas Seções III e IV. A metodologia e a base de dados utilizada são descritas na Seção V, os resultados obtidos são apresentados e discutidos na Seção VI. Finalmente, a conclusão sobre os resultados e trabalhos futuros são apresentados na Seção VII.

## II. Transformada Wavelet

De forma geral, um sinal pode ser considerado como a superposição de diferentes estruturas ocorrendo sobre diferentes escalas de tempo em diferentes tempos (ou escalas espaciais em diferentes localizações). A ideia básica da análise de wavelet consiste em expressar um sinal como uma combinação linear de um conjunto de funções, que são obtidas pela translação e dilatação de uma função especifica chamada wavelet mãe. A decomposição de um sinal conduz para um conjunto de coeficientes chamados de coeficientes de wavelet. Estes são definidos como produto interno do sinal pelo conjunto de funções base. Sendo esse conjunto construído com base nas funções Wavelet mãe, que tem suporte finito e decai rapidamente para fora de um determinado intervalo [24].

Para o cálculo dos coeficientes da transformada wavelet de um sinal, não é necessário utilizar uma função wavelet conhecida. Os coeficientes podem calculados a partir da utilização de 2 filtros de resposta impulsiva finita (filtros FIR), um filtro passa-baixa $g$ e um filtro passa-alta $h$. A esse processo de decomposição wavelet é dado o nome de Algoritmo de Mallat [25]. A saída desse algoritmo é composta por 2 novos sinais: coeficientes de aproximação, representando as baixas frequências do sinal obtidas pelo filtro $g$ , e coeficientes de detalhes, representando as altas frequências do sinal, obtidas pelo filtro $h$. Esse processo pode ser visto na figura 1.

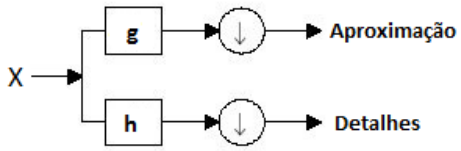



Fig. 1. Decomposição Wavelet 1D.

Uma imagem decomposta pela transformada wavelet é descrita por um conjunto de sub-imagens em diferentes resoluções e bandas de frequências. O método mais comum para realizar a decomposição Wavelet 2D é através do esquema de pirâmide [26]. Nesse esquema a imagem de entrada é decomposta em $k$ níveis, gerando um conjunto de sub-imagens $LL_k, LH_k, HL_k, HH_k, LL_{k-1}, LH_{k-1}, HL_{k-1}, HH_{k-1}, \cdots, HH_1$. A sub-imagem $LL$ representa a aproximação da imagem, já as sub-imagens $LH, HL, HH$ representam os detalhes da imagem na direção horizontal, vertical e diagonal, respectivamente. A decomposição Wavelet 2D pode ser vista na figura 2 e uma imagem decomposta pode ser vista na figura 3 .

## III. Support Vector Machine

As Máquinas de Vetores Suporte (SVM) foram propostas por Vapnik [23], como um classificador para problemas lineares e não-lineares e vêm sendo utilizada para a construção de sistemas biométricos baseados em face [18], [27], [20]. De acordo com [28] os resultados que são obtidos pelo uso das SVMs geralmente é superior, ou em alguns casos equivalentes, a outras técnicas de reconhecimento de padrões.

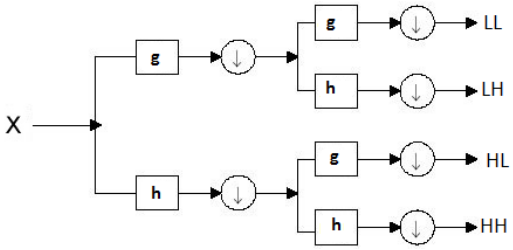



Fig. 2. Decomposição Wavelet 2D.

Fig. 3. Exemplo de imagem decomposta pela WT usando nível 1.

Considere um conjunto de dados de treinamento $\{x_i, y_i\}_{i=1}^{N}$ com $x_i \in \mathcal{R}^m$ e $y_i \in \pm 1$ representando o dado de entrada e a classe desejada para a instância $i$ respectivamente. A SVM busca construir uma fronteira de decisão, $sign[f(x;w)]$, onde $f(x;w) = w^T\phi(x) + b$. $f(x;w)$ é uma aproximação para o mapeamento $y$, com $w \in R^m$ e $\phi(\cdot) : R^n \rightarrow R^m$ é uma função que mapeia o espaço de entrada para um espaço de maior dimensionalidade, conhecido como espaço de característica. Os parâmetros $b$ e $w$ podem ser obtidos através do seguinte problema de otimização restrito [23]:

$$\min_{w,\beta,\xi} J(w,\beta,\xi) = \frac{1}{2}w^T w + C\sum_{i=1}^{N}\xi_i, \qquad (1)$$

onde $y_i[w^T\phi(x_i)+b] \geq 1-\xi_i, \xi_i \geq 0, i = 1,\cdots,N$, e $C$ é um parâmetro que estabelece o compromisso entre a complexidade do modelo e o erro de treinamento.

A Eq. 1 pode ser escrita através de sua representação dual [23]

$$\max_{\alpha} J(\alpha) = \sum_{i=1}^{N}\alpha_i - \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j \phi(x_i)^T\phi(x_j), \quad (2)$$

sujeito a $\sum_{i=1}^{N}\alpha_i y_i = 0$ e $0 \leq \alpha_i \leq C$, para $i = 1,\cdots,N$. Para obtermos $\phi(x_i)^T\phi(x_j)$, nós não necessitamos calcular $\phi(x_i)$ ou $\phi(x_j)$ explicitamente. Ao invés disso, para algum $\phi$, nós podemos projetar uma matriz de kernel $K$ tal que $K(x_i,x_j) = \phi(x_i)^T\phi(x_j)$. As funções kernel satisfazem o Teorema de Mercer [28] e fornecem uma maneira rápida e implícita para o cálculo de $\phi(x_i)^T\phi(x_j)$. Dentre as possíveis funções de kernels, escolhemos o Kernel RBF: $K(x_i,x_j) = \exp(-\frac{(x_i-x_j)^T(x_i-x_j)}{2\sigma^2})$, onde $\sigma^2$ denota a variância da função kernel, que necessita ser adequadamente ajustada. Com base nisso, a superfície de decisão pode ser reescrita como $f(x;w) = \sum_{i=1}^{N}\alpha_i y_i K(x_i,x_j) + b$.

Após o treinamento da SVM, os dados que satisfazem a restrição $\alpha_i \geq 0$ são chamados de *vetores suporte*, que são os pontos localizados sobre a fronteira de separação entre as duas classes, sendo que o número de vetores suporte é geralmente muito menor que a quantidade de dados. Um dado vetor de teste $x \in \mathcal{R}^m$ é classificado da seguinte forma: $sign(\sum_{i=1}^{N}\alpha_i y_i K(x,x_i) + b)$

## IV. Comitê de Máquinas

Como visto na Seção III o desempenho da SVM está diretamente relacionado à definição do valor do parâmetro do Kernel ($\sigma$) e do parâmetro $C$. Caso os parâmetros não sejam definidos adequadamente, o modelo construído pode apresentar uma degradação da capacidade de generalização. Na ausência de uma metodologia sistemática e de baixo custo para proposição de modelos computacionais otimamente especificados em SVM, os comitês de máquinas, particularmente os comitês de máquinas de vetores suporte, se apresentam como alternativas promissoras [29]. Existem versões estáticas de comitês, na forma de ensembles de componentes, e versões dinâmicas, na forma de misturas de especialistas.

O termo ensemble é aquele comumente empregado na combinação de um conjunto de componentes sintetizados para executar a mesma tarefa. Neste caso, cada componente representa isoladamente uma solução candidata para o problema de classificação, podendo cada solução ser obtida por meios distintos e independentes entre si. Já na mistura de especialista, a tarefa é decomposta em várias sub-tarefas, e a solução da tarefa completa requer a contribuição de todos ou um subconjunto dos especialistas. Logo, um especialista isolado não representa uma solução candidata para o problema de classificação como um todo, sendo fundamental a composição dos especialistas.

Neste artigo optamos pela implementação de Ensembles de SVMs (veja figura 4), onde as SVMs são treinadas usando diferentes coeficientes de wavelet obtidos a partir de diferentes funções Wavelet e com diferentes valores de $\sigma$. Quanto à combinação das SVMs, a abordagem adotada é o voto baseado na pluralidade ou voto majoritário, neste caso, um exemplo é classificado de acordo com a classe majoritária atribuída pelas SVMs individuais.

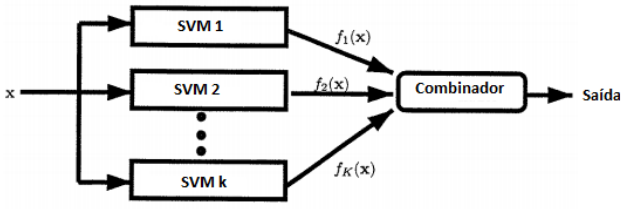



Fig. 4. Exemplo da estrutura de um Ensemble.

Para a construção do Ensemble foram utilizadas diferentes abordagens: i) combinação das melhores SVMs obtidas para cada função Wavelet, ii) combinação das SVMs obtidas para cada função Wavelet, considerando o mesmo parâmetro $\sigma$, iii) combinação das SVMs obtidas para cada função Wavelet considerando que os parâmetros da função kernel estivesse definido dentro de um intervalo de $\sigma$ e iv) combinação das

SVMs obtidas usando uma estratégia construtiva. A partir de um reservatório de SVMs candidatas a compor o ensemble, inicialmente a estratégia construtiva cria um ordenada das SVMs segundo o erro quadrático médio crescente. Em seguida, seleciona a melhor SVM para compor o ensemble e a cada passo adiciona uma nova SVM. Caso este novo comitê apresente um desempenho superior quando comparado com o comitê anterior, esta SVM é mantida, caso contrário, é retirada e a próxima SVM da lista ordenada é inserida. Este processo continua até que todas as SVMs da lista ordenada tenham sido testadas. Todas estas estratégias podem ser consideradas como uma forma de fusão do modulo de decisão. Outra forma de fusão realizada foi através da concatenação de todos os coeficientes extraídos pelas funções wavelets em um único vetor de características, o qual foi utilizado como entrada para uma única SVM. Neste caso, teríamos a fusão no nível de característica.

## V. METODOLOGIA

Nessa seção será descrita a base de dados utilizada, processo de extração de características, a configuração das SVMs e os modelos de Comitê.

### A. Base de Dados e Extração de Características

Para realizarmos a aplicação da metodologia aqui proposta, utilizamos a base de dados Yale [30], umas das bases mais utilizadas para sistemas de reconhecimento facial. Esta base contém 165 imagens, composta de 15 usuários e 11 imagens por usuário, cada imagem apresenta variações em relação expressões e iluminação. Na primeira etapa aplicamos o algoritmo de Viola e Jones [31] para extrair as faces e então redimensionamos todas as imagens para a resolução de 128×128 pixels. Na figura 5 apresentamos algumas das imagens que compõe a base de dados Yale.

Fig. 5. Exemplo de imagens da base de dados Yale.

Com as imagens das faces extraídas, aplicamos a transformada Wavelet, usando 3 níveis de decomposição, para extrair os coeficientes que serão utilizados como vetor de características. Nesse artigo, consideramos somente os coeficientes das imagens $LL, HL, LHeHH$ do último nível de decomposição.

Na Tabela I é apresentado o tamanho de cada um dos vetores de características utilizados, sendo que a última linha desta tabela corresponde ao tamanho do vetor de característica obtido pela concatenação de todos os coeficientes extraídos.

### B. Configuração das SVMs

A SVM é um classificador binário, ou seja, não pode ser utilizado diretamente em problemas com múltiplas classes. De forma a utilizar a SVM para o nosso problema, utilizamos a estratégia One-Vs-One. Conforme já mencionado, para que a SVM ofereça um bom desempenho é necessário que os seus parâmetros sejam escolhidos cuidadosamente. Como as SVMs são mais sensíveis ao parâmetro $\sigma$ da função Kernel RBF, neste artigo, mantivemos o valor de C igual a 1000. Em [32] é apresentada uma estratégia para a seleção dos parâmetros de uma SVM, com base nesse estudo decidimos pelo uso de $\sigma$ como sendo $2^i$ com $i = -2, -1, \cdots, 15$. Sendo que para cada valor de $i$ foi aplicada a estratégia de treinamento de validação cruzada 10-fold.

TABELA I. QUANTIDADE DE COEFICIENTES PARA CADA ABORDAGEM UTILIZADA.

| Wavelet | Quantidade de Coeficientes |
|---|---|
| Daubechies 2 | 1296 |
| Daubechies 4 | 1936 |
| Symlet 3 | 1600 |
| Symlet 4 | 1936 |
| Symlet 5 | 2116 |
| Combinação dos Coefficientes | 8884 |

### C. Geração dos componentes do Ensemble

Neste artigo, quatro estratégias para a geração dos componentes foram realizadas.

**Melhor resultado para cada função Wavelet:**
Selecionamos a melhor SVM obtida a partir de cada uma das funções Wavelet e combinamos as SVMs usando voto majoritário;

**Combinação por parâmetro:**
Selecionamos um intervalo de valores para $\sigma$ e para cada valor selecionamos as SVMs treinadas para cada função de Wavelet e as combinamos usando voto majoritário;

**Combinação por intervalo de parâmetro:**
Selecionamos um intervalo de valores para $\sigma$ e selecionamos todas as SVMs treinadas para estes valores e as combinamos usando voto majoritário;

**Estratégia construtiva:**
Escolhemos a melhor SVM individual para iniciar o comitê, independente do parâmetro ou da função Wavelet, então a cada passo adicionamos um novo componente se este produz uma melhora no resultado, caso contrário este componente não é adicionado, em seguida testamos um próximo componente.

## VI. RESULTADOS

Na Tabela II é apresentado os resultados obtidos pelas melhores SVMs geradas partir dos coeficientes das funções Wavelets utilizadas para decompor as imagens.

TABELA II. RESULTADOS OBTIDOS COM A TRANSFORMADA WAVELET

| Wavelet | $\sigma$ | Erro Total | Desvio Padrão |
|---|---|---|---|
| Daubechies 2 | 64 | 11,91% | 12,36% |
| Daubechies 4 | 64 | 15,64% | 13,95% |
| Symlet 3 | 128 | 14,92% | 14,39% |
| Symlet 4 | 128 | 14,92% | 13,60% |
| Symlet 5 | 64 | 14,92% | 14,92% |
| Fusão de Características | 128 | 14,33% | 13,98% |

Em relação aos resultados alcançados pelas SVMs individualmente usando os coeficientes Wavelet como vetores de características, o melhor modelo foi produzido usando a função

Wavelet Daubecchies de ordem 2, com erro igual a 11,91% e desvio padrão de 12,36%. Na Figura 6 é apresentado todos os resultados obtidos para todos os valores de $\sigma$. Nota-se nesse caso, que os melhores resultados em termos de erro estão localizados no intervalo $[2^5, 2^{10}]$ para $\sigma$.

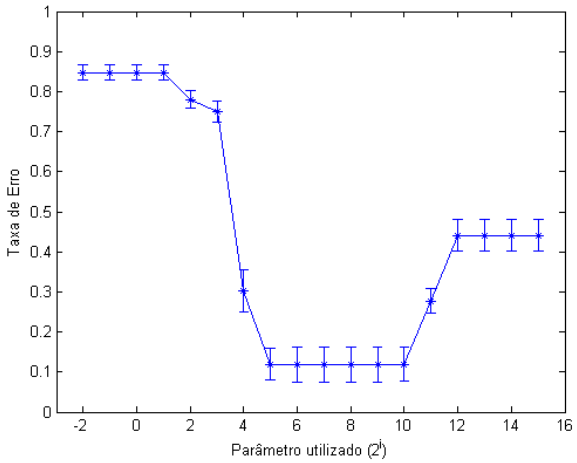



Fig. 6. Resultado obtido para DB2.

As Figuras 7(a), 7(b), 7(c) e 7(d) apresentam os resultados obtidos para as outras funções Wavelets. Analisando estas figuras, pode-se observar que os melhores resultados foram obtidos no intervalo $[2^6, 2^{10}]$ para $\sigma$. Comparando os intervalos obtidos para cada função wavelet fica claro que existe uma região, na qual o desempenho das SVMs são muito similares. Entretanto, exige uma região na qual o desempenho é muito ruim. Isto mostra a necessidade de selecionar adequadamente este parâmetro.

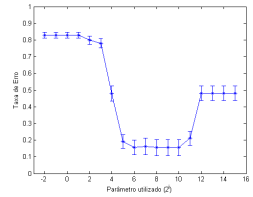
Na Tabela III apresentamos os resultados obtidos para as diferentes abordagens adotadas na construção dos Ensemble. Para a construção do Ensemble usando um intervalo de $sigma$, optou-se pela utilização das SVMs geradas com os valores de $\sigma$ entre $[2^6, 2^{10}]$. Este intervalo foi utilizado levando em conta os resultados individuais apresentado nas figuras 6, 7(a), 7(b), 7(c), 7(d) e 7(e). Para cada estratégia utilizada algumas observações são descritas abaixo:

**Melhor $\sigma$:**
Selecionamos as SVMs da Tabela II e as combinamos com voto majoritário;

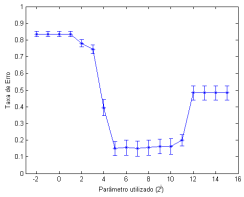
**Combinação por $\sigma$:**
Para cada valor de $\sigma$ entre $[2^6, 2^{10}]$, combinamos todas as SVMs obtidas, treinadas com o mesmo valor de $\sigma$, com voto majoritário;

**Combinação por intervalo de $\sigma$:**
Selecionamos todas as SVMs treinadas com os valores de $\sigma$ entre $[2^6, 2^{10}]$ e as combinamos com voto majoritário;

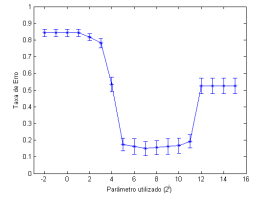
**Abordagem construtiva:**
Ordenamos todas as SVMs treinadas com $\sigma$ entre $[2^6, 2^{10}]$ de acordo com sua taxa de erro, selecionamos a melhor e a cada passo adicionamos uma SVM se o resultado melhora, caso contrário, não adicionamos.



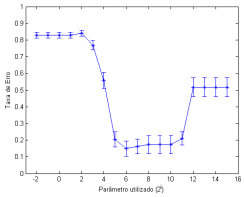
(a) Resultado obtido para DB4



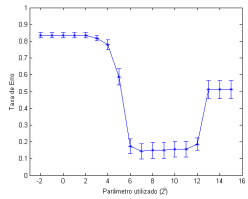
(b) Resultado obtido para Sym3



(c) Resultado obtido para Sym4



(d) Resultado obtido para Sym5



(e) Resultado obtido para a fusão de características

Fig. 7. Resultados obtido com combinação das SVMs para cada valor de $\sigma$.

TABELA III. RESULTADOS OBTIDOS USANDO AS ESTRATÉGIAS DE ENSEMBLE.

| Estratégia de Comitê | $\sigma$ | Erro Total | Desvio Padrão |
|---|---|---|---|
| Melhor $\sigma$ | Tabela II | 14,30% | 14,18% |
| Combinação por $\sigma$ | 64 e 1024 | 13,13% | 13,25% |
| Combinação por Intervalo de $\sigma$ | [64,1024] | 12,54% | 13,15% |
| Estratégia Construtiva | | 11,32% | 13,96% |

Analisando os resultados obtidos com o Ensemble podemos verificar que estes foram superiores a aqueles obtidos pelos componentes individuais. A única exceção foi obtida quando a SVM foi treinada com a função Wavelet DB2. Isto pode ser justificado pelo fato que esta função produziu uma SVM com menor taxa de erro. Logo, se não houver nenhum processo de seleção das componentes, é esperado que o resultado da combinação do melhor modelo com outros modelos não produza bons resultados.

A abordagem construtiva, composta por apenas 5 SVMs, sendo todas treinadas sobre os coeficientes da função Wavelet DB2, alcançou um resultado superior as SVMs treinadas individualmente. Isto mostra que se conseguirmos selecionar adequadamente um subconjunto de componentes candidatas, podemos superar o desempenho inclusive do melhor modelo.

Com a fusão no nível de característica, pode-se observar um desempenho muito ruim. Isto mostra que apesar das SVMs não serem sensíveis à dimensionalidade do espaço de entrada, a simples concatenação do vetor de característica não é uma abordagem apropriada. Sendo que possivelmente os coeficientes de algumas funções Wavelet podem estar dificultando a construção do hiperplano de separação das classes.

## VII. Conclusão

Nesse artigo, nós realizamos uma análise do desempenho dos sistemas de reconhecimento facial usando os coeficientes extraídos da transformada Wavelet como entradas para as SVMs. Além da análise do desempenho individual de cada SVM, foram propostas algumas abordagens para geração de componentes de Comitês de Máquinas.

Em relação aos resultados obtidos com a utilização dos sistemas individuais observamos que melhor resultado obtido foi através da utilização da função Wavelet DB2, que alcançou uma taxa de erro igual a 11,91%, mostrando que essa função consegue gerar um conjunto de características com maior capacidade discriminativa.

Já em relação ao emprego de Comitês de Máquina, podemos ver que os resultados obtidos foram superiores a quase todos os modelos obtidos individualmente, com exceção para a função Wavelet DB2. No entanto, ao utilizarmos uma estratégia construtiva conseguimos superar todos os modelos individualmente.

Como trabalhos futuros, planejamos a utilização de diferentes técnicas para extração de características e também a construção de outros modelos de Comitês de Máquina com o uso de diferentes classificadores como Least-Squares SVMs e Relevance SVMs.

## Referências


[1] A. K. Jain, P. Flynn, and A. A. Ross, *Handbook of Biometrics*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2007.

[2] K. Cao, X. Yang, X. Chen, Y. Zang, J. Liang, and J. Tian, "A novel ant colony optimization algorithm for large-distorted fingerprint matching," *Pattern Recognition*, vol. 45, no. 1, pp. 151 – 161, 2012.

[3] A. Batool and A. Tariq, "Computerized system for fingerprint identification for biometric security," in *Multitopic Conference (INMIC), 2011 IEEE 14th International*. IEEE, 2011, pp. 102–106.

[4] D. M. Rankin, B. W. Scotney, P. J. Morrow, and B. K. Pierscionek, "Iris recognition failure over time: The effects of texture," *Pattern Recogn.*, vol. 45, no. 1, pp. 145–150, Jan. 2012.

[5] H. Zhang, N. M. Nasrabadi, Y. Zhang, and T. S. Huang, "Joint dynamic sparse representation for multi-view face recognition," *Pattern Recogn.*, vol. 45, no. 4, pp. 1290–1298, Apr. 2012.

[6] M. A. Lone, S. Zakariya, and R. Ali, "Automatic face recognition system by combining four individual algorithms," in *Computational Intelligence and Communication Networks (CICN), 2011 International Conference on*. IEEE, 2011, pp. 222–226.

[7] R. Paranjape, J. Mahovsky, L. Benedicenti, and Z. Koles', "The electroencephalogram as a biometric," in *Electrical and Computer Engineering, 2001. Canadian Conference on*, vol. 2, 2001, pp. 1363–1366 vol.2.

[8] S.-K. Yeom, H.-I. Suk, and S.-W. Lee, "Person authentication from neural activity of face-specific visual self-representation," *Pattern Recognition*, vol. 46, no. 4, pp. 1159 – 1169, 2013.

[9] A. Lourenço, H. Silva, and A. Fred, "Ecg-based biometrics: A real time classification approach," in *Machine Learning for Signal Processing (MLSP), 2012 IEEE International Workshop on*. IEEE, 2012, pp. 1–6.

[10] H. Silva, A. Lourenço, F. Canento, A. Fred, and N. Raposo, "Ecg biometrics: Principles and applications," in *Proc. of the 6th BIOSIGNALS Conf*, 2013.

[11] A. Jain, A. Ross, and S. Prabhakar, "An introduction to biometric recognition," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 14, no. 1, pp. 4–20, 2004.

[12] L. Hong and A. Jain, "Integrating faces and fingerprints for personal identification," *IEEE transactions on pattern analysis and machine intelligence*, vol. 20, pp. 1295–1307, 1998.

[13] M. Mazloom, S. Kasaei, and H. A. Neissi, "Construction and application of svm model and wavelet-pca for face recognition," in *Proceedings of the 2009 Second International Conference on Computer and Electrical Engineering - Volume 01*, ser. ICCEE '09. Washington, DC, USA: IEEE Computer Society, 2009, pp. 391–398.

[14] A. Bansal, K. Mehta, and S. Arora, "Face recognition using pca and lda algorithm," in *Advanced Computing Communication Technologies (ACCT), 2012 Second International Conference on*, Jan 2012, pp. 251–254.

[15] H. K. Ekenel and B. Sankur, "Multiresolution face recognition," *Image and Vision Computing*, vol. 23, no. 5, pp. 469 – 477, 2005.

[16] K.-c. Chung, S. C. Kee, and S. R. Kim, "Face recognition using principal component analysis of gabor filter responses," in *Proceedings of the International Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems*, ser. RATFG-RTS '99. Washington, DC, USA: IEEE Computer Society, 1999, pp. 53–.

[17] Y. Gao, Y. Wang, X. Feng, and X. Zhou, "Face recognition using most discriminative local and global features," in *Proceedings of the 18th International Conference on Pattern Recognition - Volume 01*, ser. ICPR '06. Washington, DC, USA: IEEE Computer Society, 2006, pp. 351–354.

[18] M. Mazloom and S. Ayat, "Combinational method for face recognition: Wavelet, pca and ann," in *Digital Image Computing: Techniques and Applications (DICTA), 2008*, Dec 2008, pp. 90–95.

[19] C. Feng-jun, W. Zhi-liang, X. Zheng-guang, and X. Jiang, "Facial expression recognition based on wavelet energy distribution feature and neural network ensemble," in *Proceedings of the 2009 WRI Global Congress on Intelligent Systems - Volume 02*, ser. GCIS '09. Washington, DC, USA: IEEE Computer Society, 2009, pp. 122–126.

[20] Y. Wang, J.-P. Li, J. Lin, and L. Liu, "The contourlet transfrom and svm classification for face recognition," in *Apperceiving Computing and Intelligence Analysis, 2008. ICACIA 2008. International Conference on*, Dec 2008, pp. 208–211.

[21] V. Kabeer, T. M. Thasleema, and N. K. Narayanan, "Face recognition using state space parameters and k-nn classifier," in *Innovations in Information Technology, 2007. IIT '07. 4th International Conference on*, Nov 2007, pp. 476–480.

[22] E. Demir, L. Akarun, and E. Alpaydin, "Two-stage approach for pose invariant face recognition," in *Acoustics, Speech, and Signal Processing. Proceedings of the 2000 IEEE International Conference on*, vol. 6, 2000, pp. 2342–2344 vol.4.

[23] V. N. Vapnik, *The nature of statistical learning theory*. New York, NY, USA: Springer-Verlag New York, Inc., 1995.

[24] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies, "Image coding using wavelet transform." *IEEE Transactions on Image Processing*, vol. 1, no. 2, pp. 205–220, 1992.

[25] S. Mallat, "A theory for multiresolution signal decomposition: the wavelet representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 7, pp. 674–693, Jul 1989.

[26] ——, *A Wavelet Tour of Signal Processing: The Sparse Way.*, 3rd ed. Academic Press, 2008, ch. 7, pp. 410–418.

[27] P. Jiang and Y. Li, "A hybrid face recognition algorithm based on wt, nmfs and svm," in *Cybernetics and Intelligent Systems, 2008 IEEE Conference on*, Sept 2008, pp. 734–737.

[28] C. J. C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Min. Knowl. Discov.*, vol. 2, no. 2, pp. 121–167, Jun. 1998.

[29] S. Haykin, *Neural Networks and Learning Machines*. Prentice Hall, 2008.

[30] Yale. Face database: http://vision.ucsd.edu/content/yale-face-database.

[31] P. Viola and M. Jones, "Robust real-time face detection," *International Journal of Comp. Vision*, vol. 57, no. 2, pp. 137–154, 2004.

[32] O. Chapelle, V. Vapnik, O. Bousquet, and S. Mukherjee, "Choosing multiple parameters for support vector machines," *Mach. Learn.*, vol. 46, no. 1-3, pp. 131–159, Mar. 2002.

# ECG Biometric Recognition based on fiducial features using Support Vector Machines

Felipe G. S. Teodoro, Daniel M. M. da Costa, Sarajane M. Peres, Clodoaldo A. M. Lima
Programa de Pós-Graduação em Sistemas de Informação, Escola de Artes, Ciências e Humanidades, Universidade de São Paulo
Av. Arlindo Bettio, 1000, 03828-000 São Paulo, SP, Brasil
Email:{fteodoro, daniel.moura.costa, sarajane, c.lima}@usp.br

***Resumo*—The electrocardiogram signal (ECG) is a non-invasive, inexpensive, safe and fast method for cardiac disease diagnosis. Recently, some studies have shown the feasibility of using ECG as a biometric signal. The validity of such an assertion is based on uniqueness represented by physiological and geometrical differences among ECG signals from hearts of different individuals. Biometric recognition based on ECG signals is commonly built using two main approaches for feature extraction: fiducial and non-fiducial. Fiducial methods use points of interest within single heartbeat waveforms, while non-fiducial aim at extracting discriminative information without localizing reference points. The aim of this paper is to evaluate the impact of the fiducial feature set on recognition performance when Support Vector Machine (SVM) is adopted to solve ECG multi-class classification problems. Overall, experimental results using the PTB (Physikalisch-Technische Bundesanstalt) diagnostic ECG database demonstrated that the choice of the feature type is an important factor to take into account if the final classifier's accuracy is a requirement to be pursued.**

Keywords: Biometric System, Electrocardiogram, Fiducial points, Support Vector Machine.

## I. Introdução

Nos últimos anos, sinais biológicos vêm sendo estudados para identificação de pessoas. Esses sinais diferem das abordagens clássicas de biometria porque constituem-se em uma modalidade de informação que está ligada à uma função biológica. Trata-se de uma abordagem que utiliza sinais biológicos como Eletrocardiograma (ECG), Eletroencefalograma (EEG) e Resposta Eletrodermal (EDR) como entrada para sistemas de identificação ou autenticação [1]. Esses sinais são produzidos pelo coração, cérebro e sistema nervoso autônomo, respectivamente, e são capturados usando equipamentos não invasivos. A abordagem básica para emprego destes sinais em biometria consiste em gravá-los e usá-los como assinatura biométrica.

O Eletrocardiograma (ECG) é um sinal não invasivo que mede as atividades elétricas do coração, fornecendo informação sobre o seu estado de saúde. Análise de ECG, como técnica de diagnóstico, tem sido uma área de pesquisa bastante ativa nas últimas décadas. Recentemente, vários trabalhos sugeriram a possibilidade de usar o ECG como uma modalidade biométrica [2], [3], [4]. Diferenças na estrutura do coração, tais como geometria do peito, posição e tamanho geram características únicas em um sinal de ECG que podem ser usadas como modalidade biométrica. Além disso, o sinal de ECG é capaz de fornecer um indicador sobre o fato do usuário a ser identificado estar vivo, diminuindo a possibilidade de falsificação no processo de identificação ou autenticação.

Os métodos propostos para reconhecimento biométrico baseado em ECG podem ser agrupados de acordo com o número de canais de dados, configuração operacional, método de extração de características e o tipo de classificador. Em relação ao método de extração de características, os métodos podem ser classificados em fiduciais, não fiduciais ou híbridos. Os métodos fiduciais extraem características dinâmicas e temporais do sinal de ECG [2], [5], enquanto os métodos não fiduciais capturam alguma característica do sinal através de estratégias como as transformadas *wavelets* [6] e coeficientes de autocorrelação [7]. Os métodos híbridos fazem a combinação de ambas as abordagens [8].

Em [3] foi proposto um conjunto formado por 36 características que podem ser extraídas a partir do sinal de ECG. Neste artigo, nós investigamos o impacto da informação contida nessas características, em termos de taxa de reconhecimento, sobre o desempenho da Máquina de Vetor Suporte (SVM) usada como classificador. A escolha da SVM foi baseada nos excelentes resultados evidenciados na literatura quando da sua aplicação em vários problemas de classificação. A fim de encontrar um subconjunto de características com maior poder discriminatório foi empregada a Análise de Componentes Principais. O desempenho desse subconjunto foi examinado e contrastado em relação ao conjunto total de características extraídas.

Este artigo está organizado nas seguintes seções: uma descrição formal do sinal de ECG, considerando o contexto de reconhecimento biométrico, é apresentada na Seção II; na Seção III são descritos os aspectos relacionados com os sistemas biométricos baseados em ECG, como a extração de características e as técnicas mais comumente empregadas para a tarefa de classificação e trabalhos correlatos; uma revisão sobre a SVM é descrita na Seção IV; a metodologia proposta é apresentada na Seção V e os resultados obtidos são apresentados na Seção VI. Finalmente na Seção VII as conclusões são descritas.

## II. Eletrocardiograma

O ECG é uma gravação das variações dos potenciais elétricos gerados pela atividade cardíaca. Esse sinal é capturado por eletrodos colocados sobre a superfície da pele em determinados pontos do corpo, de forma que é possível verificar a sequência e intensidade de ondas que representam a despolarização e polarização do átrio e ventrículo, medidas em unidades de mili volts (mV).

Um sinal de ECG normal é composto pelas ondas P, Q, R, S e T. A onda P é detectada quando a despolarização acontece, pouco antes da sístole atrial. Seguindo a onda P aparecem as ondas Q, R e S, também chamadas de complexo QRS, que indicam a despolarização dos ventrículos, antes da sístole ventricular. Finalmente, a onda T representa a fase de polarização dos ventrículos, que indica o começo da diástole ventricular. A diástole atrial no ciclo cardíaco acontece no mesmo tempo do complexo QRS, entretanto uma onda T atrial é raramente registrada durante um ECG.

Para cada intervalo entre as ondas, é possível extrair a duração da sístole: intervalo de tempo P-Q ou P-R, o qual indica o período do começo da contração do átrio, e intervalo Q-T, o qual corresponde ao tempo que decorre desde a despolarização até o fim da repolarização dos ventrículos. A taxa cardíaca do sinal de ECG pode ser calculada obtendo o intervalo entre dois complexos QRS em sequência.

## III. Sistemas Biométricos baseado em ECG

Um sistema de reconhecimento biométrico baseado no sinal de ECG é apresentado na figura 1. Primeiro, o sinal de ECG é adquirido, pré-processado e convertido para um formato digital, sendo então filtrado. Após a filtragem do sinal é realizada a etapa de detecção dos complexos QRS do sinal. Em seguida são aplicadas técnicas de extração de características sobre os complexos QRS detectados, gerando um vetor de características a ser utilizado como entrada para o classificador. Após a construção do classificador, o usuário pode ser reconhecido [9].

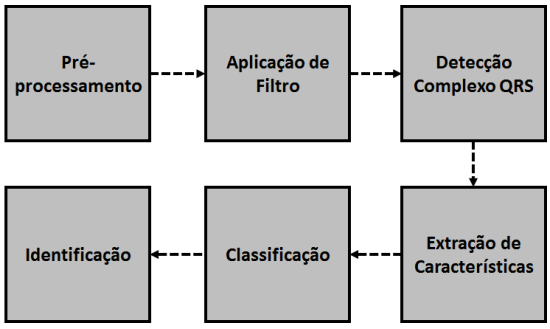



Figura 1. Diagrama de blocos de um sistema biométrico baseado em ECG [10].

### A. Extração de Características

A maioria dos estudos propostos na literatura sobre ECG como modalidade biométrica emprega um único canal de captura. Em [2] foi mostrado que um único canal contém informação suficiente para permitir o reconhecimento biométrico. O estudo realizado neste artigo segue esta metodologia.

Como citado na Seção VI, baseado nas informações extraídas do sinal de ECG, é possível classificar os sistemas de reconhecimento baseado em ECG como métodos fiduciais, não fiduciais e híbridos. Os métodos fiduciais extraem medidas temporais, amplitudes, áreas, ângulos ou características dinâmicas (através dos batimentos cardíacos) do complexo QRS do sinal de ECG. As características incluem, mas não são limitadas, comprimento das ondas P, R, T, a distância entre os picos da onda, área das ondas e informação sobre suas inclinações [4].

Para extração das características fiduciais é necessário realizar a segmentação dos batimentos cardíacos e detecção do complexo QRS. Técnicas de processamento de sinal digital têm sido utilizadas visando esse objetivo. Neste artigo é discutida a utilização do algoritmo de Pan-Tompkins para fazer a detecção dos complexos QRS [11].

### B. Detecção de QRS

O algoritmo Pan-Tompkins [11] é descrito na Figura 2. Primeiro, dois filtros (um filtro passa baixa e um filtro passa alta) são aplicados de forma a destacar as características do sinal que são referentes aos complexos QRS, ao mesmo tempo em que diminui as informações irrelevantes. Esses filtros juntos formam um filtro passa-banda de largura de banda que varia de 5 a 11 Hz. Em seguida, um filtro derivativo é aplicado ao sinal, o qual minimiza o ruído e também enfatiza as informações provenientes dos complexos QRS. Na próxima etapa, os valores resultantes do filtro derivativo são elevados ao quadrado, enfatizando ainda mais os complexos QRS. Posteriormente, um integrador de janela móvel é aplicado, o qual calcula a média das últimas 30 amostras do sinal. Após a aplicação do integrador de janela móvel, é possível identificar alguns picos, os quais correspondem ao pico R. Em seguida, alguns limiares são ajustados automaticamente para detectar o pico das ondas Q e S. Quando o intervalo entre dois picos R é menor que 360 ms, uma análise é realizada para determinar se o complexo QRS foi identificado corretamente ou se ele corresponde a uma onda T. Se a inclinação máxima que ocorre nessa onda é menor que metade da onda QRS anterior, então essa inclinação é identificada como uma onda T, caso contrário classificamos o intervalo como um novo complexo QRS.

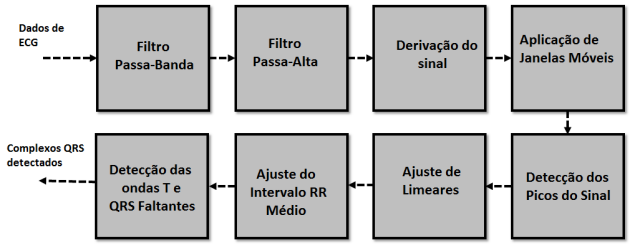



Figura 2. Estágios de Detecção do complexo QRS do algoritmo de Pan-Tompkins.

O algoritmo de Pan-Tompkins não menciona nada a respeito da detecção da onda P. Neste artigo, foi realizada uma busca para analisar $m$ pontos antes da onda Q de modo a detectar o máximo local, o qual representa o pico da onda P. De modo a melhorar a detecção do pico da onda T foi realizada uma análise de $n$ pontos após a onda S, de forma localizar o máximo local, que é o pico da onda T.

Com o intuito de melhorar a precisão dos valores obtidos para os picos das ondas PQRST, foi empregada uma janela deslizante sobre o sinal de EEG contendo 5 batimentos. Todos estes batimentos foram centralizados em relação ao pico R e foi gerado um novo batimento. De forma geral, para cada complexo QRS$(i)$ é calculado um novo QRS usando a média

dos pontos do QRS$(i)$ até o QRS$(i+k-1)$, sendo $k$ o número de QRS que foram utilizados para o cálculo, como mostrado na figura 3.

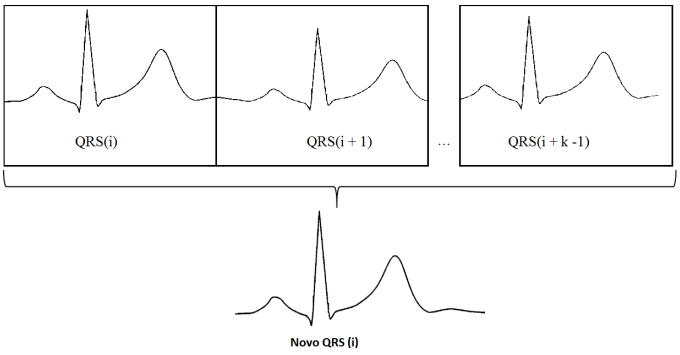



Figura 3. Geração do novo sinal ECG.

A Tabela I apresenta as características extraídas do sinal de ECG. A título de ilustração estas características são mostradas na figura 4.

Tabela I. CONJUNTO DE CARACTERÍSTICAS FIDUCIAIS.

| **Características** | |
|---|---|
| **Intervalos** | QR, RS, PQ, PR, RT, PS, QT, PT |
| **Amplitudes** | QRS, PT, SP, QP, PR, TR, QS, QT, ST, SR |
| **Ângulo dos Segmentos** | $\widehat{PQR}$, $\widehat{QRS}$, $\widehat{RST}$ |
| **Área dos polígonos formados a partir das coordenadas** | $\{x(P), y(P)\}$, $\{x(P), y(R)\}$ , $\{x(R), y(R)\}$ |
| | $\{x(P), y(P)\}$, $\{x(Q), y(Q)\}$ , $\{x(P), y(Q)\}$ |
| | $\{x(P), y(P)\}$, $\{x(R), y(R)\}$ , $\{x(Q), y(Q)\}$ |
| | $\{x(R), y(R)\}$, $\{x(S), y(S)\}$ , $\{x(T), y(T)\}$ |
| | $\{x(S), y(S)\}$, $\{x(T), y(S)\}$ , $\{x(T), y(T)\}$ |
| | $\{x(T), y(T)\}$, $\{x(T), y(R)\}$ , $\{x(R), y(R)\}$ |
| **Área do polígono formado a partir das coordenadas do complexo QRS** | $\{x(Q), y(Q)\}$, $\{x(R), y(R)\}$ , $\{x(S), y(S)\}$ |
| **Soma das características** | Soma de todas as áreas |
| **Distância Euclidiana** | $\{x(T), y(R)\}$, $\{x(Q), y(Q)\}$ |
| | $\{x(P), y(R)\}$, $\{x(S), y(S)\}$ |

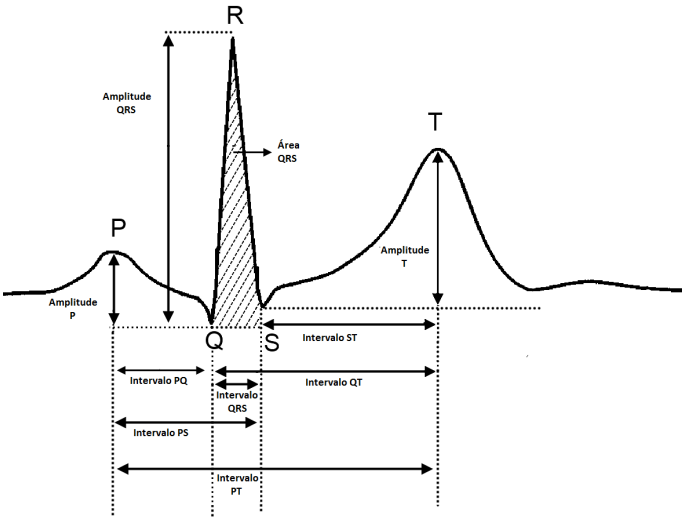



Figura 4. Características extraídas do complexo QRS.

### C. Análise de Componentes Principais

Análise de Componentes Principais (PCA), também conhecido como Transformada de Karhunen-Loeve, é um método de análise estatística dos dados, cujo objetivo é promover a redução da dimensionalidade, maximizando a variabilidade dos dados [12]. De forma geral, o PCA é uma transformação linear na qual um novo sistema de coordenadas para os dados é escolhido de forma que a maior variância entre as projeções dos dados com dimensionalidade $d$ recaia sobre o primeiro eixo, a segunda maior variância sobre o segundo eixo e assim por diante. Desta forma, pretende-se escolher um número de eixos $m$, tal que $m < d$, onde reside a maior variabilidade dos dados.

A projeção de $x$ na direção de $w$ pode ser calculada como:

$$z = w^T x$$

As direções de projeção podem ser obtidas por meio dos seguintes passos:

- Calcule a média de $x$: $\overline{x} = \frac{1}{N}\sum_{i=1}^{N} x_i$.
- Calcule a matriz de covariância: $\Sigma = \frac{\sum_{i=1}^{n}(x_i - \overline{x})^T(x_i - \overline{x})}{n}$.
- Calcule os autovetores e autovalores da matriz de covariância $\Sigma$.
- Selecione os $m$ autovetores que correspondem aos $m$ maiores autovalores.
- Projete os dados nas direções selecionadas.

Após essa transformação linear, os dados estão representados no espaço $m$-dimensional, sendo $m < d$.

### D. Trabalhos Relacionados

De acordo com [4] diversos trabalhos empregaram SVMs com o objetivo de identificação utilizando sinal de ECG. Esses trabalhos apresentaram uma taxa de reconhecimento acima de $80\%$, independente do método de pré-processamento ou de extração de características utilizado.

Em [13], informações temporais e cepstral de cada batimento cardíaco foram utilizadas como conjunto de características para realizar a identificação de usuários. Nesse trabalho, dois classificadores foram utilizados em conjunto, SVM e Modelo de Mistura de Gaussianas (GMM). O conjunto de características para a SVM foi construído usando os coeficientes dos polinômios de Hermite usado para capturar a forma do sinal de ECG. Para as GMM foi utilizada os coeficientes cepstrais obtidos do sinal de ECG. Análise de Discriminante Linear foi empregada para reduzir a dimensionalidade das características extraídas. As saídas dos dois classificadores foram combinadas de forma a produzir uma única saída. Os autores obtiveram uma taxa de reconhecimento acima de $90\%$.

De acordo com os estudos de [2], [14], [15], métodos de redução de dimensionalidade, como PCA, Análise de Componentes Independentes e Análise de Fatores, podem melhorar a taxa de reconhecimento do classificador, selecionando as melhores características a serem utilizadas no processo de

identificação. Em todos esses trabalhos foi utilizado PCA para reduzir a dimensionalidade dos dados.

Em [10] foi realizado uma comparação entre dois classificadores, SVM e $k$-Vizinhos mais Próximos para reconhecimento biométrico. SVM obteve o melhor desempenho em termos de taxa de reconhecimento. Ye, Coimbra e Kumar [16] empregaram SVM, com a função *kernel* gaussiana de base radial, sendo que o valor do parâmetro do *kernel* foi selecionado através da abordagem validação cruzada com 10-fold. Os autores obtiveram uma taxa média de acerto de $88.34\%$, considerando todos os experimentos realizados.

Existem outros tipos de classificadores que tem sido utilizado na identificação de pessoas utilizando ECG [4], tais como: $k$-Vizinhos mais Próximos, Análise de Discriminante Linear (LDA), Redes Neurais Artificiais (NNs), Modelos de Classificadores Generativos.

## IV. Máquinas de Vetores Suporte

Máquina de Vetores Suporte (SVM) é uma das principais metodologias para problemas de classificação, sendo empregada em diversos domínios. Geralmente é indicada para o uso em problemas com dados de alta dimensionalidade [10]. Nesta seção, o emprego de SVM em problemas de classificação é descrito.

Considere um problema de classificação com um conjunto de treinamento $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$, com entrada $\mathbf{x}_i \in \Re^m$ e saída $y_i \in \{\pm 1\}$. Seja um mapeamento não linear $\phi : \Re^m \longrightarrow \Re^n$, onde $n$ é muito maior que $m$ de modo que os vetores de entrada são mapeados em um espaço de alta dimensionalidade [17]. Nesse espaço (chamado espaço de característica), a SVM constrói um hiperplano $\mathbf{w}^T\phi(\mathbf{x}) + b$, onde a margem entre as classes é maximizada. Pode-se mostrar que $\mathbf{w}$, para esse hiperplano ótimo, pode ser definido como a combinação linear $\phi(x)$ tal que $\mathbf{w} = \sum_{i=1}^N \alpha_i y_i \phi(\mathbf{x}_i)$ [18].

Na formulação das SVMs [19], o hiperplano de separação ótimo $\mathbf{w}$ pode ser determinado minimizando a seguinte função:

$$\min_{w,b,\xi_i} J(w, b, , \xi_i) = \frac{1}{2}(\mathbf{w}^T\mathbf{w}) + C\sum_{i=1}^{N} \xi_i, \qquad (1)$$

sujeita às restrições:

$$y_i[\mathbf{w}^T\phi(\mathbf{x_i}) + b] \geq 1 - \xi, i = 1, \ldots, N. \qquad (2)$$

onde $C$ estabelece o compromisso entre a complexidade do modelo e o erro de treinamento e $\xi_i, i = 1, \ldots, N$, são variáveis de folga, medindo a diferença (erro) entre a saída desejada $y_i$ e a produzida pela SVM.

Aplicando a técnica de Lagrange ao problema de otimização da equação 1, juntamente com as restrições da expressão 2, resulta um problema quadrático com restrições, cuja solução é o ponto de sela do funcional lagrangeano [19]:

$$\max_{\alpha} J(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i\alpha_j y_i y_j \phi(\mathbf{x}_i)^T\phi(\mathbf{x}_j) \qquad (3)$$

sujeita à $\sum_{i=1}^N \alpha_i y_i = 0$ e $0 \leq \alpha_i \leq C$, para $i = 1, \ldots, N$, onde $\alpha_i (i = 1, \cdots, N)$ são os multiplicadores de Lagrange.

Para obter $\phi(\mathbf{x}_i)^T\phi(\mathbf{x}_j)$ na equação 3, não é necessário calcular $\phi(\mathbf{x}_i)$ ou $\phi(\mathbf{x}_j)$ explicitamente. Ao invés disso para algum $\phi$, é possível construir uma matriz de *kernel* $K$ tal que $K(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^T\phi(\mathbf{x}_j)$ [17].

Os *kernels* são utilizados para realizar o mapeamento não linear dos dados de entrada para um espaço de características de alta-dimensionalidade de uma maneira computacionalmente eficiente. Diferentes funções *kernel* dão origem a diferentes espaços de características, influenciando na capacidade de generalização do classificador resultante. Então, podemos definir que um passo importante a ser considerado na construção de SVMs é como escolher a melhor função *kernel* e o valor do seu parâmetro para cada problema.

Entre diversos tipos de *kernel*[17], os resultados mostrados na Seção VI foram obtidos com o uso do *kernel* RBF (Função de Base Radial):

$$K_{RBF}(\mathbf{x}_i, \mathbf{x}_j) = exp\left(-\frac{\parallel \mathbf{x}_i - \mathbf{x}_j \parallel^2}{2\sigma^2}\right).$$

onde $\sigma$ representa a variância, e o *kernel* Linear:

$$K_{Linear}(\mathbf{x}_i, \mathbf{x}_j) = x_i^T x_j.$$

Usando uma função *kernel*, a função objetiva 3 pode ser reescrita como:

$$\max_{\alpha} J(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i\alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) \qquad (4)$$

Para os exemplos de treinamento que estão localizados na fronteira de decisão, o respectivo $\alpha_i$ de cada exemplo é maior que 0, como mostrado pelo Teorema de Kuhn-Tucker [18]. Esses exemplos são conhecidos como vetores suporte, sendo que a quantidade de vetores suporte geralmente é muito menor que $N$ e é ao erro de generalização do classificador [19]. Um vetor de teste $\mathbf{x} \in \Re^m$ é então classificado de acordo com:

$$f(n) = sign[\mathbf{w}^T\phi(\mathbf{x}) + b] = sign(\sum_{i=1}^{N} \alpha_i y_i K(\mathbf{x}, \mathbf{x_i}) + b)$$

.

De acordo com [20] uma estratégia comum para a validação do classificador SVM é aplicar a estratégia de validação cruzada. A validação cruzada consiste em dividir o conjunto de treinamento do classificador em $C$ subconjuntos (*folds*) de tamanhos iguais. Então é realizado um processo de treinamento e teste $C$ vezes, sendo que a cada vez um subconjunto $c_i$ é selecionado para ser testado, enquanto que os outros subconjuntos são usados para o treinamento do classificador. Ao final dessa abordagem obtém-se uma análise mais precisa sobre o desempenho do classificador para todos os elementos do conjunto de dados original.

## V. METODOLOGIA PROPOSTA

Nessa seção, são fornecidos os detalhes sobre o conjunto de dados utilizado nos experimentos e como esses experimentos foram configurados. Os resultados são apresentados em termos da taxa de reconhecimento alcançada pelas SVMs, considerando como vetor de entrada o conjunto de características fiduciais extraído do sinal de ECG e as características produzidas pelo emprego do PCA. Para cada método de extração de características, é também apresentado o valor ótimo obtido para o parâmetro de *kernel*. Neste artigo, a abordagem *um contra um* foi aplicado para adaptar a SVM (um classificador binário) para um problema com múltiplas classes.

### A. Descrição do Conjunto de Dados

De forma a avaliar o desempenho obtido com a metodologia proposta, foi utilizada uma base de ECG pública, a PTB ECG Database, que contém o sinal de ECG obtido de 290 pessoas. Os sinais foram obtidos de 209 homens e 81 mulheres com idade entre 17 e 87 anos. O número de gravações por pessoa varia de 2 (coletados em um único dia) a 20 (coletados periodicamente durante 6 meses) [21]. Cada sinal inclui os sinais brutos e filtrados, apenas os sinais filtrados foram usados nos experimentos apresentados neste artigo.

### B. Pré-processamento

O pré-processamento é um dos passos mais importantes em um sistema biométrico. Ele é composto por três fases: filtragem, normalização e segmentação. Na filtragem, um filtro passa-banda Butterworth de ordem 4 com frequência de corte de 1Hz e 40Hz[8] foi utilizado de forma a remover os ruídos presentes no sinal. Em seguida, para cada indivíduo, o ECG médio foi subtraído de cada amostra do sinal de ECG de modo a remover o efeito de deslocamento do sinal. Posteriormente, o sinal é dividido pelo desvio padrão. Ao final desse processo, tem-se um sinal filtrado e normalizado com média 0 e desvio padrão 1. A última etapa corresponde a detecção dos complexos QRS. Para cada usuário com batimento normal, foram detectados os complexos QRS usando o algoritmo de Pan-Tompkins.

Como mencionado anteriormente na Seção III, dois métodos diferentes foram utilizados para gerar o vetor de características: o primeiro método utiliza o conjunto de características fiduciais descritos na Seção III-A; o segundo utiliza a abordagem PCA sobre esse conjunto de características fiduciais de modo a realizar uma redução de dimensionalidade do vetor de característica gerado, utilizando diferentes quantidades de componentes principais. O desempenho do modelo classificador foi avaliado de acordo com a taxa de reconhecimento.

### C. Configuração de Experimentos

Originalmente, as SVMs foram projetadas para problemas de classificação binária. Diferentes métodos têm sido propostos de modo a adaptá-las para problemas com múltiplas classes. Neste artigo, de modo a utilizar a SVM para um problema com múltiplas classes, a abordagem *um contra um* foi adotada. Como mencionado anteriormente, o desempenho obtido pelas SVMs depende de uma escolha apropriada dos hiperparâmetros, tais como, valor do parâmetro do *kernel* ($\sigma$) e parâmetro de regularização $C$. Neste artigo o parâmetro de regularização $C$ foi mantido constante, com um valor igual a 100. Em [22] é apresentada uma abordagem para a definição do valor do parâmetro do *kernel*, sendo que a opção adotada aqui foi utilizar valores de $\sigma$ como $2^i$, com $i = -10, -9, \cdots, 14, 15$ para o *kernel* RBF. Para o *kernel* Linear não há nenhum parâmetro a ser definido. Para cada valor de $i$ foi aplicado o processo de validação cruzada com 10-*folds* para obter uma medida de desempenho mais confiável.

## VI. EXPERIMENTOS E RESULTADOS

Analisando os resultados na Tabela II, verifica-se que os modelos obtidos, independente do tipo de *kernel*, forneceram resultados bastante satisfatórios. A taxa mínima de acerto foi de 93.03% de reconhecimento, esta é superior a algumas abordagens descritas na Seção III-D. Os melhores resultados foram alcançados utilizando todo o conjunto de características extraídas dos pontos fiduciais, o qual produziu uma taxa de reconhecimento maior que 99%. Isto mostra que as características extraídas apresentam uma boa capacidade discriminatória.

Tabela II. TAXA DE RECONHECIMENTO MÉDIO OBTIDA COM OS MELHORES MODELOS DE SVM.

| Kernel | Estratégia | # de Características | $\sigma$ | Taxa de Reconhecimento % |
|---|---|---|---|---|
| RBF | Todas as Características | 31 | 8,16 | 99,6793±0,3919 |
| | Complexo QRS | 7 | 0,5 | 99,4304±0,6957 |
| | PCA | 2 | 0,25 | 95,7617±1,7966 |
| | | 3 | 0,25 | 98,2194±0,5035 |
| | | 4 | 0,5 | 98,6112±0,5921 |
| | | 5 | 1,2 | 99,4657±0,4196 |
| | | 10 | 16 | 99,5370±0,4456 |
| Linear | Todas as Características | 31 | - | 99,6082±0,3919 |
| | Complexo QRS | 7 | - | 99,2168±0,5761 |
| | PCA | 2 | - | 93,0203±1,9920 |
| | | 3 | - | 97,1858±1,0710 |
| | | 4 | - | 98,1837±0,6811 |
| | | 5 | - | 99,3945±0,4128 |
| | | 10 | - | 99,5014±0,4178 |

Do ponto de vista da função *kernel*, os melhores resultados foram obtidos considerando o *kernel* RBF, independente do tipo de vetor de característica utilizado. O conjunto de parâmetros adequado para o *kernel* RBF ficou no intervalo $[2^{-2}, 2^4]$. Isto mostra que, caso seja utilizado o *kernel* RBF, é importante que o parâmetro $\sigma$ seja selecionado adequadamente. Além disso, este intervalo é importante, pois pode ser empregado como ponto de partida para outros conjuntos de dados.

O melhor sistema de reconhecimento foi obtido usando o *kernel* RBF, com todas as características fiduciais. Neste caso, a taxa de reconhecimento obtida foi de $99,67\%$ de acerto, para $\sigma$ igual a $8$ ou a $16$.

## VII. CONCLUSÃO

Neste trabalho foi apresentado um estudo sobre o impacto das características extraídas do sinal de ECG baseada nos pontos fiduciais e nas características extraídas pelo PCA. Como classificador foi empregado o modelo de aprendizado de máquina SVM e adotado a estratégia um contra um para tratar problema com múltiplas classes.

Analisando os resultados obtidos, observa-se que o sinal ECG para identificação de usuários possui alta potencialidade,

sendo que foram obtidos resultados com taxa de reconhecimento superior a 93%. Os melhores resultados foram obtidos usando *kernel* RBF e considerando todas as características extraídas, neste caso obtivemos uma taxa reconhecimento de $99,07\%$.

Nos experimentos realizados, o número de componentes selecionadas via PCA não produziu resultados superiores a aqueles obtidos usando todas as características. Isto mostra que as características extraídas são importantes para o processo de reconhecimento.

Como trabalho futuro, pretende-se avaliar a adoção de outras máquinas de aprendizado, como Least-Square SVMs e Máquinas de Aprendizado Extremo, para a tarefa de reconhecimento biométrico com base em sinal de ECG.

## REFERÊNCIAS


[1] A. Riera, S. Dunne, I. Cester, and G. Ruffini, "Starfast: a wire-less wearable eeg/ecg biometric system based on the enobio sensor," in *Proceedings of the International Workshop on Wearable Micro and Nanosystems for Personalised Health*, 2008.

[2] L. Biel, O. Pettersson, L. Philipson, and P. Wide, "Ecg analysis: a new approach in human identification." *IEEE Transactions on Instrumentation and Measurement*, vol. 50, no. 3, pp. 808–812, 2001.

[3] M. Tantawi, K. Revett, M. Tolba, and A. Salem, "A novel feature set for deployment in ecg based biometrics," in *Computer Engineering Systems (ICCES), 2012 Seventh International Conference on*, 2012, pp. 186–191.

[4] I. Odinaka, P.-H. Lai, A. D. Kaplan, J. A. O'Sullivan, E. J. Sirevaag, and J. W. Rohrbaugh, "Ecg biometric recognition: A comparative analysis." *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 6, pp. 1812–1824, 2012.

[5] J. M. Irvine and S. A. Israel, "A sequential procedure for individual identity verification using ecg," *EURASIP J. Adv. Signal Process*, vol. 2009, pp. 3:1–3:13, Jan. 2009.

[6] A. D. C. Chan, M. Hamdy, A. Badre, and V. Badee, "Wavelet distance measure for person identification using electrocardiograms," *IEEE Transactions on Instrumentation and Measurement*, vol. 57, no. 2, pp. 248–253, 2008.

[7] K. Plataniotis, D. Hatzinakos, and J. Lee, "Ecg biometric recognition without fiducial detection," in *Biometric Consortium Conference, 2006 Biometrics Symposium: Special Session on Research at the*, 2006, pp. 1–6.

[8] Y. Wang, F. Agrafioti, D. Hatzinakos, and K. N. Plataniotis, "Analysis of human electrocardiogram for biometric recognition," *EURASIP J. Adv. Signal Process*, vol. 2008, Jan. 2008.

[9] F. Canento, A. Lourenço, H. Silva, and A. Fred, "Review and Comparison of Real Time Electrocardiogram Segmentation Algorithms for Biometric Applications," in *Proceedings of the 6th Int'l Conference on Health Informatics (HEALTHINF)*, 2013.

[10] A. Lourenço, H. Silva, and A. Fred, "ECG-Based Biometrics: A Real Time Classification Approach," in *Proceedings of the 22nd IEEE International Workshop on Machine Learning for Signal Processing*, 2012.

[11] J. Pan and W. J. Tompkins, "A real-time qrs detection algorithm," *IEEE Transactions on Biomedical Engineering*, vol. BME-32, no. 3, pp. 230–236, 1985.

[12] K. Fukunaga, *Introduction to Statistical Pattern Recognition, Second Edition (Computer Science & Scientific Computing)*, 2nd ed. Academic Press, Oct. 1990.

[13] M. Li and S. S. Narayanan, "Robust ecg biometrics by fusing temporal and cepstral information," in *Proceedings of 20th International Conference on Pattern Recognition (ICPR)*, Aug. 2010.

[14] Z. Zhang and D. Wei, "A new ecg identification method using bayes' teorem," in *TENCON 2006. 2006 IEEE Region 10 Conference*, Nov 2006, pp. 1–4.

[15] M. D. Bugdol and A. W. Mitas, "Multimodal biometric system combining {ECG} and sound signals," *Pattern Recognition Letters*, vol. 38, no. 0, pp. 107 – 112, 2014.

[16] C. Ye, M. Coimbra, and B. Kumar, "Investigation of human identification using two-lead electrocardiogram (ecg) signals," in *Biometrics: Theory Applications and Systems (BTAS), 2010 Fourth IEEE International Conference on*, 2010, pp. 1–8.

[17] B. Schölkopf and A. J. Smola, *Learning with kernels : support vector machines, regularization, optimization, and beyond*, ser. Adaptive computation and machine learning. MIT Press, 2002.

[18] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*, 1st ed. Cambridge University Press, 2000.

[19] V. N. Vapnik, *Statistical Learning Theory*. Wiley-Interscience, 1998.

[20] C.-W. Hsu, C.-C. Chang, and C.-J. Lin, "A practical guide to support vector classification," Department of Computer Science, National Taiwan University, Tech. Rep., 2003.

[21] A. Goldberger, L. Amaral, and I. P. M. R. M. J. M. G. P. C.-K. S. H. Glass, L Hausdorff JM, "A real-time qrs detection algorithm," *IEEE Transactions on Biomedical Engineering*, vol. BME-32, no. 3, pp. 230–236, 1985.

[22] O. Chapelle, V. Vapnik, O. Bousquet, and S. Mukherjee, "Choosing multiple parameters for support vector machines," *Mach. Learn.*, vol. 46, no. 1-3, pp. 131–159, Mar. 2002.

# Sponsored by:

Pró-Reitoria de Pesquisa e Pós-Graduação
Universidade Federal de Uberlândia