

**Dois problemas em análise de formas
de estruturas de ramificação**

Jorge de Jesus Gomes Leandro

DISSERTAÇÃO APRESENTADA
AO
INSTITUTO DE MATEMÁTICA E ESTATÍSTICA
DA
UNIVERSIDADE DE SÃO PAULO
PARA
OBTENÇÃO DO TÍTULO
DE
MESTRE EM CIÊNCIAS

Área de Concentração: Ciência da Computação
Orientador: Prof. Dr. Roberto Marcondes Cesar Jr.

São Paulo, Julho de 2008

Dois problemas em análise de formas de estruturas de ramificação

Este exemplar corresponde à redação final da dissertação devidamente corrigida e defendida por Jorge de Jesus Gomes Leandro e aprovada pela Comissão Julgadora.

Banca Examinadora:

- Prof. Dr. Roberto Marcondes Cesar Jr. (orientador) - IME-USP.
- Prof. Dr. Marcos Craizer - Departamento de Matemática - PUC-Rio.
- Prof. Dr. Marcel P. Jackowski - IME-USP.

Agradecimentos

Pelo dom da vida e o anelo por desafios que a move em mim, sou grato ao Todo-Poderoso, o Criador.

Por inspirar-me o apreço pelos valores que verdadeiramente enobrecem o homem, sou grato aos meus pais, Dulce e Satiro.

Pelo apoio, a compreensão e a resignação, durante os longos períodos solitários que minha ausência lhes impôs, sou grato a minha esposa Sheila e filha Julia.

Pela oportunidade e a confiança em mim depositada, sou grato ao meu orientador, Prof. Roberto Marcondes César Jr.

Pela oportunidade e as valiosas idéias compartilhadas, sou grato ao colaborador, Prof. Luciano da Fontoura Costa.

Pela licença, sempre que necessária, sou grato às chefias do Departamento de Física Nuclear - IFUSP e a minha Coordenadora Prof^a. Nobuko Ueta. Pela cumplicidade, agradeço à colega Wanda G. P. Engel.

A todos os que colaboraram direta ou indiretamente para a conclusão deste trabalho, em particular aos ouvidos atentos dos colegas Evandro Drigo e Ricardo Ichiwaki.

Tudo o que te vier à mão para fazer, faze-o conforme as tuas forças, porque no além, para onde tu vais, não há obra, nem projetos, nem conhecimento, nem sabedoria alguma.

Eclesiastes 9 : 10. Rei Salomão.

Se os teus olhos se perderem nas estrelas, não verás as pedras no caminho.

Provérbio chinês.

Resumo

O presente texto descreve métodos e apresenta resultados do projeto de mestrado intitulado **Dois Problemas em Análise de Formas de Estruturas de Ramificação**. Ambos os problemas abordados estão relacionados às sub-áreas da Análise de Formas denominadas Caracterização e Descrição de Formas.

O primeiro problema consiste na investigação de um conjunto de características propostas para distinguir, primeiramente, entre estruturas de ramificação de vasos sanguíneos em imagens de retina segmentadas manualmente e automaticamente. A seguir, as mesmas características são aplicadas para discernir entre estruturas de ramificação de vasos sanguíneos em imagens de retina com e sem retinopatia diabética proliferativa (*Proliferative Diabetic Retinopathy* - PDR). A PDR é uma das patologias associadas à diabetes, que pode culminar na cegueira do indivíduo. Diagnósticos são possíveis por meio de imagens de fundo de olho e, quando efetuados precocemente, viabilizam intervenções oportunas evitando a perda da visão.

Neste trabalho, 27 imagens digitais de fundo de olho foram segmentadas por dois processos distintos, isto é, segmentação *manual* por um especialista e a segmentação *automática*, mediante a transformada contínua *Wavelet - CWT* e classificadores estatísticos [92]. Visando à caracterização destas formas, um conjunto de 08 características foi proposto. Este conjunto foi formado por três grupos, a saber: descritores tradicionais geométricos (Área, Perímetro e Circularidade), descritores associados à transformada wavelet (2º momento estatístico da distribuição de módulos da *CWT*, Entropia de Orientação da distribuição de fases da *CWT* e Curvatura) e um descritor fractal (Dimensão de Correlação - Global e Mediana). Uma Análise Discriminante Linear **LDA** revelou que as características geométricas tradicionais não detectam o início da retinopatia diabética proliferativa. A maior capacidade discriminante individual foi exibida pela Curvatura, com área sob a curva *ROC* de 0.76. Um subconjunto com 6 características apresentou grande capacidade discriminante com área sob a curva *ROC* de 0.90.

O segundo problema diz respeito à extração de contorno de estruturas de ramificação bidimensionais de neurônios tridimensionais. Este trabalho contribui originalmente com uma solução para este problema, propondo dois algoritmos desenvolvidos para Rastreamento de Ramos e Extração do

Contorno Paramétrico de estruturas de ramificação, capazes de transpor regiões críticas formadas por cruzamentos ocasionados pela projeção de estruturas 3D no plano das imagens 2D. Grande parte dos métodos baseados em contorno para análise de formas de estruturas de ramificação de células neuronais não produz representações corretas destas formas, devido à presença de sobreposições entre processos neuronais, levando os algoritmos tradicionais de extração de contorno a ignorar as regiões mais internas destas estruturas, gerando representações incompletas. O sistema proposto neste trabalho foi desenvolvido objetivando a solução do problema de extração de contorno, mesmo na presença de múltiplas sobreposições.

Inicialmente, a imagem de entrada é pré-processada, gerando um esqueleto 8-conexo com ramos de um pixel de largura, um conjunto de sementes de sub-árvores dendríticas e um conjunto de regiões críticas (bifurcações e cruzamentos). Para cada sub-árvore, o algoritmo de rastreamento rotula todos os pixels válidos de um ramo, até chegar em uma região crítica, onde o algoritmo decide a direção em que deve continuar o rastreamento. Nosso algoritmo mostrou-se robusto, mesmo quando aplicado a imagens com segmentos paralelos muito próximos. Resultados obtidos com imagens reais (neurônios) são apresentados.

Palavras-chave: retina, neurociência, células ganglionares, reconhecimento de padrões, visão computacional, análise de formas, características.

Abstract

This document describes methods and presents results from the Master of Science's research project in computer science entitled **Two Problems in Shape Analysis of Branching Structures**. Both tackled problems herein are related to Shape Analysis sub-fields, namely Characterization and Description of shapes.

The former problem consists of an investigation on a proposed set of features aimed at discriminating, firstly, between blood vessels branching structures manually and automatically segmented. In the sequel, the same features are used to assess their discriminative capability in distinguishing between blood vessels branching structures with and without proliferative diabetic retinopathy (PDR). The PDR is a pathology related to diabetes, which may lead to the blindness. Diagnosis is possible through optic fundus image analysis, which may allow timely interventions preventing vision loss.

In this work, 27 digital optic fundus images were segmented by two distinct segmentation processes, i.e.: *manual* segmentation carried out by an especialist and *automated* segmentation, through the *CWT* (Continuous Wavelet Transform) and statistical classifiers [92]. In order to characterize such shapes, a set of 8 features has been proposed. The aforementioned set was comprised of three features groups, that is: traditional geometric descriptors (Area, Perimeter and Circularity), wavelet-based descriptors (2^{nd} statistical moment from the *CWT* Modulus distribution, Orientation Entropy from the *CWT* Phase distribution and Curvature) and a fractal descriptor (Correlation Dimension - global and median). Linear Discriminant Analysis **LDA** revealed that the traditional geometric features are not able to detect early proliferative diabetic retinopathy. The largest singular discriminant capability was shown by the Curvature, with area under the *ROC* curve of 0.76. A subset of 6 features presented a good discriminating power with area under the curve of 0.90.

The second problem concerns contour extraction from *2D* branching structures of *3D* neurons. This work contributes with an original solution for such a problem, proposing two algorithms devised for Branches Tracking and Branching Structures Contour Extraction. The proposed algorithms are able to traverse *critical regions* implied by the projection of *3D* structures onto a *2D* image plane. Most of contour-based methods intended to shape analysis of neuronal branching structures fall

short of yielding proper shape representations, owing to the presence of overlapings among neuronal processes, causing the traditional algorithms for contour following to ignore the innermost regions, thus generating incomplete representations. The proposed framework system was developed aiming at the solution of the contour extraction problem, even in the presence of multiple overlapings.

The input image is pre-processed, so as to obtain an 8-connected *skeleton* with one-pixel wide *branches*, a set of *seeds* of dendritic sub-trees and a set of *critical regions* (bifurcations, crossings and superpositions). For each sub-tree, the Branches Tracking Algorithm labels all valid pixels of a *branch*, until reaching a *critical region*, where the algorithm decides about the direction to go on with the tracking. Our algorithm has shown robustness, even in images plenty of very close parallel segments. Results with real images (neurons) are presented.

Keywords: retina, neuroscience, ganglion cells, pattern recognition, computer vision, shape analysis, features.

Sumário

Lista de Abreviaturas	xi
Lista de Símbolos	xiii
Lista de Figuras	xvii
Lista de Tabelas	xix
Lista de Algoritmos	xxi
1 Introdução	1
1.1 Preâmbulo	1
1.2 Objetivos	5
1.3 Organização do Texto	5
2 Análise de Formas - Parte I - Vasos Sangüíneos	7
2.1 Caracterização de Formas	7
2.2 Características	8
2.3 Análise de Formas: desempenho de segmentador	9
2.3.1 Área	9
2.3.2 Perímetro	10
2.3.3 Circularidade	10
2.3.4 Características derivadas de Transformadas <i>Wavelet</i>	10
2.3.4.1 Curvatura	12
2.3.4.2 Entropia de Orientação	13
2.3.4.3 2º Momento Estatístico da Magnitude	14
2.3.5 Dimensão de Correlação - ν	14
2.3.5.1 Um panorama sobre conceitos de dimensão	14

2.3.5.2	A análise fractal aplicada a imagens de retina	15
2.3.5.3	A dimensão de correlação como alternativa à dimensão fractal	16
2.3.5.4	Algoritmos para o cálculo da Dimensão de Correlação	21
2.4	Resultados	25
2.4.1	Validação do Segmentador Automático	25
2.4.1.1	Comparação entre segmentadores por curvas <i>ROC</i>	28
2.4.1.2	Comparação preliminar entre segmentadores por Coeficiente de Correlação e MSE	28
2.4.1.3	Comparação entre Segmentadores por Teste de Hipóteses	31
	Fundamentos de Testes Estatísticos de Hipóteses.	32
	Testando nossas hipóteses.	33
2.4.2	Validação de Características	34
2.4.2.1	Teste de Hipóteses para avaliar o poder discriminante das características	34
	Testando nossas hipóteses.	34
2.4.2.2	Curvas ROC para avaliar o poder discriminante individual das características	36
	Fundamentos da Teoria de Reconhecimento de Padrões - Classificação.	36
2.4.2.3	Curvas ROC para avaliar o poder discriminante de conjuntos de características	41
	Seleção Seqüencial Regressiva ¹ :	42
	Seleção Seqüencial Progressiva ² :	42
	Análise Discriminante Linear ³ :	43
	Classificador Bayesiano Ingênuo	44
3	Análise de Formas - Parte II - Neurônios	47
3.1	Motivação	47
3.2	Conceitos	53
3.2.1	Tipos de Células Ganglionares de Retina	53
3.2.2	Terminologia	55
3.3	Um panorama dos métodos	56
3.4	Os algoritmos	60
3.4.1	Pré-processamento	60
3.4.2	Rastreamento de Ramos	63
3.4.2.1	Avaliação da Continuidade da Orientação da Tangente	64

¹SBS - Sequential Backward Selection

²SFS - Sequential Forward Selection

³LDA - Linear Discriminant Analysis

3.4.2.2	Classificação de Região Crítica	65
3.4.3	Extração de Contorno Paramétrico	66
3.4.3.1	Encontrando o primeiro pixel	67
3.4.3.2	Encontrando o próximo pixel	67
3.4.3.3	Transpondo regiões críticas	68
3.5	Resultados	71
4	Conclusão	81
4.1	Comentários Finais	81
4.1.1	Comentários sobre a Análise de Formas de Vasos Sangüíneos	81
4.1.2	Comentários sobre a Análise de Formas de Neurônios	82
4.2	Trabalhos Futuros	84
4.2.1	Análise de Formas de Vasos Sangüíneos - perspectivas	84
4.2.2	Análise de Formas de Neurônios - perspectivas	84
A	Tabelas com Resultados da Análise 1	87
B	Avaliando a similaridade entre segmentações	93
C	Integral de Correlação - Pseudocódigo	97
D	Pré-Processamento - Pseudocódigo	99
E	BTA - Pseudocódigo	101
F	BSCEA - Pseudocódigo	109
G	Caracterização de Regiões Críticas	113
H	Publicações	115
H.1	<i>Classification of Pathology in Diabetic Eye Disease</i>	116
H.2	<i>Automated segmentation of retinal blood vessels and identification of proliferative diabetic retinopathy</i>	122
H.3	<i>Determining the branching structure of 3D structures from respective 2D projections</i>	132
H.4	<i>Automatic Contour Extraction from 2D Neuron Images</i>	141
Referências Bibliográficas		169
Índice Remissivo		177

Lista de Abreviaturas

AUC	Área Sob a Curva (<i>Area Under the Curve</i>)
BSCEA	Algoritmo de Extração de Contorno (<i>Branching Structure Contour Extraction Algorithm</i>).
BTA	Algoritmo de Rastreamento de Ramos (<i>Branches Tracking Algorithm</i>).
CFT	Transformada contínua de Fourier (<i>Continuous Fourier transform</i>).
CWT	Transformada Contínua Wavelet (<i>Continuous Wavelet Transform</i>).
DFT	Transformada discreta de Fourier (<i>Discrete Fourier transform</i>).
DLA	Agregação por Difusão Limitada (<i>Diffusion Limited Aggregation</i>).
f.d.p.	Função Densidade de Probabilidade.
FIG	Figura
FP	Falso Positivo (<i>False Positive</i>)
FPR	Taxa de Falsos Positivos (<i>False Positive Rate</i>)
GPA	<i>Grassberger-Proccacia Algorithm</i>

IBM	<i>International Business Machines</i>
LDA	Análise Discriminante Linear (<i>Linear Discriminant Analysis</i>).
MIT	<i>Massachusetts Institute of Technology</i>
MSE	Desvio Quadrático Médio (<i>Mean Squared Error</i>)
PDR	Retinopatia Diabética Proliferativa (<i>Proliferative Diabetic Retinopathy</i>).
ROC	Característica de Operação do Receptor (<i>Receiver Operating Characteristic</i>).
SBS	Seleção Seqüencial Regressiva (<i>Sequential Backward Selection</i>)
SFS	Seleção Seqüencial Progressiva (<i>Sequential Forward Selection</i>)
STARE	<i>Structured Analysis of the Retina</i>
STFT	Transformada de Fourier de tempo reduzido (<i>Short-time Fourier transform</i>).
TAB	Tabela
TP	Verdadeiro Positivo (<i>True Positive</i>)
TPR	Taxa de Verdadeiros Positivos (<i>True Positive Rate</i>)
v.a.	Variável Aleatória.

Lista de Símbolos

A	Área.
P	Perímetro.
C	Circularidade.
L	Número de linhas de uma imagem.
C	Número de colunas de uma imagem.
Ψ	Transformada de Fourier de ψ .
ψ	Função wavelet mãe ou analisadora.
\mathbb{R}^2	Plano real.
$*$	Complexo conjugado.
ω	Freqüênci angular.
\mathbf{x}	Vetor sobre f .
\mathbf{b}	Vetor deslocamento.
θ	Parâmetro de rotação.
a	Parâmetro de dilatação.
f	Imagen .
L^1	Espaço das funções integráveis.
L^2	Espaço das funções quadraticamente integráveis definidas em \mathbb{R}^2 .
T_ψ	Transformada contínua wavelet.
C_ψ	Constante de normalização.
ψ^*	Conjugado complexo de ψ .
$O(f(N))$	Notação para a ordem de funções com crescimento assintótico limitado superiormente por alguma função $f(N)$ a menos de uma constante multiplicativa.

$\Theta(f(N))$	Notação para a ordem de funções com crescimento assintótico limitado superiormente e inferiormente por alguma função $f(N)$ a menos de uma constante multiplicativa.
Δt	Resolução no domínio do tempo.
Δf	Resolução no domínio da freqüência.
Q	Fator de qualidade.
$1D$	Unidimensional.
$2D$	Bidimensional.
$3D$	Tridimensional.
$\nabla_g(x, y)$	Operador gradiente aplicado a uma função gaussiana no ponto (x, y) .
\log_2	Função logaritmo na base 2.
$G(f_x, f_y)$	Transformada de Fourier da função gaussiana $g(x,y)$.
\exp	Função exponencial.
$\vec{P}(t)$	Função vetor deslocamento de uma partícula puntual.
$\ddot{\vec{P}}(t)$	Função vetor aceleração de uma partícula puntual.
$k(t)$	Função curvatura.
E	Função Entropia.
m_q^M	Momento Estatístico de ordem q .
ν	Dimensão de Correlação.
\mathbb{R}^N	Espaço Euclidiano de dimensão N .
d	Dimensão Euclidiana.
d_T	Dimensão Topológica.
d_H	Dimensão de <i>Hausdorff-Besicovitch</i> .
d_C	Dimensão de Capacidade.
d_F	Dimensão Fractal.
σ	Dimensão de Informação.
$d_C(\delta)$	Dimensão δ -Capacidade.
$d_H(\delta)$	Dimensão δ -Hausdorff.
d_P	Dimensão Puntual.
d_μ	Dimensão Probabilística ou de Medida Natural.
F	Dimensão do Espaço de Imersão.
$C(\epsilon)$	Integral de Correlação.
$X_i(t)$	Série Temporal associada a um atrator.

$\Theta(x)$	Função <i>Heaviside</i> .
$c(r)$	Função de Correlação.
$\delta^F(.)$	Delta de Dirac no espaço F -dimensional.
$\binom{N}{2}$	Combinação de N dois a dois.
Δr	Raio de alcance de um ponto de referência.
r_{ij}	Distância entre os pontos X_i e X_j .
D_i	Conjunto das distâncias do ponto X_i a todos os outros.
ω_1	Classe das segmentações manuais produzidas por um especialista.
ω_2	Classe das segmentações automáticas produzidas por um software.
F^{ω_j}	Espaço de Características para a classe ω_j .
$f_k^{\vec{\omega}_j T}$	Vetor de Características do espaço F^{ω_j} referente à imagem k .
$f_{k,i}^{\omega_j}$	Componente i do Vetor de Características $f_k^{\vec{\omega}_j T}$.
$\phi_{k,i}^{\omega_j}$	Notação para a i -ésima característica referente à imagem k da classe ω_j .
r	Coeficiente de Correlação de <i>Pearson</i> .
H_0	Hipótese Nula.
H_1	Hipótese Alternativa.
$\bar{\phi}_i^{\omega_j}$	Média da i -ésima característica na classe ω_j .
σ_j	Desvio padrão para uma dada característica na classe ω_j .
t_j	Estatística para a Distribuição <i>t-student</i> para a classe ω_j .
ω_{2p}	Classe de segmentações automáticas de árvores venosas com PDR.
ω_{2n}	Classe de segmentações automáticas de árvores venosas sem PDR.
$\hat{\phi}_i$	Transformada normal da característica ϕ_i .
α	Nível de Significância.
α_i	Ação tomada.
λ_{ij}	Custo associado à ação tomada α_i , dado que o estado da natureza é ω_j .
$R(\alpha_i \vec{f}_k)$	Risco devido à ação tomada α_i dado o vetor \vec{f}_k .
$J_i(.)$	Critério de separabilidade.
$g(\vec{f})$	Função Discriminante.
$J(\omega)$	Função Critério de Fisher.
S_B	Matriz de Espalhamento Inter-classes.
S_W	Matriz de Espalhamento Intra-classe.
\hat{v}_0	Vetor diretor incidente.
\hat{v}_j	j-ésimo Vetor diretor emergente.

β	Ângulo entre vetores incidente e emergente.
γ	Ângulo entre vetores emergentes.
s_i	i-ésima Região Crítica.
E_i	Conjunto de vetores adjacentes à s_i .
$ E_i $	Cardinalidade de E_i .
$E(k)$	k-ésimo pixel do contorno E .
d_{cp}	direção do pixel atual para o anterior.
P_n	Ponteiro relativo ao pixel $E(n)$.
V_n	Último pixel do segmento atual.
$d_{E(n) \leftrightarrow V_n}$	Relação de orientação entre os pixels $E(n)$ e V_n .
D_{max}	Distância máxima assumida entre regiões críticas.

Listas de Figuras

1.1	Ubiqüidade de estruturas de ramificação - 1	2
1.2	Ubiqüidade de estruturas de ramificação - 2	3
2.1	Integral de Correlação e Estimativas	22
2.2	Características derivadas <i>CWT</i> em cores	26
2.3	Sobreposição de segmentações e descritores	27
2.4	Comparação entre segmentações: Área e Entropia	31
2.5	Sobreposição de verossimilhanças e curva ROC	39
2.6	Curva ROC para as 6 melhores características	45
3.1	Exemplos de imagens 2D de neurônios, obtidos por <i>camera lucida</i> , considerados neste trabalho. Neurônio tipo α (a). Neurônio tipo Y (b).	49
3.2	Vizinhança definida pela regra da cadeia e usada pelos algoritmos <i>BTA</i> e <i>BSCEA</i>	50
3.3	Áreas inacessíveis para extração de contornos	50
3.4	Contornos: Paramétrico e Wavelet	52
3.5	Representação de auto-intersecção	54
3.6	Terminologia adotada	56
3.7	<i>Caracterização de Regiões Críticas</i>	57
3.8	Gabarito <i>Hit-or-Miss</i> para gerar esqueleto de largura um	60
3.9	Segmentação do Soma	61
3.10	Resultados do pré-processamento	62
3.11	Árvore de Decisão com regras de classificação	66
3.12	Contornando uma <i>bifurcação</i>	69
3.13	Contornando uma <i>sobreposição</i>	70
3.14	Resultados obtidos pelo <i>BTA</i> - 1	71
3.15	Resultados obtidos pelo <i>BTA</i> - 2	72
3.16	Resultados obtidos pelo <i>BTA</i> - Dendritos	72

3.17 Neurônio Alfa Rotulado	73
3.18 Neurônio Ypsilon Rotulado	74
3.19 Comparação entre resultados para imagem <i>alfa</i>	75
3.20 Comparação entre resultados para imagem <i>ypsililon</i>	76
3.21 Contorno de Neurônio Alfa	77
3.22 Contorno de Neurônio Ypsilon	78
3.23 Efeito aglutinante da Busca em Largura	79
4.1 Resultados para uma Imagem de Retina	85
B.1 Comparação entre segmentações: Área, Perímetro, Circularidade e 2º Momento . . .	94
B.2 Comparação entre segmentações: Entropia, Curvatura, ν mediana e global	95

Listas de Tabelas

2.1	Resultados para r e MSE	30
2.2	Resultados do teste-t: Manual vs Automático	33
2.3	Resultados do teste-t: PDR vs Não-PDR	35
2.4	Tabela de Contingência	40
2.5	Capacidade de Classificação - Individualmente	41
2.6	Capacidade de Classificação - Subconjuntos	45
3.1	Resumo de Conceitos.	58
A.1	Espaço de Características - Segmentação Manual	88
A.2	Espaço de Características - Segmentação Automática	89
A.3	Espaço de Características - PDR	90
A.4	Espaço de Características - Não-PDR	91

Listas de Algoritmos

1	Estimação da Integral de Correlação	97
2	Pré-processamento	99
3	LabelTree - Rotulador de Árvores Dendríticas	101
4	LabelBranch - Rotulador de Ramos	102
5	LabelBranchQueueSec - Rotula administrando fila secundária	102
6	IsValid - Verifica se o pixel é válido	103
7	LabelBranchMyStack - rotula administrando pilha	104
8	LabelBranchClassAndCont - Classificação de Região Crítica e Avaliação de Continuidade	105
9	LabelBranchClassAndContA - parte A	105
10	LabelBranchClassAndContB - parte B	106
11	LabelBranchClassAndContC - parte C	107
12	ComputeVectorsAndinnerProducts - Calcula Produtos Internos	108
13	MyContFol - Extrator de Contorno BSCEA	110
14	MyFindNext - Encontra o próximo pixel	111

Introdução

1.1 Preâmbulo

A ubiquidade de estruturas de ramificação na natureza é notória [102]. Desde agregados moleculares em Química e bacias hidrográficas em geologia [28, 97], a algas unicelulares, plantas, pulmões [68], neurônios [12] ou redes vasculares [49] no campo biológico, ou ainda descargas elétricas atmosféricas em Física [71], os padrões de ramificação estão presentes em todo lugar e em todas as escalas. As Figs. 1.1 e 1.2 ilustram diversos exemplos deste fato. Em particular, a Fig.1.1(b)¹ apresenta formações geológicas de redes em vales em padrões ramificados na superfície do planeta Marte, sugerindo a possibilidade da existência de fluxo de água em épocas remotas. Descargas elétricas atmosféricas também apresentam padrões ramificados, como o registrado no lago Ontario, Canadá pela Fig.1.1(c)². Em química, os padrões ramificados surgem nas estruturas crescidas em processos de agregação por difusão limitada (*Diffusion-Limited Aggregation - DLA*), mediante elektrodeposição em soluções aquosas, como o padrão ilustrado na Fig.1.2(c)³ obtido a partir de uma solução de sulfato de cobre. Padrões ramificados também são encontrados nas redes das vias aéreas e vasos sanguíneos presentes nos pulmões, como ilustrado na Fig.1.2(d)⁴. Embora tais sistemas pareçam pertencer a diferentes mundos, uma inspeção mais detalhada revela que estes conceitos demonstram-se surpreendentemente similares.

A similaridade de forma entre sistemas ramificados, tanto os biológicos quanto os não-biológicos, como os exemplificados nas Fig. 1.1 e Fig. 1.2, têm se revelado em comparações efetuadas entre sistemas de ramificação em folhas e em rios, demonstrando uma similaridade estatística [101]. Aparentemente, redes de rios tendem a seguir padrões estatísticos [76]. Em padrões de ramificação de sistemas geológicos e geofísicos, têm sido encontradas similaridades também [102]. Ainda, comparações têm sido feitas entre os sistemas de ramificação de corais do Caribe, folhas e rios [89].

¹<http://www.solarviews.com/> - Copyright *Calvin J. Hamilton*

²<http://streetnine.com> - Copyright *Joseph O. Holmes*

³Cortesia *Kevin R. Johnson* - Licença *GFDL*

⁴Cortesia *Prof. Ewald R. Weibel* - *Institute of Anatomy, University of Berne*.



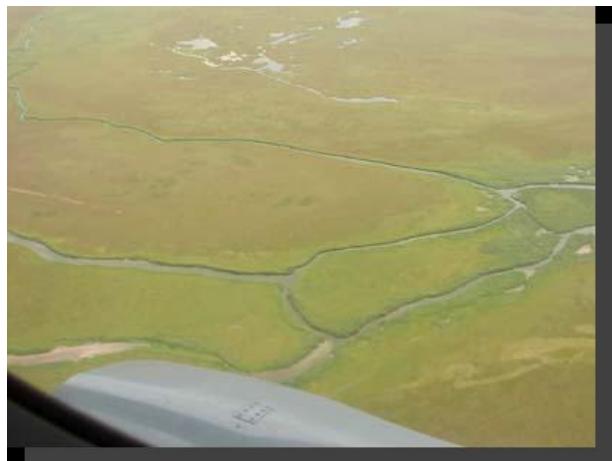
(a)



(b)

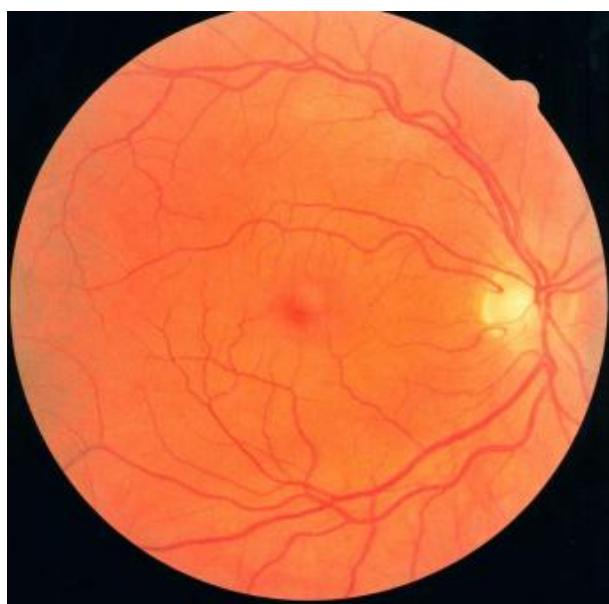


(c)

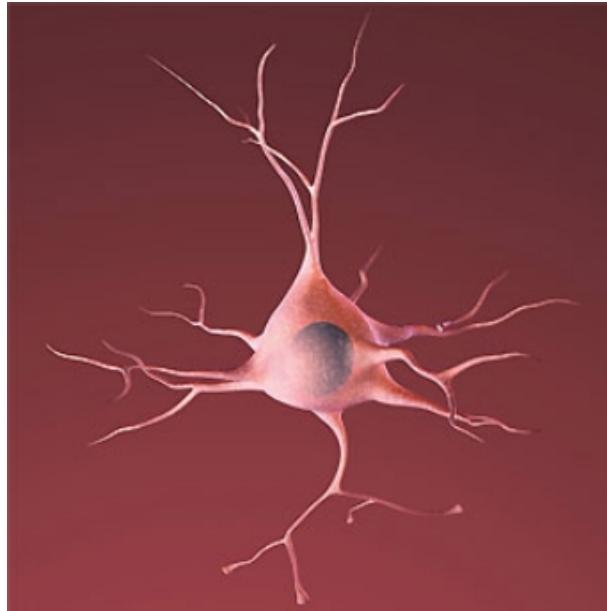


(d)

Figura 1.1: Exemplos da ubiqüidade de estruturas de ramificação na natureza: uma árvore (a), a presença de redes em vales denotam a possibilidade de fluxo de água na superfície do planeta Marte em tempos remotos - Cortesia *Calvin J. Hamilton* (b), as descargas elétricas atmosféricas (c), uma bacia hidrográfica (d)



(a)



(b)



(c)



(d)

Figura 1.2: Mais exemplos da ubiqüidade de estruturas de ramificação na natureza: a retina humana (a), representação 3D de um neurônio saudável - Cortesia *Alzheimer's Disease Education and Referral (ADEAR) Center, um serviço do National Institute on Aging*. (b), agregação por difusão limitada (DLA) obtida por eletrodeposição de solução de sulfato de cobre - Cortesia *Kevin R. Johnson* (c), Modelo de vias aéreas e vasos sangüíneos do Institute of Anatomy, University of Berne, cortesia *Prof. Ewald R. Weibel* (d).

Nota-se, assim, a existência de paradigmas universais [35].

O córtex visual dispõe de habilidades cognitivas que o habilitam a identificar características peculiares nas mais variadas formas, em particular nestas estruturas de ramificação. A partir daquelas características, torna-se possível avaliar a "complexidade" da forma em questão, classificando-a finalmente. Sempre que se faz necessária a automação da análise de informações visuais, considerando fatores envolvidos na atividade, tais como risco, repetitibilidade e precisão [25], a *Visão Computacional* é solicitada.

Grosso modo, a Visão Computacional estuda métodos que viabilizem um sistema artificial de visão, que seja capaz de desempenhar funções usualmente atribuídas ao sentido da visão, com competência similar ou superior, tais como a detecção e a identificação de objetos. Surge então a questão quanto à possibilidade de encontrar algoritmos capazes de reproduzir essa habilidade, para massiva investigação de algumas estruturas específicas.

Em particular, dois problemas em análise de formas de estruturas de ramificação, aparentemente distintos, encontram-se quando investigados à luz da **Análise de Formas**, uma sub-área da Visão Computacional: o intrigante problema de classificar objetivamente uma célula neuronal quanto a sua forma, para estabelecer relações entre sua função e forma, e o problema de classificar estruturas vasculares obtidas de imagens de retina, a fim de automatizar longos e fastidiosos processos de análise para estas, até então realizados somente por especialistas em neurociência e oftalmologia, mediante o uso de critérios subjetivos.

No que tange à solução destes problemas, o primeiro desafio consiste em descobrir descritores matemáticos, ou combinações destes, capazes de capturar quantitativamente, ou ainda modelar matematicamente, informações subjetivas contidas em imagens de estruturas ramificadas. O próximo desafio inclui o desenvolvimento de algoritmos eficientes para estes modelos, quando não houver, e sua implementação, para finalmente, com os resultados fornecidos por descritores adequados, distinguir entre categorias de estruturas por meio da teoria de reconhecimento de padrões [29].

Esta dissertação descreve o projeto de mestrado em que foram estudados e desenvolvidos métodos com vistas à solução dos dois desafios supramencionados. Na primeira parte deste trabalho abordamos o primeiro desafio, propondo uma combinação - conjunto de descritores - capaz de capturar a complexidade em imagens com padrões ramificados do sistema vascular da retina humana, culminando com a discriminação entre estruturas ramificadas com e sem neovascularização, um indicativo da presença de patologias. Esta parte foi desenvolvida, durante trabalhos em colaboração com H. F. Jelinek ⁵, M. J. Cree ⁶, A. Luckie ⁷ e J. V. B. Soares ⁸ [55, 56]. Na segunda parte deste trabalho abordamos o segundo desafio, desenvolvendo um algoritmo para extração do contorno paramétrico em estruturas de ramificação de células neurais, para as quais o algoritmo tradicional de extração de contornos é impreciso. [60]. Esta parte foi desenvolvida, durante trabalhos em colaboração com

⁵ School of Community Health, Charles Sturt University, Wilson Street, Albury 2640, Australia

⁶ Department of Engineering, University of Waikato, Hillcrest Road, Hamilton 3240, New Zealand

⁷ Albury Eye Clinic, Swift Street, Albury 2640, Australia

⁸ Departamento de Ciéncia da Computação, Instituto de Matemática e Estatística, Universidade de São Paulo

L. da F. Costa ⁹ [60, 61, 62, 63].

1.2 Objetivos

Os principais objetivos deste texto são:

- descrever os métodos estudados e desenvolvidos para análise de estruturas de ramificação de sistemas vasculares em imagens de retinopatia;
- descrever os algoritmos desenvolvidos para:
 - rastreamento de ramos em células neuronais,
 - extração do contorno paramétrico de células neuronais.

1.3 Organização do Texto

O presente texto é organizado como descrito a seguir. O Capítulo 2 apresenta o problema de análise de formas de estruturas de ramificação de sistemas vasculares em imagens de retina. As estruturas de ramificação são esqueletos obtidos a partir de dois processos de segmentação aplicados sobre as imagens originais, a saber: segmentação manual por um especialista e segmentação automática usando *Wavelets* e Classificadores Estatísticos. Neste capítulo é apresentado e discutido o conjunto proposto de sete descritores para caracterizar a complexidade daqueles esqueletos, visando a:

- avaliação do segmentador automático, comparando objetivamente segmentações automáticas e manuais;
- avaliação do poder diagnóstico destas características, buscando a discriminação entre sistemas vasculares com e sem retinopatia diabética progressiva.

Finalmente será apresentada uma avaliação quantitativa sobre a adequação deste conjunto de características como descritores de complexidade para estruturas de ramificação de esqueletos de sistemas vasculares de retina humana.

O Capítulo 3 apresenta o problema de análise de formas em estruturas de ramificação de células neuronais representadas em imagens bidimensionais. Uma vez introduzida a motivação para este estudo, será convencionada uma terminologia usada por todo o texto, e far-se-á uma descrição panorâmica e conceitual dos métodos. Em seguida, serão descritos os algoritmos desenvolvidos para o pré-processamento, o rastreamento de ramos e a extração do contorno paramétrico dos neurônios.

O Capítulo 4 encerra com as conclusões deste trabalho e perspectivas de trabalhos futuros.

Apêndices com tabelas, pseudo-códigos, figuras e publicações seguem anexos a este documento.

⁹Departamento de Física e Informática - Instituto de Física de São Carlos - USP

Capítulo 2

Análise de Formas de Estruturas de Ramificação - Parte I - Vasos Sangüíneos

A análise do fundo do olho para diagnóstico de patologias vasculares e não-vasculares tem sido amplamente utilizada pela comunidade médica. Exames por inspeção da árvore vascular da retina podem indicar hipertensão, diabetes, arteriosclerose, doenças cardíacas e derrames. A retinopatia diabética é uma das maiores causas da perda de visão em adultos, devido a alterações na estrutura e distribuição de vasos sangüíneos, tais como o crescimento de novos vasos (retinopatia diabética proliferativa), exigindo trabalhosas análises de especialistas.

O trabalho para diminuir os efeitos da retinopatia diabética inclui a aquisição e análise de imagens de fundo de olho em intervalos regulares (6 meses a um ano). O diagnóstico em tempo das mudanças de padrões em vasos sangüíneos em fundo de olho propiciam intervenções oportunas que podem impedir grandes perdas de visão [56].

Para antecipar a detecção de processos iniciais de mudanças nos padrões vasculares em fundo de olho em larga escala, a análise de imagens assistida por computadores tem sido introduzida. O primeiro desafio consiste em segmentar as árvores vasculares nas imagens de fundo de olho e obter destas seu esqueleto, o qual preserva a informação da complexidade da forma em questão. Dentre as diversas abordagens propostas para resolver este problema, ressaltamos o sistema de **Segmentação de Vasos Sangüíneos em Imagens de Retina usando Wavelets e Classificadores Estatísticos**, tema da dissertação de mestrado apresentada em 2006 por *João Vitor Baldini Soares* [92]. Destes esqueletos extraem-se características visando à classificação automática daquelas estruturas.

2.1 Caracterização de Formas

Queremos classificar entidades biológicas ramificadas, especialmente redes vasculares e neurônios, conforme informações morfológicas intrínsecas àquelas. O problema em questão poderia ser formulado da seguinte maneira: dada uma amostra desconhecida de uma destas entidades, doravante

chamada *forma*, atribua o rótulo da classe mais provável à amostra. Para tanto, usamos uma das abordagens da *Teoria de Reconhecimento de Padrões*, isto é, medir uma série de *características* (*features*) ou *atributos* da forma, tais como *área*, *perímetro*, *número de cantos*, *medidas baseadas em curvatura*, *orientação preferencial de contornos*, etc e submeter o *espaço de características* a classificadores automáticos [29]. Fogem ao escopo deste estudo os meandros da aquisição e detecção das formas. Ressaltamos, apenas que as mesmas são representadas por imagens binárias.

2.2 Características

Descrever uma forma por um vetor de características calculadas a partir daquela pressupõe a suficiência deste conjunto de características para representar a informação constante na mesma [25], o que nem sempre é verdade, daí a importância das pesquisas.

Em particular, devido à onipresença das estruturas de ramificação na natureza, pesquisas têm sido conduzidas nos diversos domínios do conhecimento objetivando a descrição e caracterização destas formas. A análise de estruturas de redes de rios, por exemplo, conta com o conceito de ordem de canal introduzido por Horton e Strahler, além de outras grandezas consideradas leis estruturais como a *Lei de Tokunaga* e a *Lei de Horton* [28]. A morfologia de sistemas vasculares de retina, por sua vez, tem sido analisada por características tais como comprimento, ângulos, ordem de ramificação, assimetria de árvores, energia de dobramento, dimensão fractal, diagramas de Sholl e área de influência, entre outros [56]. Células neuronais têm sido analisadas basicamente por três famílias de descritores morfológicos, a saber: geometria diferencial (e.g.: comprimento de arco, curvatura, orientações no campo tangente, contorno e pontos de bifurcação), eixos de simetria (dendrogramas, número de ramos, número de hierarquias, ângulos e comprimentos de processos) e complexidade (dimensões fractais multiescala, área de interface entre membrana e meio externo, superfície de contato e área de influência) [26].

No entanto, face à inexistência de meios *a priori* para identificar características adequadas ao estudo de variações nos padrões do sistema vascular de retina causados por patologias, permanece o desafio de prosseguir com pesquisas tanto na busca por novas características com grande poder de discriminação e diagnóstico preciso [56], quanto na busca por um conjunto de características conhecidas, em que as formas de interesse sejam bem discriminadas, em outras palavras, um conjunto ótimo de características efetivas que preserve a separabilidade entre classes, com respeito a um classificador *Bayesiano*, o qual garante o erro mínimo para as distribuições dadas [39]. A mesma consideração se aplica à caracterização de células neuronais, pois há inúmeros descritores morfológicos possíveis, mas não há regras bem definidas ou procedimentos para selecioná-los. Por isso, a seleção de características praticamente se baseia nas particularidades do problema, no conhecimento prévio deste, na experiência em neuromorfologia, além do conhecimento sobre a potencialidade de certas características de expressar satisfatoriamente as propriedades das formas de interesse [26].

2.3 Análise de Formas aplicada à Validação de Segmentador Automático e Identificação de *PDR* em Sistemas Vasculares de Retina

A seguinte análise foi publicada em trabalho de colaboração com Jelinek *et al* [56], em que minha participação contribui com a descrição das características, sua interpretação física, cálculos e implementações e análise de complexidade de algoritmos. Neste trabalho, 27 imagens digitais de vasos sanguíneos da retina contrastados por fluorescência foram segmentadas manualmente por um especialista e automaticamente por transformadas *Wavelet* e classificadores estatísticos [93, 94, 92]. Destas 27 imagens, 16 continham retinopatia diabética proliferativa (Proliferative Diabetic Retinopathy - PDR) e 11 não continham *PDR*, mas retinopatia de fundo e outras patologias. O conjunto de 54 segmentações obtidas pelos dois métodos foi analisado pelo conjunto de características proposto para discriminar as estruturas de ramificação de interesse na primeira parte deste trabalho. Estas características são descritas a seguir, tendo sido selecionadas com base nos mesmos critérios mencionados anteriormente [26, 51], aplicados na seleção das características usadas na análise morfométrica de células ganglionares de gato em trabalho anterior [53], isto é, *Área*, *Perímetro*, *Circularidade* [25], *Curvatura* [19], *Entropia de Orientação da Distribuição de Fases da CWT* [64], *2º Momento Estatístico da Distribuição de Módulos da Transformada Wavelet* [3, 4] e *Dimensão de Correlação* [51].

2.3.1 Área

Para estimar a área A de uma forma representada numa imagem binária, basta varrer toda a imagem contando o número de pixels objeto nela presentes, o que consome tempo linear no tamanho N da imagem, isto é, $\Theta(N)$, onde $N = L \times C$, sendo L e C o número de linhas e de colunas da imagem, respectivamente. Apesar desta abordagem simples para estimar a área, esta característica é um indicador imediato da magnitude de ocupação da forma, porém indireta de complexidade. Considere o conjunto de imagens binárias distintas de N pixels, dos quais K são pixels objeto e $N - K$ são pixels de fundo. A cardinalidade deste conjunto é $\binom{N}{K}$, pois este número equivale ao número de k -combinações de um conjunto de N elementos. Portanto, existem $\binom{N}{K}$ imagens distintas com N pixels, e K pixels objeto, para as quais a área é exatamente a mesma: $K!$ De fato, da Teoria de Contagem [22] vale a seguinte identidade ¹:

$$\sum_{K=0}^N \binom{N}{K} = 2^N \quad (2.1)$$

que é precisamente o número de imagens binárias distintas possíveis com N pixels. Note o caráter *degenerado* da Área como característica, cujo resultado é o mesmo para $\binom{N}{K}$ formas completamente diferentes.

¹Um caso particular da expansão binomial em que $x = y = 1$

2.3.2 Perímetro

A mera contagem do número de pixels obtidos por um detector de bordas aplicado a uma forma representada numa imagem binária é a mais simples abordagem para o problema de estimar o comprimento de arco de uma curva amostrada, ou contorno. Os detectores de bordas, isto é, operadores diferenciais tais como *Sobel* e *Prewitt*, ou mesmo os *Gradientes Morfológicos*, provêem o contorno representado como imagem. Neste caso, o cálculo do perímetro consome tempo $\Theta(N)$, onde $N = L \times C$, sendo L e C o número de linhas e de colunas da imagem, respectivamente. Os algoritmos tradicionais de extração de contorno baseados no código da cadeia [25] fornecem o contorno representado como um sinal paramétrico, que pode ser armazenado em lista ou vetor de tamanho M . Neste caso, o cálculo do perímetro consome tempo $\Theta(M)$.

2.3.3 Circularidade

A circularidade é um descritor facilmente estimado, e tem a vantagem de ser adimensional. Pode ser interpretado como o grau de dissimilaridade entre a forma dada e uma circunferência, sendo definida por:

$$C = \frac{P^2}{A}$$

Assim como a Área, e o Perímetro, a Circularidade apresenta a propriedade de ser invariante por rotações, sendo calculada em tempo $\Theta(N)$, onde $N = L \times C$, sendo L e C o número de linhas e de colunas da imagem, respectivamente, pois envolve o cálculo da área.

2.3.4 Características derivadas de Transformadas Wavelet

No que se segue, apresentamos uma breve revisão sobre a teoria de transformadas Wavelet, a partir das quais obtemos outras características. Basicamente, as chamadas *Wavelets* são famílias de funções especiais que viabilizam transformações do tipo *tempo-freqüência*, localizando transientes no tempo e na freqüência com boa resolução. Versões transladadas e dilatadas da função Wavelet original podem ser interpretadas como bancos de filtros para análise de sinais [17], ou como funções de base de um espaço vetorial em que o sinal de interesse será projetado [2]. Tendo sido originalmente projetadas para analisar adequadamente sinais não-estacionários e inhomogêneos, a análise tempo-escala surgiu para resolver situações difíceis para a Análise de *Fourier*, baseada na *transformada wavelet contínua*, ou *continuous wavelet transform (CWT)*.

A CWT é uma ferramenta poderosa e versátil que tem sido aplicada em muitos problemas diferentes em processamento de imagens, desde codificação de imagens [85] até análise de formas [25]. Este sucesso deve-se grandemente ao fato de que as wavelets são especialmente adequadas para detectar singularidades (e.g. bordas) em sinais [46], extraíndo freqüências instantâneas [3], e executando análise fractal e multifractal. Em Análise de Formas, as transformadas Wavelet encontram diversas aplicações, em especial as duas aqui citadas: segmentação de sistemas vasculares de imagens de retinopatia diabética [64, 65, 93, 94, 92] e caracterização de neurônios [52, 53].

Além disso, o uso da transformada wavelet com a wavelet Morlet, também chamada de *Gabor wavelet*, tem desempenhado um papel fundamental no aumento da nossa compreensão do processamento visual em diferentes contextos desde detecção de características até rastreamento de face. Esta wavelet também tem se revelado como ferramenta útil para a segmentação de imagens de fundo de olho usando um esquema baseado em limiarização de uma escala [51, 64]. É possível ainda melhorar os resultados integrando a informação de diversas escalas e outros parâmetros da wavelet [65].

A notação e as definições nesta seção seguem [4]. O plano real $\mathbb{R} \times \mathbb{R}$ é denotado como \mathbb{R}^2 e os vetores são representados em negrito, e.g. $\mathbf{x}, \mathbf{b} \in \mathbb{R}^2$. Seja $f \in L^2$ uma imagem representada como uma função quadraticamente integrável (i.e. energia finita) definida sobre \mathbb{R}^2 . A transformada contínua wavelet $T_\psi(\mathbf{b}, \theta, a)(\mathbf{x})$ é definida como:

$$T_\psi(\mathbf{b}, \theta, a)(\mathbf{x}) = C_\psi^{-1/2} \frac{1}{a} \int \psi^*(a^{-1}r_{-\theta}(\mathbf{x} - \mathbf{b})) f(\mathbf{x}) d^2\mathbf{x} \quad (2.2)$$

onde C_ψ , ψ , \mathbf{b} , θ e a denotam a constante de normalização, wavelet mãe, o vetor deslocamento e os parâmetros de rotação angular e dilatação, respectivamente. ψ^* denota o conjugado complexo. A wavelet mãe, ou analisadora, deve satisfazer essencialmente duas condições, a saber:

- I. $\psi \in L^1 \cap L^2$, isto é, ψ deve ser bem localizada no domínio do tempo e da freqüência (suporte compacto);
- II. $\int_{-\infty}^{+\infty} dt \psi(t) = 0$, ou seja, ψ deve ser admissível, o que é equivalente à condição de ter média zero.

Assim como a Transformada Janelada de Fourier (*Short Time Fourier Transform - STFT*), a transformada wavelet também está sujeita ao princípio da incerteza de Heisenberg, que estabelece um limite inferior para a resolução nos domínios do tempo e da freqüência, isto é, $\Delta t \Delta f \geq (4\pi)^{-1}$. Isto significa que o produto da largura de banda no tempo e na freqüência não pode ser arbitrariamente pequeno. Por outro lado, ao contrário da *STFT*, as wavelets constituem-se em filtros com fator de qualidade $Q = \frac{\Delta f}{f}$ constante, ou seja, a resolução no tempo aumenta com a freqüência central das wavelets [85], de tal maneira que a proporção entre f e Δf mantém-se constante. Isto evidencia o caráter de filtro local das wavelets com fator de qualidade constante, indicando sua eficiência com o aumento da freqüência, ou a redução da escala. Tal propriedade torna a wavelet efetiva para detecção e análise de propriedades locais e singularidades [3].

Não obstante o enorme leque de aplicações das transformadas wavelet em Visão Computacional, dentre elas a segmentação de vasos sanguíneos [56], o presente texto foca a aplicação da transformada wavelet para caracterização de formas de estruturas de ramificação. Neste trabalho, um sinal de interesse é uma imagem 2D, podendo ser interpretada como um campo escalar 2D. Queremos detectar singularidades nas imagens, ou seja, seus pixels de contorno, onde ocorrem variações abruptas nos valores dos pixels. Considerando o exposto, usamos, neste trabalho, a 1ª das wavelets da

família de *Wavelets Hermitianas* (a família de derivadas da Gaussiana) [66], com a qual obtemos o campo vetorial gradiente a partir da imagem, análogo 2D da 1^a derivada da função gaussiana. O gradiente aponta na direção de máxima variação do respectivo campo escalar [59]. Vale ressaltar que a função gaussiana $g(x, y)$ propriamente não é uma wavelet, por não satisfazer a condição de admissibilidade [2] e também porque $g(x, y) \notin L^1 \cap L^2$, mas todas as suas derivadas o são.

A 1^a wavelet hermitiana é definida como:

$$\psi_g(x, y) = \nabla g(x, y) = \hat{i} \frac{\partial g(x, y)}{\partial x} + \hat{j} \frac{\partial g(x, y)}{\partial y} \quad (2.3)$$

Como a gaussiana pode ser usada como um filtro de suavização, a equação 2.3 define a transformada wavelet 2D como o gradiente da imagem, suavizada por versões dilatadas da gaussiana, ou ainda como a convolução entre o gradiente da gaussiana 2D e a imagem [4]. Assim, torna-se possível a implementação dessas transformações com um algoritmo que consome tempo $\Theta(N \log_2 N)$, onde N é o tamanho da imagem. Basta efetuar tais operações no domínio de Fourier, onde vale a *propriedade da derivada* [25], e também o *Teorema da Convolução*, segundo o qual um produto de convolução no domínio do tempo equivale a um produto interno no domínio da freqüência.

Usando a propriedade da derivada da transformada de *Fourier*, obtemos:

$$\frac{\partial g(x, y)}{\partial x} = (j2\pi f_x) G(f_x) \quad (2.4)$$

$$\frac{\partial g(x, y)}{\partial y} = (j2\pi f_y) G(f_y) \quad (2.5)$$

onde a gaussiana é dada por:

$$g_\sigma(x, y) = \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right)$$

Considerando que a Wavelet aplicada é o Gradiente da imagem borrada, esta transformada foi implementada mediante o uso de dois operadores diferenciais dx e dy , aproveitando o recurso de manipulação de matrizes em *Matlab®* e a propriedade da derivada da Transformada de Fourier, calculando assim a *CWT* como um produto interno no domínio da freqüência.

2.3.4.1 Curvatura

A natureza geométrica local de uma curva, ou a velocidade com que esta muda de direção é uma característica muito poderosa. As derivadas utilizadas no cálculo da curvatura podem ser obtidas derivando-se a própria imagem do contorno da forma (ex: curvatura de esqueletos de vasos de imagens de retinopatia pelas equações (2.4) e (2.5) - Seção 2.3.4.) ou derivando os *Sinais Paramétricos do Contorno* da forma (ex: curvatura dos neurônios - Capítulo 3). Se o contorno da forma for interpretado como a descrição da trajetória de uma partícula puntual no tempo,

$\vec{P}(t)$, então a aceleração da partícula é dada por $\ddot{\vec{P}}(t) = (\ddot{\vec{P}}_t(t), \ddot{\vec{P}}_r(t))$, onde $\ddot{\vec{P}}_t(t)$ é a componente denominada aceleração tangencial, e $\ddot{\vec{P}}_r(t)$ é a componente denominada aceleração radial, que indica a variação de direção. Se o contorno for parametrizado por comprimento de arco, então $\dot{\vec{P}}(t) = 1$, logo $\ddot{\vec{P}}_t(t) = 0$, e podemos definir a curvatura $|k(t)| = \|\ddot{\vec{P}}_r\|$.

Considerando a representação paramétrica do contorno de uma forma dado por $\vec{p}(t) = (\vec{x}(t), \vec{y}(t))$, a curvatura pode ser definida como [25] :

$$k(t) = \frac{\dot{x}(t)\ddot{y}(t) - \dot{y}(t)\ddot{x}(t)}{\left(x^2(t) + y^2(t)\right)^{\frac{3}{2}}} \quad (2.6)$$

Então, dadas as derivadas parciais, os cálculos da curvatura requerem tempo adicional $\Theta(M)$, desde que os sinais paramétricos sejam armazenados numa estrutura de dados de tamanho M , conforme discussão na Seção 2.3.2.

2.3.4.2 Entropia de Orientação

A Entropia de um sistema físico foi introduzida pelo físico *Ludwig Boltzman* no contexto da Mecânica Estatística, e mais tarde encontrou aplicações também na teoria da informação desenvolvida pelo cientista do *Bell Labs*, *Claude Elwood Shannon* [90] graduado em engenharia e matemática e conhecido como o pai da teoria da informação. Shannon demonstrou que seu conceito de entropia de informação era equivalente à deficiência no conteúdo de informação de uma mensagem. De acordo com a segunda lei da termodinâmica, a entropia é o grau de aleatoriedade em qualquer sistema e sempre é crescente. Conforme descrito na seção 2.3.4, a transformação resultante da aplicação da 1^a Wavelet Hermitiana sobre o campo escalar definido pela imagem nada mais é que um campo vetorial. A fase dos vetores deste campo indicam a direção da máxima variação do campo escalar, enquanto a magnitude deles representa a máxima taxa de variação [59]. Assim, podemos considerar características associadas às direções e às magnitudes dos vetores definidos pela transformada. Aqui, o conceito de entropia é aplicado para obter a entropia de orientação da distribuição angular do campo vetorial gradiente definido pela transformada da imagem pela 1^a Wavelet Hermitiana descrita em 2.3.4. Neste caso, a entropia de orientação indica o grau de desorganização direcional na imagem. Se a estrutura de ramificação privilegia alguma direção, a entropia de orientação o indicará. A entropia é definida como:

$$E = - \sum_{i \in K} p_i \ln(p_i) \quad (2.7)$$

em que p_i é a freqüência de ocorrência de algum vetor com orientação na direção θ_i e K é o conjunto de *bins* do histograma. Este cálculo leva tempo proporcional ao número de *bins* do histograma, isto é, $\Theta(|K|)$.

2.3.4.3 2º Momento Estatístico do Módulo da Transformada Contínua Wavelet

Medidas estatísticas de dispersão de dados são freqüentemente utilizadas para avaliar a existência de tendências no espaço amostral. Assim como a entropia identifica tendências direcionais capturadas pela transformada wavelet, o 2º Momento Estatístico avalia a dispersão da distribuição de magnitudes destas variações direcionais, dadas pelos coeficientes do Módulo da **CWT**, interpretados neste contexto como as intensidades destas variações direcionais.

O momento central de ordem q da distribuição de probabilidades de uma variável aleatória pode ser definido matematicamente como $m_q = E((X - \mu)^q)$. Se h_i é o i -ésimo *bin* de um histograma representando a freqüência relativa de valores de uma determinada grandeza, com cada *bin* centrado em i , então o momento estatístico de ordem q do histograma é definido por [25, 53]:

$$m_q^M = \sum_i i^q h_i \quad (2.8)$$

2.3.5 Dimensão de Correlação - ν

2.3.5.1 Um panorama sobre conceitos de dimensão

O conceito de dimensão mais familiar ao senso comum é o denominado *Dimensão Euclidiana* (d) no espaço euclidiano \mathbb{R}^N , definido como o número de parâmetros independentes (coordenadas) necessários para especificar-se a posição de um dado objeto. Por exemplo, $d = 0$ para um ponto, $d = 1$ para uma curva, $d = 2$ para uma superfície e $d = 3$ para um volume. A *Dimensão Topológica* - d_T , por sua vez, é uma generalização da dimensão Euclidiana, definida, simplificadamente, como $d_T = 0$ para um ponto, e, para qualquer outro objeto, como uma unidade a mais do que a dimensão de um número finito de objetos necessários para subdividi-lo em partes. Assim, uma linha tem $d_T = 1$, pois é necessário um ponto, que tem dimensão zero, para dividi-la em duas partes; um plano tem $d_T = 2$, já que é necessária uma linha, que tem dimensão um, para dividi-lo em duas partes, e assim por diante [91]. Em Álgebra Linear, o número de elementos da base de um espaço vetorial V é chamado de dimensão de V [8], enquanto que em Física, diz-se que para os sistemas holonômicos, ou seja, aqueles em que os vínculos podem ser expressos como equações nas coordenadas, o número de graus de liberdade é igual ao número mínimo de coordenadas (dimensões) necessárias para especificar-se a configuração de um sistema [96]. A natureza inteira inerente a estes conceitos de dimensão os tornam inapropriados para a descrição de estruturas complexas, no que diz respeito à ocupação espacial, e que podem exibir auto-similaridade ou invariância através de escalas. Em seu trabalho seminal [70] publicado em 1960, Mandelbrot - um matemático da IBM - batizou estruturas com estas propriedades como *Fractais*², definindo outrossim a dimensão fractal d_F como um conjunto cuja dimensão de Hausdorff-Besicovitch d_H é estritamente maior que a dimensão topológica d_T . Esta definição é considerada insatisfatória, por excluir muitas estruturas que também poderiam ser consideradas fractais [30].

²Fractus do Latim, significando fração, quebrado

Na verdade, normalmente os cientistas reconhecem um fractal quando vêem um, mas não há uma definição universalmente aceita. Em geral, admite-se que fractais têm algumas ou todas das seguintes propriedades: estrutura complexa num amplo intervalo de escalas de comprimento, repetição de estruturas em diferentes escalas de comprimento (auto-similaridade), e uma dimensão fractal, portanto não inteira. [1]. A rigor, existem dois tipos de fractais, a saber: *fractais regulares (determinísticos)*, em que se verifica a auto-similaridade através de escalas *ad infinitum*, tais como os gerados por algoritmos em computadores, e os fractais encontrados na natureza, também chamados de *fractais aleatórios*, nos quais a auto-similaridade é apenas estatística, sendo melhor descrita como *invariância*, em vez de auto-similaridade, já que os fractais aleatórios apresentam invariância em um intervalo finito de escalas, limitado inferiormente pela resolução do método e superiormente pelo tamanho da estrutura [74].

Ademais, apesar da dimensão de Hausdorff-Besicovitch d_H admitir valores não-inteiros [47], podendo quantificar a fractalidade destes objetos, sua definição formal é deveras complicada e os cálculos que envolvem sua obtenção são demasiadamente complexos [25], o que torna sua aplicação inviável. Por esta razão, a dimensão de capacidade d_C , introduzida por Kolmogorov [58] mais tarde e considerada uma versão simplificada da d_H [32], popularizou-se de tal forma, que os termos d_F , d_C e d_H têm sido usados indistintamente como sinônimos para a dimensão fractal.

2.3.5.2 A análise fractal aplicada a imagens de retina

Entre 1984 e 1988, Kretzer propôs um modelo para o processo de formação de vasos na retina, onde a demanda por oxigênio é maior do que em quaisquer outros tecidos do corpo. Qualquer deficiência na circulação nesta região resulta em prejuízo funcional, danos ao tecido e o surgimento de patologias como retinopatia diabética e hipertensão vascular, podendo culminar em cegueira [31]. O modelo de Kretzer supõe que a diminuição do fluxo de oxigênio através da retina pelos vasos é compensada pela angiogênese no interior da retina, já que os fotoreceptores em amadurecimento requerem mais oxigênio, ocasionando um decréscimo na provisão de oxigênio para retina e causando uma estimulação para a liberação de fatores angiogênicos. Isto implicaria na difusão de fatores angiogênicos no plano da retina, que seria então o processo físico responsável pelo desenvolvimento da padrões de vasos na retina.

Em 1989, Family *et al.* testaram a hipótese de Kretzer, em seu trabalho pioneiro de análise geométrica de árvores vasculares na retina humana [31]. Para tanto, Family lançou mão de análise fractal, usando o método *mass-radius* e a função correlação de pares, cujos resultados exibiram auto-similaridade num intervalo de escalas com dimensão fractal $d_F \approx 1.7$. Curiosamente, esta é justamente a dimensão fractal encontrada na análise de imagens de *DLA* (processos crescidos mediante agregação por difusão limitada - Diffusion Limited Aggregation) e também em fractais de Laplace, levando Kretzer a concluir, por analogia, que a rede de vasos da retina é fractal e que havia obtido suporte quantitativo validando a hipótese de que a difusão limitada de fatores angiogênicos seria o processo fundamental na formação de vasos no plano da retina [74]. Desde então, a dimensão

fractal tem sido amplamente utilizada para análise de imagens de retina.

Não obstante, duas controvérsias basicamente foram levantadas com respeito à aplicação da dimensão fractal a estruturas biológicas. A primeira se refere a inferências que podem ser feitas sobre o mecanismo de formação de uma certa estrutura, a partir da determinação de sua dimensão fractal apenas [74], posto que "esta é uma conclusão injustificada, obtida a partir de uma mera similaridade visual [sic]" [78]. De fato, a semelhança visual entre as estruturas e o fato destas apresentarem o mesmo valor de dimensão fractal não implica necessariamente que ambas tenham sido originadas pelo mesmo processo. No entanto, esta ressalva não descarta o uso de fractais para analisar imagens de retina, por exemplo, como outros autores ressaltam. Assim como com outras características, dimensão fractal pode ser usada desde que apenas e tão somente como um descritor de complexidade da forma, sem a presunção de qualquer mecanismo de geração de forma ou função [54]. Desta forma, o objetivo seria apenas distinguir entre estruturas de ramificação diferentes, com vistas a sua posterior classificação.

A segunda controvérsia questiona a validade da aplicação da dimensão fractal a estruturas biológicas, já que a auto-similaridade através de escalas é uma propriedade de alguns fractais ideais, isto é, fractais determinísticos gerados por Leis Lineares de Escala [78], propriedade esta não verificada em estruturas encontradas na natureza, ou seja, fractais aleatórios. Em particular, *Panico* e *Sterling* mostraram em seu trabalho [80] que células neuronais ganglionares e vasos sanguíneos de retina não apresentavam maior característica de auto-similaridade do que os controles não-fractais por eles testados, concluindo que estas estruturas de ramificação não seriam fractais, mas sim curvas de preenchimento de espaço (*space-filling curves*). Estas conclusões de *Panico* e *Sterling* foram contestadas por outros autores, lembrando que as curvas de Peano, ou curvas de preenchimento de espaço, também são fractais, considerando sua auto-similaridade, muito embora apresentem dimensão 2, e que fractais também são definidos pela complexidade com que preenchem o espaço, sendo mais complexos tanto maior sua capacidade de preenchimento do espaço [54]. Muitos fractais apresentam graus de auto-similaridade, exibindo a semelhança entre partes e o todo, mas esta semelhança nem sempre é estrita sob o ponto de vista euclidiano, podendo ser aproximada ou estatística [30]. Além disso, nenhuma representação por meio de imagens digitais é rigorosamente fractal, já que imagens digitais possuem resolução finita - dados discretos - e, consequentemente, um limite inferior para observação de auto-similaridade. Ainda, a similaridade *ad infinitum* não é observada em estruturas na natureza, sendo limitada inferiormente pela escala atômica, e superiormente pelo tamanho do objeto [25].

2.3.5.3 A dimensão de correlação como alternativa à dimensão fractal

O conceito de fractalidade introduzido por Mandelbrot envolve a percepção de auto-similaridade (invariância) num dado objeto, isto é, a observação de que o objeto seja "fraturável" em partes menores similares ao todo. Esta relação recursiva de similaridade entre partes, ou subconjuntos, poderia ser considerada uma particularidade da idéia de que haveria um sistema dinâmico subjacente à

geometria do conjunto. Tal associação entre um objeto - ou conjunto - e um sistema dinâmico a ele subjacente estaria entre as razões que explicariam o alcance que o conceito e a linguagem dos fractais têm atingido em diversos ramos da ciência, haja vista o objetivo fundamental da ciência de modelar fenômenos naturais, mediante dinâmicas intrínsecas aos mesmos [104]. Conseqüentemente, assim como a Geometria Euclidiana tem servido como linguagem descritiva para o movimento na Mecânica Clássica, a Geometria Fractal tem sido usada para descrever padrões produzidos por Sistemas Dinâmicos Complexos Não-Lineares, em particular, Sistemas Caóticos, tais como as trajetórias do problema de dois corpos, por exemplo, que consistem em seções cônicas (elipses, parábolas e hipérboles) [1]. Um destes padrões, um tipo especial de trajetórias no espaço de fase, foi evidenciado em 1963 pelo meteorologista do MIT Edward Lorenz, quando publicou sua demonstração de que comportamentos complexos e aperiódicos podem resultar de sistemas físicos determinísticos com alguns graus de liberdade, e que sistemas dinâmicos dissipativos, os quais normalmente apresentam muitos graus de liberdade podem, depois de um tempo transiente inicial, estabelecer um estado em que somente poucos graus de liberdade são relevantes para descrever a dinâmica do sistema [67, 99]. Neste estado pós-transiente, a trajetória do sistema através do espaço de fase é confinada a subconjuntos de baixa-dimensão do espaço de fase disponível [99], representações da dinâmica de um sistema no espaço de fase denominadas Atratores. Quando estes assumem características que se repetem em muitas escalas de tempo ou espaço, ou seja, propriedades de fractais de auto-similaridade em várias escalas e dimensão não-inteira, eles são ditos *Atratores Estranhos*, e podem ser caracterizados pela dimensão fractal [43], posto que a d_T se faz inadequada para descrevê-los. Disto, entende-se que a *estranheza* em Sistemas Dinâmicos, corresponde à *fractalidade* em Geometria Fractal [43]. Ademais, atratores estranhos são representações da dinâmica de um sistema no espaço de fase, em que os movimentos são aperiódicos, complexos e tipicamente caóticos [99]. Com a descoberta de Lorenz, atualmente um sistema dinâmico complexo pode ser classificado, conforme a variação temporal da descrição de seus estados, em [40, 79]:

- **estacionário:** ponto fixo, solução constante;
- **periódico:** solução periódica;
- **quase-periódico:** solução por combinação linear de funções periódicas;
- **estocástico:** flutuações aleatórias;
- **caos determinístico:** determinado por um sistema de equações não-lineares, no entanto imprevisível e aperiódico.

Uma vez que a dimensão pode ser a propriedade mais fundamental de um atrator [32, 79] e já que atratores estranhos exibem natureza fractal [43, 1], como discutido anteriormente, diversos métodos da Geometria Fractal têm sido aplicados a fim de estimar a dimensão não-inteira destes conjuntos [99], bem como conceitos e métodos especificamente desenvolvidos para estimar a dimensão

de atratores, por meio do cálculo de distâncias entre pares de pontos [98]. Uma relação entre os diversos conceitos de dimensão não-inteiras foi estabelecida por Farmer em 1983, que classificou as dimensões conhecidas até aquele momento em duas categorias [32]:

I. Dimensões Métricas, definidas em termos de uma métrica apenas, ou conceito de distância:

- Dimensão de Capacidade $[d_C]$
- Dimensão de Hausdorff-Besicovitch $[d_H]$

II. Dimensões Probabilísticas d_μ , ou Dimensões de Medida Natural, definidas em termos de uma métrica e de probabilidades:

- Dimensão de Informação $[\sigma]$
- Dimensão δ -Capacidade $[d_C(\delta)]$
- Dimensão δ -Hausdorff $[d_H(\delta)]$
- Dimensão Puntual $[d_P]$

Farmer chegou a esta classificação, após conjecturar e constatar que [32]:

$$d_C = d_H \equiv d_F \quad (2.9)$$

onde d_F é a dimensão fractal, e que:

$$\sigma = d_C(\delta) = d_H(\delta) = d_P \equiv d_\mu \quad (2.10)$$

onde d_μ é dimensão de medida natural.

De fato, a identidade 2.9 explica a razão pela qual os diversos métodos para estimar a d_C , tais como *box-counting*, *mass-radius* e *Minkowsky-sausage*, têm sido usados com sucesso na prática, como alternativas à complicada e inviável d_H . No entanto, estes métodos basicamente contam o número de hipercubos não vazios que cobrem a região do espaço em que o fractal está imerso, consequentemente quanto maior a dimensão F do espaço de imersão, mais hipercubos serão necessárias para cobri-lo, e portanto mais tempo necessário para contá-los [38, 45, 43]; em particular, já foi mostrado que o *box-counting* é inviável para cálculo da dimensão de atratores [45]. Some-se a isto o fato de que a d_F não leva em conta a densidade dos pontos da estrutura fractal, ou seja, a distribuição possivelmente não uniforme dos pontos ao longo da estrutura [44], por isso as dimensões probabilísticas trazem mais informação com respeito aos atratores [32]. O problema de identificar atratores resultantes de soluções periódicas pode ser facilmente resolvido mediante a análise de Fourier. No entanto também faz-se necessária a distinção entre fenômenos com movimento aparentemente errático, os quais podem ser devidos a um movimento determinístico (*Caos*) ou a um movimento estocástico (*ruido*), além da caracterização e distinção entre atratores estranhos diferentes [44]. Em face às

desvantagens supracitadas com respeito à d_C , e aos problemas de identificação e caracterização de atratores, como alternativa às medidas métricas, *Grassberger* e *Proccacia* propuseram a *Dimensão de Correlação* em trabalho também publicado em 1983, a qual popularizou-se como a ferramenta mais utilizada para detectar caos em dados experimentais, desde a publicação de seu algoritmo **GPA** [43, 44]. A Dimensão de Correlação ν é essencialmente uma medida de densidade do atrator dentro do espaço de fases de dimensão F (graus de liberdade) em que está imerso. Em sistemas estocásticos, a ν diverge com F e em sistemas caóticos determinísticos, ν converge para um valor, tornando-se independente de F [44]. Além disso, *Grassberger* e *Proccacia* também demonstraram que:

$$\nu \leq \sigma \leq d_F \leq F \quad (2.11)$$

onde ν é a dimensão de correlação, σ é a dimensão de informação [32], d_F é a dimensão fractal e F é a dimensão do espaço de imersão.

O conjunto de desigualdades 2.11 estabelece um limite justo para a maioria dos casos, além de um limiar inferior para dimensão fractal, com a vantagem de a dimensão de correlação ν ser muito mais fácil de calcular do que σ ou d_F [44].

A idéia é calcular uma Integral de Correlação $C(\epsilon)$, que pode ser interpretada como a probabilidade de dois pontos arbitrários da trajetória de um atrator estranho serem encontrados a uma distância, no máximo, igual a um dado ϵ . Por esta razão, a dimensão de correlação também é uma dimensão probabilística.

No caso particular de Análise de Formas de Estruturas de Ramificação, a dimensão de correlação ν pode ser aplicada, se considerarmos a estrutura como sendo a reconstrução de um atrator estranho no espaço de fase, com um sistema dinâmico subjacente caracterizado por caos determinístico, portanto um sistema dinâmico complexo, e que deve ter uma série temporal associada a ele [99], do tipo:

$$X_i(t) = X(t + i\tau)_{i=1}^N \quad (2.12)$$

Então, calculam-se as separações (distâncias Euclidianas) entre cada par dos N pontos de dados do atrator, a fim de ordená-las posteriormente em *bins* de largura $\Delta\epsilon$ proporcional a ϵ [95]. Em outras palavras, estima-se a correlação espacial existente entre os pontos deste atrator por meio da função Integral de Correlação $C(\epsilon)$, a qual pode ser definida conceitualmente como [44]:

$$C(\epsilon) = \lim_{N \rightarrow \infty} \frac{2}{N^2} \times \{ \text{número de pares } (i, j) \text{ cuja distância } |X_i - X_j|, \text{ é menor do que } \epsilon \} \quad (2.13)$$

Formalmente, a Integral de Correlação $C(\epsilon)$ é definida como [43, 44]:

$$C(\epsilon) = \lim_{N \rightarrow \infty} \frac{2}{N^2} \sum_{i=1}^N \sum_{j=i+1}^N \Theta(\epsilon - |X_i - X_j|) \quad (2.14)$$

onde a função *Heaviside* $\Theta(x)$ se escreve:

$$\Theta(\epsilon - |X_i - X_j|) = \begin{cases} 1 & \text{se } 0 \leq (\epsilon - |X_i - X_j|) \\ 0 & \text{se } 0 > (\epsilon - |X_i - X_j|) \end{cases} \quad (2.15)$$

A Integral de Correlação $C(\epsilon)$ também pode ser definida em termos da função de correlação usual [44]:

$$C(\epsilon) = \int_0^\epsilon c(r) d^F r \quad (2.16)$$

onde a integral de correlação usual $c(r)$ se define em termos da função delta de *Dirac*:

$$c(r) = \lim_{N \rightarrow \infty} \frac{2}{N^2} \sum_{\substack{i,j=1 \\ i \neq j}}^N \delta^F(r - (|(X_i - X_j)|)) \quad (2.17)$$

Verifica-se que a Integral de Correlação (2.14) obedece uma lei de potências do tipo:

$$C(\epsilon) \propto \epsilon^\nu \quad (2.18)$$

onde o expoente ν é a *Dimensão de Correlação*, que pode ser obtida estimando-se a inclinação da curva $\log(C(\epsilon))$ vs $\log(\epsilon)$ (Fig. 2.1(a)). Como esta curva não é propriamente uma linha reta, uma regressão linear por mínimos quadrados para toda a curva resultaria numa estimativa bastante grosseira da porção linear da mesma. De fato, esta curva apresenta três regiões [25]; na primeira região o comportamento fractal não é evidente, já que a escala de análise ϵ é muito pequena e $\nu \rightarrow 1$. Na região intermediária, a invariância (auto-similaridade) se evidencia, com $\nu > 1$. Por fim, na última região, as escalas ϵ de análise são maiores que o objeto, e este passa a ser observado como um ponto, por isso $\nu \rightarrow 0$. Vale ressaltar que a curva $\log(C(\epsilon))$ vs $\log(\epsilon)$ é uma curva discreta, portanto uma concatenação de segmentos de reta. A partir desta curva, estamos interessados em obter duas medidas, a saber:

- ν_{global}
- $\nu_{mediana}$

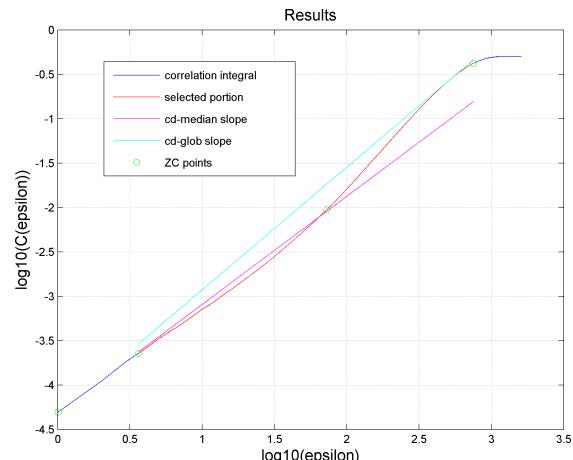
A Dimensão de Correlação Global [51] diz respeito à estimativa da inclinação da região intermediária da curva $\log(C(\epsilon))$ vs $\log(\epsilon)$, ajustando uma reta entre dois pontos apenas: o 2º e o último pontos, enquanto que a Dimensão de Correlação Mediana está relacionada à mediana

das inclinações de todos os segmentos de reta que compõem a curva entre o 2º e o último pontos(Fig. 2.1(a)). Obter tais inclinações significa diferenciar o sinal $\log(C(\epsilon))$, e para tanto usamos uma variação do método introduzido por Marr-Hildreth originalmente proposto para detecção de bordas em imagens, em que buscam-se pontos de cruzamento por zero na 2ª derivada da imagem suavizada por uma gaussiana [72, 42](Fig. 2.1(b)). Estas derivadas podem ser obtidas por meio da propriedade de diferenciação no domínio de Fourier usando convoluções; convoluir um sinal com uma gaussiana e diferenciar o resultado equivale à convolução do sinal original pela derivada da gaussiana [25]. Como a família de derivadas da gaussiana são as *Wavelets Hermitianas* [66], estamos em última análise interessados em procurar por cruzamentos por zero [51, 69] na transformada *wavelet* entre a *Wavelet Hermitiana* de ordem 3 e o sinal $\log(C(\epsilon))$. Sabe-se, do Cálculo Diferencial, que procurar pelos pontos críticos (máximos e mínimos locais e pontos de inflexão) de uma função é equivalente a procurar pelos cruzamentos por zero em sua derivada de primeira ordem. Em nosso caso, a primeira derivada do sinal $\log(C(\epsilon))$ suavizado por uma gaussiana fornece a inclinação de cada segmento constituinte da curva e a segunda derivada detecta os pontos críticos na primeira derivada, ou seja, as extremidades dos segmentos. Então a segunda derivada é o nossa função de interesse, logo mediante a 3ª derivada do sinal $\log(C(\epsilon))$, tomamos conhecimento dos pontos onde há cruzamento por zero [42, 51, 72], ou seja, os pontos onde o sinal apresenta mudanças abruptas de inclinação, como se vê na Fig. 2.1(b). Um vetor contendo os extremos de cada um destes segmentos viabiliza o cálculo imediato das inclinações de todos os segmentos, seguindo-se pela estimação da mediana , isto é, a ($\nu_{mediana}$). Desconsiderando, agora, o primeiro e último segmentos, traçamos uma longa reta entre o segundo e o último pontos de cruzamento por zero e estimamos a inclinação desta reta [51], ou seja, a ν_{global} . Estes resultados são ilustrados na Fig. 2.1(a)

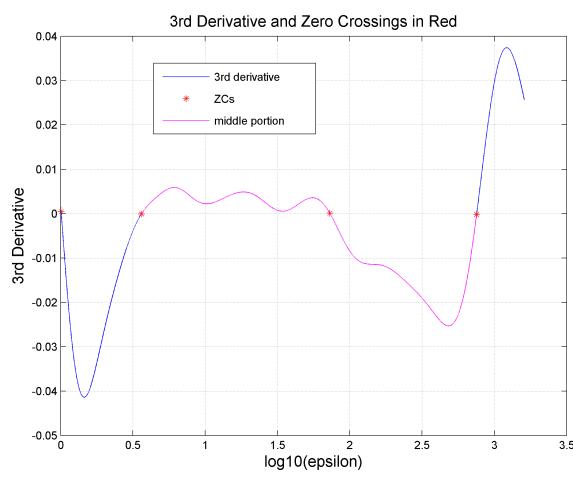
2.3.5.4 Algoritmos para o cálculo da Dimensão de Correlação

Desde o trabalho de Grassberger e Proccacia [43, 44], diversos algoritmos têm sido propostos com o intuito de contornar o problema do longo tempo consumido por (2.14) devido ao cálculo de $\binom{N}{2}$ distâncias, ou seja, $O(N^2)$. A maioria das propostas de algoritmos sugere que nem todas as distâncias sejam efetivamente calculadas, mas apenas algumas. A seguir, descrevemos brevemente as idéias centrais de alguns destes algoritmos.

Farmer *et al.* propuseram um algoritmo **GPA** modificado, introduzindo a definição de pontos de referência [32], segundo o qual , não é necessário calcular as distâncias entre cada um dos N pontos e os $(N - 1)$ restantes, mas sim entre cada um dos N e alguns dos $(N - 1)$ pontos, chamados pontos de referência. Estes pontos de referência devem ser escolhidos de forma a manter a estabilidade do atrator. Mais tarde, Theiler propôs um complicado algoritmo chamado *box-assisted CD* [99], o qual não calcula as $\binom{N}{2}$ distâncias, mas apenas distâncias entre pontos que estejam dentro de caixas de lado r_0 , ou entre estes e pontos em caixas adjacentes àquela. O parâmetro r_0 é um limiar superior que delimitaria o início da terceira região da curva $\log(C(\epsilon))$ vs $\log(\epsilon)$, como discutido anteriormente, a partir do qual, estão as distâncias desnecessárias para calcular a dimensão de correlação. A análise



(a)



(b)

Figura 2.1: (a) Integral de Correlação e Resultados: retas com inclinações dadas pela dimensão de correlação mediana e dimensão de correlação global (b) A terceira derivada obtida pela transformada wavelet da Integral de Correlação e os Cruzamentos por Zero

de complexidade do *box-assisted* ν demonstra que o mesmo consome tempo $O(N \log(N))$, entretanto seu desempenho é fortemente atrelado à escolha de parâmetros. Bingham e Kot aproveitaram a idéia de Theiler de calcular apenas distâncias abaixo de um certo limiar, mas propuseram uma árvore binária multidimensional e busca binária [7] como alternativa à lista de caixas ordenadas lexicograficamente e busca linear do algoritmo de Theiler, melhorando o desempenho deste último. Miao *et al.* apresentaram o algoritmo chamado Δr neighborhood **GPA** [75], que, alega-se, aumenta significativamente a velocidade do cálculo da ν . Seu método usa a definição de *pontos de referência* proposta por Farmer, desconsiderando algumas distâncias, mas não ordena todos os pontos com distâncias inferiores a um certo ϵ , pois observaram a redundância de cálculos em diversos nós da árvore de recorrência produzida por este algoritmo. Propuseram considerar a distância para pontos dentro de um Δr pequeno como sendo iguais, e estimam o tempo do algoritmo em $O(N * \frac{N_{ref}}{q^2})$, onde N_{ref} são *pontos de referência*, e q é o número de elementos que caem em média dentro do intervalo Δr . Os critérios para escolha dos *pontos de referência* e para decidir quão pequenos devem ser os Δr não ficaram claros nesse trabalho. No presente trabalho, algumas destas modificações foram incorporadas, mas não todas.

Modificação 1: distâncias entre pontos de referência. Como as trajetórias são visitadas com a mesma freqüência em toda a extensão do nosso atrator (densidade uniforme de pontos), obtemos nossos pontos de referência por amostragem uniforme. Diferentemente de Farmer e Miao [32, 75], calculamos as distâncias entre estes pontos de referência apenas, e não entre estes e todos os demais.

Modificação 2: simetria entre pontos [32, 75]. Sejam $r_{ij} = |X_i - X_j|$ as distâncias entre os pontos X_i e X_j , como definido na equação 2.15. Sabendo que distâncias $r_{ij} = r_{ji}$, basta considerar as r_{ij} , $\forall j > i$.

Modificação 3: cálculos parciais das distâncias. Considerando um número médio típico de pontos nestes atratores da ordem de 2×10^4 , calcular as $\binom{N}{2}$ distâncias de uma vez requer espaço para armazenamento da ordem de 4×10^8 , isto é, 400 Mbytes de memória! Então, em vez de calcular todos os elementos do conjunto de distâncias D , para depois distribuí-las num histograma de $C(\epsilon)$, efetuamos cálculos parciais, computando a cada iteração em i somente as distâncias do subconjunto $D_i = \{r_{ij}\}$, $\forall j > i$, e contando-as no histograma de $C(\epsilon)$. Quando $i = N - 1$ e $j = N$, temos $D_{N-1} = \{r_{(N-1),N}\}$. Portanto, todas as distâncias terão sido calculadas e devidamente contadas no histograma de $C(\epsilon)$.

Modificação 4: contagens parciais das distâncias. Os valores extremos - mínimo e máximo - dos *bins* do histograma de $C(\epsilon)$ seriam respectivamente 1 e *Diag*, onde *Diag* é o comprimento da diagonal da imagem. Para cada subconjunto D_i , encontrar o *bin* a ser incrementado para cada $r_{ij} \in D_i$ levaria tempo $O(|D_i| \times Diag)$, pois cada r_{ij} seria comparado, desnecessariamente, com cada ϵ possível para efeito de contagem. Em vez disso, as distâncias r_{ij} são ordenadas

em tempo $|D_i| \ln(|D_i|)$, e o vetor de distâncias v_i - representando o subconjunto D_i - é varrido uma única vez, sendo cada $r_{ij} \in D_i$ contado no bin $\epsilon = \lceil r_{ij} \rceil$.

Como se vê, o cálculo da função Integral de Correlação para cada ϵ constitui a etapa mais crítica do processo de obtenção da Dimensão de Correlação. A implementação para esta etapa, usada neste trabalho, é descrita em pseudo-código no Apêndice C.

2.4 Resultados

A partir das transformadas wavelet das imagens de segmentação manual e automática, com **retinopatia diabética proliferativa PDR** e sem *PDR*, foram obtidas distribuições dos coeficientes e das fases, ou seja, *Módulos da Transformada Wavelet*, *Ângulos da Transformada Wavelet* respectivamente, bem como as respectivas curvas das integrais de correlação.

Na Fig. 2.2 vê-se a representação em cores de características obtidas a partir da *CWT*, usando a 1^a *wavelet* hermitiana (Gradiente da Gaussiana). A aplicação deste operador sobre o campo escalar definido pelas imagens de segmentações resulta em campos vetoriais. As escalas coloridas ressaltam a similaridade dos respectivos valores em pontos diferentes da imagem. Em particular, note a variação da cor associada à variação de curvatura na aproximação das extremidades e dos pontos de cruzamento das segmentações (Fig. 2.2(e)).

Em cada grupo de imagens ilustradas na Fig. 2.3, vêm-se a sobreposição das segmentações manual e automática sobre as respectivas imagens originais, a curva da integral de correlação na 2^a linha e o histograma da distribuição dos ângulos da transformada na 3^a linha.

Assim, do histograma da distribuição dos *Ângulos da Transformada Wavelet*, calculamos a *Entropia de Orientação* e da curva da integral de correlação, podemos calcular as *Dimensões de Correlação Global* (ν_{global}) e *Mediana* (ν_{mediana}).

Os resultados dos cálculos das características para as segmentações manuais e automáticas encontram-se no Apêndice A, tabelas A.1 e A.2 respectivamente. Imagens com representações dos resultados encontram-se publicados em http://www.vision.ime.usp.br/~australia/projects/angio_08_08_out/3/. Estes resultados também foram apresentados em [56].

2.4.1 Validação do Segmentador Automático: comparação entre segmentações manuais e automáticas

No problema abordado nesta seção, temos dois grupos de resultados de segmentações, isto é, o grupo de esqueletos produzido pela segmentação manual de um especialista (ω_1) e o grupo de esqueletos produzido pela segmentação automática por meio de transformadas *Wavelet* e classificadores estatísticos (ω_2). Obviamente, esperaria-se que os resultados fornecidos por um segmentador automático ideal coincidissem com os resultados fornecidos por especialistas, havendo neste caso uma classe de segmentação apenas. Então a pergunta premente é: os grupos ω_1 e ω_2 pertencem a uma mesma classe ou pertencem a classes distintas? O processo de verificação do nível de coincidência entre os resultados obtidos e os esperados constitui a etapa de validação dos métodos de segmentação, primordial na avaliação dos resultados obtidos pelos mesmos.

A mais simples destas avaliações, claramente, é dada pela inspeção visual, segundo a qual, especialistas clínicos avaliaram as segmentações automáticas, observando um casamento considerado bom entre estas e as segmentações manuais. No entanto, são desejáveis meios objetivos para avaliar os resultados da segmentação, viabilizando análises múltiplas e em grande escala [51].

A seguir, apresentamos algumas análises comparativas entre estes resultados, com vistas à va-

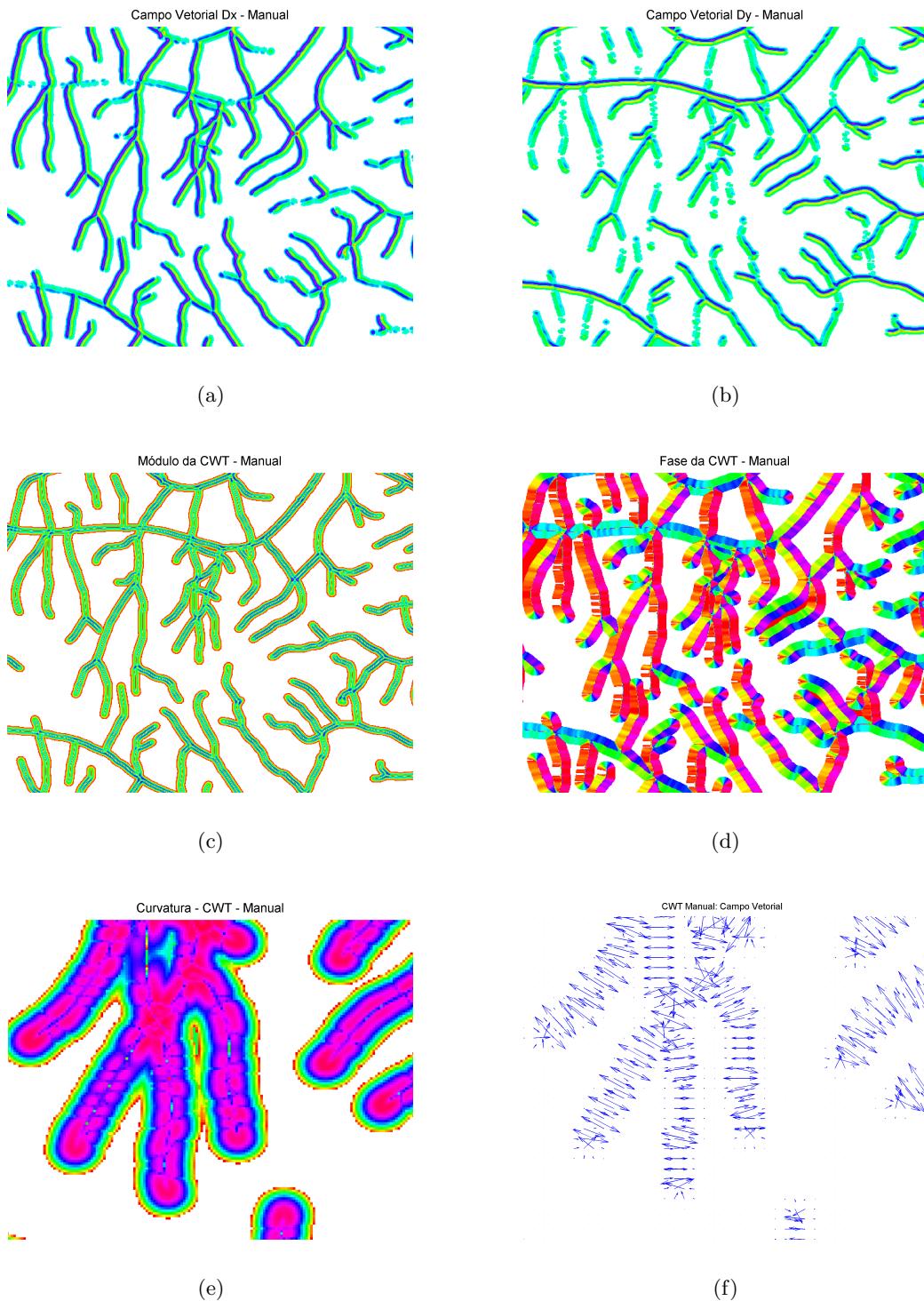


Figura 2.2: Representação em cores de características derivadas da *CWT*, aplicando a 1^{a} *wavelet* hermitiana (Gradiente da Gaussiana) ao campo escalar definido pela imagem da segmentação manual, resultando num campo vetorial. A escala de cores enfatiza a similaridade dos valores em pontos diferentes da imagem. (a) Componente dx (b) Componente dy (c) Módulo (d) Fase (e) Curvatura (f) Campo Vetorial

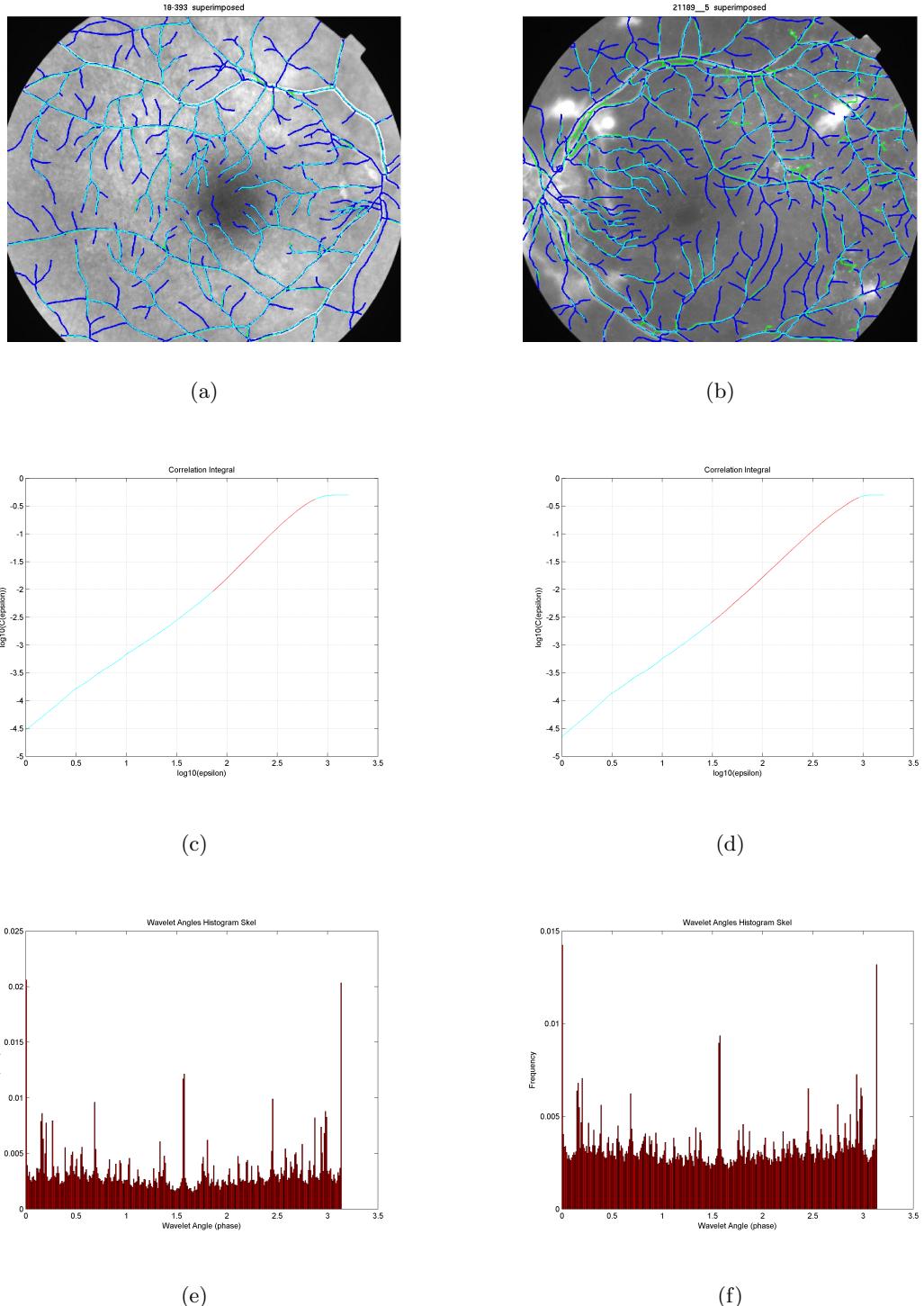


Figura 2.3: Segmentação manual (azul) sobreposta à segmentação automática (ciano) sobre a imagem original: Imagem 18-393 (a) (sem PDR). Imagem 20-201 (b) (com PDR). Integral de Correlação - Sem PDR (c). Integral de Correlação - Com PDR (d). Distribuição de fase da CWT - Sem PDR (e). Distribuição de fase da CWT - Com PDR (f)

lidação do método de segmentação por transformadas *Wavelet* e classificadores estatísticos por meio de curvas *ROC*, coeficiente de correlação de Pearson, desvio quadrático médio, teste de hipóteses e análise discriminante linear (LDA).

Ainda, uma estratégia de comparação entre segmentações automáticas com e sem *PDR* é apresentada.

2.4.1.1 Comparação entre segmentadores por curvas *ROC*

Uma cuidadosa análise com o emprego de curvas *ROC* - (*Receiver Operating Characteristic*) do segmentador baseado em *Wavelets* e Classificadores Estatísticos [93, 94, 92] demonstrou que este superou os resultados obtidos por outros métodos, tendo sido o método, cujos resultados mais se aproximaram das segmentações manuais, apesar da ocorrência de falsos positivos em regiões específicas das imagens [92]. Essa abordagem baseia-se na relação entre os sucessos e alarmes falsos do segmentador, isto é, verdadeiros positivos e falsos positivos, para diferentes limiarizações aplicadas sobre o espaço de probabilidades *a posteriori* [33].

Esta avaliação apontou a superioridade deste segmentador em relação aos demais no estado-da-arte [92]. Não obstante este resultado, subsiste a pergunta quanto à suficiência do grau de similaridade atingido entre os grupos ω_1 e ω_2 para que estas sejam consideradas uma única classe - caso ideal - ou não. O grau de similaridade entre ω_1 e ω_2 será quantificado mediante a Análise de Formas, com o emprego do conjunto de características proposto e descrito na Seção 2.2.

2.4.1.2 Comparação Preliminar entre segmentadores por Coeficiente de Correlação de Pearson e o Desvio Quadrático Médio (MSE)

Preliminarmente, uma comparação entre segmentações manuais (grupo ω_1) e automáticas (grupo ω_2) foi efetuada usando as características descritas na Seção 2.2, analogamente ao que foi publicado em [51]. Considere os espaços de características F^{ω_j} , com $j \in \{1, 2\}$, organizados como matrizes, seguindo a notação adotada em [25], cujas linhas são formadas pelos vetores de características $f_k^{\vec{\omega}_j T}$, para $k = 1, 2, \dots, N$, $N = |\omega_j| = 27$ isto é:

$$\mathbf{F}^{\omega_j} = \begin{pmatrix} \longleftarrow & f_1^{\vec{\omega}_j T} & \longrightarrow \\ \longleftarrow & f_2^{\vec{\omega}_j T} & \longrightarrow \\ \vdots & & \vdots \\ \longleftarrow & f_k^{\vec{\omega}_j T} & \longrightarrow \\ \vdots & & \vdots \\ \longleftarrow & f_N^{\vec{\omega}_j T} & \longrightarrow \end{pmatrix} \quad (2.19)$$

onde as componentes do vetor de características relativo a uma certa imagem k pertencente ao

grupo ω_j são dadas por:

$$f_k^{\vec{\omega}_j} = \begin{pmatrix} f_{k,1}^{\omega_j} \equiv \text{Área} \\ f_{k,2}^{\omega_j} \equiv \text{Perímetro} \\ f_{k,3}^{\omega_j} \equiv \text{Circularidade} \\ f_{k,4}^{\omega_j} \equiv 2^o \text{ momento} \\ f_{k,5}^{\omega_j} \equiv \text{Entropia} \\ f_{k,6}^{\omega_j} \equiv \text{Curvatura} \\ f_{k,7}^{\omega_j} \equiv \nu_{\text{mediana}} \\ f_{k,8}^{\omega_j} \equiv \nu_{\text{global}} \end{pmatrix} \quad (2.20)$$

Note que a coluna i da matriz do espaço de características F^{ω_j} fornece o vetor coluna que denotaremos por $\phi_i^{\vec{\omega}_j}$, em que cada componente $\phi_{k,i}^{\omega_j}$ indexada por $k = 1, 2, \dots, 27$ corresponde ao valor da i -ésima característica para a k -é-sima imagem pertencente ao grupo ω_j , ou seja:

$$\phi_i^{\vec{\omega}_j} = \begin{pmatrix} \phi_{1,i}^{\omega_j} \\ \phi_{2,i}^{\omega_j} \\ \vdots \\ \phi_{k,i}^{\omega_j} \\ \vdots \\ \phi_{27,i}^{\omega_j} \end{pmatrix} \quad (2.21)$$

Assim, temos nos vetores $\phi_i^{\vec{\omega}_j}$ uma lista ordenada contendo os valores calculados para a característica i das 27 imagens no grupo ω_j .

Para avaliar então os resultados obtidos das segmentações do grupo ω_2 em relação às segmentações do grupo ω_1 [51], calculamos duas medidas de associação entre cada par de características $(\phi_{k,i}^{\omega_1}, \phi_{k,i}^{\omega_2})$, para todas as imagens $k = 1, 2, \dots, 27$ e todas as características $i = 1, 2, \dots, 8$, isto é, o *Coeficiente de Correlação de Pearson - (r)* e o *Desvio Quadrático Médio - (Mean Squared Error - MSE)*.

No contexto desta análise, o Coeficiente de Correlação de Pearson (r) pode ser escrito como [83]:

$$r = \text{corr}(\phi_i^{\omega_1}, \phi_i^{\omega_2}) = \frac{\sum_{k=1}^N (\phi_{k,i}^{\omega_1} - \bar{\phi}_i^{\omega_1})(\phi_{k,i}^{\omega_2} - \bar{\phi}_i^{\omega_2})}{\left[\sum_{k=1}^N (\phi_{k,i}^{\omega_1} - \bar{\phi}_i^{\omega_1})^2 \sum_{k=1}^N (\phi_{k,i}^{\omega_2} - \bar{\phi}_i^{\omega_2})^2 \right]^{1/2}} \quad (2.22)$$

Vale salientar que os valores de r são mapeados no intervalo $[-1, 1]$. Para $r = 0$, diz-se que não há correlação entre os dados. Quando $r > 0$ diz-se que os dados são correlacionados positivamente, e para $r = 1$ tem-se a "correlação positiva completa". Quando $r < 0$ diz-se que os dados são

Tabela 2.1: Resultados para r e MSE

características	r	MSE
Área	0.798	0.388
Perímetro	0.797	0.390
Circularidade	0.796	0.393
2º momento	0.394	1.167
Entropia	0.142	1.652
Curvatura	0.206	1.529
Dimensão de Correlação Mediana (ν_{mediana})	-0.452	2.796
Dimensão de Correlação Global (ν_{global})	0.523	0.918

correlacionados negativamente, e para $r = -1$ tem-se a "correlação negativa completa" [83].

O *Desvio Quadrático Médio* - (*Mean Squared Error* - MSE), por sua vez, é definido em Teoria da Utilidade como a função perda, que no contexto desta análise pode ser escrita como:

$$MSE = L(\phi_i^{\omega_1}, \phi_i^{\omega_2}) = E[(\phi_i^{\omega_1} - \phi_i^{\omega_2})^2] \quad (2.23)$$

O MSE foi proposto por Carl Friedrich Gauss como uma medida de perda, incurácia ou imprecisão. Gauss admitiu que esta medida pode ser considerada arbitrária, mas defende sua aplicação pela simplicidade matemática e conveniência da mesma [15]. Em problemas de decisão em que os estimadores não são tendenciosos (*unbiased*), o MSE é a própria variância, o que explica em parte sua popularidade entre os estatísticos, além de sua relação com a Teoria dos Mínimos Quadrados [6].

Os resultados obtidos para r e MSE são listados na tabela 2.1. Observou-se alta correlação, isto é, $r > 0.79$ somente para área, perímetro e circularidade e baixa correlação, isto é $r < 0.53$, para as demais seqüências de características. Ainda, obtivemos baixo desvio, isto é, $MSE < 1$, para as características área, perímetro, circularidade e ν_{global} , e alto desvio, isto é, $MSE > 1$ para as demais características. Destes resultados inferimos que existe uma diferença perceptível entre os resultados das segmentações manuais e automáticas, pois a correlação é baixa e o desvio é alto para a maioria das características. Na Fig. 2.4 vêem-se a representação gráfica da concordância entre as segmentações dada pela Área (Fig. 2.4(a)) e a discordância dada pela Entropia (Fig. 2.4(b)). Os gráficos para todas as características são apresentados no Apêndice B.

As altas correlações e baixos desvios para área, perímetro e circularidade podem ser explicadas como uma relação de proporcionalidade direta entre o número de pixels de cada segmentação, ou seja, quando diminuem os pixels contados por cada uma destas características para a segmentação manual, também é observada uma diminuição na contagem da segmentação automática. Os valores de r e MSE sugerem baixa concordância entre as segmentações manual e automática neste experimento, ao contrário do que se observou nos resultados publicados em [51]. Esta diferença

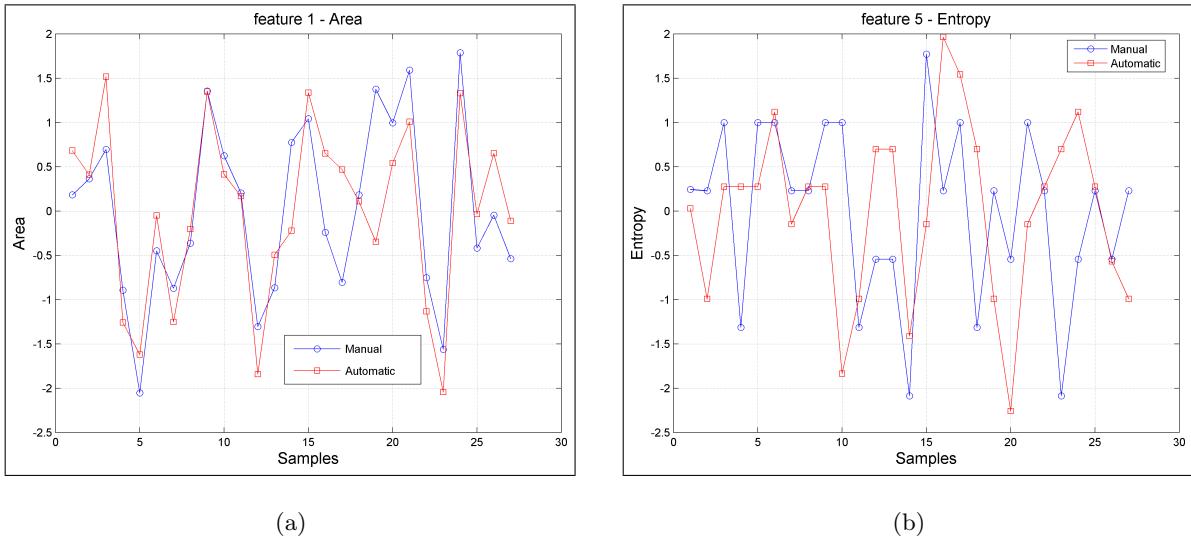


Figura 2.4: (a) A característica Área apresenta alta correlação e baixo desvio quadrático médio, (b) enquanto a característica Entropia apresenta baixa correlação e alto desvio quadrático médio.

possivelmente está relacionada essencialmente a dois fatores:

- I. as imagens naquele trabalho eram do tipo não-midriáticas, que não capturam tantos detalhes quanto as usadas neste trabalho, isto é, angiografias com contraste fluorescente.
 - II. das 57 imagens naquele trabalho, apenas 15 eram de diabéticos, nenhum deles apresentando neovascularização, anormalidades clínicas, microaneurismas, hemorragias, etc, ou seja, imagens pertencentes a uma classe apenas, enquanto que as imagens neste trabalho representavam indivíduos com neovascularização e sem neovascularização.

Temos assim, a primeira constatação de que a segmentação manual possui mais pixels que a segmentação automática, sendo, portanto, mais detalhada.

2.4.1.3 Comparação entre Segmentadores por Teste Estatístico de Hipóteses

As conclusões da análise desenvolvida na seção anterior apontam para a dissimilaridade entre os grupos, ou seja, a existência de duas classes de segmentações ω_1 e ω_2 . Assumindo então, que as características adotadas têm poder discriminante suficiente para discernir entre estas classes, podemos testar as distribuições de valores, para cada característica, obtidas para a ω_1 e ω_2 , a fim de investigarmos a hipótese de que as duas distribuições apresentam a mesma média, ou em outras palavras, as duas amostras pertencem a mesma população. Para tanto, precisamos estimar a diferença entre as médias, levando em conta a força e a significância desta diferença, já que uma diferença de médias pode ser pequena comparada ao desvio padrão, mas muito significante se o espaço amostral for grande, ou pode ser moderadamente grande, mas não significante, se o espaço

amostral for pequeno. A escala mais apropriada para medir a distância entre médias não é o número de desvios padrão, mas sim o número de erros padrão, o qual mede a exatidão com a qual a média amostral estima o valor verdadeiro [83]. O erro padrão, ou desvio padrão da média, é dado pela razão entre o desvio padrão e a raiz quadrada do tamanho da amostra. Para testarmos se há diferença significante entre as médias $\bar{\phi}_1^i$ e $\bar{\phi}_2^i$ dos valores obtidos para a característica i aplicada às classes ω_1 e ω_2 , formulamos, então, o problema como um teste estatístico de hipóteses, em que:

H_0 : As médias $\bar{\phi}_1^i$ e $\bar{\phi}_2^i$ não diferem significantemente.

H_1 : As médias $\bar{\phi}_1^i$ e $\bar{\phi}_2^i$ diferem significantemente.

Sendo H_0 a hipótese nula e H_1 a hipótese alternativa.

Fundamentos de Testes Estatísticos de Hipóteses. No que se segue, formalizamos brevemente a idéia subjacente ao teste de hipóteses mencionado acima, para resolver este problema de decisão, no contexto de reconhecimento de padrões, adotando a notação usada em [100] e [81]. Seja x uma variável aleatória com distribuição densidade de probabilidade $p(x)$, supostamente conhecida, descrita por um parâmetro desconhecido θ . Queremos testar as hipóteses:

H_0 : $\theta = \theta_0$

H_1 : $\theta \neq \theta_0$

Sejam x_i , $i = 1, 2, \dots, N$ as realizações da v.a. x . Seja $q = f(x_1, x_2, \dots, x_N)$ uma v.a. (estatística de teste), onde $f(\cdot)$ é uma função escolhida conforme o problema específico, de maneira que a f.d.p. de q , em termos de θ , seja dada por $p_q(q; \theta)$. Seja D o intervalo denominado *Região de Aceitação*, em que q pode cair com alta probabilidade, pela hipótese H_0 , e \bar{D} o seu complemento, denominado *Região Crítica*. Obviamente, se q cair no intervalo D , a hipótese H_0 é aceita, caso contrário H_0 é rejeitada. Então, se H_0 é verdadeira, a probabilidade de cometer um erro na tomada de decisão é dada por $P(q \in \bar{D} | H_0) \equiv \alpha$, isto é, a probabilidade do valor da estatística de teste q cair na Região Crítica, dado que H_0 é verdadeira, é igual ao nível de significância α , que expressa a confiabilidade da decisão tomada; quanto maior seu valor, menor a confiabilidade. Usualmente, o nível de significância α é arbitrariamente pré-determinado, definindo consequentemente a Região Crítica \bar{D}_α e o seu complemento D_α , ou seja, para um nível de significância α , o intervalo de aceitação $D_\alpha \equiv [-x_\alpha, x_\alpha]$ é o intervalo no qual o valor calculado para a estatística q cai com probabilidade igual a $1 - \alpha$. Tendo adotado um nível de significância α , os limites da Região de Aceitação encontram-se tabelados [100].

Pela abordagem descrita, considera-se um Intervalo de Confiança, comparando-se valores calculados para a estatística q com os limites tabelados da Região Crítica, para um dado nível de significância α arbitrariamente escolhido. Alternativa e equivalentemente, outra abordagem para este processo considera a probabilidade de observar-se um valor para a estatística calculada q , tão

Tabela 2.2: Resultados do teste-t com amostras independentes e variâncias diferentes. Os $t_{\alpha;\nu}$ correspondem aos valores tabelados [34] para t_i ao nível de significância α e ν_i graus de liberdade (Eq. 2.25).

Característica	ω_1		ω_2		t_i	ν_i	$t_{\alpha;\nu_i}$	p-valor
	média	desvio padrão	média	desvio padrão				
Área	102203	23189	24565	5201	-16,98	29	4,506	< 0.0001
Perímetro	37177	8363	24421	5166	-6,74	44	4,269	< 0.0001
Circularidade	13524	3017	24279	5131	9,39	43	4,295	< 0.0001
2º momento	4362	25	1080	24	-492,95	52	4,212	< 0.0001
Entropia	5,63	0,01	5,62	0,02	-0,68	41	4,321	0.50
Curvatura	178,0	3,9	133,6	4,3	-39,86	52	4,212	< 0.0001
$\nu_{mediana}$	1,68	0,1	1,42	0,1	-11,76	50	4,221	< 0.0001
ν_{global}	1,58	0.04	1,49	0,04	-8,26	52	4,212	< 0.0001

extremo ou mais do que o obtido para a amostra em questão. Esta probabilidade é denominada Nível Descritivo do teste, também conhecida como nível- p ou p -valor. Por esta abordagem, H_0 é rejeitada caso $p \leq \alpha$, e aceita caso contrário.

Testando nossas hipóteses. No contexto de nossa aplicação, estamos interessados em verificar a hipótese de que não há diferença estatística significante entre as médias das características obtidas para as segmentações manuais (classe ω_1) e as automáticas (classe ω_2). Para tanto, formulamos as hipóteses H_0 e H_1 :

$$H_0: \Delta\phi_i = \bar{\phi}_i^{\omega_1} - \bar{\phi}_i^{\omega_2} = 0, \forall i \in \{1, \dots, 8\}$$

$$H_1: \Delta\phi_i = \bar{\phi}_i^{\omega_1} - \bar{\phi}_i^{\omega_2} \neq 0, \forall i \in \{1, \dots, 8\}$$

Assumindo que as classes ω_1 e ω_2 são independentes e que suas variâncias populacionais σ_1^2 e σ_2^2 são desconhecidas e diferentes, temos o caso em que o teste mais adequado é o denominado *teste-t*, cuja estatística q apresenta uma distribuição *t-student* com ν graus de liberdade. Neste caso, a estatística q é dada por [83, 23]:

$$t_i = \frac{\bar{\phi}_i^{\omega_1} - \bar{\phi}_i^{\omega_2}}{\sqrt{\frac{s_{\phi_i^{\omega_1}}^2}{n_1} + \frac{s_{\phi_i^{\omega_2}}^2}{n_2}}} \quad (2.24)$$

e o número de graus de liberdade da distribuição *t-student* é dado por:

$$\nu_i = \frac{(A + B)^2}{\frac{A^2}{n_1-1} + \frac{B^2}{n_2-1}}, \text{ onde } A = \frac{s_{\phi_i^{\omega_1}}^2}{n_1}, \text{ e } B = \frac{s_{\phi_i^{\omega_2}}^2}{n_2} \quad (2.25)$$

Mediante os resultados do teste- t , registrados na tabela 2.2, a hipótese nula H_0 foi rejeitada ao nível de significância $\alpha = 0.0001$, pois para todas as características, salvo a Entropia, tem-se $|t_i| > t_{\alpha;\nu_i}$. Isto significa que $|t_i|$ cai na região Crítica, para todas as características, exceto a Entropia, portanto as diferenças entre as médias de todas as características são estatisticamente significantes ($p < \alpha = 0.001$), exceto para a entropia, com $p = 0.50 > \alpha$, conforme calculado pelo sistema R [41]. Estas diferenças surgem, porque mais vasos são identificados pelo observador do que pelo procedimento automático [56], corroborando a constatação na avaliação preliminar descrita no início da Seção 2.4.1.2.

2.4.2 Validação de Características: avaliação do poder preditivo para distinção entre estruturas com PDR e sem PDR

Muito embora as análises de validação discutidas na Seção 2.4.1 pelo conjunto de características proposto tenham apontado para a dissimilaridade entre as segmentações manuais e as automáticas, também estamos interessados no poder preditivo destas mesmas características para distinção entre segmentações automáticas com PDR e sem PDR. Ainda que o segmentador automático tenha perdido parte das informações, isto é, detalhes das árvores venosas que foram capturados nas segmentações manuais, é possível que os detalhes capturados pela segmentação automática sejam suficientes para discernir entre imagens de retina com PDR e sem PDR. Assim, procedemos a seguir a validação individual e em conjunto das características usadas para este fim, lançando mão de teste de hipóteses e curvas ROC.

2.4.2.1 Teste de hipóteses para avaliar o poder discriminante das características entre segmentações com PDR e sem PDR

Na Seção 2.4.1.3, testamos se há diferença entre as médias de características aplicadas a ω_1 e ω_2 , assumindo que as características são suficientemente discriminantes para separar aquelas classes. Nesta seção, consideramos duas subclasses ω_j de segmentações em ω_2 , isto é, $\omega_{2p} \cup \omega_{2n} = \omega_2$, ou em palavras, segmentações automáticas de árvores venosas com PDR (ω_{2p}) e sem PDR (ω_{2n}). Para estas classes, abordamos agora o problema inverso, isto é, assumimos a dissimilaridade entre ω_{2p} e ω_{2n} , com base em conhecimento *a priori*, e empregamos o teste estatístico de hipóteses para avaliar a eficácia das características adotadas, como meio de proceder uma pré-seleção de características, testando a capacidade discriminante de cada característica independentemente [100], com vistas a uma possível redução de dimensionalidade do espaço de características.

Testando nossas hipóteses. Estamos agora interessados em verificar a hipótese de que há diferença estatística significante entre as médias das características obtidas para as segmentações automáticas com PDR (classe ω_{2p}) e sem PDR (classe ω_{2n}). Para tanto, formulamos as hipóteses H_0 e H_1 :

$$H_0: \Delta\phi_i = \bar{\phi}_i^{\omega_{2p}} - \bar{\phi}_i^{\omega_{2n}} = 0, \forall i \in \{1, \dots, 8\}$$

$$H_1: \Delta\phi_i = \bar{\phi}_i^{\omega_{2p}} - \bar{\phi}_i^{\omega_{2n}} \neq 0, \forall i \in \{1, \dots, 8\}$$

Tabela 2.3: Resultados do teste-t com amostras independentes e variâncias diferentes, entre as médias de segmentações com PDR e sem PDR.

Característica	ω_{2n}		ω_{2p}		t_i	ν_i	p -valor	AUC
	média	desvio padrão	média	desvio padrão				
Área	0,038	0,82	-0,026	1,1	0,169	23	0,87	0,49
Perímetro	0,040	0,82	-0,027	1,1	0,178	23	0,86	0,49
Circularidade	0,041	0,82	-0,029	1,1	0,186	23	0,85	0,49
2º momento	-0,355	0,63	0,244	1,1	-1,739	19	0,095	0,72
Entropia	-0,416	1,28	0,286	0,6	-1,672	20	0,12	0,70
Curvatura	-0,456	0,85	0,313	1,0	-2,152	25	0,042	0,76
$\nu_{mediana}$	-0,441	1,05	0,303	0,9	-1,937	25	0,068	0,70
ν_{global}	-0,062	0,82	0,043	1,1	-0,279	23	0,77	0,49

Diferentes características usualmente mensuram aspectos diferentes das formas, por conseguinte podem cobrir magnitudes de ordens diferentes, estando também em unidades diferentes. A fim de compará-las, submetemos estas características à chamada transformada normal [25], tornando-as adimensionais e viabilizando comparações entre as mesmas, em termos de métrica no espaço de características. A transformada normal se define como:

$$\hat{\phi}_i = \frac{\phi_i - \mu_i}{\sigma_i} \quad (2.26)$$

onde $\hat{\phi}_i$ é a i ésima característica normalizada, μ_i é a i ésima característica original, μ_i é a média da i ésima característica e σ_i seu desvio padrão. Desta transformação resultam características com média zero e desvio padrão unitário.

Assumindo que as classes ω_{2p} e ω_{2n} são independentes e que suas variâncias populacionais σ_{2p}^2 e σ_{2n}^2 são desconhecidas e diferentes, temos analogamente à Seção 2.4.1.3, o caso em que se aplica o teste- t , cuja estatística q apresenta uma distribuição t -student com ν graus de liberdade, conforme as Eq. 2.27 e 2.28.

$$t_i = \frac{\bar{\phi}_i^{\omega_{2p}} - \bar{\phi}_i^{\omega_{2n}}}{\sqrt{\frac{s_{\phi_i^{\omega_{2p}}}^2}{n_{2p}} + \frac{s_{\phi_i^{\omega_{2n}}}^2}{n_{2n}}}} \quad (2.27)$$

e o número de graus de liberdade da distribuição t -student é dado por:

$$\nu_i = \frac{(A + B)^2}{\frac{A^2}{n_{2p}-1} + \frac{B^2}{n_{2n}-1}}, \text{ onde } A = \frac{s_{\phi_i^{\omega_{2p}}}^2}{n_{2p}}, \text{ e } B = \frac{s_{\phi_i^{\omega_{2n}}}^2}{n_{2n}} \quad (2.28)$$

Mediante os resultados do teste- t , registrados na tabela 2.3, a hipótese nula H_0 foi aceita ao nível de significância $\alpha = 0,05$, pois para todas as características, salvo a Curvatura, tem-se $p > \alpha = 0,05$. Isto significa que a probabilidade (p -valor) de obter-se um valor calculado para a estatística t tão extremo ou mais que o observado é grande, dado que H_0 é verdadeira, para os testes com todas as características, exceto a Curvatura, que apresenta p -valor inferior ao nível de significância α . Este resultado sugere que não temos evidências suficientes a partir dos dados para rejeitar H_0 , portanto concluímos que as diferenças entre as médias de todas as características não são estatisticamente significantes ($p > \alpha = 0,05$), exceto para a curvatura ($p = 0,042 < \alpha$). Pelos resultados apresentados por este método, apenas a curvatura seria selecionada. Note que, relaxando o nível de significância para $\alpha = 0,15$, poderíamos rejeitar H_0 ao menos para as características derivadas das transformadas *Wavelet*. O fato de não verificarmos diferença significante para quase todas as características pode ser devido a três razões:

- I. a segmentação automática perde detalhes importantes para distinguir entre as classes de imagens ω_{2p} e ω_{2n} .
- II. as características consideradas não têm poder discriminante suficiente para distinguir entre as classes de imagens ω_{2p} e ω_{2n} .
- III. para o nível de significância desejado, as médias não estão afastadas o suficiente, a fim de garantir diferenças estatisticamente significantes.

No entanto, ainda que H_0 fosse rejeitada para todas as características, se as respectivas médias não estiverem afastadas o suficiente, este resultado não garantiria a capacidade de discriminação daquelas características, haja vista a possibilidade do espalhamento em torno da média ser grande o bastante para causar a sobreposição entre classes [100]. Considerando o exposto, avaliaremos os resultados de dois classificadores (LDA e Naive-Bayes) aplicados ao espaço de características, por meio de curvas ROC, que levam em conta a sobreposição entre classes.

2.4.2.2 Curvas ROC para avaliar o poder discriminante individual das características entre segmentações com PDR e sem PDR

Fundamentos da Teoria de Reconhecimento de Padrões - Classificação. No que se segue, comentamos brevemente a derivação de regras de decisão ótimas, a partir da condição de mínimo risco e da Regra de Bayes. A seguir, descrevemos as chamadas curvas ROC como avaliadores de desempenho de classificadores, explicitando sua dependência daquelas regras de decisão ótimas, escritas como razões entre verossimilhanças.

Em Teoria de Decisão, um risco é um custo esperado e as melhores ações são aquelas tomadas, mediante regras de decisão, que incorrem em mínimo risco local, garantindo assim, mínimo risco global. Para fins de notação, considere agora o espaço de características F , formado pelos subespaços

de características representados pelas Eq. 2.19 e Eq. 2.20, dado pela matriz:

$$F = \begin{pmatrix} F^{\omega_1} \\ F^{\omega_2} \end{pmatrix} \quad (2.29)$$

Uma regra de decisão é uma função $\alpha : F \rightarrow \{\alpha_1, \alpha_2, \dots, \alpha_a\}$, que para cada vetor de características $\vec{f}_k \in F$, $k \in \{1, 2, 3, \dots, 54\}$, decide por uma ação $\alpha_i \in \{\alpha_1, \alpha_2, \dots, \alpha_a\}$. Se a ação $\alpha(\vec{f}_k) = \alpha_i$ escolhida é tal que incorra no custo $\lambda_{ij} = \lambda(\alpha_i | \omega_j)$, devido à ação tomada α_i , dado que o estado da natureza é ω_j , então o risco desta ação é uma combinação linear dos custos para todas as c classes ω_j , ponderadas pelas respectivas *posterioris*, isto é:

$$R(\alpha_i | \vec{f}_k) = \sum_{j=1}^c \lambda(\alpha_i | \omega_j) P(\omega_j | \vec{f}_k) \quad (2.30)$$

Sendo a função risco global a soma dos riscos para todo \vec{f}_k , ou seja:

$$R = \int R(\alpha(\vec{f}_k) | \vec{f}_k) p(\vec{f}_k) d\vec{f}_k \quad (2.31)$$

minimizar este risco global significa escolher uma ação α_{min} para cada \vec{f}_k que minimize o risco dado pela Eq. 2.30, isto é:

$$\alpha_{min} = \arg \min_{\alpha_i} (R(\alpha_i | \vec{f}_k)) \quad (2.32)$$

Para o caso particular de duas classes, em que se insere nosso problema, a ação α_1 pode ser classificar um vetor de características em ω_{2p} e α_2 a de classificar um vetor de características em ω_{2n} . Neste caso, temos, a partir da Eq. 2.30, os riscos para cada decisão equivocada:

$$\begin{aligned} R(\alpha_1 | \vec{f}_k) &= \lambda_{11} P(\omega_{2p} | \vec{f}_k) + \lambda_{12} P(\omega_{2n} | \vec{f}_k) \\ R(\alpha_2 | \vec{f}_k) &= \lambda_{21} P(\omega_{2p} | \vec{f}_k) + \lambda_{22} P(\omega_{2n} | \vec{f}_k) \end{aligned} \quad (2.33)$$

e podemos escrever uma regra de decisão que minimize o risco, tomando:

$$\begin{array}{ll} \text{Se } R(\alpha_1 | \vec{f}_k) < R(\alpha_2 | \vec{f}_k) & \text{escolha } \omega_{2p} \\ \text{caso contrário} & \text{escolha } \omega_{2n} \end{array} . \quad (2.34)$$

Substituindo os riscos dados pela Eq. 2.33 na Eq. 2.34, apresentamos a regra de decisão na forma de uma desigualdade entre probabilidades a posteriori:

$$(\lambda_{21} - \lambda_{11}) P(\omega_{2p} | \vec{f}_k) > (\lambda_{12} - \lambda_{22}) P(\omega_{2n} | \vec{f}_k) \quad (2.35)$$

Podemos usar a Regra de Bayes, para reescrever a Eq. 2.35 em outras formas. A Regra de Bayes

é definida como:

$$P(\omega_j | \vec{f}_k) = \frac{p(\vec{f}_k | \omega_j) P(\omega_j)}{p(\vec{f}_k)} \quad (2.36)$$

ou, informalmente por:

$$posterior = \frac{\text{verossimilhança} \times \text{priori}}{\text{evidência}} \quad (2.37)$$

onde, a evidência é apenas um fator de escala, e, neste caso em que temos duas classes possíveis, escreve-se como:

$$p(\vec{f}_k) = \sum_{j \in \{\omega_{2p}, \omega_{2n}\}} p(\vec{f}_k | \omega_j) P(\omega_j) \quad (2.38)$$

Na Eq. 2.36, a função densidade de probabilidade condicional de classe (de estado) $p(\vec{f}_k | \omega_j)$, também chamada *verossimilhança* de ω_j com relação ao \vec{f}_k , informa a distribuição de probabilidade para a característica \vec{f}_k quando o estado da natureza é ω_j , refletindo assim a diferença de valores observados de \vec{f}_k para distintas categorias ω_j . Interpreta-se a *priori* como o nosso conhecimento prévio sobre as probabilidades dos estados da natureza ω_j . A regra de Bayes viabiliza a assimilação da nova informação $p(\vec{f}_k | \omega_j)$, obtida pela observação do vetor de características \vec{f}_k , ao conhecimento anterior sobre os estados da natureza dados por $P(\omega_j)$, informando a densidade *a posteriori* $p(\omega_j | \vec{f}_k)$, que é a probabilidade do estado da natureza ser ω_j , dada a observação do vetor \vec{f}_k . Com este resultado, a Eq. 2.35 pode ser reescrita na forma de uma razão entre as verossimilhanças:

$$\frac{p(\vec{f}_k | \omega_{2p})}{p(\vec{f}_k | \omega_{2n})} > \frac{\lambda_{12} - \lambda_{22}}{\lambda_{21} - \lambda_{11}} \frac{P(\omega_{2n})}{P(\omega_{2p})} \quad (2.39)$$

e, para uma função custo *zero-um*, dada por:

$$\lambda(\alpha_i, \omega_j) = \begin{cases} 0 & \text{se } i = j \\ 1 & \text{se } i \neq j \end{cases} \quad i, j \in \{\omega_{2p}, \omega_{2n}\} \quad (2.40)$$

a Eq. 2.39 se torna a Regra de Decisão de Bayes:

$$\begin{array}{ll} \text{Se } p(\vec{f}_k | \omega_{2p}) p(\omega_{2p}) > p(\vec{f}_k | \omega_{2n}) p(\omega_{2n}) & \text{escolha } \omega_{2p} \\ \text{caso contrário} & \text{escolha } \omega_{2n} \end{array} \quad (2.41)$$

Do ponto de vista probabilístico, o classificador *Bayesiano* é ótimo por garantir o risco mínimo em suas decisões, combinando toda a informação disponível sobre o problema em questão, por meio da Regra de Bayes. Claramente, esta abordagem nos garante o risco mínimo, haja vista que:

$$P(error | \vec{f}_k) = \min[P(\omega_{2p} | \vec{f}_k), P(\omega_{2n} | \vec{f}_k)] \quad (2.42)$$

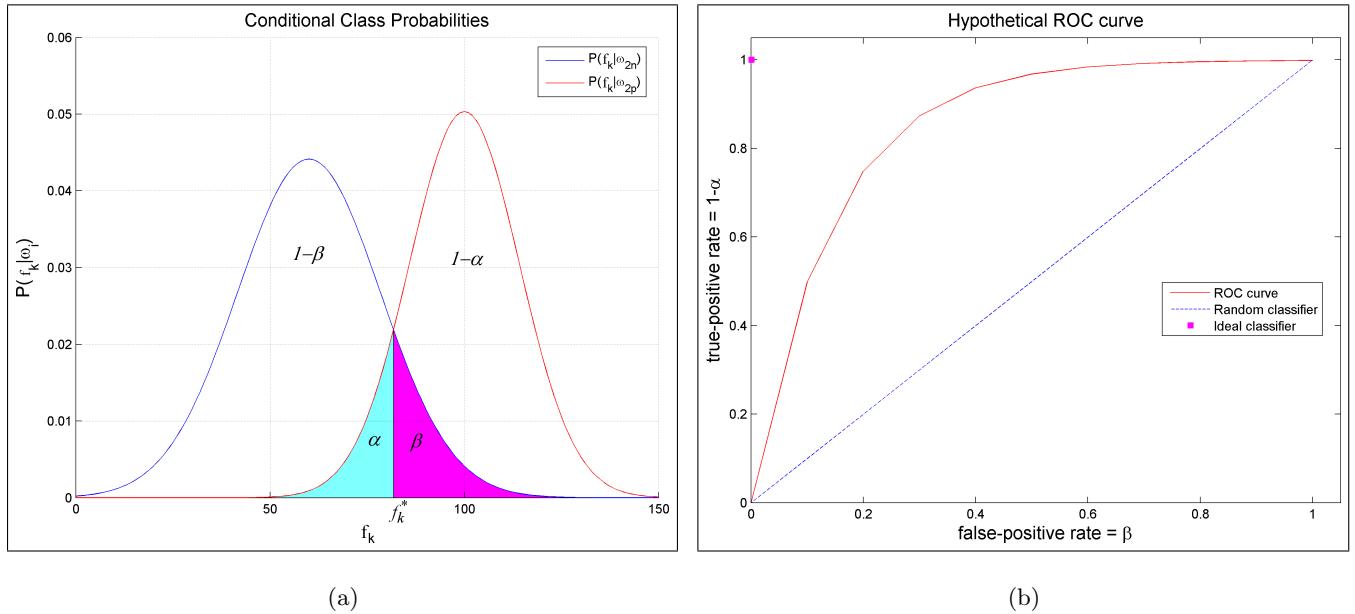


Figura 2.5: (a) Distribuições hipotéticas para o caso de apenas uma característica f_k e duas classes de segmentações: ω_{2p} e ω_{2n} . Decidimos, com probabilidade de sucesso $1 - \alpha$, pela classe ω_{2p} , se o valor observado cai à direita do limiar f_k^* , com a probabilidade de alarme falso associada dada por β ; a área sombreada em cor *magenta*. Analogamente, decidimos, com probabilidade de sucesso $1 - \beta$, pela classe ω_{2n} , se o valor observado cai à esquerda do limiar, com a probabilidade de alarme falso associada dada por α ; a área sombreada em cor *cyan*. (b) Representação de uma típica curva ROC, em que cada ponto (TP, FP) representa o desempenho de um classificador correspondente a um critério de decisão, isto é, a um limiar diferente.

Considere a Fig. 2.5(a) meramente ilustrativa, em que as verossimilhanças hipotéticas das classes ω_{2p} e ω_{2n} em relação a apenas uma característica f_k (e não um vetor de características) são representadas. Observe que as *verossimilhanças* sobrepõem-se. Dada uma observação para a característica f_k , adotamos a estratégia Bayesiana de decisão (Eq. 2.39), escolhendo a classe ω_{2p} , com probabilidade de sucesso igual a $1 - \alpha$, caso o valor observado caia à direita do limiar f_k^* , ou decidindo pela classe ω_{2n} , com probabilidade de sucesso igual a $1 - \beta$, caso contrário. A cada decisão, tem-se os erros associados $\beta = \int_{f_k^*}^{+\infty} p(f_k|\omega_{2n}) df_k$ e $\alpha = \int_{-\infty}^{f_k^*} p(f_k|\omega_{2p}) df_k$, respectivamente.

Em se tratando de Classificação Supervisionada, os rótulos das instâncias $\vec{f}_j \in F$ são conhecidos, podendo ser ω_{2p} (positivo), ou ω_{2n} (negativo). Os tais são ditos *rótulos verdadeiros* de classe, e denominados *positivo* e *negativo*, usualmente denotados por $\{p, n\}$ nos casos de classificação com 2 classes. Os *rótulos previstos* pelo classificador, por sua vez, são denominados $\{Y, N\}$, uma notação adotada para diferenciar rótulos verdadeiros de rótulos previstos [33]. Deste modo, um classificador nada mais é que um mapeamento $C : F \rightarrow \{Y, N\}$, ou seja, uma transformação C definida sobre o espaço de características, que associa um rótulo previsto pelo modelo de classificação a cada instância do espaço de características. Considerando a definição de probabilidade em termos de freqüência relativa [23], note que a probabilidade de sucesso ($1 - \alpha$) do classificador (Fig. 2.5(a))

Tabela 2.4: Tabela de Contingência - Matriz de Confusão

		Classe Real	
		Positivo (p)	Negativo (n)
Classe prevista	Positivo (Y)	Verdadeiros Positivos (TP)	Falsos Positivos (FP)
	Negativo (N)	Falsos Negativos (FN)	Verdadeiros Negativos (TN)

pode ser estimada como a fração de verdadeiros positivos ³, e a probabilidade de alarme falso (β) também pode ser estimada como a fração de falsos positivos ⁴. Estas proporções são métricas de desempenho de classificadores, obtidas a partir da tabela de contingência ⁵ em Teoria de Detecção de Sinais [27], fundamentais para a construção das chamadas curvas características de operação do receptor (*ROC curves*) ⁶.

Curvas ROC (Fig. 2.5(b)) são gráficos utilizados para visualizar, organizar e selecionar classificadores segundo seu desempenho, extensamente utilizados em Teoria de Detecção de Sinais e Diagnósticos Médicos, como meio de representar o compromisso entre taxas de sucesso e taxas de alarme falso em resultados obtidos por classificadores [33]. Variando o limiar f_k^* no gráfico das verossimilhanças (Fig. 2.5(a)), obtemos novos pares de pontos (TP, FP) para a curva ROC, tipificada em vermelho na Fig. 2.5(b), em cujo eixo das abscissas figura a taxa de falsos positivos FPR, e no eixo das ordenadas a taxa de verdadeiros positivos (TPR), para as quais valem as relações de equivalência [29, 100]:

$$\begin{aligned} FPR &\equiv (\beta) \equiv P(f_k > f_k^* | f_k \in \omega_{2n}) \\ TPR &\equiv (1 - \alpha) \equiv P(f_k > f_k^* | f_k \in \omega_{2p}) \end{aligned} . \quad (2.43)$$

Importante ressaltar que a Fig. 2.5(a) ilustra as verossimilhanças das duas classes em relação a uma característica apenas; no nosso caso, temos um vetor composto por 8 características. Vale lembrar ainda, que o limiar f_k^* sobre as verossimilhanças só mantém uma relação unívoca com o limiar θ sobre as razões entre verossimilhanças, desde que as verossimilhanças apresentem a mesma variância para ambas as classes. Do contrário, esta relação unívoca não se mantém, e para um limiar θ sobre as razões entre verossimilhanças haverá vários limiares f_k^* definindo **intervalos**, ou regiões intercaladas de classificação na abscissa. Como se vê no 2º membro da Eq. 2.39, a variação destes limiares pode ser devida a variações dos custos de classificações incorretas ou às probabilidades *a priori* [103]. Além disso, quando as variâncias das funções densidade de probabilidade de classe são diferentes, as curvas *ROC* não são convexas, nem simétricas [100, 29]. Sendo a curva *ROC* um gráfico bidimensional, faz-se necessária a definição de uma medida de desempenho de classificadores,

³TPR: True Positive Rate

⁴FPR: False Positive Rate

⁵ou matriz de confusão

⁶Receiver Operating Characteristic curves

Tabela 2.5: Análise da Capacidade Individual de Classificação das Características. Valores aproximados.

AUC - Área Sob a Curva ROC							
área	perímetro	circularidade	2º momento	entropia	curvatura	$\nu_{mediana}$	ν_{global}
0,49	0,49	0,49	0,7	0,7	0,7	0,7	0,49

usando as curvas *ROC*, que sejam invariantes com os critérios de seleção, probabilidades *a priori* e que incluam análise de custo/benefício [11], como um valor escalar que represente o desempenho esperado [33]. A medida que atende a estas necessidades e tem sido comumente utilizada é a denominada **AUC** (Área sob a Curva) ⁷.

Apesar de apenas a curvatura exibir poder discriminante entre as classes ω_{2p} e ω_{2n} , consoante a análise na Seção 2.4.1.3, queremos avaliar a habilidade de classificação das características, e para tanto calculamos a Área sob a Curva (AUC) das curvas *ROC* para cada uma das características estudadas. Os resultados obtidos para a área sob as curvas ROC para as 8 características individualmente são apresentados na Tab. 2.5.

Observamos que os descritores geométricos Área, Perímetro e Circularidade, além da Dimensão de Correlação Global, apresentaram $AUC = 0,49$ em todos os casos, denotando essencialmente aleatoriedade, portanto incapacidade de classificação. As características derivadas da transformada *Wavelet*, isto é, o Segundo Momento Estatístico da Norma da Transformada, a Entropia de Orientação da Fase da Transformada e a Curvatura, assim como a Dimensão de Correlação Mediana demonstram alguma capacidade de classificação, apresentando $AUC \approx 0,7$. Este resultado sugere que estas características são mais sensíveis aos padrões ramificados dos vasos sanguíneos na retina. Note que esta análise considerou as características isoladamente, no entanto o uso da combinação de características com um classificador apropriado pode resultar numa capacidade preditiva superior [39]. Na próxima seção, descrevemos a análise mediante a combinação de características.

2.4.2.3 Curvas ROC para avaliar o poder discriminante de conjuntos de características entre segmentações com PDR e sem PDR

Na Seção 2.4.2.2, as características são avaliadas individualmente quanto a sua capacidade de classificação, isto é, são tratadas como escalares. Uma vantagem naquela análise é o custo computacional reduzido, por outro lado as características como escalares podem não ser efetivas para analisar problemas complexos e para os casos em que há características com alta correlação mútua [100]. Além disso, é possível que características com baixo poder discriminante sejam mais eficazes quando combinadas [39]. Na presente seção, analisamos a capacidade de classificação de vetores de características, o que obviamente encarece a análise, pois em nosso caso a dimensão destes vetores pode variar entre 1 e 8. Temos aqui, um problema de seleção de subconjuntos de características [39] que, sob o prisma de otimização, deveria ser tratado com a análise de todas as combinações de carac-

⁷AUC - Area Under the Curve

terísticas em questão [100], em outras palavras, os vetores de características deveriam ser formados por todas as k -combinações de $|F| = 8$ características, o que configura claramente um problema de combinatória de complexidade fatorial. Para contornar este problema de complexidade computacional em seleção de características, surgiram algumas técnicas ótimas e subótimas de busca, capazes de evitar a enumeração exaustiva de todos os subconjuntos. Por outro lado, não garantem a seleção do melhor subconjunto [39]. Neste trabalho foram empregadas duas destas técnicas, a saber Seleção Seqüencial Regressiva e Seleção Seqüencial Progressiva de características. Os algoritmos [39, 100] simplificados dos métodos de busca mencionados, são descritos brevemente a seguir:

Seleção Seqüencial Regressiva⁸: Inicialmente, adotamos um critério $J_i(\cdot)$ de separabilidade entre classes, e denotamos seu valor para um subconjunto, por exemplo $\{f_2, f_5\} \subset F = \{f_1, \dots, f_8\}$, por $J_2(\{f_2, f_5\})$, onde o índice 2 designa a dimensão do subconjunto. Neste trabalho, o critério de separabilidade adotado corresponde à AUC medida para a curva ROC obtida para cada classificação. A partir do conjunto F , **eliminamos** uma característica, obtendo todas as suas $\binom{8}{7}$ combinações, e calculamos $J_7(\cdot)$ para cada um destes subconjuntos. O subconjunto que apresentar melhor $J_7(\cdot)$, dentre os $\binom{8}{7}$ subconjuntos de F , é selecionado como melhor solução de dimensão 7, e deste obtemos os $\binom{7}{6}$ subconjuntos de dimensão 6, para os quais calculamos $J_6(\cdot)$ e elegemos a melhor solução de dimensão 6. Este processo é repetido recursivamente, **decrementando** a dimensão dos subconjuntos, até que seja obtida a melhor solução de dimensão 2, dentre as $\binom{3}{2}$ soluções possíveis. Tal procedimento é dito subótimo, pois não garante que a solução ótima de dimensão i tenha necessariamente se originado da solução ótima de dimensão $(i + 1)$ [39].

Seleção Seqüencial Progressiva⁹: Analogamente ao *SBS*, obtemos os valores $J_i(\cdot)$ de subconjuntos de características, mas agora começando de conjuntos de dimensão 1, isto é, contendo apenas uma característica; o subconjunto de dimensão 1 que apresentar melhor $J_1(\cdot)$ é selecionado como melhor solução de dimensão 1. A este conjunto, **acrescentamos** uma característica, obtendo 7 subconjuntos de dimensão 2, e calculamos $J_2(\cdot)$ para cada qual. O subconjunto que apresentar melhor $J_2(\cdot)$ é selecionado como melhor solução de dimensão 2. Este processo é repetido recursivamente, **incrementando** a dimensão dos subconjuntos, até que seja obtida a melhor solução de dimensão 7. O procedimento descrito é considerado subótimo, por descartar características com menor poder discriminante nas subsequentes combinações de características, o que não garante que características descartadas não poderiam apresentar maior eficácia de classificação em outras combinações [39].

Com os subespaços de dimensões reduzidas, passamos à etapa final de avaliação dos subespaços de características, mediante o cálculo da Área sob a Curva ROC, para dois classificadores, são eles:

- I. Análise Discriminante Linear, segundo a qual projetam-se os vetores N -dimensionais sobre a linha em que estes são melhor separados, na direção do vetor para o qual a Função Critério de Fisher é máxima, determinando a seguir o limiar de separação nesta linha de projeção.

⁸SBS - Sequential Backward Selection

⁹SFS - Sequential Forward Selection

II. Classificador Bayesiano Ingênuo, lançando mão da regra de Bayes, sob a hipótese de independência entre as características.

A seguir, descrevemos estas abordagens de classificação.

Análise Discriminante Linear¹⁰: Uma função discriminante é uma função que define uma superfície de decisão no espaço de características [9]. A família de funções discriminantes constituídas por combinações lineares das características em questão são ditas Funções Discriminantes Lineares [103] e são definidas como:

$$g(\vec{f}) = \vec{w}^T \vec{f} + w_0 = \sum_{i=1}^N w_i f_i + w_0 \quad (2.44)$$

onde \vec{f} são vetores de características N -dimensionais, \vec{w} é um vetor com parâmetros de ponderação ajustáveis [9, 103] e w_0 é um limiar de peso.

A Eq. 2.44 define um hiperplano com versor normal na direção de \vec{w} , a uma distância perpendicular $\frac{|w_0|}{|\vec{w}|}$ da origem. O valor de $g(\vec{f})$ corresponde à medida da distância perpendicular de \vec{f} ao hiperplano [103].

Tais funções são empregadas em Análise Discriminante Linear (LDA) com vistas à classificação de vetores de características multidimensionais; note que, em nosso caso, N corresponde às dimensões dos subespaços de características obtidos por SBS e SFS, constantes na Tab. 2.6.

Usualmente, aplicam-se técnicas de redução de dimensionalidade, antes da etapa de classificação, por quanto espaços de características multidimensionais podem tornar a tarefa de classificação mais complexa [9], embora também possam conter mais informações [9]. Para o problema de 2 classes, o Discriminante Linear de Fisher figura entre as técnicas de redução dimensionalidade mais utilizadas, segundo o qual busca-se a direção \vec{w} , na qual as projeções dos vetores de características de diferentes classes alcançam melhor separação [9, 103]. Para tanto, Fisher propôs o critério de maximização da razão entre o espalhamento intra-classe e o espalhamento inter-classes, conhecido como Função Critério de Fisher [103], que pode ser definida conceitualmente como:

$$J(w) = \frac{\text{distância quadrática entre médias}}{\text{variância}} \quad (2.45)$$

ou formalmente como [103]:

$$J(w) = \frac{w^T S_B w}{w^T S_w w} \quad (2.46)$$

onde, S_w e S_B são as matrizes de espalhamento intra-classes e entre-classes, respectivamente, dadas

¹⁰LDA - Linear Discriminant Analysis

por [9]:

$$S_w = \sum_{\vec{f}_k \in \omega_{2p}}^N (\vec{f}_k - \bar{f}^{\omega_{2p}})(\vec{f}_k - \bar{f}^{\omega_{2p}})^T + \sum_{\vec{f}_k \in \omega_{2n}}^N (\vec{f}_k - \bar{f}^{\omega_{2n}})(\vec{f}_k - \bar{f}^{\omega_{2n}})^T \quad (2.47)$$

$$S_B = (\bar{f}^{\omega_{2p}} - \bar{f}^{\omega_{2n}})(\bar{f}^{\omega_{2p}} - \bar{f}^{\omega_{2n}})^T \quad (2.48)$$

Obter w que maximiza a Eq. 2.46 significa apenas encontrar a direção que maximiza a separação entre a projeção das classes, e não uma regra de decisão, mapeando vetores multidimensionais em uma dimensão, o que pode facilitar a tarefa de classificação [103]. Então o problema de classificação reduz-se a determinar uma regra de alocação, encontrar o limiar w_0 , um ponto na direção ótima que separe os pontos projetados [29, 103], de tal maneira que possamos atribuir um rótulo ω_j ao vetor \vec{f} se [103]:

$$\vec{w}^T \vec{f} + w_0 > 0$$

Classificador Bayesiano Ingênuo Nos casos em que a dependência entre características é ignorada, adota-se, em geral, o classificador Bayesiano Ingênuo, que nada mais é que o Classificador Bayesiano descrito na Seção 2.4.2.2, para o qual vale a regra dada pela Eq. 2.41, com a suposição de independência condicional entre as características, dadas as classes [29, 105].

Assim, as verossimilhanças se escrevem:

$$p(\vec{f}_k | \omega_j) = p(f_{k,1}, f_{k,2}, \dots, f_{k,8} | \omega_j) = \prod_{i=1}^8 p(f_{k,i} | \omega_j), \quad \text{onde } \omega_j \in \{\omega_{2p}, \omega_{2n}\} \quad (2.49)$$

As 54 amostras do nosso conjunto de treinamento descritas em subespaços de características de dimensões reduzidas, obtidos pelas técnicas de seleção de características SBS e SFS, foram submetidas aos dois classificadores supramencionados. O desempenho de ambos os classificadores sobre os diversos subespaços de dimensões reduzidas foi avaliado mediante curvas ROC e as respectivas AUC calculadas. O desempenho dos classificadores, medido pelas AUC, foi usado como critério de separabilidade para seleção de características. A Tab. 2.6 sintetiza os principais resultados obtidos. Observe que as características tradicionais Área, Perímetro e Circularidade (linhas 3 e 4 - Tab. 2.6) não atingem poder de discriminação com significância estatística melhor do que uma classificação aleatória para nenhum dos dois classificadores testados, enquanto as características derivadas das transformadas *Wavelets* atingem, com significância estatística, poder de discriminação entre estruturas com e sem PDR para ambos os classificadores [56].

A Fig. 2.6 apresenta as curvas ROC ajustadas pelo modelo paramétrico binormal de Metz, obtidas para o classificador da LDA, sobre o subespaço com a melhor combinação de 6 características e sobre o subespaço com as 3 piores características (geométricas). Com o melhor 6-subespaço, 3

Tabela 2.6: Análise da Capacidade de Classificação de Subespaços de Características. Valores aproximados.

Subespaço	Dimensão	Seleção	Classificador	AUC	Intervalo de Confiança
$\nu_{mediana}$ e 2º momento	2	BFS-SBS	LDA	0,77	0,56-0,91
área, perímetro, entropia, curvatura, $\nu_{mediana}$ e ν_{global}	6	BFS-SBS	LDA	0,90	0,73-0,97
área, perímetro e circularidade	3	-	LDA	0,61	0,37-0,82
área, perímetro e circularidade	3	-	Bayes Ingênuo	0,58	0,37-0,78
área, 2º momento, entropia e ν_{global}	4	SFS-SBS	Bayes Ingênuo	0,87	0,67-0,96

imagens foram incorretamente classificadas, das quais, a primeira era de baixa qualidade (opaca), a segunda apresentava um laço venoso e a terceira era de boa qualidade, mas com neovascularização na periferia da retina e nas vizinhanças do disco óptico. Este classificador obteve a marca de 15 verdadeiros positivos das 16 imagens, sucesso de 94% (sensibilidade ou $TPR = 0,94$), e 2 falsos positivos de 11 imagens, resultando uma especificidade de 82% (ou $FPR = 0,18$).

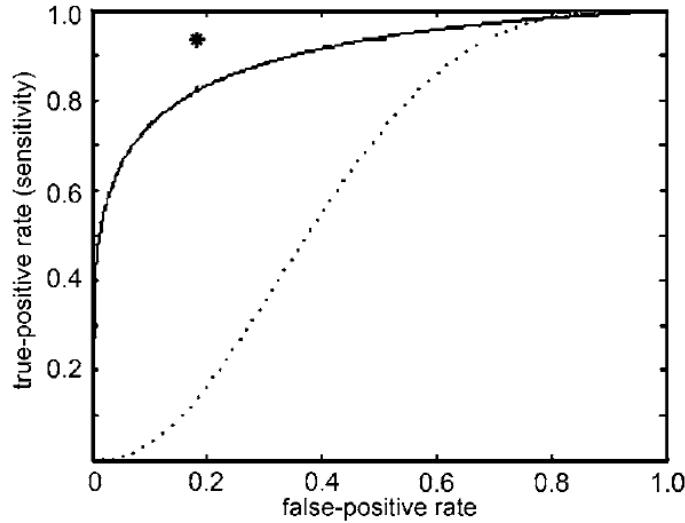


Figura 2.6: Curva ROC obtida para imagens segmentadas automaticamente usando um subespaço composto pelas 6 melhores características (curva sólida), comparada ao subespaço composto por área, perímetro e circularidade (curva pontilhada) e ao ponto de operação (asterisco).

Capítulo 3

Análise de Formas de Estruturas de Ramificação - Parte *II* - Neurônios

3.1 Motivação

Neurônios são células organizadas em caminhos sinalizadores, que se comunicam mediante transmissões sinápticas nas interfaces entre ramificações de dendritos e axônios, constituindo unidades sinalizadoras do cérebro (e.g. [57]). Estímulos e alterações biológicas estabelecem uma dinâmica de constante rearranjo das interconexões que caracterizam as redes neurais biológicas. Em grande parte, a quantidade de conexões que um neurônio pode receber está diretamente relacionada à forma de suas árvores dendríticas, que servem como alvos para o crescimento de axônios. Por esta razão, a compreensão do funcionamento dos circuitos neuronais está intimamente relacionada com a adequada caracterização da forma dos neurônios que os compõem (e.g. [24, 26, 86]). Ademais, são poucos os sistemas biológicos tão intrigantes quanto o sistema nervoso, haja vista a relação entre o fenótipo e o genótipo durante o desenvolvimento animal e a influência tanto do conteúdo genético quanto de estímulos e mutações causadas por alterações bioquímicas sobre a forma de células neuronais. Some-se a isto o fato de a morfologia neuronal ter se tornado um recurso crítico para a pesquisa em neurociência e uma relevante ferramenta utilizada para diagnósticos. Conseqüentemente, somente mediante a correta representação e caracterização das formas dos neurônios, que a relação entre fenótipo neuronal, genética, filogenética, neurologia comparativa e desenvolvimento animal tornar-se-á compreensível [86].

A análise de formas de células neuronais, como área de pesquisa, estabeleceu-se recentemente. Abordagens as mais variadas baseiam-se na caracterização geométrica e na conectividade entre células neuronais [26, 54, 16, 77, 86]. Particularmente, algumas das medidas mais efetivas requerem a extração do contorno de células. Por exemplo, a extração adequada do contorno de imagens 2D de neurônios fornece sinais paramétricos, dos quais características podem ser calculadas. Estas características podem sugerir diferenças entre as formas dos neurônios. Neurônios com formas distintas

tendem a pertencer a diferentes classes, com diferentes funções [53]. Nossa sistema contribuiria para automatizar a tarefa de classificar imagens 2D de neurônios. Não obstante a importância do problema de contornar ramos emaranhados em imagens bidimensionais, não pudemos encontrar quaisquer abordagens relacionadas ao problema na literatura.

Pelo que é de nosso conhecimento, o único trabalho similar descrito na literatura é [87, 88], que apresenta um método semi-automático para separar veias e artérias em árvores vasculares de imagens de retina. A rotulação de vasos, que também são estruturas ramificadas, é uma tarefa crucial para a resolução deste problema. No método de Rothaus *et al.*, rótulos manualmente atribuídos a alguns vasos inicialmente são propagados por todo o grafo vascular G , uma representação do esqueleto da imagem. Tais rótulos definem um conjunto de vínculos, que são representados por um grafo dual G' . Em cada região crítica (*bifurcação/cruzamento*), uma fórmula booleana sintetiza as relações de vínculo existentes entre *segmentos de ramos* adjacentes. Tal formulação permite que o problema seja interpretado como um problema de satisfabilidade, passível de solução por uma versão adaptada do algoritmo **AC-3**. À medida que os rótulos são propagados, contradições podem ocorrer, ocasionando conflitos. Neste caso, os vínculos entre segmentos de vasos são revistos num procedimento de *backtracking*, e G' é localmente manipulado por meio de uma operação sobre grafos, dentre três previamente definidas. Assim, o algoritmo adaptado **AC-3** provê uma rotulação consistente para todo o grafo. A despeito da existência de algumas semelhanças entre as abordagens propostas (principalmente o uso de esqueletos e a análise de casos de bifurcações e cruzamentos), a metodologia aqui introduzida tem uma abordagem diferente. O sistema de Rothaus *et al.* faz análises globais, sempre que um conflito acontece, rearranjando todo o grafo em um procedimento de *backtracking*, enquanto nosso método executa somente análises locais da topologia da imagem de maneira seqüencial, sem recorrer a representações por grafos, tampouco a *backtracking*. Como o sistema deles foi especialmente projetado para rotular árvores vasculares, o número de rótulos disponível é limitado a dois: veias e artérias. Ao contrário, nossos algoritmos produzem uma atribuição de um rótulo distinto para cada ramo existente e um rótulo distinto para cada árvore dendrítica, independente do número de árvores dendríticas existentes no neurônio (ver Fig. 3.16). Além disso, ao segmentar cada ramo do neurônio, nosso método permite análises estatísticas como contagem de ramos e medição do comprimento da ramos. Ainda, nosso sistema provê finalmente o contorno paramétrico de toda a estrutura, com o que é possível extrair características a serem usadas por um classificador.

Em geral, uma imagem é obtida por algum dispositivo óptico de imageamento, como uma câmera fotográfica, correspondendo a uma representação incompleta da informação originalmente presente no objeto real [17]. Por esta razão, as imagens são comumente privadas de alguma informação, como a informação relativa à profundidade, um problema oriundo da supressão da terceira dimensão do objeto real 3D ocasionada pela projeção no plano 2D. No que tange às *imagens com formas complexas*, tais como neurônios, a informação de profundidade é crucial para o entendimento das estruturas na imagem. As formas bidimensionais presentes nestas imagens podem ser representadas de diversas

maneiras, dentre as quais destacam-se três tipos, a saber, representação (i) por Transformadas, (ii) por Regiões e (iii) por Contornos. Cada uma destas representações oferece vantagens para extração de características específicas.

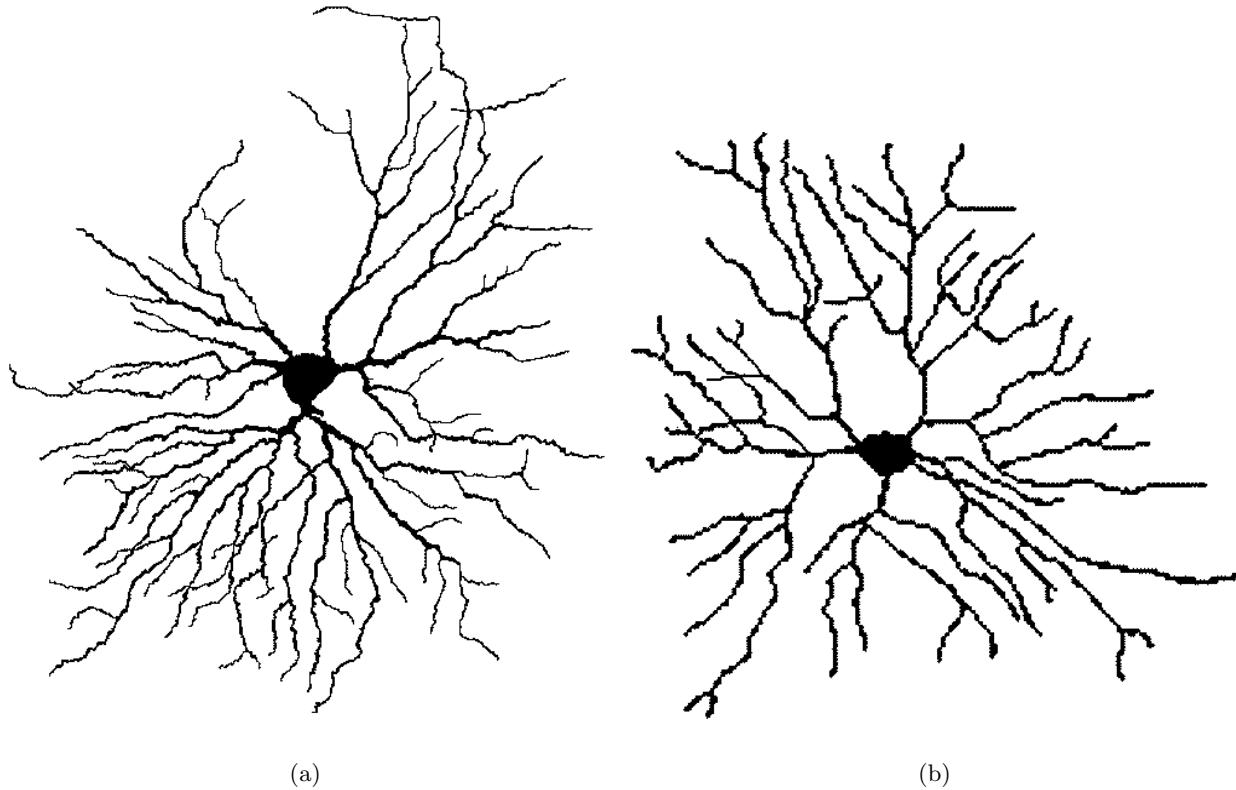


Figura 3.1: Exemplos de imagens 2D de neurônios, obtidos por *camera lucida*, considerados neste trabalho. Neurônio tipo α (a). Neurônio tipo Y (b).

Usualmente, neurônios tridimensionais são projetados no espaço bidimensional (e.g. camera lucida), assim o contorno das células pode ser representado como curvas paramétricas unidimensionais [48] (Fig. 3.1). Contornos, por sua vez, podem ser obtidos como (iii – a) aproximações de curvas (ex: splines, snakes, etc), (iii – b) conjunto de pontos (ex: pontos do gradiente morfológico) ou (iii – c) contorno paramétrico [25]. Sempre que um contorno [48] obtido a partir de uma *forma* puder ser interpretado como uma curva paramétrica, informações importantes, como a orientação dos campos tangente e normal ao longo dos contornos podem ser obtidas, permitindo a estimativa da curvatura do contorno (e.g. [19, 26]), que é conhecida por trazer informações particularmente valiosas sobre a forma do neurônio, tais como sua energia de dobramento e concavidade. Outra aplicação possível em Neurociência e o uso do contorno paramétrico de neurônios para obter automaticamente dendrogramas neurais [20].

Uma falha típica na extração de contornos paramétricos de estruturas de ramificação consiste

3	2	1
4	C	8
5	6	7

Figura 3.2: Vizinhança definida pela regra da cadeia e usada pelos algoritmos *BTA* e *BSCEA*

no fato de que o algoritmo de extração de contornos baseado no código da cadeia [37] (Fig.3.2) não consegue atravessar regiões delimitadas por cruzamentos (ver Fig.3.3), (devido à projeção do espaço 3D para o espaço 2D). Como resultado, somente o contorno mais externo da célula pode ser obtido, enquanto estruturas mais internas permanecem inacessíveis. Este fato é ilustrado na Fig. 3.3, em que regiões acinzentadas representam áreas fora do alcance dos algoritmos tradicionais de extração de contorno, produzindo somente a curva em vermelho como contorno.

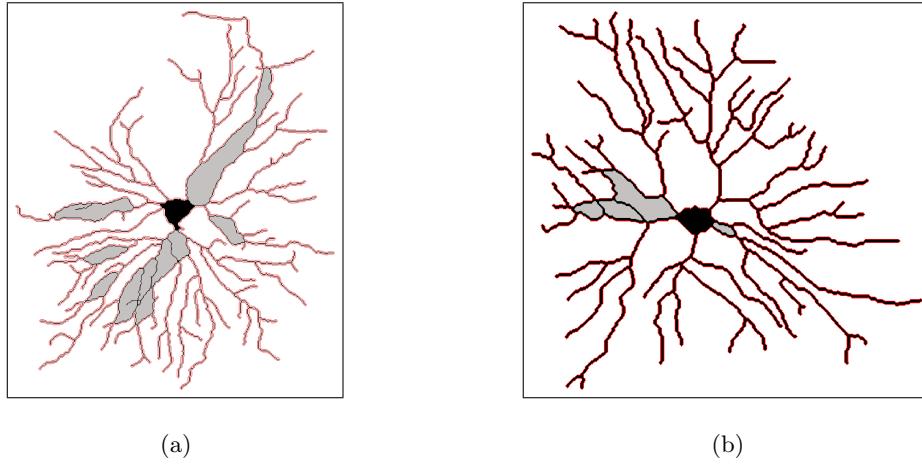


Figura 3.3: Esqueletos dos neurônios (preto) e respectivos contornos (vermelho). Áreas sombreadas representam regiões inacessíveis para o algoritmo de extração de contornos baseado no código da cadeia.

Qualquer característica global calculada sobre um contorno incompleto fornecerá informação imprecisa. Considerando o exposto, algoritmos de extração de contorno baseados no código da cadeia não conseguem lidar com curvas que não são de Jordan [14], isto é, curvas com autointersecção, o que os torna incapazes de acompanhar ramos emaranhados. Até a introdução do presente método, este problema encontrava-se aberto. Na Fig. 3.5 (adaptada da tese [18]), o autor ilustra o caso típico de uma intersecção numa estrutura de ramificação e sua influência no problema de extração de contorno. Neste exemplo, as estruturas de ramificação foram editadas manualmente, a fim de obter contornos mais próximos da formas originais.

Em resumo, algoritmos de extração de contorno baseados no código da cadeia funcionam como

descrito a seguir. Inicialmente, o algoritmo detecta um primeiro pixel pertencente ao contorno externo do objeto. Normalmente este pixel é obtido por uma varredura *raster* da imagem, até que um pixel de fundo, que também é vizinho de um pixel objeto, seja encontrado. Então, o algoritmo procura o próximo pixel de contorno, sondando a vizinhança do pixel de contorno atual, de acordo com a ordem definida pelo código da cadeia. Esta etapa é executada a cada iteração, sobrepondo a máscara ilustrada na Fig. 3.2 ao pixel atual e visitando seus vizinhos na ordem 1, 2, ⋯, 8, de maneira a decidir qual candidato deveria ser escolhido como o próximo pixel de contorno. Do segundo pixel em diante, o algoritmo leva em conta a direção do pixel de contorno anterior ao pixel atual, para definir o ponto de partida do próximo procedimento de sondagem. Assim fazendo, o algoritmo contorna todo o objeto, até revisitar o primeiro pixel, quando a tarefa terá sido completada. Mais detalhes são documentados numa descrição completa em [25].

Poder-se-ia alegar que as sobreposições causadas pela projeção de estruturas complexas tridimensionais no plano bidimensional são passíveis de solução durante a etapa de aquisição, usando, para tanto, o microscópio confocal inventado na década de 50 por Marvin Minsky. Este microscópio, criado para visualizar a interconexão entre células do sistema nervoso humano, é capaz de iluminar amostras ponto-a-ponto, minimizando o espalhamento da luz oriunda de regiões fora do plano de interesse, além de rejeitar feixes de luz fora de foco, mediante o uso de um orifício confocal [82]. Isto lhe permite varrer finas seções de uma amostra para posterior reconstrução 3D da mesma. De fato, a microscopia confocal viabiliza a obtenção de dados 3D de células neurais. No entanto, tais dados são usualmente fornecidos como apresentação volumétrica, enquanto reconstruções de superfície freqüentemente envolvem casos em que a continuação dos segmentos é ambígua, causadas por orientações e posições semelhantes. Há muitas situações em que a reconstrução automática da superfície é praticamente impossível a partir de dados confocais. Portanto, os resultados relatados neste trabalho também podem ser úteis como extensão para estes casos 3D. Todavia, existem ainda outros aspectos relevantes com respeito à importância e aplicabilidade de nossa contribuição. Em primeiro lugar, há dezenas de outras técnicas de microscopia que não conseguem produzir imagens 3D, mas somente 2D, causando necessariamente a sobreposição de ramos neurais que podem ser tratados pelo nosso método. Estas técnicas de microscopia são normalmente necessárias, em vez da microscopia confocal, por serem capazes de revelar propriedades específicas dos tecidos e estruturas analisadas, que não poderiam ser capturadas pelo método confocal. Em segundo lugar, o método relatado pode ser aplicado a diversos outros problemas, considerando a ubiqüidade das estruturas de ramificação na natureza, principalmente problemas médicos envolvendo segmentos emaranhados em imagens 2D. Uma das mais importantes situações é a representação e análise da vascularização na retina, que necessariamente implica na sobreposição de artérias e veias. Esta sobreposição acontece em inúmeros métodos de aquisição de imagem de sistemas vasculares de retina, tais como fotografias não-midriáticas, fotografias *red-free* e angiogramas fluorescentes. Pela mesma razão que a análise de neurônios é importante para sua classificação apropriada, a inspeção de vasos da retina, auxiliada por computadores, é importante para a detecção de diversas patologias e diagnóstico da

diabetes, doenças cardiovasculares, hipertensão, derrame e arterioesclerose [94]. Os algoritmos aqui propostos também podem ser aplicados a imagens de retinopatia (Fig. 4.1), para caracterização daquelas, conforme as considerações constantes no Capítulo 2 desta dissertação.

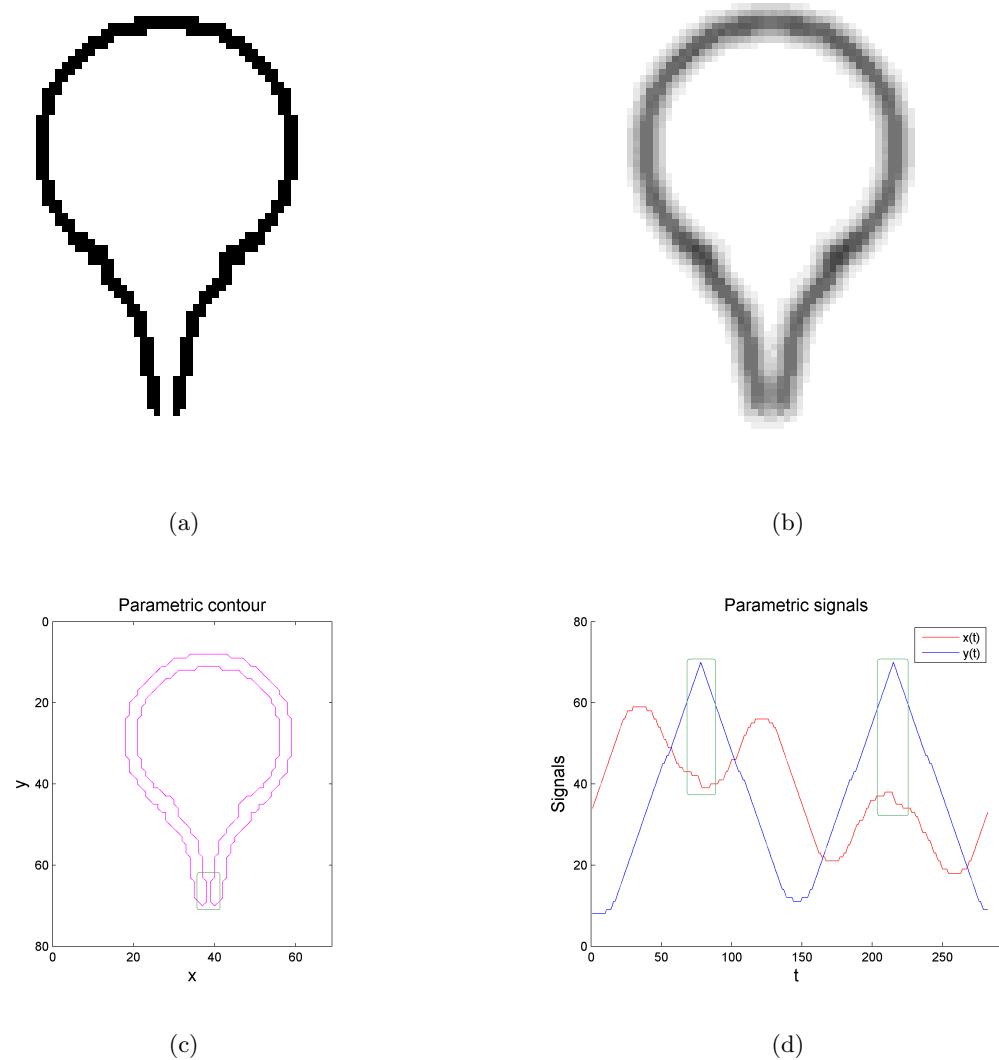


Figura 3.4: Diferenças entre o contorno obtido por derivação e o contorno paramétrico. (a) Estruturas próximas na forma original (b) afetam-se mutuamente na etapa de suavização para obtenção do contorno usando a Wavelet Hermitiana. (c) Por outro lado, as estruturas que se apresentam próximas na representação do contorno (elipse verde), (d) aparecem distantes na representação por sinais paramétricos ($x(t)$ e $y(t)$).

Vale ressaltar que na Seção 2.3.4 obtivemos o contorno de uma forma a partir da transformada wavelet, usando como wavelet analisadora a 1^a wavelet *Hermitiana*; o resultado foi o gradiente da forma original suavizada por uma gaussiana. Neste capítulo, apresentamos uma abordagem diferente daquela, isto é, a extração do contorno paramétrico da forma. A Figura 3.4 ilustra uma

diferença entre as abordagens de extração de contorno por transformada wavelet e por contorno paramétrico. Note que dois pontos próximos na imagem (Figs. 3.4(a) e 3.4(b)) podem estar distantes parametricamente (Figs. 3.4(c) e 3.4(d)). A proximidade entre estruturas na forma original pode causar influência mútua na etapa de suavização para obtenção do contorno usando a Wavelet Hermitiana, dependendo da dispersão da gaussiana, como se vê na Fig. 3.4(b), por outro lado as mesmas estruturas aparecem distantes na representação por contorno paramétrico, como ilustrado na Fig. 3.4(d). Esta diferença entre as abordagens se deve ao fato de o suporte da transformada Wavelet ser a própria imagem, enquanto o suporte do contorno paramétrico $c(t) = (x(t), y(t))$ é o parâmetro t .

Este capítulo visa a apresentar uma solução desenvolvida para este problema, de tal maneira que representações paramétricas de estruturas de ramificação possam ser analisadas. Assim como a maioria dos métodos computacionais, o método proposto envolve a combinação efetiva de diversas técnicas previamente conhecidas. Dentre os diversos critérios incorporados, destacamos particularmente o uso da similaridade ao longo da orientação da tangente como meio para identificar a continuação apropriada para cada processo neuronal, quando estes atravessam regiões críticas. O princípio fundamental da continuação da tangente, assim como a combinação de métodos definem aqui um novo método. Os algoritmos introduzidos representam algumas das principais e originais contribuições deste trabalho, são eles: Algoritmo de Rastreamento de Ramos ou *Branches Tracking Algorithm (BTA)* e o Algoritmo de Extração de Contorno de Estruturas de Ramificação ou *Branching Structures Contour Extraction Algorithm (BSCEA)*. Em resumo, o *BTA* é um algoritmo projetado para segmentar cada ramo distinto de uma imagem 2D de neurônio, além do soma e regiões de intersecção. Por sua vez, o *BSCEA* é um algoritmo com o objetivo de extrair o contorno paramétrico de uma imagem 2D de neurônio, tendo a saída do *BTA* como sua entrada.

Privilegiando o bom entendimento, este capítulo foi estruturado em níveis crescentes de detalhes, desenvolvendo-se como segue. A Seção 3.3 contém uma visão global do sistema proposto, que é mais detalhada na Seção 3.4. Os resultados experimentais são apresentados na Seção 3.5. O texto é concluído no Capítulo 4, identificando as principais contribuições, assim como possibilidades para trabalhos futuros. Descrições de mais baixo nível foram deixadas para o Apêndice G e pseudo-códigos para os Apêndices D, E e F.

3.2 Conceitos

3.2.1 Tipos de Células Ganglionares de Retina

Células Ganglionares de Retina (RGC) são tipos de células neuronais presentes na retina e responsáveis pela transferência da informação visual que chega para ser processada em diferentes regiões do cérebro. As RGC podem ser ainda subdivididas em grupos de tipos, de acordo com critérios morfológicos e fisiológicos [10]. Abordar o problema da relação entre forma (morfologia) e função (fisiologia) em RGC parte do pressuposto amplamente aceito de que a forma de células neuronais são de alguma maneira determinantes das respectivas funções. Uma hipótese alternativa seria a de

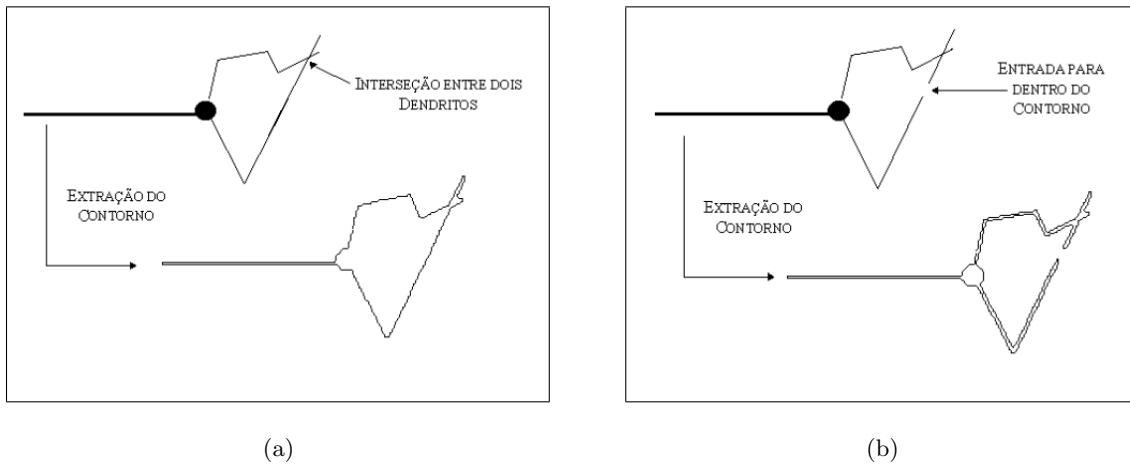


Figura 3.5: (a) Ilustração esquemática de uma auto-intersecção e o problema resultante na etapa extração do contorno; (b) O contorno editado permite que o algoritmo de extração de contorno gere um contorno mais significativo em relação à forma do neurônio original. Adaptada de [18]

que as formas de células neuronais não têm influência sobre suas funções, mas sim a maneira em que arranjos sinápticos entre células ocorrem [52]. A fim de entender melhor esta relação intrigante, a análise de formas de células neuronais, por meio de algoritmos computacionais, tem desempenhado um papel importante mais do que nunca [53].

Mesmo antes da invenção de procedimentos anatômicos para análise, as medidas eletrofisiológicas apontavam para a existência de dois tipos de células de retina, considerando seus campos receptivos concentricamente organizados: células X com uma pequena área de resposta linear e células Y com uma grande área de resposta não-linear. Pesquisas modernas baseadas em estratégias anatômicas revelam a existência de outros tipos de células ganglionares, principalmente levando em conta os níveis de ramificação e larguras de árvores dendríticas [73]. Todos os estudos executados até agora, à luz da análise de formas, têm consolidado a suposição de que células com morfologia estrutural distintas também apresentam funções fisiológicas diferentes [73]. As correlações entre células Y e α e células X e β foram descobertas há muito, com relação tanto às características morfológicas quanto fisiológicas, por gravações eletrofisiológicas e exames histológicos de células [10, 53]. Além disso, acredita-se atualmente que pode haver mais do que 23 tipos de células ganglionares na retina de gato, morfologicamente falando [52]. Normalmente, as RCC são visualmente classificadas com base no tamanho do corpo das células e na forma das árvores dendríticas. Grosso modo, células α são caracterizadas por corpos celulares grandes e campos dendríticos amplos, enquanto células ϵ possuem corpos celulares medianos e campos dendríticos amplos. Trabalhos anteriores corroboraram as semelhanças entre células α e δ [53]. Ainda, foram relatadas semelhanças entre as RGC de gatos e de furões [36].

3.2.2 Terminologia

Um neurônio pode ser descrito como uma composição de suas partes funcionais, i.e., **soma**, **dendritos** ou **árvores dendríticas** e **axônio**, conforme a *Doutrina do Neurônio* [84]. Para os objetivos deste trabalho, estendemos esta terminologia para a Análise de Formas, subdividindo as árvores dendríticas em partes constituintes como segue (veja Fig. 3.6 e Tab. 3.1):

- *Pontos*
- *Linhas*
- *Regiões Críticas*
- *Coleções*

A categoria *Pontos* é composta por três classes de pontos extremos: *sementes primárias*, *sementes secundárias* e *terminações*. Cada ponto extremo é classificado de acordo com sua localização, i.e. uma *semente primária* é localizada como um ponto de junção entre uma *árvore dendrítica* e o *soma*, enquanto uma *semente secundária* é localizada como um ponto de junção entre uma *região crítica* e uma *subárvore dendrítica*. *Terminações* são pontos de extremidade dos *ramos*.

A categoria *Linhas* é composta por duas classes de linhas: *segmentos* e *ramos*. Cada linha é classificada considerando seus pontos extremos, i.e. um *segmento* pode se originar de uma *semente primária* ou de uma *semente secundária*, mas não necessariamente possuindo uma *terminação*. Assim, os *segmentos* são linhas de pixels delimitados por um par de pequenas estruturas, por exemplo uma *semente* e uma *região crítica*, ou duas *regiões críticas* ou ainda uma *região crítica* e uma *terminação*. Ao contrário, um *ramo* pode se originar de uma *semente primária* ou *semente secundária*, tendo necessariamente uma *terminação*. Segue obviamente de tal definição, que um *ramo* é uma composição, feita de uma cadeia de *segmentos*, emergindo de uma *semente* e estendendo-se até uma *terminação*, como ilustrado na Fig. 3.6. Além disso, *Segmentos* ainda podem ser subdivididos dependendo de suas relações de orientação a uma *região crítica*. Analisando a forma de um neurônio de dentro para fora, isto é, do *soma* em direção a suas *terminações*, um *segmento* que chega a uma *região crítica* é dito *segmento incidente*, enquanto um *segmento* que sai de uma *região crítica* é dito *segmento emergente*.

A categoria *Regiões Críticas* é composta por aglomerados de pixels onde *ramos* se encontram. Esta categoria inclui três classes de regiões: *bifurcações*, *cruzamentos* e *sobreposições*. Cada *Região Crítica* é classificada levando em conta o número de *segmentos* a ela adjacentes e as mútuas relações de orientação, além da relação de proximidade entre a *região crítica* atual e alguma outra na vizinhança. Estas regras de classificação serão detalhadas mais adiante. Em uma **bifurcação**, um *segmento incidente* freqüentemente se divide em dois *segmentos emergentes* com diferentes orientações, como ilustrado na Fig. 3.7-(a-b). Ocasionalmente, podem surgir **bifurcações** muito próximas umas das outras. O exame destas regiões numa escala maior pode sugerir a existência de

apenas uma *região crítica*, em que um *segmento incidente* se divide em três *segmentos emergentes*, como mostrado na Fig.3.7-(c-d). Semelhantemente, numa **Sobreposição**, um *segmento incidente* se divide em três *segmentos emergentes*, dois dos quais em orientações normalmente distintas e opostas, como pode ser visto na Fig. 3.7-(e). Se as **sobreposições** não fossem consideradas, elas poderiam ser localmente confundidas como duas **bifurcações** muito próximas, ligadas por um *segmento* extremamente curto.

Por fim, um **Cruzamento** é um aglomerado de pixels onde um *segmento incidente* se divide em três *segmentos emergentes*, dois dos quais em orientações muito diferentes e necessariamente opostas, como representado na Fig.3.7-(f).

A categoria *Coleções* é constituída simplesmente pela união dos objetos supracitados. Uma *Árvore Dendrítica* é uma coleção de *ramos* oriundos do *soma*. Doravante, a coleção de *Árvores Dendríticas*, que é o neurônio sem o *soma*, será tratada por *Periferia*.

Estes conceitos encontram-se resumidos na Tab. 3.1.

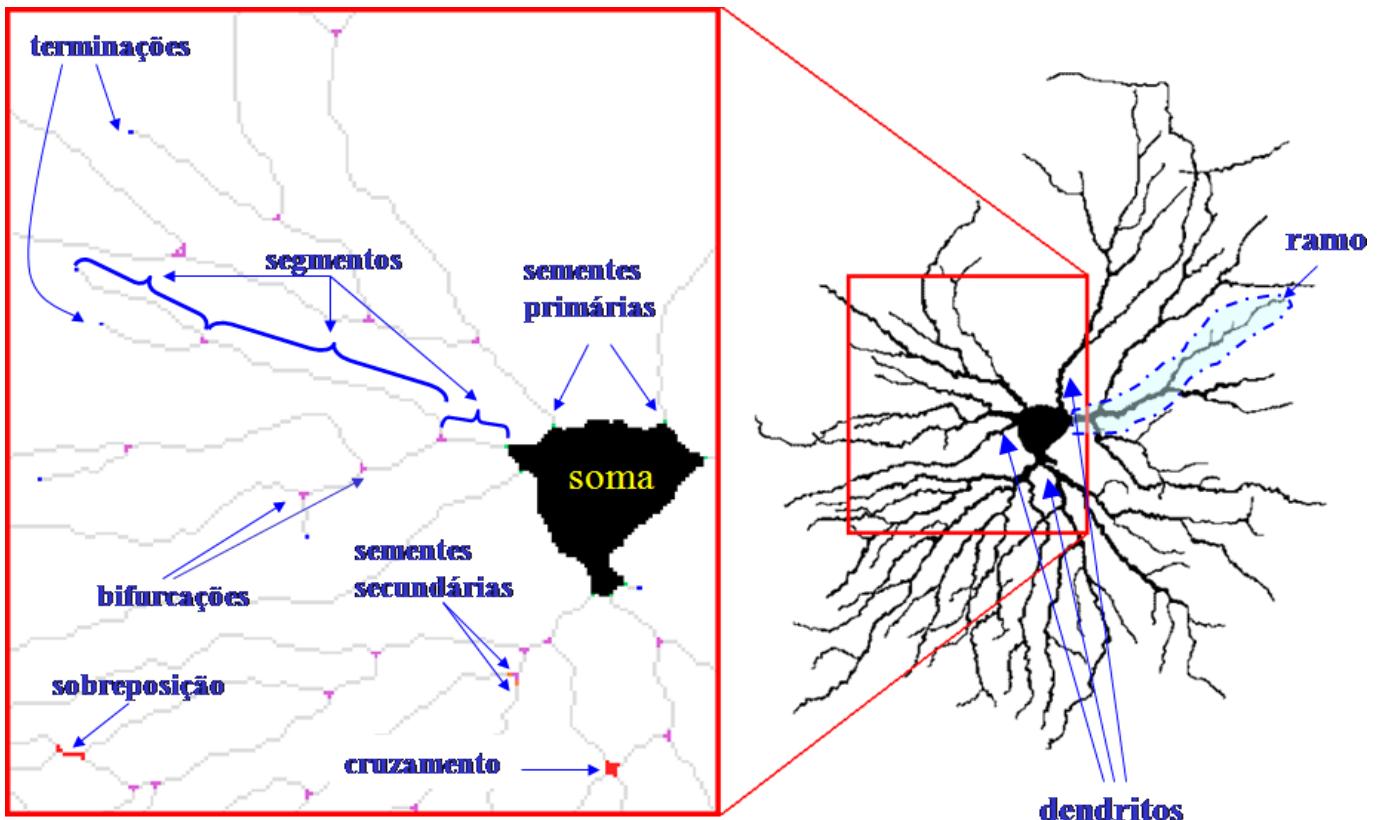


Figura 3.6: O detalhe ilustra a terminologia estendida adotada neste trabalho: *ramos*, *segmentos*, *sementes*, *terminações* e *regiões críticas*. Adaptada de [61,63]

3.3 Um panorama dos métodos

Em linhas gerais, o método proposto envolve as seguintes três etapas:

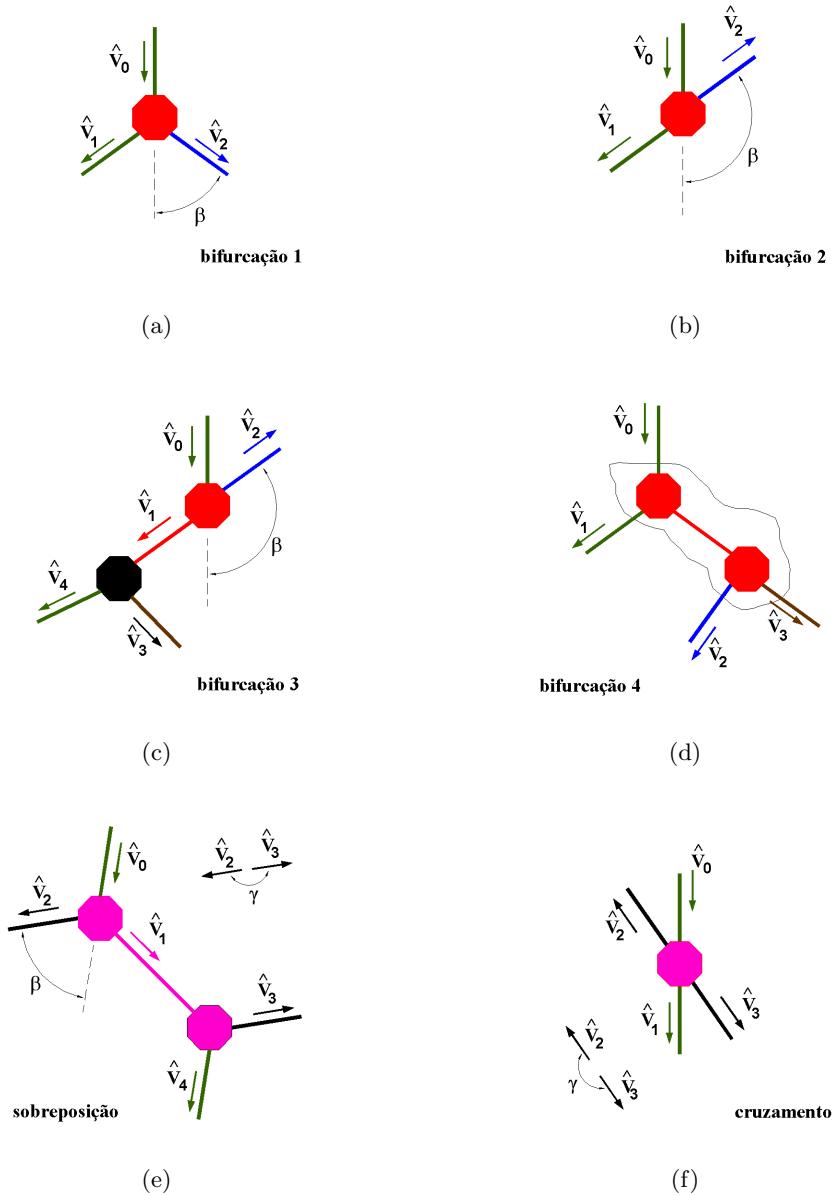


Figura 3.7: As regras de classificação de *regiões críticas* levam em conta o ângulo β entre o *vetor diretor incidente* - \hat{v}_0 - e o *vetor diretor emergente* - \hat{v}_2 , o ângulo γ entre quaisquer pares de *vetores diretores emergentes* e a cardinalidade $|E_1|$ do conjunto de *vetores diretores emergentes* E_1 relacionados à *região crítica* atual s_1 . (a) Se uma *região crítica* apresentar $|E_1| = 2$ e $\beta < 90^\circ$ ela é imediatamente classificada como uma *bifurcação 1*, (b) contudo se $\beta \geq 90^\circ$ ela pode ser uma *bifurcação 2* ou (c) mesmo uma *bifurcação 3*, contanto que haja outra *região crítica* nas vizinhanças. (d) Uma *bifurcação 4* apresenta $|E_1| = 3$, mas não há nenhum $\gamma \approx 180^\circ$. (e) Uma *sobreposição* aparece com $|E_1| = 2$ e outra *região crítica* s_2 nas vizinhanças, além de $\gamma \approx 180^\circ$. (f) Finalmente, um *cruzamento* tem $|E_1| = 3$ e $\gamma \approx 180^\circ$. Note que uma *bifurcação 1* poderia ser confundida com uma *bifurcação 4* ou uma *sobreposição* e uma *bifurcação 2* poderia ser confundida com uma *bifurcação 3* se todas as regras não forem consideradas.

Tabela 3.1: Resumo de Conceitos.

TERMO	DESCRIÇÃO
Pontos	
semente	primária
	secundária
terminação	ponto final de um ramo
Linhas	
segmento	linha de pixels delimitada por outras estruturas
segmento incidente	segmento que chega a uma região crítica
segmento emergente	segmento que sai de uma região crítica
ramo	cadeia de segmentos
Regiões Críticas (<i>CR</i>)	
bifurcação	aglomerado de pixels onde um segmento incidente se divide em dois; um dos quais em outra direção
cruzamento	aglomerado de pixels onde um segmento incidente se divide em três; dois dos quais em outra direção
sobreposição	aglomerado de bifurcações onde um segmento incidente se divide em três; dois dos quais em outra direção
Coleções	
árvore dendrítica	coleção de ramos oriundos de um soma
periferia	coleção de árvores dendríticas (excluindo o soma)
esqueleto	esqueleto de um pixel de largura da periferia

- **Algoritmo de Pré-processamento da Entrada :** a imagem de entrada é pré-processada mediante operações de morfologia matemática, produzindo como saída seus componentes separados:
 - *imagem do esqueleto da periferia*, doravante denominado apenas como *esqueleto*
 - *imagem das regiões críticas*
 - *imagem das terminações*
 - *imagem do soma*
 - *fila de sementes primárias*
- **Algoritmo de Rastreamento de Ramos (*Branch Tracking Algorithm - BTA*):** o *BTA* é aplicado para cada *semente primária* presente na *fila*. Ele inicia o processo de rotulação no

segmento adjacente à *semente primária*. Ao atingir uma *região crítica*, o *segmento* atual já foi totalmente rotulado, assim uma decisão deve ser tomada com respeito ao próximo *segmento* onde deve continuar o rastreamento. Além de encontrar o *segmento* ótimo para prosseguir, o algoritmo também identifica a *região crítica* atual como uma *bifurcação*, uma *sobreposição* ou um *cruzamento*. Se a *região crítica* atual for uma *bifurcação*, então o *BTA* armazena a respectiva *semente secundária* numa fila auxiliar, caso contrário o *BTA* armazena, em ponteiros, os endereços do último ponto do *segmento* atual e o primeiro ponto do próximo *segmento*. Assim fazendo, o *BTA* rotula todos os *segmentos* que compõem cada *ramo* dendrítico de maneira recursiva, até chegar em alguma *terminação*.

- **Algoritmo de Extração de Contorno de Estrutura de Ramificação (Branching Structure Contour Extraction Algorithm - BSCEA)** : o **BSCEA** basicamente segue a forma do neurônio resultante da união entre o *esqueleto rotulado* e o *soma*, entrando em todas as regiões mais internas da forma. Durante o processo de contorno, sempre que uma região de *bifurcação* for encontrada, o *BSCEA* contorna a forma "por fora", como o algoritmo tradicional faria. Por outro lado, sempre que um *cruzamento* ou uma *sobreposição* forem encontradas, o *BSCEA* contorna a forma "por dentro", atravessando a *região crítica* atual, por meio dos endereços armazenados em ponteiros pelo *BTA*. Finalmente, o *BTA* devolve como resultado os sinais do contorno paramétrico $x(t)$ e $y(t)$ (Fig.3.19(b)) e uma imagem do contorno (Fig.3.19(d)).

Estes procedimentos são detalhados nas Seções 3.4.1, 3.4.2 e 3.4.3, respectivamente.

3.4 Os algoritmos

3.4.1 Pré-processamento

Inicialmente, uma seqüência de operações de pré-processamento com transformações por morfologia matemática é aplicada sobre a imagem de entrada, de maneira a separar os componentes da imagem do neurônio, isto é a imagem do *esqueleto* composta pelos *ramos* 8-conexos com um pixel de largura, imagem das *regiões críticas*, imagem das *terminações*, imagem do *soma* e uma *fila* para as *sementes primárias*. Vale ressaltar que a restrição sobre os *ramos* do *esqueleto* de ter um pixel de largura é obrigatória, a fim de garantir a operação adequada do algoritmo de rastreamento.

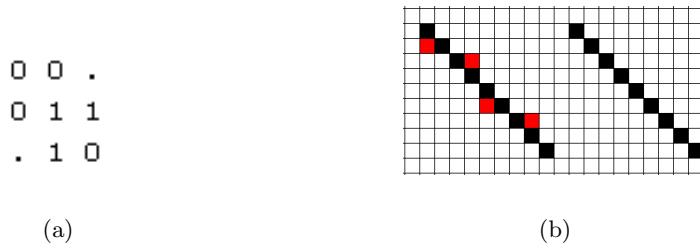


Figura 3.8: (a) Gabarito *Hit-or-Miss* usado para filtrar o esqueleto podado, resultando no *esqueleto* 8-conexo com *ramos* de largura um. (b) Os pixels em vermelho do lado esquerdo devem ser removidos resultando na estrutura essencial do lado direito.

Todos os elementos estruturantes usados são planos e centrados em suas origens, i.e., seus centróides. Para segmentar o *soma*, a seguinte seqüência de operações morfológicas, ilustrada na Fig. 3.9, é aplicada:

- I. erosão da imagem por um disco de raio 3, eliminando boa parte das árvores dendríticas (Fig. 3.9(a));
- II. filtragem por área (*areaopen*) do resultado da operação anterior; todas as componentes conexas que sobreviveram à erosão, mas têm área inferior a 50 são eliminados (Fig. 3.9(b));
- III. dilatação do resultado da operação anterior por um disco de raio 1; assim fazendo, o soma considerado para o restante do processamento será um pouco menor que o original. De fato, tudo se passa como se estivéssemos subtraindo a borda do soma (gradiente morfológico). Mais adiante, as sementes primárias serão detectadas como os pixels adjacentes a este soma reduzido, portanto pertencentes à borda externa deste, e consequentemente à borda interna do soma original (Fig. 3.9(c)).

Sabe-se que as formas de *soma* não seguem um padrão claro, o que torna sua segmentação crítica. Tal problema pode ser solucionado usando a transformada da distância [21]. Pixels ruidosos em torno do *soma* são eliminados por meio da operação "area opening" sobre o *esqueleto* com uma vizinhança definida por um quadrado 3×3 .

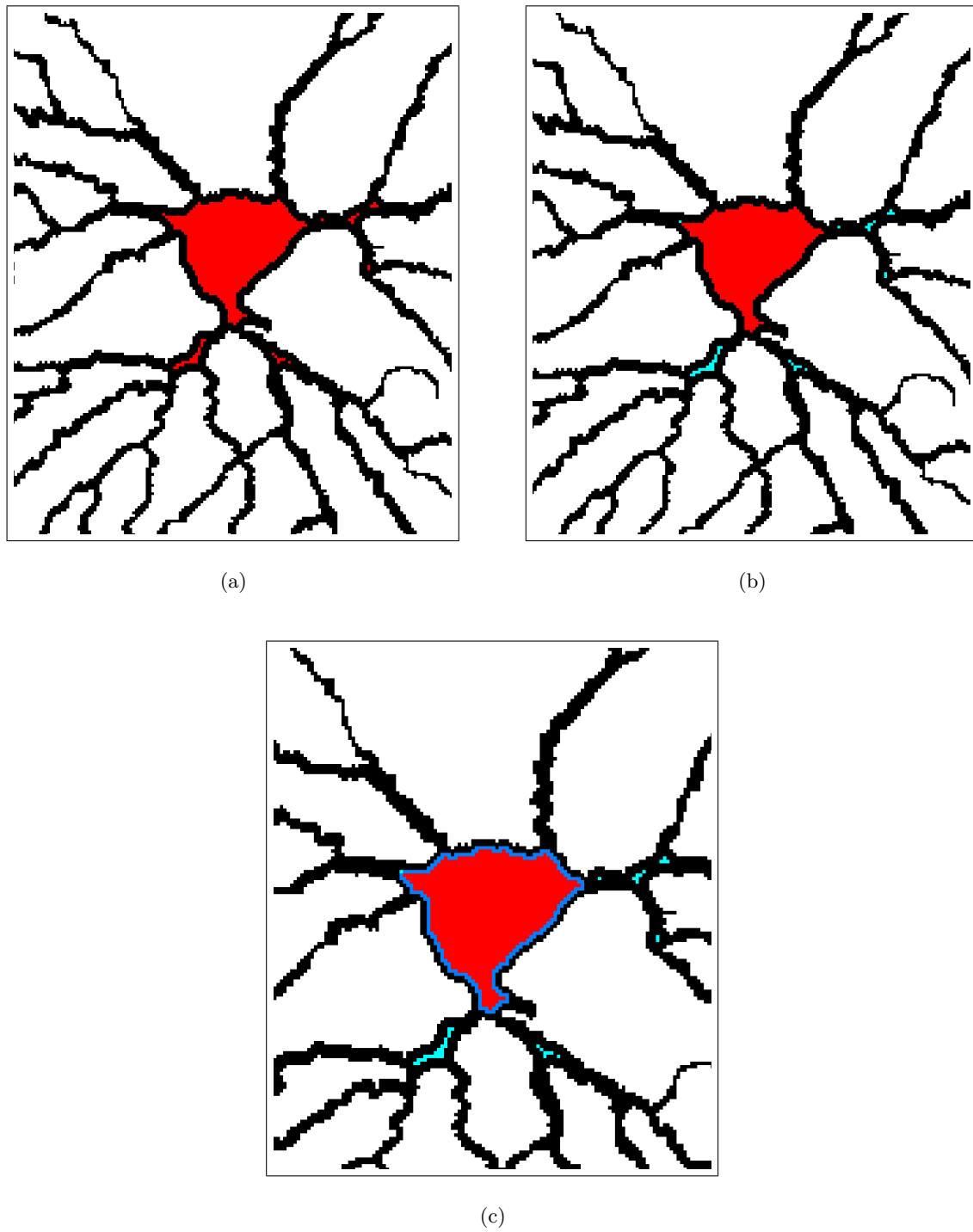


Figura 3.9: Seqüência de operações para segmentação do soma: (a) Da erosão da imagem original (preto) por um disco de raio 3 resulta o soma e algum ruído (vermelho). (b) O ruído (ciano) é removido por uma filtragem por área; todos os componentes conexos com área menor do que 50 são eliminados. (c) O resultado (vermelho) dilatado por um disco de raio 1 produz o soma (vermelho e azul) ligeiramente menor que o original (preto). Sementes primárias, detectadas posteriormente, serão adjacentes à borda interna (azul) do soma segmentado.

Note que as dimensões dos elementos estruturantes foram obtidos empiricamente, especialmente para as imagens usadas neste trabalho. Três imagens foram usadas para calibrar estes parâmetros: uma imagem de neurônio do tipo *alpha* de 500×598 , uma imagem de neurônio do tipo *delta* de 475×511 e uma imagem de neurônio do tipo *epsilon* de 768×712 . Em particular, as Fig. 3.14(a) e 3.15(a) exibem imagens de um neurônio 2D do tipo *alpha* de 500×595 e um neurônio 2D do tipo *ypsilon* de 500×491 . Considerando que o processamento de imagens baseado em Morfologia Matemática depende das dimensões e das formas dos elementos estruturantes, escolher o elemento estruturante adequado é uma tarefa importante. Portanto, para tamanhos de imagens diferentes das supracitadas, faz-se necessário testar discos com diâmetros diferentes, seguindo a seqüência de operações descrita. Talvez o gabarito mais crítico e difícil de encontrar seja o ilustrado na Fig. 3.8(a) para a operação *Hit-or-Miss*.

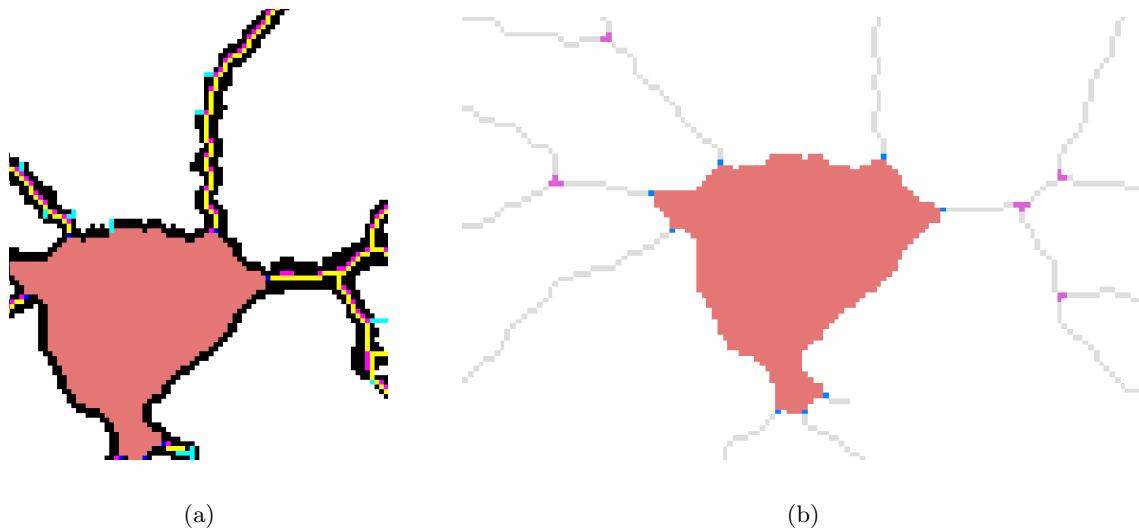


Figura 3.10: (a) A imagem apresenta uma sobreposição de algumas etapas do pré-processamento. Inicialmente, o soma (vermelho) é segmentado da imagem original (preto). Em seguida, a imagem original (preto) é esqueletonizada por afinamento (amarelo, ciano e magenta). Então, ocorre uma primeira filtragem, por meio da qual o esqueleto resultante é podado (amarelo e magenta), eliminando pixels ruidosos (ciano). A seguir, ocorre uma segunda filtragem (*Hit-or-Miss*), mediante o que, pixels redundantes (magenta) são eliminados, resultando o esqueleto 8-conexo de largura um (amarelo). Finalmente são segmentadas as sementes primárias (azul). (b) Soma (vermelho), Esqueleto 8-conexo com ramos de largura um (cinza), sementes primárias (azul) e algumas regiões críticas (magenta).

3.4.2 Rastreamento de Ramos

Basicamente, o principal objetivo desta etapa é rotular cada ramo dendrítico como um objeto em si. Isto é feito rotulando pixel por pixel de um dado *ramo*, um por vez. Considerando a natureza seqüencial deste processamento, este problema pode ser descrito como estimar as coordenadas espaciais (x, y) de cada pixel subseqüente num *ramo*. Como isto é análogo aos problemas de rastreamento [5] na literatura, este algoritmo é chamado de algoritmo de rastreamento de ramos (*Branches Tracking Algorithm (BTA)*).

O *Rastreamento* é normalmente dividido nas etapas denominadas *Previsão*, *Medição/Observação* e *Atualização/Correção/Assimilação* [5]. Na etapa de *Previsão*, o algoritmo estima o próximo estado do sistema, através de um modelo de dinâmica. Na etapa de *Medição*, o algoritmo sonda o sistema buscando estados plausíveis na vizinhança do estado atual, neste caso pixels válidos, por meio da medição de observáveis (aqui as coordenadas espaciais (x, y) dos pixels). Na etapa de *Atualização*, o algoritmo combina ambas as informações coletadas nos dois estágios precedentes, mediante uma combinação linear, que resulta numa estimativa ótima para o próximo estado.

Portanto, em termos de *Rastreamento*, as etapas de *Previsão* e *Medição* no *BTA* são executadas num único passo, pela varredura numa vizinhança de conectividade 8, usando o código da cadeia. A etapa de *Atualização* do *BTA* está relacionada à rotulação do pixel.

Por outro lado, na vizinhança das *regiões críticas*, a etapa de *Previsão* considera a orientação da tangente da trajetória, seguida pelo algoritmo de rastreamento quando ele chega numa *região crítica*. Esta orientação da tangente é dada pelo cálculo do *vetor diretor incidente* \hat{v}_0 . A etapa de *Medição* leva em conta as orientações da tangente de possíveis trajetórias para continuar com o algoritmo de rastreamento quando este passa pela *região crítica*. Estas orientações de tangente são dadas pelo cálculo dos *vetores diretores emergentes* \hat{v}_i . Uma detalhada e ilustrada explanação desse processo, baseado em busca em largura, pode ser encontrado no Apêndice H.4. Note que este problema é análogo ao de rastreamento de um objeto em uma seqüência de vídeo, em uma situação em que o objeto desaparece, caracterizando uma oclusão (ex: um objeto esconde-se sob algo).

O procedimento de *Atualização* finalmente mescla as duas informações usando produtos internos e empilhando adequadamente o pixel ótimo. Para resumir, o *vetor diretor incidente* \hat{v}_0 é comparado com cada *vetor diretor emergente* \hat{v}_i por operações de produtos internos. O produto interno para cada par \hat{v}_0, \hat{v}_i resulta num escalar s . O *vetor diretor emergente* \hat{v}_i para o qual s é máximo fornece ao *BTA* a orientação adequada para prosseguir.

O *BTA* é composto principalmente dos seguintes laços aninhados:

- o laço mais externo relativo à fila de *sementes primárias*.
- o laço mais interno administra duas estruturas de dados:
 - a fila de *sementes secundárias*.
 - uma *pilha* para armazenar vizinhos válidos.

Vale mencionar que, para nossos objetivos, *pixels válidos* são definidos como pixels objeto não-rotulados e não-críticos. Então, para cada *semente primária*, o *BTA* começa pelo empilhamento subsequente de cada pixel de um *segmento* a ser rotulado mais tarde, até que uma *terminação* ou uma *região crítica* seja atingida. Neste ponto, o *BTA* pode executar até duas tarefas, a saber: *Avaliação da Continuidade da Orientação da Tangente* e *Classificação de Regiões Críticas*. A primeira tarefa sempre é executada, enquanto a segunda é executada desde que a *região crítica* atual não tenha sido classificada ainda.

3.4.2.1 Avaliação da Continuidade da Orientação da Tangente

Como discutido na seção 3.4.2, esta etapa também compreende *Previsão*, *Medição* e *Atualização*. Ao chegar numa *região crítica*, a *Previsão* é efetuada pelo cômputo do *vetor diretor incidente* \hat{v}_0 . Na etapa de *Medição*, o algoritmo calcula todos os *vetores diretores emergentes* \hat{v}_i . Finalmente, a etapa de *Atualização* combina as saídas da *Previsão* e *Medição*, a fim de estimar o melhor *segmento* candidato entre todas as alternativas para continuar com o procedimento de rastreamento, calculando produtos internos entre o *vetor diretor incidente* \hat{v}_0 e cada *vetor diretor emergente* \hat{v}_i . O pixel inicial do *segmento ótimo* para prosseguir é finalmente empilhado e rotulado.

Ainda, a origem do caminho alternativo é considerado como uma *semente secundária*, que é a semente de um ramo secundário a ser enfileirado no caso de uma *bifurcação* ser detectada. Por outro lado, se uma *sobreposição* ou um *cruzamento* for detectado, os endereços do ponto inicial do próximo *segmento* V_{n+1} e do último ponto do *segmento* atual V_n (Fig. 3.13(b)) são armazenados no *Mapa de Ponteiros*.

Todos os *ramos* secundários são rotulados recursivamente, até que não haja mais nenhum *ramo* não rotulado oriundo de qualquer *ramo secundário* pertencente à subárvore dendrítica originária da *semente primária* atual. O melhor *segmento* candidato para prosseguir com a rotulação depois da *região crítica* é escolhido de acordo com uma comparação entre o *vetor diretor incidente* \hat{v}_0 e cada *vetor diretor emergente* \hat{v}_i (observe que estes dois vetores têm magnitude unitária) por meio de produtos internos, conforme a Eq. 3.1:

$$k = \arg \max_i (\langle \hat{v}_0, \hat{v}_i \rangle) \quad (3.1)$$

em que o índice k está associado ao vetor diretor \hat{v}_k dentre os \hat{v}_i , para o qual o produto interno é máximo. Os pontos extremos de cada vetor são determinados por uma *Busca em Largura* [22]. Sempre que uma *região crítica* é encontrada, a *Busca em Largura* é acionada e todos os pixels vizinhos frontais são iterativamente enfileirados numa fila auxiliar, enquanto a *região crítica* corrente é atravessada. A cada iteração da *Busca em Largura*, a fila auxiliar é executada em busca de pixels críticos. A condição de parada da *Busca em Largura* é estabelecida previamente como um número C de varreduras consecutivas da fila auxiliar, sem encontrar qualquer pixel crítico. Este procedimento é detalhado num exemplo nos Apêndices H.3 e H.4.

3.4.2.2 Classificação de Região Crítica

Enquanto as orientações dos vetores diretores tangentes são avaliadas em cada *região crítica*, como descrito na seção 3.4.2.1, o *BTA* também reúne informação suficiente para classificar a *região crítica* corrente em uma das 6 diferentes classes (veja Fig. 3.7), que foram identificadas como sendo críticas nos esqueletos. A *Classificação de Regiões Críticas* é um conceito crucial para o funcionamento adequado do nosso algoritmo de *Extração de Contorno* apresentado na seção 3.4.3. Embora a *Classificação de Regiões Críticas* não seja um algoritmo por si mesmo, ela é parte integrante do *BTA*. Portanto, cada *região crítica* pode ser classificada conforme algumas regras especiais. A árvore de decisão ilustrada na Fig. 3.11 detalha as regras de classificação e a ordem em que as mesmas são consideradas. Em resumo, as *regras de classificação* levam em conta o ângulo β entre o *vetor diretor incidente* - \hat{v}_0 - na *região crítica* atual s_1 e seu *vetor diretor emergente* - \hat{v}_2 , o ângulo γ entre quaisquer pares de *vectores diretores emergentes* \hat{v}_i, \hat{v}_j e a cardinalidade $|E_1|$ do conjunto de *vectores diretores emergentes* E_1 relacionados à *região crítica* atual s_1 . Uma descrição detalhada pode ser encontrada no Apêndice G e Fig. 3.11.

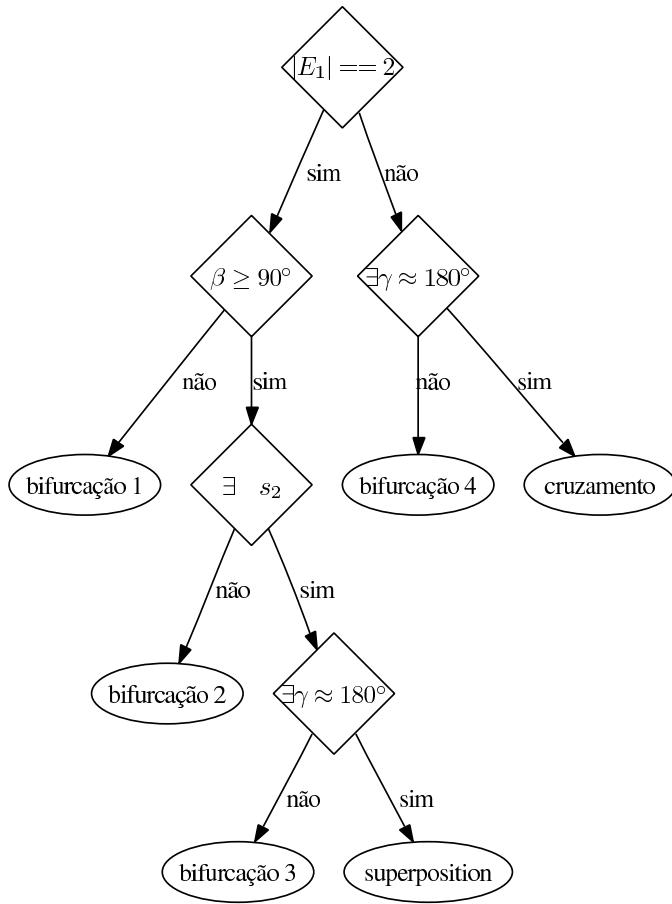


Figura 3.11: Árvore de Decisão representando a seqüência de regras aplicadas para representar a *região crítica* e outra na vizinhança no conjunto S de *regiões críticas* (see Fig. 3.7). As variáveis E_1 e E_2 representam os conjuntos de **vetores diretores** unitários relativos a s_1 e s_2 respectivamente. A variável β é o ângulo entre *vetor diretor incidente* \hat{v}_0 e o *vetor diretor emergente* \hat{v}_2 , enquanto que a variável γ é o ângulo entre quaisquer dois *vetores diretores emergentes*.

3.4.3 Extração de Contorno Paramétrico

Considere as seguintes convenções para a descrição do *Algoritmo de Extração de Contornos de Estruturas de Ramificação (BSCEA)*:

- i *entrada*: união das imagens do *esqueleto rotulado* e do *soma*.
- ii sistema de coordenadas no canto superior esquerdo.
- iii direções relacionadas ao pixel atual C são identificadas de acordo com o código da cadeia mostrado na Fig. 3.2.
- iv a *entrada* é seguida no sentido anti-horário.

v todos os N pontos do contorno paramétrico são armazenados numa estrutura de dados adequada $E(1..N)$. Cada elemento $E(n)$ guarda as coordenadas do n -ésimo ponto do contorno, i.e. $E(n).x$ e $E(n).y$. Quando o contorno se fechar, $E(1)$ é igual a $E(N)$, e portanto temos um sinal periódico.

Considerando o exposto, *BSCEA* é composto pelas etapas principais descritas a seguir:

3.4.3.1 Encontrando o primeiro pixel

O **BSCEA** inicia por uma varredura *raster*, i.e., da esquerda para a direita, de cima para baixo, em busca do primeiro pixel do contorno $E(1)$, que deve ser o primeiro pixel de fundo encontrado, que também é um vizinho de um pixel objeto. Na seqüência, o *BSCEA* contorna toda a forma, até retornar ao primeiro pixel, fechando assim o ciclo e tendo $E(1) = E(N)$.

3.4.3.2 Encontrando o próximo pixel

Do segundo pixel em diante, o código da cadeia (Fig. 3.2) será usado para varrer a vizinhança do pixel atual. Fazendo isso, o segundo pixel do contorno será o primeiro vizinho na seqüência do código da cadeia 4, 5, 6, 7, 8, que também é um pixel de fundo e um vizinho de um pixel de objeto. Aqui, esta varredura em busca do próximo pixel é feita analogamente àquela do algoritmo tradicional de extração de contorno [25]. Além da relação de direção entre os pixels *anterior* e *atual* (ver Fig. 3.12(a)) para decidir propriamente sobre o próximo pixel de contorno, deve-se considerar a transição entre rótulos, a fim de saber-se se o *BSCEA* está contornando um *ramo*, o *soma* ou uma *região crítica*. A estratégia de contorno do *BSCEA* depende da estrutura específica a ser contornada. Então, os principais parâmetros do *BSCEA* são:

- o pixel atual do contorno $E(n)$;
- a direção d_{cp} do pixel atual para o anterior;
- o rótulo do pixel anterior;
- o rótulo do pixel atual.

Com o pixel $E(n)$ e a direção d_{cp} , o *BSCEA* pode identificar o ponto de início da varredura da vizinhança em busca do próximo pixel [25], de acordo com a seqüência do código da cadeia (Fig. 3.2).

Como a entrada do *BSCEA* é uma união das imagens do *esqueleto rotulado* e do *soma*, é necessário adotar uma política para encontrar o próximo pixel em cada caso. Assim, o *BSCEA* considera o caso de contornar um *esqueleto* 8-conexo com *ramos* de largura um como o caso padrão, tomando o primeiro pixel de fundo que também é vizinho de um pixel objeto na vizinhança definida pelo código da cadeia. Por outro lado, o *BSCEA* considera o caso de contornar o *soma* como um caso particular, tomando o último pixel, em vez do primeiro, a ser incluído como contorno.

Em assim fazendo, o *BSCEA* pode contornar um *esqueleto* 8-conexo com *ramos* de largura um, enquanto preserva a característica do algoritmo tradicional de extração de contorno de resolver o problema de contornar entradas eventuais de um pixel de largura dentro do *soma*, consequentemente possibilitando que o contorno volte sobre si mesmo [25].

3.4.3.3 Transpondo regiões críticas

Faz-se necessário adotar uma estratégia para o processamento das *regiões críticas*, conforme suas classes, como descrito na seção 3.4.2.2. Regiões classificadas como *Bifurcação* deveriam ser contornadas por *fora*, enquanto aquelas classificadas como *Sobreposição* ou *Cruzamento* deveriam ser contornadas por *dentro*, por meio dos endereços de ponteiros anotados na estrutura de dados *Mapa de Ponteiros* durante a etapa de rastreamento. Dito isto, a integração entre o *soma* e o *esqueleto rotulado* é vital para a extração de contorno bem sucedida, já que a mesma garante o fechamento do contorno.

O *BSCEA* lida com ambos os casos levando em conta os rótulos dos pixels anterior e corrente, que trazem informações valiosas com respeito às seguintes situações:

- *caso 1*: o **BSCEA** está contornando algum *ramo*
 - tome o 1º candidato na seqüência do código da cadeia.
- *caso 2*: o **BSCEA** está numa transição entre um *ramo* e o *soma*
 - tome o *último* candidato na seqüência do código da cadeia.
- *caso 3*: o **BSCEA** está numa transição entre o *ramo* e uma *região crítica*
 - (a) se a *região crítica* for uma *bifurcação*, "contorne-a por fora" (veja Fig.3.12).
 - (b) se a *região crítica* for uma *sobreposição* ou um *cruzamento*, "contorne-o por dentro", (veja Fig.3.13).

No caso 3-a, "contornar por fora" significa contornar a forma com o algoritmo tradicional de extração de contorno faria [25], como mostrado na Fig.3.12.

No caso 3-b, "contornar por dentro" significa:

- sondar a vizinhança do pixel atual $E(n)$ buscando o respectivo pontoiro P_n na estrutura de dados *Mapa de Ponteiros*;
- determinar a relação de direção $d_{E(n) \leftrightarrow V_n}$ entre $E(n)$ e o último pixel V_n do *segmento* atual;
- acessar o ponto inicial do próximo *segmento* V_{n+1} , apontado por P_n ;
- assumir $d_{E(n+1) \leftrightarrow V_{n+1}} = d_{E(n) \leftrightarrow V_n}$ e encontrar E_{n+1} ;

- preencher a lacuna no contorno sobre a *região crítica* com uma linha digital entre $E(n)$ e $E(n + 1)$, usando o algoritmo de Bresenham [13].

Note que o *BSCEA* não consegue saber quais pixels na *sobreposição* ou *cruzamento* pertencem a um *ramo* ou ao outro, já que a projeção do neurônio 3D sobre o plano 2D suprime esta informação. Este problema é contornado substituindo os pixels compartilhados na *região crítica* por dois segmentos que se interceptam obtidos pelo algoritmo de Bresenham. Este caso é ilustrado na Fig.3.13.

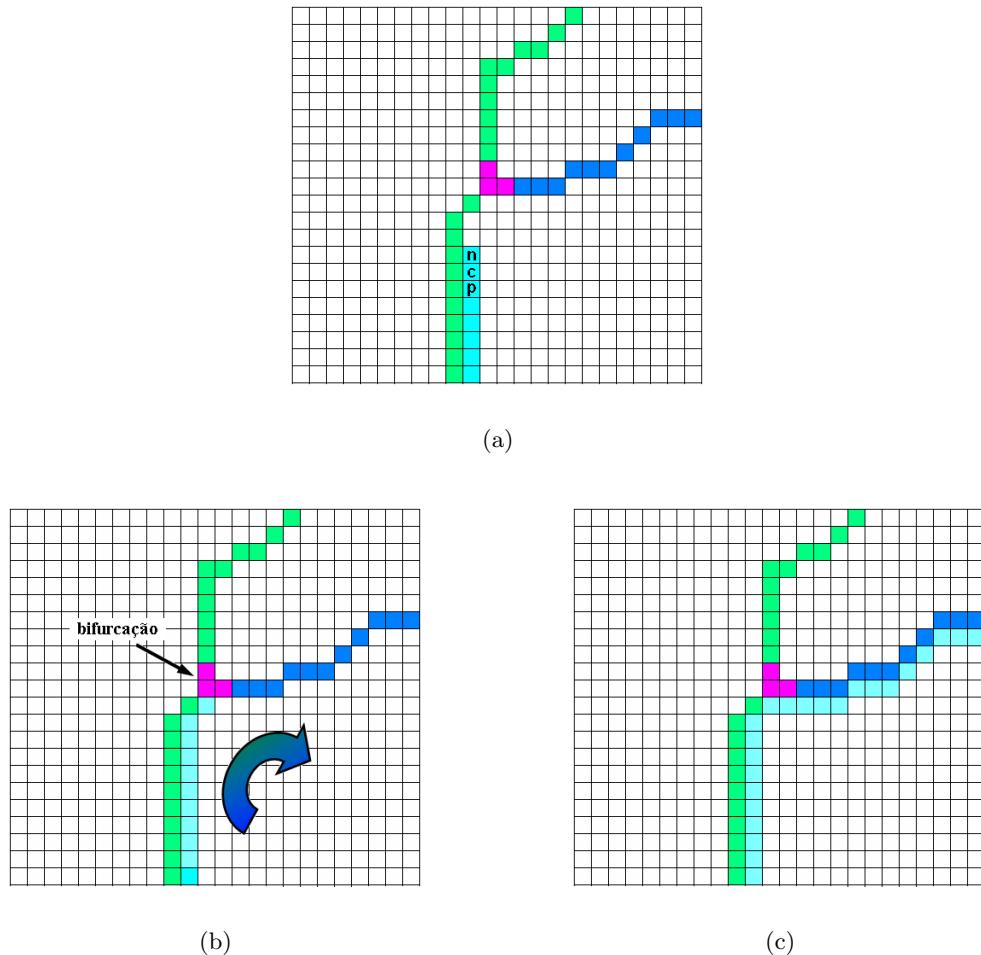


Figura 3.12: (a) Acima: Relações de direção relevantes entre pixels: Pixels Anterior (**p**), Atual (**c**) e Próximo (**n**) representados numa iteração do *BSCEA*. A relação de direção d_{pc} entre os pixels **p** e **c**, além dos rótulos associados aos *segmentos*, determina o próximo pixel **n**. Abaixo, a representação do processo de contorno de uma bifurcação: *Ramos* aparecerem rotulados em azul e verde, enquanto a *região crítica* previamente classificada como uma *bifurcação* aparece em magenta. O contorno é exibido em marrom. (b) Detectando a transição entre rótulos, o *BSCEA* identifica que chegou em uma *bifurcação*, assim decidindo contornar a forma por fora. (c) Tendo atravessado a *região crítica*, ele continua até atingir outra *região crítica*.

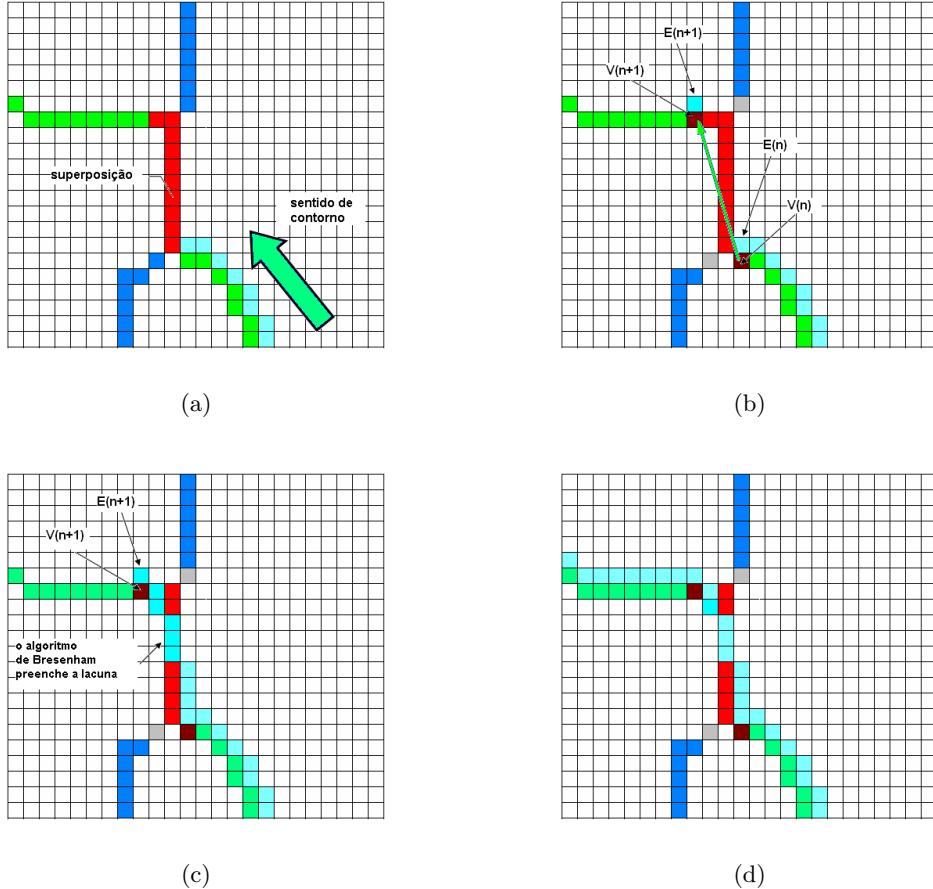


Figura 3.13: Contornando uma *sobreposição*. Ramos aparecem rotulados em azul e verde, enquanto a *região crítica* previamente classificada como uma *sobreposição* aparece em vermelho. O contorno é mostrado em marrom. (a) Detectando a transição entre rótulos, o *BSCEA* identifica que chegou em uma *sobreposição*, assim decidindo contornar a forma por dentro. (b) Em primeiro lugar, o *BSCEA* procura um ponteiro P_n na vizinhança associado ao último pixel do *segmento* atual V_n e então o pixel V_{n+1} apontado por P_n . Subseqüentemente o algoritmo determina a relação de direção $d_{E(n) \leftrightarrow V_n}$. Supondo que $d_{E(n+1) \leftrightarrow V_{n+1}} = d_{E(n) \leftrightarrow V_n}$, ele encontra o ponto inicial $E(n+1)$ do próximo *segmento*. (c) O algoritmo de Bresenham é aplicado para traçar uma linha digital entre $E(n)$ and $E(n+1)$, preenchendo a lacuna no sinal do contorno paramétrico sobre a *região crítica*. (d) Depois de passar pela *região crítica*, ele prossegue até que atingir outra *região crítica*.

3.5 Resultados

Todos os algoritmos descritos neste trabalho foram implementados em *scripts* para a plataforma *Matlab*[®] usando a biblioteca *SDC Morphology Toolbox para MatLab*[™] [50]. O método foi avaliado com as imagens 2D obtidas por *camera lucida* de células ganglionares de retina de gato, da base de imagens *webofneurons*¹.

As Fig. 3.14 e 3.15 mostram imagens rotuladas (coluna direita) obtidas de neurônios do tipo alpha e ypsilon (coluna esquerda). Novos rótulos foram atribuídos aos segmentos dendríticos oriundos de *ramos*. O algoritmo é capaz de distinguir entre classes de *regiões críticas*, o que se comprova pela correta atribuição de rótulos aos *segmentos* emergentes de tais estruturas. Note como os casos de paralelismo muito próximo fazem com que o *BTA* rotule *segmentos* aglutinados como regiões de *sobreposição*. Além disso, sobreposições muito longas também serão rotuladas como regiões de *sobreposição*, contanto que tal sobreposição seja menor do que D_{max} , que é o mínimo comprimento permitido para um caminho entre duas *regiões críticas*. *Bifurcações* do tipo 1, extremamente próximas, podem ser rotuladas como uma *bifurcação* do tipo 4 (Fig. 3.23).

Todas as *bifurcações*, *sobreposições* e *cruzamentos* foram corretamente rotuladas. Resultados para o *Algoritmo de Extração de contornos de Estruturas de Ramificação - BSCEA* são apresentados nas Fig. 3.21 e 3.22, onde pode-se ver o traço do contorno paramétrico para a forma. Os respectivos sinais paramétricos $x(t)$ e $y(t)$ são mostrados na Fig. 3.19(b).

Ainda, o algoritmo *BTA* é capaz de rotular todas as árvores dendríticas presentes na imagem 2D de neurônio, simultaneamente à rotulação de ramos, como pode-se ver na Fig. 3.16.

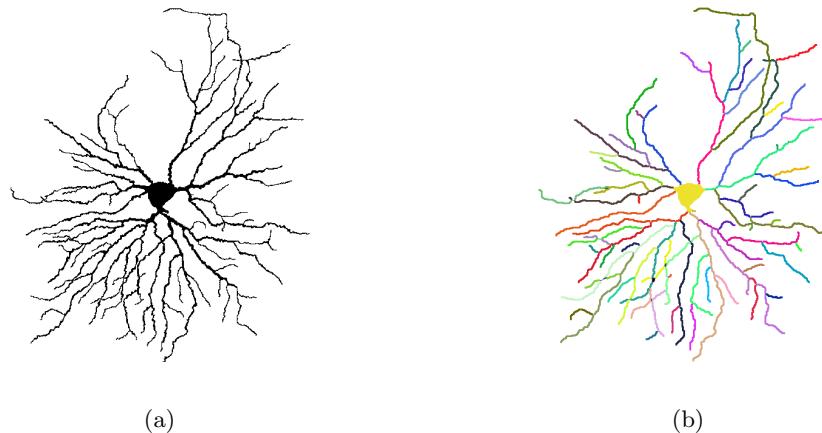


Figura 3.14: Resultados obtidos pelo *BTA - Algoritmo de Rastreamento de Ramos* quando aplicado a uma imagem de neurônio do tipo *alfa*. (a) Imagem de neurônio tipo *Alfa* e a respectiva (b) imagem rotulada. *Ramos* distintos aparecem em cores diferentes.

As Fig. 3.19 e 3.20 ilustram uma comparação em termos de imagens de contorno e sinais

¹<http://www.ime.usp.br/~cesar/projects/webofneurons/images/catgang/catgang.html>

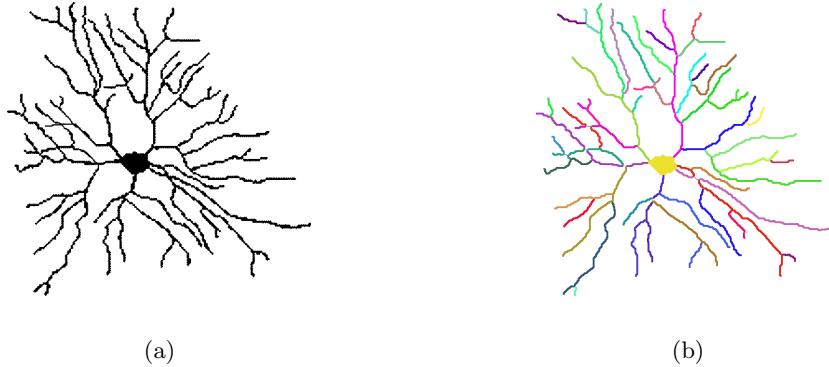


Figura 3.15: Resultados obtidos pelo *BTA - Algoritmo de Rastreamento de Ramos* quando aplicado a uma imagem de neurônio do tipo *ypsililon*. (a) Imagem de neurônio tipo *Ypsilon* e a respectiva (b) imagem rotulada. *Ramos* distintos aparecem em cores diferentes.

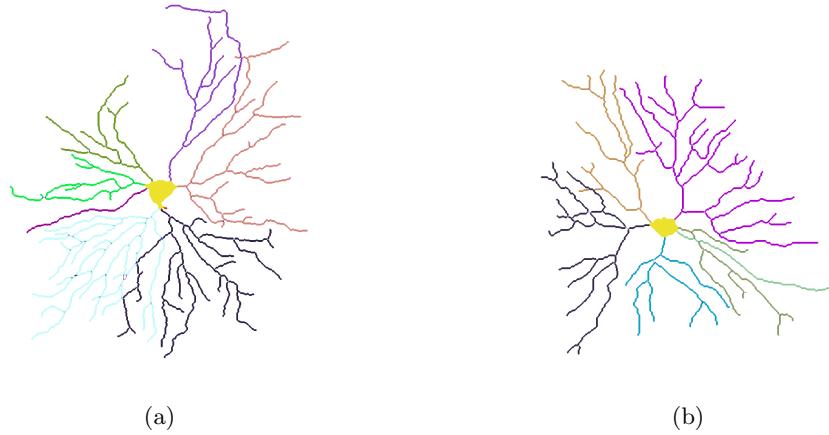


Figura 3.16: Resultados obtidos pelo *BTA - Algoritmo de Rastreamento de Ramos* quando aplicado a imagens de neurônios dos tipos *alfa* e *ypsililon*. (a) Árvores dendríticas rotuladas em neurônio tipo *Alfa* (b) Árvores dendríticas rotuladas em neurônio do tipo *Ypsilon*.

paramétricos entre os resultados obtidos usando as abordagens tradicional e o *BSCEA*. Observe nas Fig. 3.19(c) e 3.20(c) como o algoritmo tradicional não conseguiu acessar as regiões mais internas do neurônio, enquanto o *BSCEA* conseguiu pleno acesso a todos os processos neuronais, como mostrado nas Fig. 3.19(d) e 3.20(d). Mais resultados de rastreamento e rotulação obtidos pelo *BTA* para outras imagens podem ser vistos nos Apêndices H.3 e H.4 e mais resultados de comparações entre o desempenho do algoritmo tradicional de extração de contorno e do *BSCEA* são exibidos no Apêndice H.4.

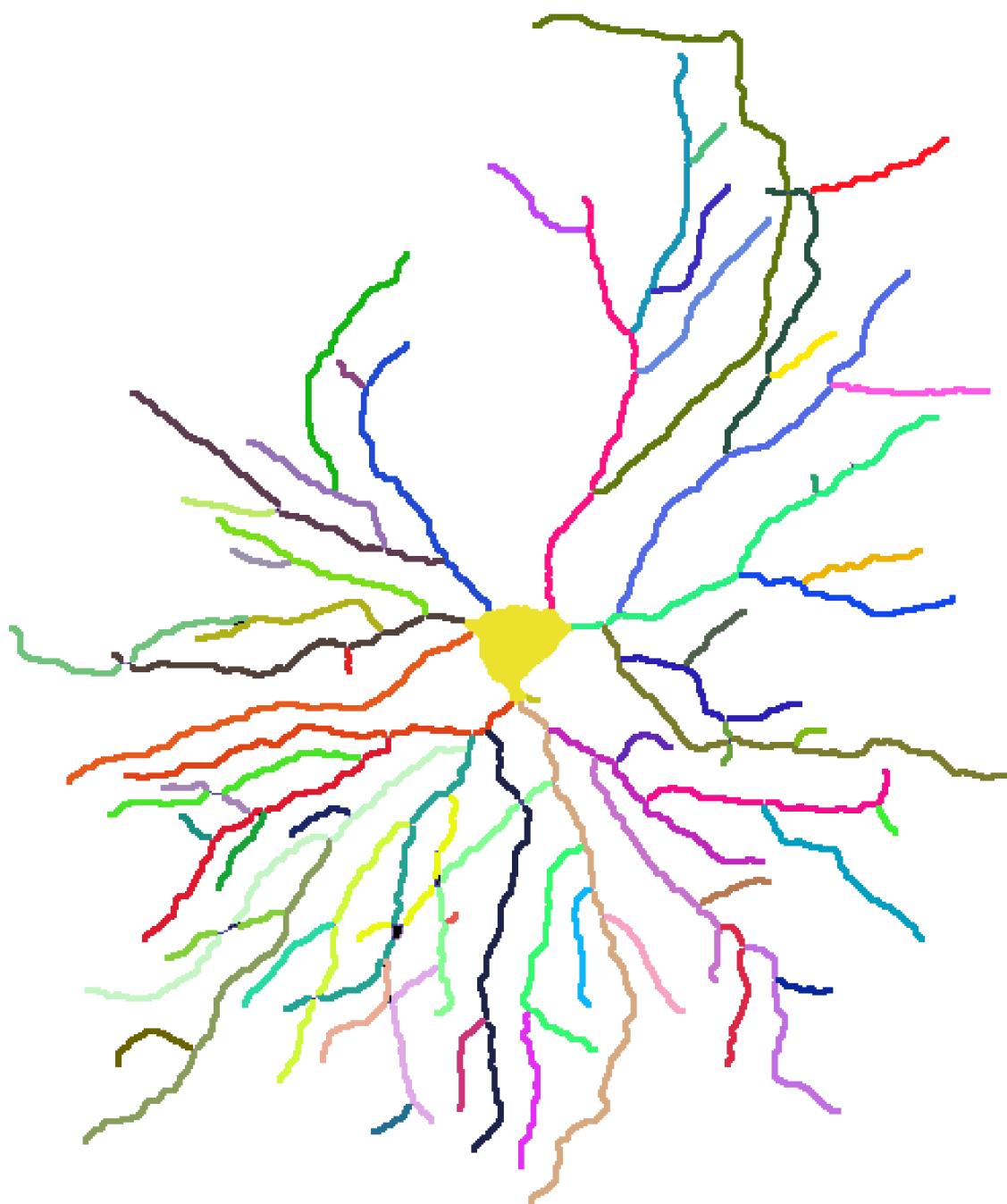


Figura 3.17: Resultados: Imagem rotulada do neurônio tipo Alfa. Segmentos de ramos diferentes recebem cores distintas e as cores foram escolhidas aleatoriamente, tendo sido dilatados para enfatizar o resultado da rotulação.



Figura 3.18: Resultados: Imagem rotulada do neurônio tipo Ypsilon. Segmentos de ramos diferentes recebem cores distintas e as cores foram escolhidas aleatoriamente, tendo sido dilatados para enfatizar o resultado da rotulação.

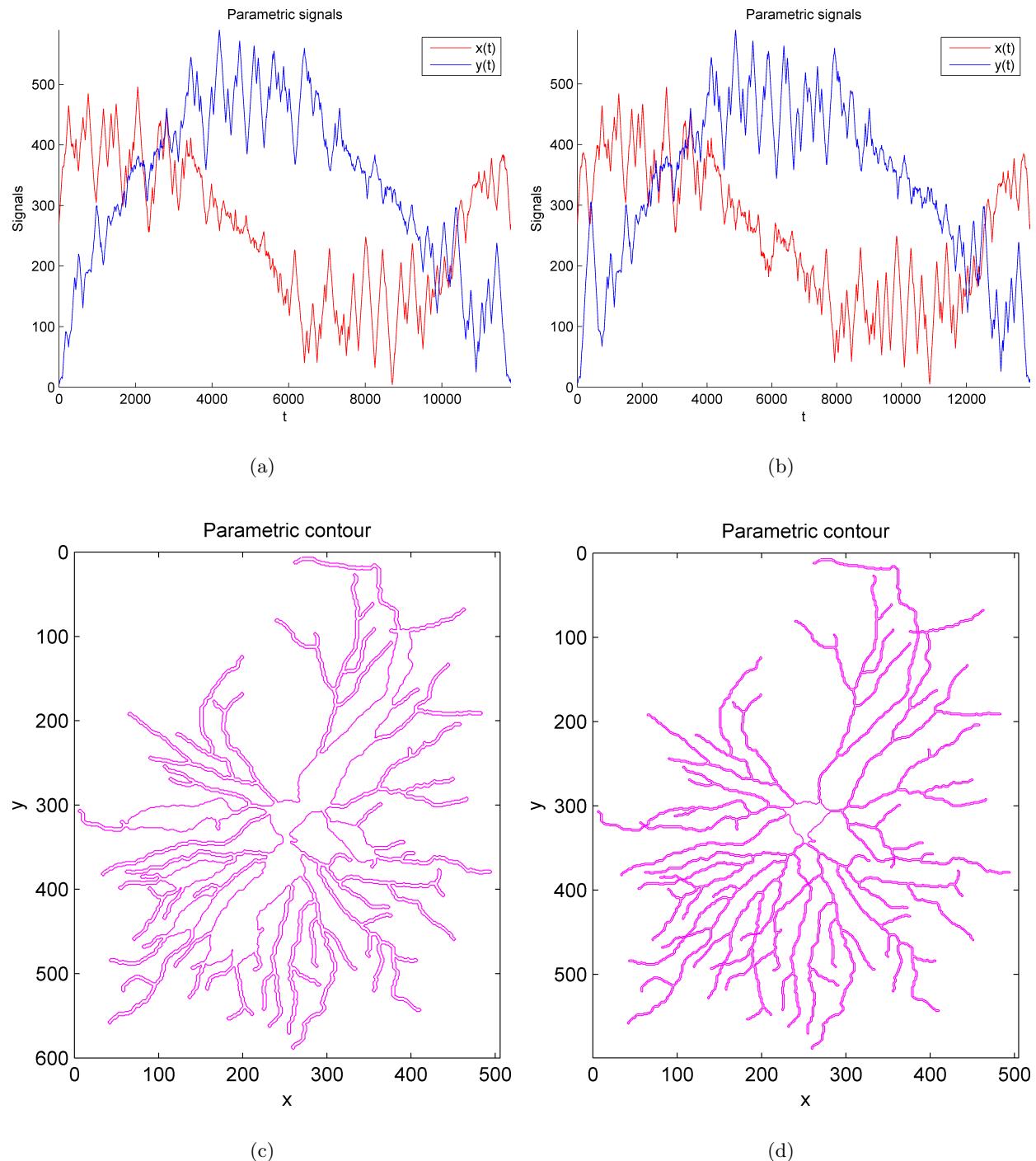


Figura 3.19: Comparação para uma imagem de neurônio do tipo *alpha* entre os resultados produzidos pelos algoritmos (a-c) Tradicional e (b-d) BSCEA. O *BSCEA* entra em todas as regiões, ultrapassando os resultados do algoritmo tradicional em mais de dois mil pontos de contorno.

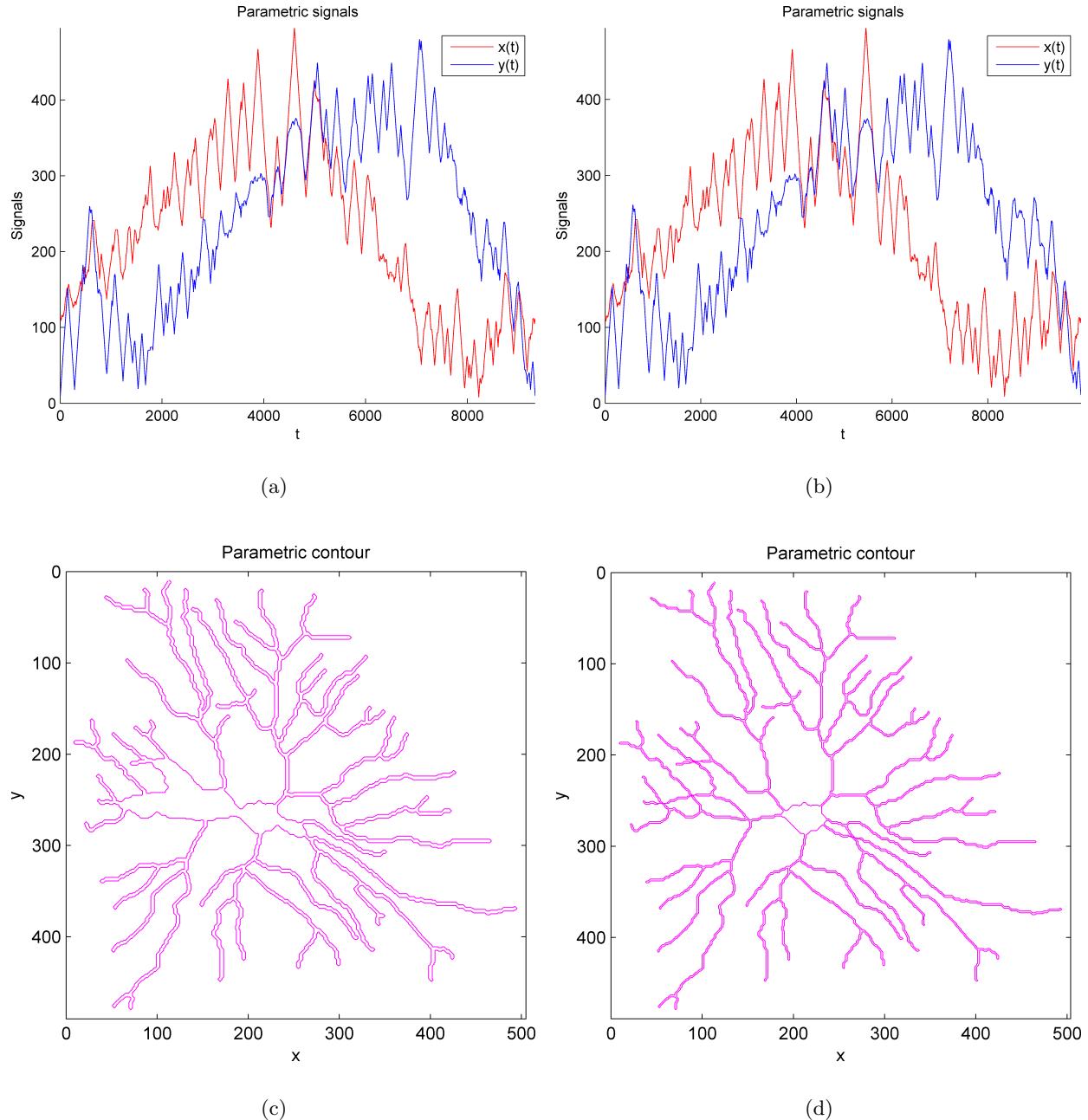


Figura 3.20: Comparação para imagem de neurônio do *ypsilone* entre os resultados produzidos pelos algoritmos (a-c) Tradicional e o (b-d) BSCEA. O *BSCEA* entra em todas as regiões, contando mais pontos que o algoritmo tradicional.

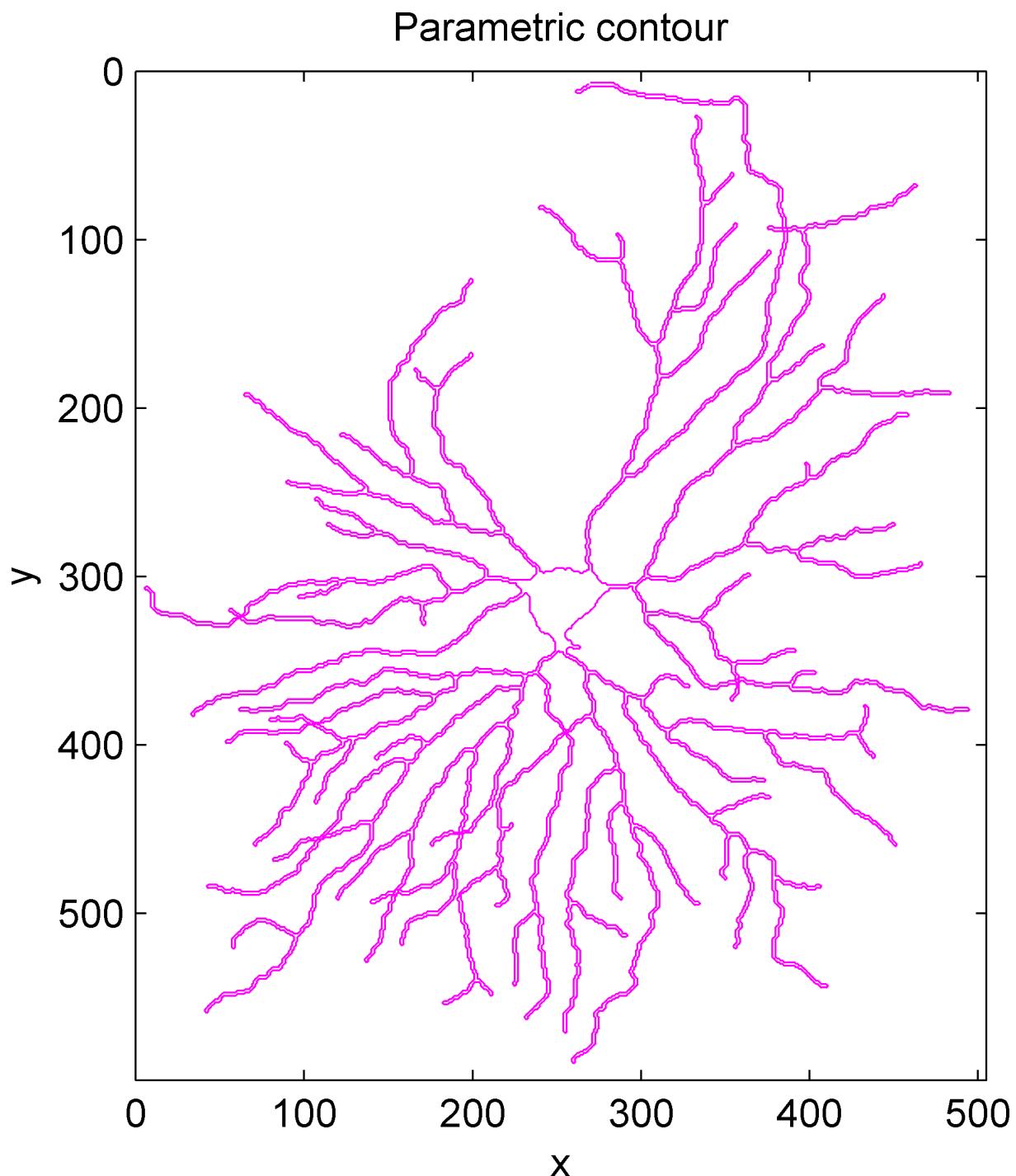


Figura 3.21: Resultados: Traço do contorno paramétrico resultante da aplicação do *BSCEA* em uma imagem de neurônio do tipo *Alfa*. Note como o *BSCEA* libera pleno acesso a todas as regiões mais internas do neurônio. Observe, também, a continuidade do contorno através das *regiões críticas*

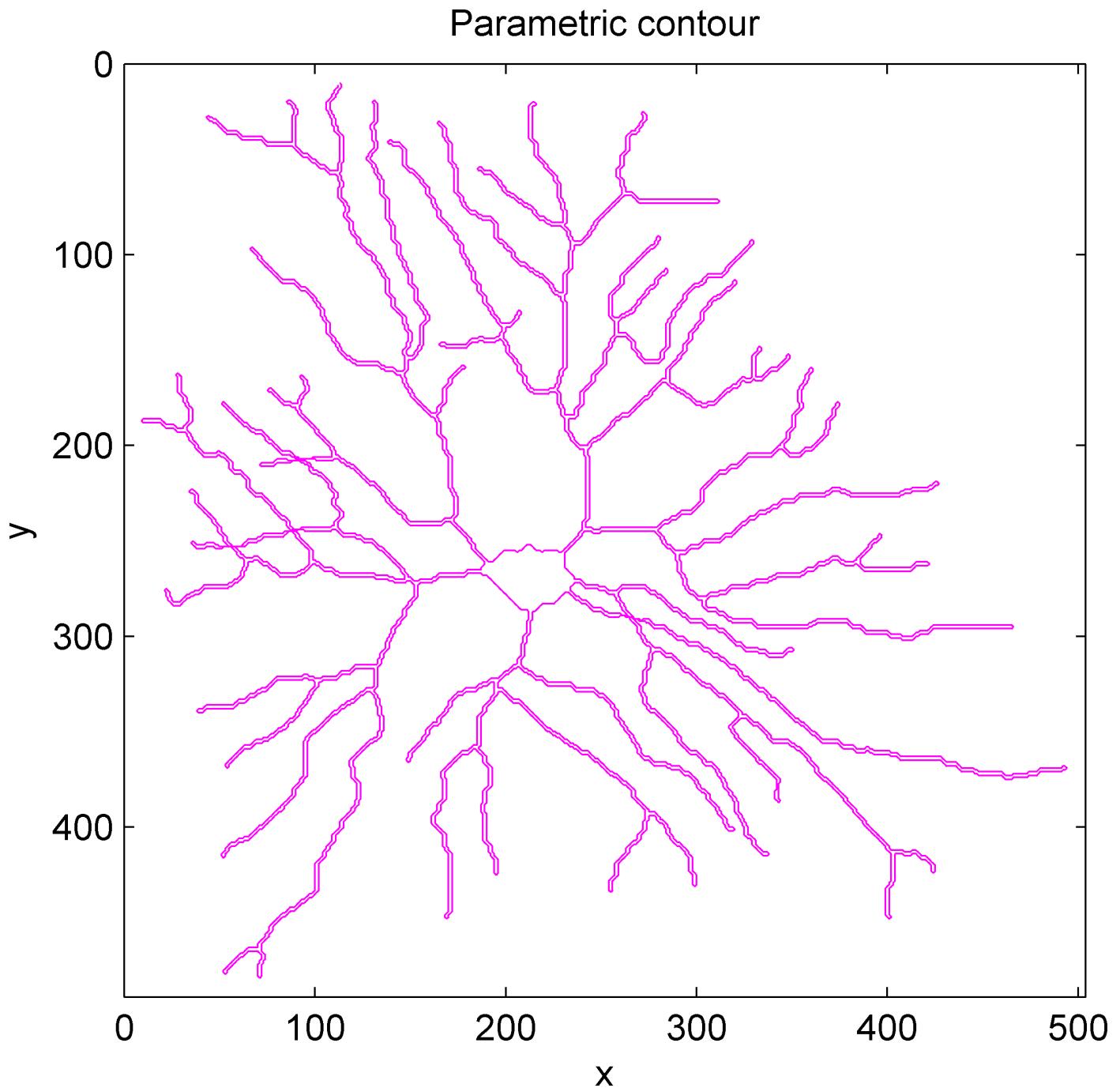
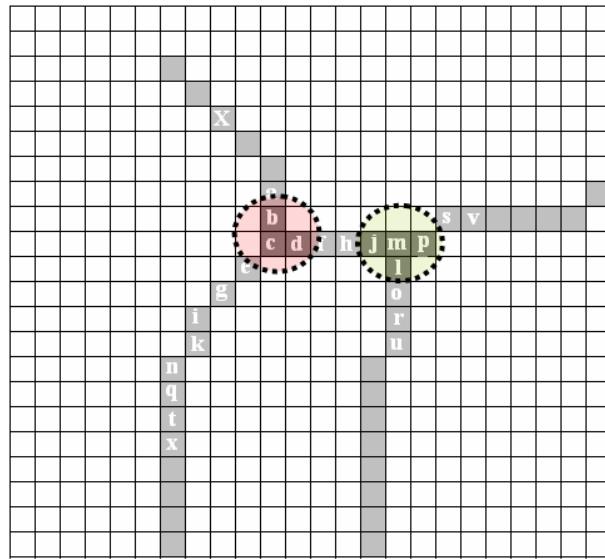
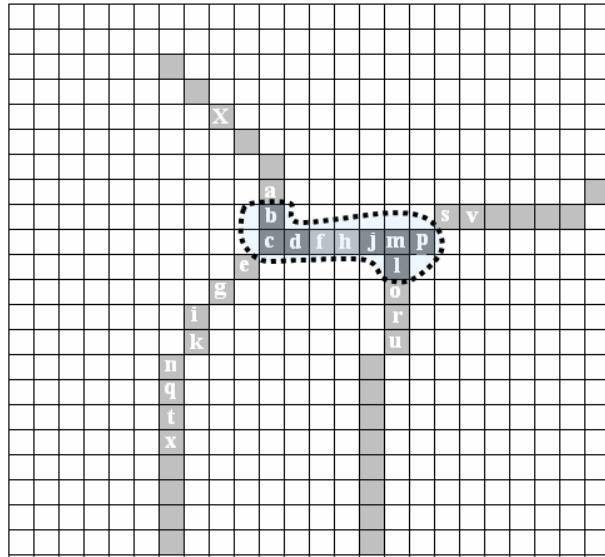


Figura 3.22: Resultados: Traço do contorno paramétrico resultante da aplicação do *BSCEA* em uma imagem de neurônio do tipo *Ypsilon*. Note como o *BSCEA* libera pleno acesso a todas as regiões mais internas do neurônio. Observe, também, a continuidade do contorno através das *regiões críticas*



(a)



(b)

Figura 3.23: (a) Duas *bifurcações* distintas do tipo 1 serão vistas como (b) uma *bifurcação* do tipo 4, uma consequência imediata do *efeito aglutinante* causado pelo algoritmo de *Busca em Largura*, ao encontrar duas bifurcações próximas, como se a análise local atual tivesse dado lugar a uma análise mais global, mudando para uma escala maior de análise.

Conclusão

4.1 Comentários Finais

4.1.1 Comentários sobre a Análise de Formas de Vasos Sangüíneos

As etapas descritas no Cap. 2 foram publicadas no *Journal of the Optical Society of America A - 2007*, em trabalho de colaboração com prof. Herbert F. Jelinek ¹, prof. Michael J. Cree ², A. Luckie ³ e João V. B. Soares ⁴ [56] (Apêndice H.2). Este trabalho seguiu como evolução daquele produzido em colaboração com o prof. Herbert F. Jelinek e Michael J. Cree, publicado nos anais do *APRS Workshop on Digital Image Computing - Pattern Recognition and Imaging for Medical Applications, Brisbane, Griffith University, 2005* [55] (Apêndice H.1).

A relevância deste trabalho reside na combinação da segmentação automática com o reconhecimento de padrões da patologia denominada retinopatia diabética progressiva, usando características derivadas da transformadas *wavelet* (2º momento, entropia), além de uma medida fractal probabilística (dimensão de correlação global e mediana). Foram usadas amostras clinicamente importantes, que incluiam imagens com baixa qualidade com cicatrizes devidas a cirurgias panretinais a laser, retinopatia progressiva diversa, neovascularização que apareceram como áreas de isquemia ou como novos vasos nas proximidades do disco óptico, ou na periferia. Os resultados obtidos com a mesma configuração de parâmetros para todas as imagens, sem a remoção do disco óptico ou pré-processamentos, por causa das propriedades da transformada *wavelet*, o que constitui um avanço em automação de segmentação análise, considerando a diminuição do tempo de processamento e o envolvimento do operador.

Apesar disso, a análise apontou para diferenças entre as segmentações manuais e automáticas (ver Fig. 2.3). Como se pode ver na base de dados *STARE*⁵, tanto a metodologia, quanto as

¹School of Community Health, Charles Sturt University, Wilson Street, Albury 2640, Australia

²Department of Engineering, University of Waikato, Hillcrest Road, Hamilton 3240, New Zealand

³Albury Eye Clinic, Swift Street, Albury 2640, Australia

⁴Departamento de Ciência da Computação, Instituto de Matemática e Estatística, Universidade de São Paulo

⁵<http://www.parl.clemson.edu/stare/>

patologias influenciam os resultados não apenas das segmentações automáticas, mas também das segmentações manuais. Acreditamos que estes erros sejam aleatoriamente distribuídos nas imagens com e sem PDR, e portanto não tenham afetado os resultados de nossa classificação.

Sabe-se que oftalmologistas conseguem identificar acertadamente a retinopatia progressiva diabética entre 80 e 90% de suas avaliações. O acerto diminui quando o meio óptico se torna opaco e na identificação da PDR em estágio inicial. Por outro lado, a análise dos não-especialistas tem uma taxa de acerto comparável ao acerto por acaso (50%).

Este trabalho demonstrou que características geométricas tradicionais (área, perímetro e circularidade), quando usadas isoladamente, são incapazes de distinguir entre imagens com e sem neovascularização em virtude da presença de PDR. Isto se deve ao caráter global destas características, que consideram as árvores venosas em seu todo, enquanto a PDR envolve a reconfiguração da geometria dos vasos, fazendo necessária a utilização de características sensíveis às propriedades de ramificação e preenchimento do espaço dos padrões venosos. Ademais, esta análise sugere que o melhor subespaço de 6 características (ver Tab. 2.6) resulta na classificação mais precisa, com 94% de sensibilidade e 82% de especificidade. As três imagens erroneamente classificadas apresentavam problemas: uma opaca, outra com PDR avançada e outra com um laço venoso e placas de Hollenhorst, além da PDR.

O uso de um detector de microaneurismas, assim como o emprego de outros classificadores em lugar de Bayes-ingênuo e LDA poderiam melhorar os resultados da classificação.

4.1.2 Comentários sobre a Análise de Formas de Neurônios

As etapas descritas nas Seções 3.4.1 e 3.4.2 foram apresentadas e publicadas nos anais do **Sibgrapi2006 - Proc. 19th SIBGRAPI - Brazilian Symposium on Computer Graphics and Image Processing** [60] (Apêndice H.3), exceptuando-se a *Classificação de Regiões Críticas* e o *Mapa de Ponteiros* introduzidos neste trabalho. A introdução destes dois últimos conceitos melhoraram o desempenho do *BTA*, de cuja saída o sucesso do *BSCEA* é fortemente dependente. A descrição destes conceitos e o algoritmo de extração de contornos *BSCEA* descrito na Seção 3.4.3 foram apresentados e publicados nos anais do *Workshop on Bio-Image Informatics: Biological Imaging, Computer Vision and Data Mining, 2008 (Santa Barbara, CA, USA)* [62] e disponibilizados no *arXiv* [63]⁶ (Apêndice H.4) e aceito pelo *Journal of Neuroscience Methods - Elsevier* [61]. A segunda parte desta dissertação foi desenvolvida sob a co-orientação do Prof. Luciano da Fontoura Costa⁷, também co-autor de ambos os artigos supra-citados.

A caracterização adequada de formas de estruturas de ramificação é um problema muito importante, desempenhando um papel central em diversas áreas em medicina e biologia, especialmente em neurociência. A compreensão atual da dinâmica fisiológica em redes neurais biológicas pode ser reforçada por meio da caracterização apropriada de formas de células neuronais, já que a quantidade

⁶<http://aps.arxiv.org/abs/0804.3234>

⁷Departamento de Física e Informática - Instituto de Física de São Carlos - USP

de sinapses e a maneira em que estes neurônios se organizam em redes estão fortemente atrelados às formas das células.

Características baseadas em contorno podem fornecer informações importantes sobre as escalas local e global das formas investigadas. A informação local pode ser recuperada mediante a análise do contorno paramétrico, enquanto a informação global pode ser adquirida a partir de medidas estatísticas considerando a forma das células como um todo.

Entretanto, a extração precisa dos contornos de neurônios não tem sido possível até o presente momento por causa da presença de sobreposições entre processos neuronais, causados pela projeção de formas de neurônios 3D em imagens 2D. Sempre que um *cruzamento* acontecer nessas imagens, o algoritmo tradicional de extração de contorno, que é baseado no código da cadeia, falha ao tentar entrar em suas regiões mais internas.

O presente trabalho descreveu um sistema original capaz de lidar apropriadamente com o problema de seguir o contorno de estruturas de ramificação, mesmo na presença de *ramos* que se interceptam. Nossa sistema é composto basicamente de três partes, isto é: (i) *Algoritmo de Pré-processamento*, (ii) *Algortimo de Rastreamento de Ramos - BTA* e (iii) *Algoritmo de Extração de Contorno de Estruturas de Ramificação - BSCEA*.

Como o sistema proposto inicia por uma série de transformações (pré-processamento) na imagem da projeção 2D de uma estrutura de ramificação 3D, a fim de obter um *esqueleto* adequado, vale mencionar que qualquer esquema de esqueletização diferente do afinamento morfológico poderia ser adotado, tal como dilatações exatas [25], transformação do eixo medial, e assim por diante, desde que um esqueleto 8-conexo com *ramos* de largura um seja obtido como resultado. Além disso, o esquema de esqueletização afetará fortemente a escolha de todos os parâmetros do pré-processamento, que, neste trabalho, foram escolhidos por tentativa e erro. Deve-se ter em mente de que a idéia principal do método é prover os algoritmos de rastreamento com um *esqueleto* adequado como entrada.

Quanto ao *BTA*, devem haver casos particulares para consideração, por exemplo imagens com alta densidade de *regiões críticas* e/ou a presença de estruturas cujas topologias possam favorecer o aparecimento de *sobreposições*. O primeiro caso, isto é, altas densidades de *regiões críticas* podem ocorrer devido às topologias específicas da forma na imagem ou devido à própria resolução da imagem, fazendo com que o *BTA* agrupe *regiões críticas* muito próximas umas das outras. Note que, a fim de satisfazer a condição de parada previamente estabelecida para a *Busca em Largura*, o *BTA* aglomerou as *bifurcações* do tipo 1 (Fig. 3.23-(a)) em um aglomerado de *bifurcações* aparecendo como uma *bifurcação* do tipo 4 (Fig. 3.23-(b)). Uma solução alternativa possível para esta armadilha poderia ser a implementação da *busca em largura* com uma estrutura de dados em árvore, além da fila auxiliar, a fim de memorizar caminhos válidos entre *extremidades de vetores diretores* e *origens*. Imediatamente após satisfazer a condição de parada, o *BTA* recuperaria os *pontos extremos dos vetores diretores* armazenados nas folhas da árvore. As respectivas *origens dos vetores diretores* seriam obtidas simplesmente escalando a árvore das folhas para cima, até atingir

os respectivos primeiros pixels não-críticos adjacentes à *região crítica* precedente. Além disso, esta estratégia permitiria desfazer ambigüidades entre duas *origens* possíveis que estejam igualmente distantes de uma *extremidade de vetor diretor*, já que a distância mais curta entre uma *extremidade de vetor diretor* e um candidato a *origem* é a única condição considerada. A estrutura de dados auxiliar em árvore acrescentaria uma informação com respeito à existência de um caminho (*segmento*) válido entre uma *extremidade de vetor diretor* e um candidato à *origem*.

No segundo caso, isto é *Sobreposições*, deve-se notar que regiões de *Sobreposição* podem ser resultado de três problemas distintos. Primeiro o *paralelismo extremamente próximo*, em que imagens contendo muitos *ramos* quase paralelos e muito próximos entre si, usualmente apresentam pequenos ciclos após a etapa de pré-processamento. Pequenos ciclos são altamente indesejáveis, pois são estruturas que podem criar muitos erros. Este problema pode ser contornado incluindo uma dilatação no pré-processamento, antes da esqueletonização. Em assim fazendo, pequenos ciclos são encolhidos em regiões fechadas de *Sobreposição*. Em segundo lugar, a resolução ruim pode criar imagens com *cruzamentos* originalmente em formato elipsóide e alongado amostradas como estruturas com forma quase linear. Depois do pré-processamento, estas estruturas também serão rotuladas como regiões de *Sobreposição*. Em último lugar, sobreposições longas também serão rotuladas como regiões de *sobreposição*, desde que tal sobreposição esteja abaixo do limiar D_{max} , que é o caminho de comprimento mais curto permitido entre duas *bifurcações*.

Isto posto, as principais contribuições originais do presente trabalho⁸ englobam o rastreamento e a extração do contorno paramétrico de estruturas de ramificação, como células neurais.

4.2 Trabalhos Futuros

4.2.1 Análise de Formas de Vasos Sanguíneos - perspectivas

Futuros desenvolvimentos incluem o aprimoramento do método, agregando um detector de microaneurismas, e empregando outros classificadores diferentes de classificador ingênuo de Bayes e o LDA.

4.2.2 Análise de Formas de Neurônios - perspectivas

Futuros desenvolvimentos incluem a extensão do método de separar células em imagens contendo múltiplas células. Diversas aplicações do método proposto neste trabalho podem ser feitas considerando imagens de redes neurais, assim como outros tipos de estruturas biológicas, tais como árvores vasculares de retina (ver Fig. 4.1).

⁸Resultados preliminares da abordagem proposta foram descritos em conferências ([60, 62]).

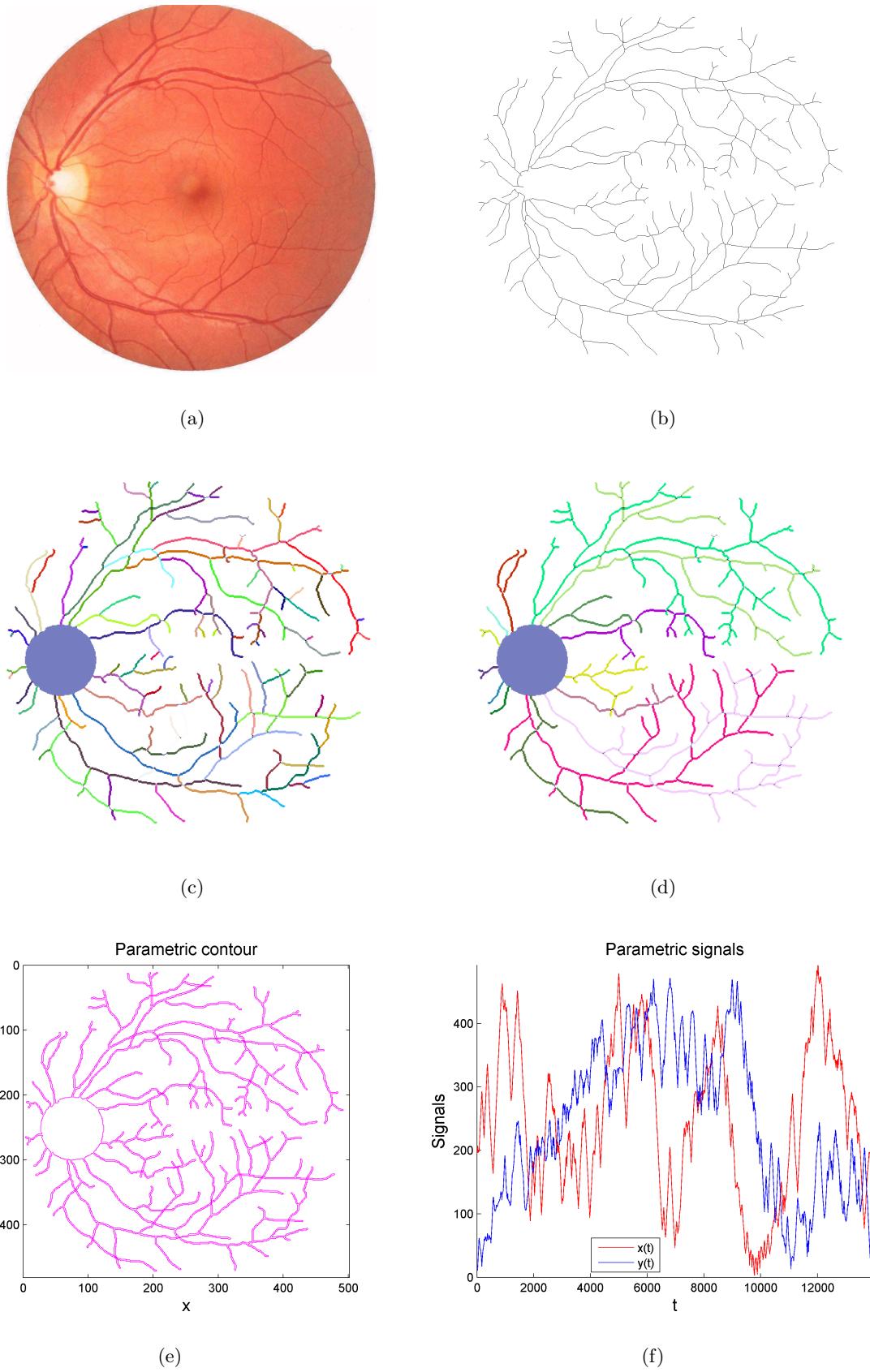


Figura 4.1: Resultados. Fotografia Não-midriática de Retina (a). Esqueleto da segmentação manual (b) Esqueleto rotulado (c). Sub-árvores rotuladas (d). Contorno Paramétrico (e). Sinais Paramétricos (f).

Apêndice **A**

Tabelas com Resultados da Análise 1

Tabela A.1: Espaço de Características - Segmentação Manual

imagem	área	perímetro	circularidade	2º momento	entropia	curvatura	$\nu_{mediana}$	ν_{global}
148643a	106402	38540	13.960	4364,05	5,63	180,52	1,52	1,59
148644a	110597	40254	14.651	4374,55	5,63	179,34	1,7	1,6
148645a	118316	42814	15.493	4360,94	5,64	182,12	1,71	1,61
159893a	81479	29702	10.827	4328,32	5,61	174,84	1,68	1,57
159894a	54551	20002	7.334	4346,71	5,64	171,7	1,52	1,47
164593a	91760	33361	12.129	4342,52	5,64	176,51	1,61	1,56
164595a	81938	29888	10.902	4382,95	5,63	172,92	1,69	1,57
180501a	93796	33766	12.156	4325,18	5,63	178,33	1,76	1,57
180502a	133617	48427	17.551	4363	5,64	181,08	1,61	1,64
18393a	116699	42263	15.306	4344,68	5,64	178,5	1,73	1,61
18403a	106943	38898	14.148	4349,88	5,61	175,64	1,76	1,6
19168a	71927	26242	9.574	4339,73	5,62	171,69	1,64	1,53
19177a	82127	30001	10.959	4348,13	5,62	173,44	1,67	1,54
19240a	120165	43823	15.982	4343,27	5,6	178,66	1,73	1,62
19349a	126343	45983	16.736	4389,79	5,65	182,94	1,77	1,63
19358a	96589	35325	12.919	4390,49	5,63	176,53	1,66	1,56
19407a	83540	30249	10.953	4366,83	5,64	178,04	1,61	1,54
19409a	106468	38695	14.063	4390,57	5,61	182,11	1,58	1,59
20201a	134124	48856	17.796	4368,88	5,63	184,96	1,75	1,64
20215a	125306	45828	16.761	4377,7	5,62	179,76	1,75	1,63
2099a	139094	50108	18.051	4378,19	5,64	184,27	1,76	1,64
211893a	84765	30967	11.313	4373,56	5,63	176,22	1,65	1,56
211894a	65943	24103	8.810	4423,62	5,6	171,57	1,62	1,52
211895a	143640	52198	18.968	4375,43	5,62	182,88	1,73	1,64
211896a	92483	33836	12.379	4383,76	5,63	175,13	1,72	1,57
739112a	101106	36775	13.376	4306,89	5,62	178,4	1,72	1,59
73913a	89751	32885	12.049	4332,43	5,63	176,73	1,64	1,58
média	102203	37177	13524	4362	5,63	178,0	1,68	1,58
variância	23189	8363	3017	25	0,01	3,9	0,1	0,04

Tabela A.2: Espaço de Características - Segmentação Automática

imagem	área	perímetro	circularidade	2º momento	entropia	curvatura	$\nu_{mediana}$	ν_{global}
148643a	28110	27942	27.775	1043,35	5,62	133,74	1,46	1,51
148644a	26689	26551	26.414	1077,14	5,6	131,82	1,39	1,53
148645a	32455	32232	32.011	1084,87	5,63	138,68	1,51	1,5
159893a	18009	17881	17.754	1085,47	5,63	137,35	1,4	1,48
159894a	16135	16045	15.956	1084,10	5,63	133,96	1,46	1,37
164593a	24302	24181	24.061	1043,43	5,65	130,63	1,48	1,44
164595a	18049	17966	17.883	1082,81	5,62	130,56	1,44	1,46
180501a	23508	23328	23.149	1086,68	5,63	135,43	1,49	1,5
180502a	31577	31394	31.212	1077,23	5,63	132,93	1,39	1,52
18393a	26712	26607	26.502	1076,05	5,58	126,7	1,31	1,48
18403a	25432	25346	25.260	1053,85	5,6	126,39	1,25	1,48
19168a	14978	14919	14.860	1084,35	5,64	129,4	1,36	1,39
19177a	21990	21867	21.745	1055,27	5,64	133,52	1,46	1,49
19240a	23403	23258	23.114	1047,86	5,59	131,41	1,27	1,47
19349a	31506	31314	31.123	1078,59	5,62	134,79	1,29	1,54
19358a	27946	27723	27.502	1093,52	5,67	137,46	1,56	1,48
19407a	26990	26736	26.484	1150,87	5,66	147,11	1,51	1,59
19409a	25130	24932	24.736	1119,29	5,64	140,15	1,55	1,48
20201a	22758	22621	22.485	1060,66	5,6	134,7	1,36	1,49
20215a	27379	27272	27.165	1074,96	5,57	128,17	1,37	1,5
2099a	29798	29669	29.541	1068,89	5,62	130,35	1,45	1,49
211893a	18674	18587	18.500	1076,42	5,63	132,88	1,42	1,48
211894a	13928	13855	13.782	1112,34	5,64	132,44	1,57	1,46
211895a	31473	31286	31.100	1088,36	5,65	135,71	1,46	1,5
211896a	24384	24249	24.115	1102,79	5,63	132,43	1,38	1,46
739112a	27948	27799	27.651	1062,75	5,61	133,87	1,48	1,53
73913a	23985	23818	23.652	1086,91	5,6	134,18	1,31	1,54
média	24565	24421	24279	1080	5,62	133,6	1,42	1,49
variância	5201	5166	5131	24	0,02	4,3	0,1	0,04

Tabela A.3: Espaço de Características - Proliferative Diabetic Retinopathy

imagem	área	perímetro	circularidade	2º momento	entropia	curvatura	$\nu_{mediana}$	ν_{global}
148643a	0,6817	0,6815	0,6813	-1,5528	0,0300	0,0364	0,4566	0,4376
148645a	1,5172	1,5120	1,5068	0,2085	0,2749	1,1884	1,0059	0,2896
159893a	-1,2606	-1,2661	-1,2717	0,2339	0,2749	0,8782	-0,2450	-0,1657
159894a	-1,6209	-1,6216	-1,6222	0,1758	0,2749	0,0877	0,4373	-2,6698
164593a	-0,0505	-0,0465	-0,0426	-1,5494	1,1192	-0,6888	0,6647	-1,0762
164595a	-1,2529	-1,2497	-1,2464	0,1211	-0,1473	-0,7051	0,2099	-0,6210
19349a	1,3347	1,3343	1,3339	-0,0579	-0,1473	0,2813	-1,4958	1,2002
19407a	0,4663	0,4481	0,4298	3,0081	1,5413	3,1541	1,0059	2,3384
19409a	0,1087	0,0988	0,0890	1,6685	0,6970	1,5311	1,4607	-0,1657
2099a	1,0063	1,0159	1,0254	-0,4694	-0,1473	-0,7540	0,3236	0,0620
211893a	-1,1327	-1,1295	-1,1262	-0,1500	0,2749	-0,1641	-0,0176	-0,1657
211894a	-2,0453	-2,0455	-2,0457	1,3737	0,6970	-0,2667	1,6881	-0,6210
211895a	1,3284	1,3289	1,3294	0,3565	1,1192	0,4958	0,4373	0,2896
211896a	-0,0348	-0,0334	-0,0320	0,9686	0,2749	-0,2690	-0,4724	-0,6210
739112a	0,6506	0,6539	0,6571	-0,7299	-0,5694	0,0668	0,6647	0,9725
73913a	-0,1115	-0,1168	-0,1222	0,2950	-0,9916	0,1390	-1,2684	1,2002
média	-0,0260	-0,027	-0,029	0,24	0,29	0,31	0,30	0,04
variância	1,1	1,1	1,1	1,1	0,65	1,0	0,87	1,1

Tabela A.4: Espaço de Características - NON-Proliferative Diabetic Retinopathy

imagem	área	perímetro	circularidade	2º momento	entropia	curvatura	$\nu_{mediana}$	ν_{global}
148644a	0,4085	0,4123	0,4160	-0,1194	-0,9916	-0,4113	-0,3587	0,9725
180501a	-0,2032	-0,2117	-0,2201	0,2852	0,2749	0,4305	0,7784	0,2896
180502a	1,3484	1,3498	1,3512	-0,1156	0,2749	-0,1524	-0,3587	0,7449
18393a	0,4129	0,4231	0,4333	-0,1657	-1,8359	-1,6052	-1,2684	-0,1657
18403a	0,1668	0,1790	0,1913	-1,1074	-0,9916	-1,6775	-1,9507	-0,1657
19168a	-1,8434	-1,8396	-1,8356	0,1864	0,6970	-0,9756	-0,6998	-2,2145
19177a	-0,4951	-0,4945	-0,4939	-1,0472	0,6970	-0,0149	0,4373	0,0620
19240a	-0,2234	-0,2252	-0,2271	-1,3615	-1,4138	-0,5069	-1,7232	-0,3933
19358a	0,6502	0,6391	0,6281	0,5754	1,9635	0,9039	1,5744	-0,1657
20201a	-0,3474	-0,3485	-0,3497	-0,8185	-0,9916	0,2603	-0,6998	0,0620
20215a	0,5411	0,5518	0,5625	-0,2119	-2,2581	-1,2624	-0,5861	0,2896
média	0,038	0,040	0,041	-0,35	-0,42	-0,46	-0,44	-0,062
variância	0,82	0,82	0,82	0,63	1,3	0,85	1,1	0,82

Apêndice **B**

Avaliando a similaridade entre segmentações

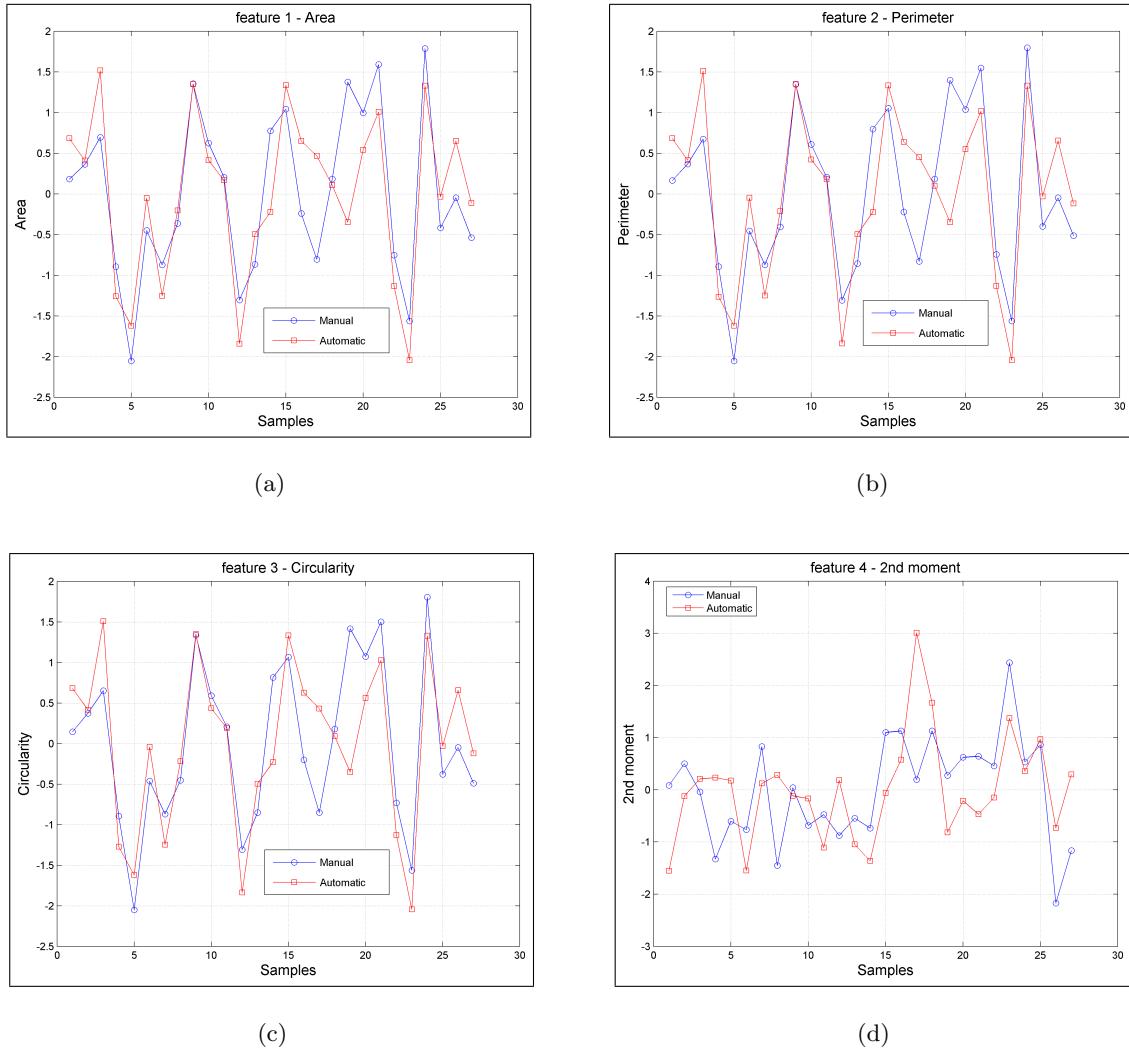


Figura B.1: Comparação entre a seqüência de valores calculados para as 27 imagens por característica: (a) Área, (b) Perímetro, (c) Circularidade e (d) 2º Momento.

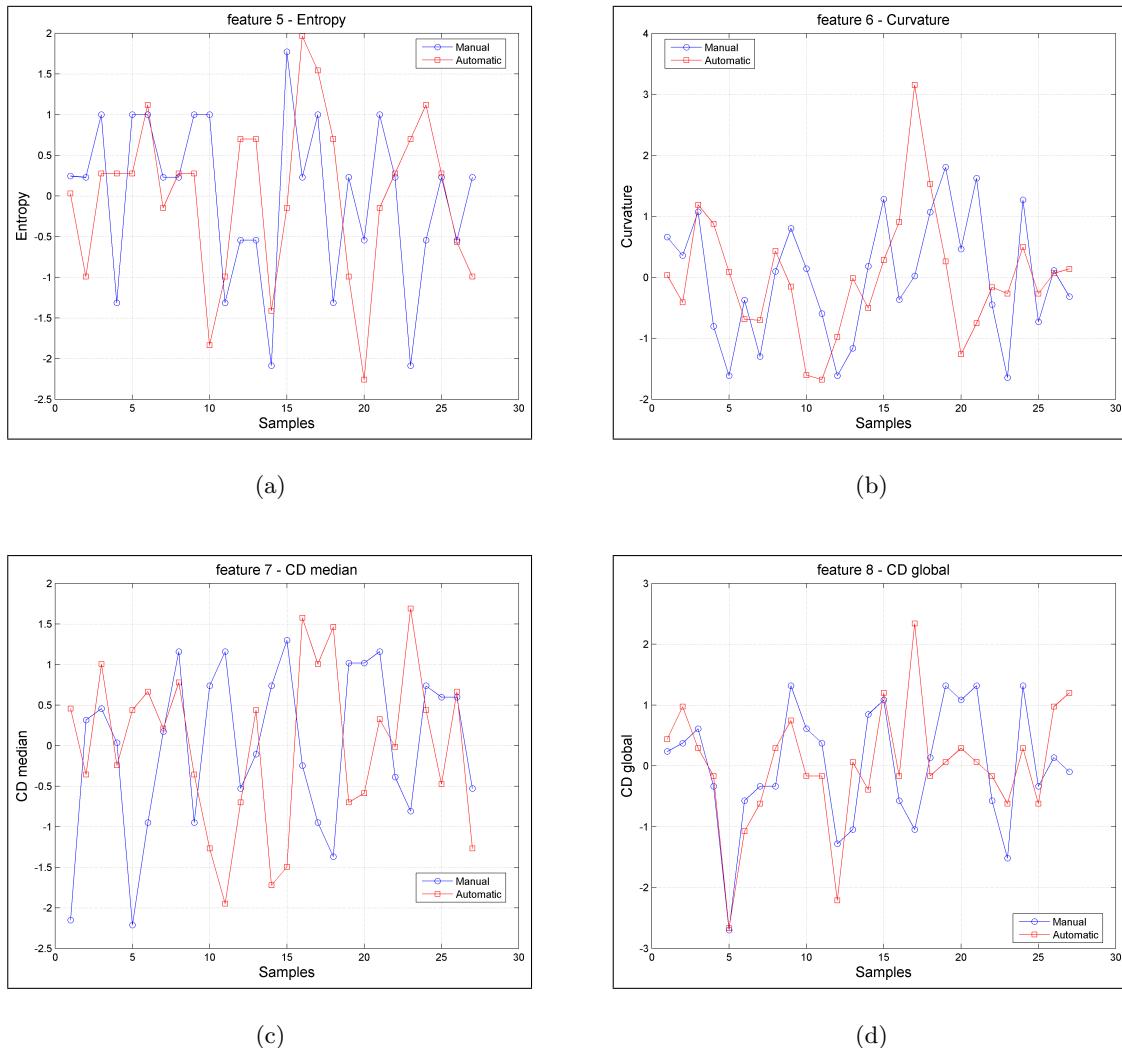


Figura B.2: Comparação entre a seqüência de valores calculados para as 27 imagens por característica: (a) Entropia, (b) Curvatura, (c) Dimensão de Correlação Mediana e (d) Dimensão de Correlação Global.

Integral de Correlação - Pseudocódigo

Algoritmo 1: Estimação da Integral de Correlação

```

1  $V[1 \dots N] \leftarrow \text{GetCoordinates}(\text{image});$ 
2 for  $i \leftarrow 1$  to  $N - 1$  do
3   for  $j \leftarrow i + 1$  to  $N$  do
4      $| d[j] \leftarrow ((V[i].x - V[j].x)^2 + (V[i].y - V[j].y)^2)^{\frac{1}{2}};$ 
5   end
6    $d \leftarrow \text{Sort}(d);$ 
7    $k \leftarrow 1; \epsilon \leftarrow 1;$ 
8   while  $k \leq \text{Length}(d)$  do
9      $| \epsilon \leftarrow \lceil d[k] \rceil;$ 
10    while  $k \leq \text{Length}(d) \wedge d[k] \leq \epsilon$  do
11       $| /* \text{Invariant: } d[1 \dots k] \leq \epsilon \leq \lceil \max(d) \rceil < \text{diag} */$ 
12       $| C[\epsilon] \leftarrow C[\epsilon] + 1; k \leftarrow k + 1;$ 
13    end
14  end
15  $\text{accumulator} \leftarrow 0;$ 
16 for  $i \leftarrow 1$  to  $\text{Length}(C)$  do
17    $| \text{accumulator} \leftarrow \text{accumulator} + C[i]; C[i] \leftarrow \text{accumulator};$ 
18 end
19 for  $i \leftarrow 1$  to  $\text{Length}(C)$  do  $C[i] \leftarrow \left( \frac{1}{N^2} * C[i] \right);$ 

```

Pré-Processamento - Pseudocódigo

Algoritmo 2: Pré-processamento

```
input : A gray level neuron image img
output : A binary image imgSkel of the 8-connected one-pixel wide skeleton, a gray level image
          imgCrossRegions of critical regions and a queue of primary seeds queuePrim

1 imgBin ← Threshold(img);
2 imgSoma ← Erode(imgBin, seDisk);
3 imgSoma ← AreaOpen(imgBin);
4 imgSoma ← Dilate(imgSoma);
5 imgBin ← AreaOpen(Subtract(imgBin,imgSoma));
6 imgBin ← Union(imgBin,imgSoma);
7 imgBin ← Dilate(imgBin);
8 imgSkel ← Thinning(imgBin);
9 imgSkel ← Pruning(imgSkel) ;
10 imgSkel ← HitorMiss(imgSkel,templateMask);
11 imgSkel ← Subtract(imgSkel, imgSoma);
12 imgextpoints ← HitorMiss(Union(imgSkel,imgSoma),extMask) ;
13 imgSeeds ← HitorMiss(imgSkel,extMask) ;
14 imgSeeds ← Subtract(imgSeeds,imgextpoints) ;
15 imgSeeds ← FilterSeeds(imgSeeds) ;
16 imgSkel ← Union(imgSkel,imgSoma);
17 queuePrim ← Find(imgSeeds);
18 imgneighborhood ← neighborhood(imgSkel);
19 imgCrossRegions ← ( imgneighborhood > 2) ;
```

BTA - Pseudocódigo

Algoritmo 3: LabelTree - Rotulador de Árvores Dendríticas

```
input : imgSkel, imgCrossRegions, queuePrim
output : imgLabeled, imgLabeledTree, imgCrossMarks,pointersMap

1 currLabel ← 0 ; currTreeLabel ← 0 ;
2 imgLabeled ← ∅ ; imgLabeledTree ← ∅ ;
3 pointersMap ← ∅ ;
4 maybeSuperposition ← False ;
5 isSuperposition ← False ;
6 outgoingRegion ← False ;
7 geqRight ← False ;
8 foreach seed ∈ queuePrim do
9   currSeed ← dequeue(queuePrim) ;
10  currLabel ← random(currLabel) ;
11  currTreeLabel ← random(currTreeLabel) ;
12  LabelBranch() ;
13 end
```

Algoritmo 4: LabelBranch - Rotulador de Ramos

```

input : imgSkel, imgLabeled, imgLabeledTree, imgCrossMarks, imgCrossRegions, pointersMap,
        currSeed, maybeSuperposition, isSuperposition, outcomingRegion, geqRight
output : imgLabeled, imgCrossMarks, pointersMap

1 myStack ← Ø ;
2 queueSec ← Ø ;
3 myCounter ← 0 ;
4 Dmax ← 10 ;
5 backPointer ← Null ;
6 if queueSec == Ø then
7   | enqueue(currSeed, queueSec) ;
8 end
9 while queueSec ≠ Ø do
10  | LabelBranchQueueSec() ;
11 end

```

Algoritmo 5: LabelBranchQueueSec - Rotula administrando fila secundária

```

input : imgSkel, imgLabeled, imgLabeledTree, imgCrossMarks, imgCrossRegions, pointersMap,
        currSeed, backPointer, maybeSuperposition, isSuperposition, outcomingRegion, geqRight
output : imgLabeled, imgCrossMarks, pointersMap

1 currSeed ← dequeue(queueSec) ;
2 if currSeed ≠ queuePrim then
3   | outcomingRegion ← True ;
4 end
5 backPointer ← Null ;
6 currLabel ← random(currLabel) ;
7 Push(currSeed,myStack) ;
8 while myStack ≠ Ø do
9   | LabelBranchMyStack() ;
10 end

```

Algoritmo 6: IsValid - Verifica se o pixel é válido

```
input  : pixel, imgCrossRegions, imgSkel, imgLabeled
output : bool
1 if IsForeground(pixel) then
2   | if  $\neg$  IsLabeled(pixel) then
3     |   | if  $\neg$  InCR(pixel) then Return True ;
4   | end
5 end
```

Algoritmo 7: LabelBranchMyStack - rotula administrando pilha

```

input : imgSkel, imgCrossRegions, pointersMap, maybeSuperposition, isSuperposition,
        outgoingRegion, geqRight
output : imgLabeled, imgCrossMarks, pointersMap

1 currPix ← Pop(myStack);
2 backPointer ← UpDateBackPointer();
3 if IsValid(currPix) then
4     imgLabeled ← Labeler(); imgLabeledTree ← Labeler();
5     if outgoingRegion then
6         criticalRegion ← ∅; outgoingRegion ← False;
7     else
8         criticalRegion ← ProbeForRegion();
9     end
10    if geqRight then
11        if myCounter < Dmax ∧ ¬ criticalRegion then
12            Push(currPix,stackCR);
13            myCounter ← myCounter +1 ;
14        else
15            if myCounter > Dmax then
16                stackCR ← ∅;
17                else if myCounter ≤ Dmax ∧ criticalRegion then
18                    Push(currPix, stackCR); maybeSuperposition ← True;
19                end
20                myCounter ← 0; geqRight ← False;
21            end
22        end
23        if criticalRegion then
24            outgoingRegion ← True; ;
25             $\hat{v}_0 \leftarrow (| currPix - backPointer |)$  ;
26             $v_0^{tip} \leftarrow currPix$  ;
27            BreadthFirstSearch();
28            if queueRegion then
29                GetVecO();
30                if outVecO then LabelBranchClassAndCont();
31            end
32        else
33            foreach pixel  $p_j \in \text{PixVicinity}(currPix)$  do
34                if IsValid( $p_j$ ) then Push( $p_j$ ,myStack);
35            end
36        end
37 end

```

Algoritmo 8: LabelBranchClassAndCont - Classificação de Região Crítica e Avaliação de Continuidade

```

input : imgSkel, imgCrossRegions, pointersMap
output : imgLabeled, imgCrossMarks, pointersMap

1 ComputeVectorsAndinnerProds();
2 if Length(innerProd) == 2 then
3   /* Classification */                                */
4   /* BIF or SUPERPOSITION */                         */
5   LabelBranchClassAndContA();                         */
6   /* Continuation */                                 */
7   LabelBranchClassAndContB();                         */
8 else
9   /* Classification */                                */
10  /* class: CROSS or close BIFs */                  */
11  LabelBranchClassAndContC();                         */
12 end

```

Algoritmo 9: LabelBranchClassAndContA - parte A

```

1 if ¬ maybeSuperposition then                               */
2   /* class: BIF. Label with 3. */                          */
3   foreach pixel  $P_i \in$  queueCR do                      */
4     | imgCrossMarks  $\leftarrow$  Labeler(imgCrossMarks, 3,  $P_i$ ) ; */
5   end
6 else
7   if isSuperposition then                                */
8     /* class: SUP. Relabel with 4. */                      */
9     foreach pixel  $P_i \in$  queueCR do                      */
10    | imgCrossMarks  $\leftarrow$  Labeler(imgCrossMarks, 4,  $P_i$ ) ; */
11    end
12    /* Relabel path between pseudo BIFs */                 */
13 else
14   /* class: BIF. Label with 3. */                          */
15   foreach pixel  $P_i \in$  queueCR do                      */
16     | imgCrossMarks  $\leftarrow$  Labeler(imgCrossMarks, 3,  $P_i$ ) ;
17     | enqueue(sideBranchRight, queueSec);
18   end
19   maybeSuperposition  $\leftarrow$  False ;
20 end
21 end

```

Algoritmo 10: LabelBranchClassAndContB - parte B

```

1 if isSuperposition then
2   /* update PointersMap */ ;
3   continuationPix ← outVecO [indexinnerMax] ;
4   pointersMap [lastSameLabelPix] ← continuationPix ;
5   pointersMap [continuationPix] ← lastSameLabelPix ;
6 end
7 indexinnerMax ← ArgMax(innerProd) ;
8 indexinnerMin ← ArgMin(innerProd) ;
9 backPointer ← Null; ;
10 outgoingRegion ← True ;
11 continuationPix ← outVecO [indexinnerMax] ;
12 currPix ← continuationPix ;
13 Push(currPix, myStack);
14 if innerProd [indexinnerMin] > 0 then
15   /* Enqueue side branch */ ;
16   if ¬ isSuperposition then
17     enqueue(outVecO [indexinnerMin] ,queueSec);
18   else
19     isSuperposition ← False ;
20 else if ¬ isSuperposition then
21   /* Caution: sidebranch dephased by an angle greater or equal to right angle */ ;
22   geqRight ← True ;
23    $\hat{v}_0^{aux} \leftarrow \hat{v}_0$  ;
24    $\hat{v}^{right} \leftarrow \hat{v}_{indexinnerMin}$  ;
25   sideBranchRight ← outVecO [indexinnerMin] ;
26   lastSameLabelPix ←  $v_0^{tip}$  ;
27 else if isSuperposition then
28   isSuperposition ← False ;
29 end

```

Algoritmo 11: LabelBranchClassAndContC - parte C

```

/* class: supposedly Cross. */  

/* Label with 4. */  

1 foreach pixel  $P_i \in \text{queueCR}$  do  

2   | if  $\text{imgCrossMarks}[P_i] == 0$  then  

3   |   |  $\text{imgCrossMarks} \leftarrow \text{Labeler}(\text{imgCrossMarks}, 4, P_i)$  ;  

4   | end  

5 end  

6  $\text{indexinnerMax} \leftarrow \text{ArgMax}(\text{innerProd})$  ; /* update PointersMap */  

7  $\text{continuationPix} \leftarrow \text{outVecO}[\text{indexinnerMax}]$  ;  

8  $\text{pointersMap}[v_0^{tip}] \leftarrow \text{continuationPix}$  ;  

9  $\text{pointersMap}[\text{continuationPix}] \leftarrow v_0^{tip}$  ;  

10  $\text{currPix} \leftarrow \text{continuationPix}$  ;  

11 Push( $\text{currPix}$ ,  $\text{myStack}$ );  

12  $\text{backPointer} \leftarrow \text{Null}$ ; ;  

13  $\text{outcomingRegion} \leftarrow \text{True}$  ;  

14 if  $\text{innerProd} \neq 1$  then /*  

15   | /* Compute inner products  

16   | forall  $\hat{v}_i, i > 0$  do /*  

17   |   |  $\text{innerProd} 3[i, i + 1] \leftarrow (\hat{v}_i * \hat{v}_{i+1})$  ;  

18   | end /* Look for opposite vectors  

19   | if  $\exists \text{innerProd } 3[i] \leq -0.7$  then /*  

20   |   | /* There are opposite vectors /*  

21   |   | /* Do nothing. It's a cross */  

22   | else /* Class: Two adjacent BIFs /*  

23   |   | /* Relabel with 3 */  

24   |   | foreach pixel  $P_i \in \text{queueCR}$  do /*  

25   |   |   |  $\text{imgCrossMarks} \leftarrow \text{Labeler}(\text{imgCrossMarks}, 3, P_i)$  ; /*  

26   |   |   | enqueue(sideBranchRight, queueSec); /*  

27   | end /* Empty pointers  

28   |  $\text{pointersMap}[v_0^{tip}] \leftarrow \text{Null}$  ; /*  

29   |  $\text{pointersMap}[\text{continuationPix}] \leftarrow \text{Null}$  ; /*  

30 end /*  

31 end /*
```

Algoritmo 12: ComputeVectorsAndinnerProducts - Calcula Produtos Internos

```

input : imgSkel, imgCrossRegions, pointersMap
output : imgLabeled, imgCrossMarks, pointersMap

1 if  $\neg$  maybeSuperposition then
2   foreach pixel  $p_i \in$  queueRegion do
3      $\hat{v}_i \leftarrow (|p_i - \text{outVecO}[i]|)$  ;
4     innerProd [i]  $\leftarrow (\hat{v}_0 * \hat{v}_i)$  ;
5   end
6 else
7   foreach pixel  $p_i \in$  queueRegion do
8      $\hat{v}_i \leftarrow (|p_i - \text{outVecO}[i]|)$  ;
9     innerProd [i]  $\leftarrow (\hat{v}_0^{aux} * \hat{v}_i)$  ;
10    innerProd 2[i]  $\leftarrow (\hat{v}^{right} * \hat{v}_i)$  ;
11  end
12  if  $\exists$  innerProd 2[i]  $\leq -0.7$  then
13    | isSuperposition  $\leftarrow$  True ;
14  end
15 end

```

Apêndice **F**

BSCEA - Pseudocódigo

Algoritmo 13: MyContFol - Extrator de Contorno BSCEA

```

input : imgSkel, imgCrossRegions, pointersMap
output : imgLabeled, imgCrossMarks, pointersMap

1 approxDPC ← True ;
2 [ rows, cols ] ← Size(imgLabeled);
   /* 1st pixel search */ 
3 firstPix ← Find(img); u[1] ← firstPix; ContourSignal ← ∅ ;
   /* 2nd pixel search */
4 found ← 0; chainCode ← 4; bool ← True ;
5 while ¬ found do
6   | up ← MyChainPoint(u[1],chainCode) ;
7   | upn ← MyChainPoint(u[1],MyMod(chainCode +1,8)) ;
8   | if ¬ img [ up ] ∧ img [ upn ] then
9     |   found ← True ;
10    | else
11      |   chainCode ← chainCode +1 ;
12    | end
13 end
14 nextPix ← up; currForePix ← upn; dcn ← chainCode; n ← 2; isCross ← False ;
15 while nextPix ≠ u(1) do
16   | u(n) ← nextPix ;
17   | ContourSignal [ nextPix ] ← 1 ;
18   | dpc ← dcn ;
19   | lastForePix ← currForePix ;
20   | MyFindNext() ;
21   | if isCross then
22     |   isCross ← False ;
23     |   lastTermPix ← ProbeForPointer() ;
24     |   nearestPix ← pointersMap [ lastTermPix ];
25     |   lastPixDir ← FindNextContPix() ;
26     |   nearestneighbor ← nearestPix + lastPixDir ;
27     |   bridgeLine ← BresenhamLine() ;
28     |   ContourSignal ← AppendLine(bridgeLine) ;
29     |   currLabel ← lastLabel ;
30     |   dpc ← dcn ;
31     |   MyFindNext() ;
32   | end
33   | n ← n +1 ;
34 end

```

Algoritmo 14: MyFindNext - Encontra o próximo pixel

```

input : imgSkel, imgCrossRegions, pointersMap
output : imgLabeled, imgCrossMarks, pointersMap

1 dcp ← MyMod( dpc +4,8) ;
2 getFirstPix ← False ;
3 for r ← 0 to 6 do
4   | de ← MyMod( dcp +r,8) ;
5   | dint ← MyMod( dcp +r +1,8) ;
6   | pe ← MyChainPoint(pc,de);
7   | pint ← MyChainPoint(pc,dint) ;
8   | if  $\neg \text{img}[ \text{pe} ] \wedge \text{img}[ \text{pint} ]$  then
9     | | lastLabel ← currLabel ;
10    | | currLabel ← img[ pint ] ;
11    | | currForePix ← pint ;
12    | | if lastLabel  $\neq$  currLabel then
13      | | | nearLabel ← ProbeForLabels() ;
14    | | end
15    | | if lastLabel  $\neq$  1  $\wedge$  currLabel == 1 then
16      | | | Print('next is a soma pixel') ;
17    | | else if lastLabel == 1  $\wedge$  currLabel  $\neq$  1 then
18      | | | Print('last one was a soma pixel') ;
19    | | else if currLabel == 3  $\mid$  lastLabel == 3  $\mid$  nearLabel == 3 then
20      | | | Print('Bif. Contour outwards!') ;
21    | | else
22      | | | Print('Cross or Sup') ;
23      | | | Print('Look for pointer') ;
24      | | | isCross ← 1 ;
25      | | | pn ← pc ;
26      | | | dcn ← dpc ;
27      | | | Return ;
28    | | end
29    /* getFirstPix true for branches only */ *
30    getFirstPix ← True ;
31    if lastLabel == 1  $\wedge$  currLabel == 1 then
32      | | getFirstPix ← False ;
33    end
34    pn ← pe ;
35    dcn ← de ;
36  end
37  if getFirstPix then Break() ;
37 end

```

Caracterização de Regiões Críticas

Seja S o conjunto de *regiões críticas* $S = \{s_1, s_2, s_3\ldots\}$, s_1 sendo a *região crítica* atual e D_{max} o caminho permitido de comprimento mais curto entre duas *regiões críticas* consecutivas.

Seja \hat{v}_0 o vetor unitário **incidente** à s_1 e sejam E_1 e E_2 os conjuntos de vetores unitários **emergentes** das *regiões críticas* próximas s_1 e s_2 respectivamente. Seja $|E_i|$ a cardinalidade de E_i . Então, uma *região crítica* deveria ser classificada em uma das seis classes ilustradas na Fig. 3.11, de acordo com as seguintes regras:

I. Bifurcação 1: Fig. 3.7(a)

$$|E_1| = 2 \text{ e} \\ <\hat{v}_0, \hat{v}_i> \gg 0, \quad \forall \hat{v}_i \in E_1$$

II. Bifurcação 2: Fig. 3.7(b)

$$|E_1| = 2 \text{ e} \\ \exists \hat{v}_i, \hat{v}_j \in E_1 \quad | \\ <\hat{v}_0, \hat{v}_i> > 0 \text{ e } <\hat{v}_0, \hat{v}_j> \leq 0 \text{ e} \\ \nexists s_2 \quad | \quad dist(s_1, s_2) < D_{max}$$

III. Bifurcação 3: Fig. 3.7(c)

$$|E_1| = 2, \text{ mas} \\ \exists s_2 | dist(s_1, s_2) < D_{max} \therefore |E_1| + |E_2| = 3 \\ \exists \hat{v}_j \in s_1 \text{ e } \hat{v}_k, \hat{v}_l \in s_2, \text{ tais que} \\ <\hat{v}_0, \hat{v}_k> \approx 1 \quad \text{e} \quad <\hat{v}_j, \hat{v}_l> \neq -1$$

IV. Bifurcação 4: Fig. 3.7(d)

$$|E_1| = 2, \text{ mas} \\ \exists s_2 | dist(s_1, s_2) < D_{max} \therefore |E_1| + |E_2| = 3 \\ \exists \hat{v}_j \in s_1 \text{ e } \hat{v}_k, \hat{v}_l \in s_2, \text{ tais que}$$

$$\begin{aligned} & \langle \hat{v}_0 \cdot \hat{v}_k \rangle \approx 1 , \quad \langle \hat{v}_j \cdot \hat{v}_l \rangle \neq -1 \\ & \text{e } \quad \langle \hat{v}_0 \cdot \hat{v}_j \rangle \leq 0 \end{aligned}$$

V. Sobreposição: Fig. 3.7(e)

$$\begin{aligned} & |E_1| = 2, \text{ mas} \\ & \exists s_2 | dist(s_1, s_2) < D_{max} : |E_1| + |E_2| = 3 \\ & \exists \hat{v}_j \in s_1 \text{ e } \hat{v}_k, \hat{v}_l \in s_2, \text{ tais que} \\ & \langle \hat{v}_0 \cdot \hat{v}_j \rangle \leq 0, \text{ e} \\ & \langle \hat{v}_0 \cdot \hat{v}_k \rangle \approx 1 \quad \text{e} \quad \langle \hat{v}_j \cdot \hat{v}_l \rangle \approx -1 \end{aligned}$$

VI. Cruzamento: Fig. 3.7(f)

$$\begin{aligned} & |E_1| = 3 \text{ e} \\ & \exists \hat{v}_j, \hat{v}_k, \hat{v}_l \in s_1 \text{ tais que} \\ & \langle \hat{v}_0 \cdot \hat{v}_k \rangle \approx 1 \quad \text{e} \quad \langle \hat{v}_j \cdot \hat{v}_l \rangle \approx -1 \end{aligned}$$

Apêndice **H**

Publicações

Seguem cópias dos principais artigos relacionados a este dissertation, nos quais colaborei como co-autor.

H.1 *Classification of Pathology in Diabetic Eye Disease*

Anexo o trabalho *Classification of Pathology in Diabetic Eye Disease* apresentado no *APRS Workshop on Digital Image Computing - Pattern Recognition and Imaging for Medical Applications, 2005, Brisbane, Griffith University*. Este trabalho descreve um sistema para extração automática de estruturas com padrões ramificados em imagens bidimensionais, obtidas de formas tridimensionais. É apresentado um algoritmo para rastreamento e rotulação de ramos, capaz de transpor regiões de cruzamentos e sobreposições, provendo pleno acesso às regiões mais internas das estruturas ramificadas, consequentemente uma caracterização completa de formas ramificadas. Este trabalho descreve a análise de formas de segmentações manuais de imagens de árvores venosas da retina, obtidas por angiografia fluorescente. Nesta análise, avalia-se um conjunto proposto com 8 características, capaz de identificar o crescimento de novos vasos no plano da retina, além de localizar a área onde a patologia ocorre.

Classification of Pathology in Diabetic Eye Disease

H. F. Jelinek¹, J. Leandro², R. M. Cesar, Jr² and M. J. Cree³

¹School of Community Health

Charles Sturt University, Albury, Australia

²Department of Computer Science

University of Sao Paulo, Brazil

³Dept. Physics and Electronic Engineering

University of Waikato, Hamilton, New Zealand

E-mail: HJelinek@csu.edu.au

Abstract

Proliferative diabetic retinopathy is a complication of diabetes that can eventually lead to blindness. Early identification of this complication reduces the risk of blindness by initiating timely treatment. We report the utility of pattern analysis tools linked with a simple linear discriminant analysis that not only identifies new vessel growth in the retinal fundus but also localises the area of pathology. Ten fluorescein images were analysed using seven feature descriptors including area, perimeter, circularity, curvature, entropy, wavelet second moment and the correlation dimension. Our results indicate that traditional features such as area or perimeter measures of neovascularisation associated with proliferative retinopathy were not sensitive enough to detect early proliferative retinopathy ($SNR = 0.76, 0.75$ respectively). The wavelet second moment provided the best discrimination with a SNR of 1.17. Combining second moment, curvature and global correlation dimension provided a 100% discrimination ($SNR = \infty$).

1 Introduction

In proliferative retinopathy new blood vessels are formed in the retina and emerge from the area of the optic disc and spread towards the macula or emerge from peripheral vessels [16]. Current prevalence of vision impairment due to retinopathy may be as high as 36% in the diabetic community. Timely intervention for diabetic retinopathy lessens the possibility of blindness [14, 18]. Any person with diabetes should expect to undergo ophthalmic examination at least annually. Initial screening and follow up assessment of the retinal fundus of diabetics is carried out by ophthalmologists, which is both expensive and time consuming when

large numbers of patients are examined [2, 19]. In addition barriers to screening in rural and remote areas exist and include distance required to travel, cost of screening and cultural reasons that often lead to indigenous people remaining in their communities rather than seeking health advice in larger urban centres. With advances in digital imaging and the development of computerised grading systems, automated reading and assessment of complications associated with the retinal fundus is becoming more sought after, especially in rural and remote areas.

Ophthalmologists have an 80 to 95% success rate in identifying proliferative retinopathy. This success rate decreases with eye obstruction such as cataract and for identifying earlier stages of proliferation without additional medical history [22]. However non-specialists perform no better than chance (50%). The National Health and Medical Research Council recommend that generally any screening procedure for identifying diabetic retinopathy needs to have a minimum sensitivity of 60% to maximise treatment outcomes and cost-effectiveness [21]. We concentrate on providing an automated procedure to assist in the identification of neovascularisation that meet NHMRC requirements, especially for rural health professionals. Automated reporting of neovascularisation involves the segmentation of the blood vessels from background in the digital image and provides an index of the stage of proliferation.

1.1 Mathematical assessment of optic fundus blood vessels

Research into automated processing of retinal fundus images has mainly concentrated on the identification of microaneurysms associated with preproliferative diabetic retinopathy [9, 15]. Mathematical techniques such as fractal analysis have been used in classification tasks as they

are able to quantify complex branching patterns including blood vessels [8, 17, 12]. Using an automated method that can detect neovascularisation with a minimum sensitivity of 60% is therefore an useful advancement as it would lessen the burden on ophthalmologists during initial population screening. The continuous wavelet transform (CWT) is a powerful and versatile tool that has been applied in many different image processing problems, including shape analysis [11].

2 Methods

2.1 Image Acquisition

Ten fluorescein angiographic retinal images (1024×1024 pixel) were obtained using a Topcon camera linked with Image 2000 software. These images were exported as TIFF images for manual tracing of the retinal vessels using the Object-Image imaging software (<http://rsb.info.nih.gov/nih-image/>) and analysed. Of the ten images five are control images (no disease present) and five are of neovascularisation (the diseased state). See Figure 1 for examples.

2.2 Morphological Feature Extraction

A number of features were measured on the vessel shapes. These included the area a , perimeter p , circularity ($c = p^2/a$) and wavelet fractal inspired measurements. These are described in the following.

2.3 Wavelet Transform Features

The wavelet transform is a mathematical tool that has been used in morphological studies of both 1D and 2D data. Instead of the 1D contour based approach of Cesar & Costa [6], we utilise the 2D approach [3]. The continuous wavelet transform (CWT) $T_\psi(\mathbf{b}, \theta, a)(\mathbf{x})$ of a retinal fundus image $f(\mathbf{x})$, with $\mathbf{x} = (x, y)$ is defined as:

$$T_\psi(\mathbf{b}, \theta, a)(\mathbf{x}) = C_\psi^{-\frac{1}{2}} \frac{1}{a} \int \psi^* (a^{-1} r_{-\theta}(\mathbf{x} - \mathbf{b})) f(\mathbf{x}) d^2 x \quad (1)$$

where C_ψ , ψ , \mathbf{b} , θ and a denote the normalising constant, analysing wavelet, the displacement vector, the rotation angle and the dilation parameter respectively, with the asterisk denoting complex conjugation, and the partial form of the wavelet transform being the position representation [1]. The scale and angle parameters (a and θ respectively) were kept fixed for some *a priori* defined values $a = a_0$ and $\theta = \theta_0$. For the analysing wavelets used in this research we employed the first derivative of the Gaussian function [3]. Therefore, we define two analysing wavelets, $\psi_1(\mathbf{x})$ and

$\psi_2(\mathbf{x})$ as partial derivatives of the Gaussian, viz

$$\psi_1(\mathbf{x}) = \frac{\partial g(\mathbf{x})}{\partial x} \quad \text{and} \quad \psi_2(\mathbf{x}) = \frac{\partial g(\mathbf{x})}{\partial y} \quad (2)$$

where $g(\mathbf{x})$ denotes the 2D Gaussian. By using ψ_1 and ψ_2 as wavelets and the wavelet transform definition in Equation 1, we calculated the gradient wavelet as

$$\mathbf{T}_\psi[f](\mathbf{b}, a) = \begin{pmatrix} T_{\psi_1}[f](\mathbf{b}, a) \\ T_{\psi_2}[f](\mathbf{b}, a) \end{pmatrix} \quad (3)$$

Here, the wavelet transform \mathbf{T}_ψ for each pair (\mathbf{b}, a) is actually a vector whose components are the respective coefficients of the wavelet transform using ψ_1 and ψ_2 as the analysing wavelets. The wavelet gradient is calculated for every pixel in the image.

From the gradient waveform, \mathbf{T}_ψ , we obtain three complementary shape features to characterise the retinal fundus blood vessel pattern, namely the second wavelet moment, 2D curvature and entropy of orientation, calculated only on the pixels located at the boundary of the vessels.

2.4 Second Wavelet Moment

In order to characterise shape complexity we have utilised the modulus of \mathbf{T}_ψ , i.e.,

$$M_\psi[f](\mathbf{b}, a) = |\mathbf{T}_\psi| = \sqrt{(T_{\psi_1})^2 + (T_{\psi_2})^2} \quad (4)$$

providing a histogram that was calculated from the modulus of the wavelet transform M_ψ , for a fixed scale value of a [5]. Taking the frequency count of the i th bin of the histogram as p_i , we define the statistical moment of order q of M_ψ as a shape complexity measure, given by

$$m_q^M = \sum_i i^q p_i \quad (5)$$

and adopted the second moment, namely $q = 2$. The wavelet calculates the gradient vector at a given pixel by looking at a neighbourhood around the pixel.

2.5 Entropy of the Orientation

From \mathbf{T}_ψ and M_ψ the respective orientation of each gradient vector may be easily calculated as the angle associated to each vector and a histogram of gradient versus orientation. In order to quantify this dispersion, we have adopted the entropy s of the orientation distribution,

$$s = - \sum_i p_i \ln p_i \quad (6)$$

where the i now indicate histogram binning with respect to orientation θ .

2.6 2D Curvature

A measure of how the gradient vectors vary locally is obtained from the wavelet transforms that compose the gradient vectors, defined as the 2D curvature. The 2D curvature is defined as:

$$k = \nabla \cdot \frac{\nabla f}{\|\nabla f\|} = \frac{f_{xx}f_y^2 - 2f_xf_yf_{xy} + f_{yy}f_x^2}{(f_x^2 + f_y^2)^{3/2}} \quad (7)$$

where f_x , f_y , f_{xx} , f_{yy} and f_{xy} denote the first partial derivatives of f with respect to x and to y , and the second partial derivatives of f also with respect to x and y . These partial derivatives are estimated using the 2D wavelet transform in the same spirit described above for the gradients

2.7 Correlation Dimension

We utilised the correlation dimension as a complexity measure as previously discussed in the literature [13]. The correlation dimension is defined by

$$D_2 = \lim_{\epsilon \rightarrow 0} \frac{\log_{10} C(\epsilon)}{\log_{10} \epsilon} \quad (8)$$

where $C(\epsilon)$ is the correlation integral calculated with an analysing disc of diameter ϵ . This procedure leads to a graph $C(\epsilon)$ versus ϵ from which a log-log plot-based line fitting is able to estimate the correlation dimension. The linear portion of the log-log slope is determined by two methods. The median correlation dimension is determined by measuring the slope for short segments within the log-log plot and taking the median value of all determined slopes. The global correlation dimension is determined by taking the wavelet transform of the log-log plot using the third derivative of the Gaussian as mother wavelet [7] to establish the end points of the linear region, and calculating the slope from the two end points.

2.8 Statistical Analysis

Basic statistical information, including mean and standard deviation, was calculated on the individual classes (control and neovascularisation) for each of the measured features. Assuming a Gaussian distribution for the underlying probability distribution for each feature measured over each class, we arrived at SNR values, which give an indication of the predictive power, for each individual feature.

To analyse the data for predictive power of the combined features for classification, linear discriminant analysis (LDA) was performed on the matrix representing the data of the images. Features were normalised so that each feature has zero mean and unit standard deviation. LDA

was performed with training and testing with the complete dataset, and with cross-validation by testing on each of the individual images with the classifier trained on the other nine images.

A forward feature selection process using LDA as the classifier was also tried to select the best features. For this we varied the constant term of the LDA discriminant function to bias the discriminant towards one class and then the other class, thus obtaining a series of sensitivity and specificity values for detecting the diseased state (neovascularisation). A receiver operating curve (ROC) was fitted to the specificity and sensitivity values according to the model described by Metz [20], and the area under the curve (AUC), which can be shown to be a monotonically increasing function of the SNR under certain not too restrictive assumptions [4], was used to test the efficacy of the classifier. At each stage the one best feature out of the remaining features was added to the subset of features currently selected and this process was continued until all features were added or no feature improved the classification.

3 Results

Table 1 provides the basic statistical results for each of the features analysed on the ten images. On the naïve assumption of underlying Gaussian probability distributions for each of the classes of each feature the second moment comes out as the best single feature with a SNR of 1.17.

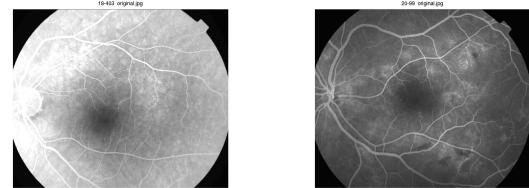


Figure 1. Fluorescein angiographic retinal images of a control patient (left) and with neovascularisation (right)

Figure 1 shows a fluorescein angiographic retinal image of a control and a neovascular retinal fundus. Figure 2 shows for the same images used in Figure 1 the wavelet gradient modulus used in the calculation of the second moment and curvature as 2D output that allows localisation of pathology.

Using LDA as a classifier, trained and tested on the full feature data set, perfect classification was achieved. This result should be treated with scepticism as the discriminant function involved the cancellation of the area, perimeter and circularity terms with each other, then multiplying that re-

Table 1. Statistical properties of the individual classes for the measured features.

Feature	Control Mean \pm SD	Neovascularisation Mean \pm SD	Discriminative Power (SNR)
Area	85000 ± 16000	100000 ± 23000	0.76
Perimeter	33200 ± 6200	39000 ± 8900	0.75
Circularity	$(13.0 \pm 2.4) \times 10^6$	$(15.1 \pm 3.5) \times 10^6$	0.73
2nd moment	4073 ± 19	4100 ± 25	1.17
Entropy	5.57 ± 0.01	5.58 ± 0.02	0.54
Curvature	290.2 ± 4.2	295.0 ± 5.2	1.02
CD ¹ median	1.63 ± 0.08	1.62 ± 0.09	0.05
CD global	1.56 ± 0.04	1.58 ± 0.06	0.54

¹ CD = correlation dimension.

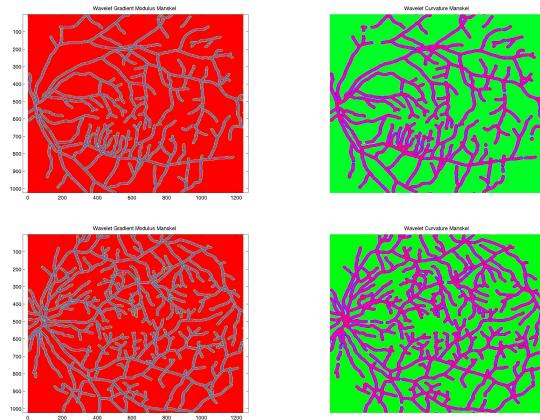


Figure 2. Colour coded image of wavelet gradient modulus (left) and curvature (right) of the control (top) and neovascular (bottom)

sult by five orders of magnitude above the remaining terms. This is the result of training on too few data.

Running the forward selection on the feature data set with LDA as the classifier identified three features, namely circularity, second moment and global correlation dimension, as all that is needed for perfect classification where the classification is both trained and tested on the full data set.

We also examined the utility of a cross-validation on the LDA where the classifier is trained with nine images and tested on the one not included in training. This is repeated for leaving each image in turn out of the training. The results of the cross-validation on the manual dataset (using all features) indicated that all neovascularisation images were correctly classified but two of the control images were not. Interestingly perfect classification is obtained by excluding the ‘area’ and ‘perimeter’ features. The removal of these features, which are highly correlated to each other and to

circularity, has improved the classification.

4 Discussion

Diabetes and its associated complications, including proliferative diabetic retinopathy, has been identified as a significant growing global public health problem. Direct screening programmes such as those based on visits to the ophthalmologist for retinal fundus assessment currently fail to screen between 15 and 62% of patients each year [10]. A large proportion of these people develop potentially sight threatening eye disease, which even at an advanced stage may not cause any symptoms, yet treatment with a laser can prevent visual loss in up to 98% of people if detected early enough [18]. An important step towards reducing the numbers of individuals seriously affected by diabetic retinopathy is to simplify the procedure used to identify the condition and ensure that early eye examinations become routine for all people with diabetes.

Traditional features such as area, perimeter and fractal dimension are not sensitive enough to discriminate between the control retinal fundi and retinal fundi displaying neovascularisation. Our feature analysis suggests that either the second wavelet moment alone or in combination with the curvature and global correlation dimension add to the accuracy of the classification.

This present study is limited in two ways. First, manually obtained vessel segmentations were used for analysis. Ideally the whole process should be automated and studies are currently been undertaken to develop an automated means of segmenting the blood vessels [7]. Second, the number of images used in this study are few, hence a question remains whether the images used truly incorporate the amount of variation present in a large population of retinal images. For this reason we only used LDA for classification. It is planned to repeat the study, with automated vessel segmentation, on a larger corpus of retinal images, and with more powerful classification algorithms.

Acknowledgements: HJ was funded for this project by

Charles Sturt University and the Australian Diabetes Association. RMC is grateful to FAPESP (99/12765-2) and to CNPq (300722/98-2, 468413/00-6)

References

- [1] J. P. Antoine, P. Carette, R. Murenzi, and B. Piette. Image analysis with two-dimensional wavelet transform. *Sig. Proc.*, 31:241–272, 1993.
- [2] R. Ariysau, P. Lee, K. Linton, L. L. Bree, S. Azen, and A. Siu. Sensitivity, specificity and predictive values of screening tests for eye conditions in a clinic-based population. *Ophthalmol.*, 103:1751–1760, 1996.
- [3] A. Arnéodo, N. Decoster, and S. G. Roux. A wavelet-based method for multifractal image analysis: I. Methodology and test applications on isotropic and anisotropic random rough surfaces. *Eur. Phys. J. B*, 15:567–600, 2000.
- [4] H. H. Barrett, C. K. Abbey, and E. Clarkson. Objective assessment of image quality. III. ROC metrics, ideal observers and likelihood-generating functions. *J. Opt. Soc. Am. A*, 15:1520–1535, 1998.
- [5] O. M. Bruno, R. M. Cesar, Jr, L. A. Consularo, and L. da F. Costa. Automatic feature selection for biological shape classification in SYNERGOS. In *Proceedings of the Brazilian Conference on Computer Graphics, Image Processing and Vision (SIBGRAPI)*, pages 363–370, Rio de Janeiro, Brazil, 1998.
- [6] R. M. Cesar, Jr and L. da F. Costa. Neural cell classification by wavelets and multiscale curvature. *Biol. Cybernet.*, 79:347–360, 1998.
- [7] R. M. Cesar, Jr and H. F. Jelinek. Segmentation of retinal fundus vasculature in nonmydriatic camera images using wavelets. In J. S. Suri and S. Laxminarayan, editors, *Angiography and Plaque Imaging*, pages 193–224. CRC, Boca Raton, Fl., 2003.
- [8] D. Cornforth, H. F. Jelinek, and L. Peichl. Fractop: A tool for automated biological image classification. In *Proceedings of the Sixth AI Australasia-Japan Workshop*, pages 141–148, Canberra, Australia, 2002.
- [9] M. J. Cree, J. A. Olson, K. C. McHardy, P. F. Sharp, and J. V. Forrester. A fully automated comparative microaneurysm digital detection system. *Eye*, 11:622–628, 1997.
- [10] M. Cummings. Screening for diabetic retinopathy. *Prac. Diab. Int.*, 19(1):5, 2002.
- [11] L. da F. Costa and R. M. Cesar, Jr. *Shape Analysis and Classification: Theory and Practice*. CRC Press, Boca Raton, FL, 2001.
- [12] A. Daxer. The fractal geometry of proliferative diabetic retinopathy: Implications for the diagnosis and the process of retinal vasculogenesis. *Curr. Eye Res.*, 12:1103–1109, 1993.
- [13] F. Family, B. R. Masters, and D. E. Platt. Fractal pattern formation in human retinal vessels. *Physica D*, 38:98–103, 1989.
- [14] U. Freudentsein and J. Verne. A national screening programme for diabetic retinopathy. *Br. Med. J.*, 323:4–5, 2001.
- [15] J. H. Hipwell, F. Strachan, J. A. Olson, K. C. McHardy, P. F. Sharp, and J. V. Forrester. Automated detection of microaneurysms in digital red-free photographs: a diabetic retinopathy screening tool. *Diabetic Medicine*, 17:588–594, 2000.
- [16] J. Kanski. *Clinical Ophthalmology: A Systematic Approach*. Butterworth-Heinemann, London, 1989.
- [17] G. Landini. Applications of fractal geometry in pathology. In P. M. Iannaccone and M. Khokha, editors, *Fractal Geometry in Biological Systems*, pages 205–245. CRC, Amsterdam, Netherlands, 1996.
- [18] S. J. Lee, C. Sicari, C. A. Harper, H. R. Taylor, and J. E. Keefe. Program for the early detection of diabetic retinopathy: A two year follow-up. *Clin. Exp. Ophthalmol.*, 29:12–25, 2001.
- [19] V. Lee, R. Kingsley, and E. Lee. The diagnosis of diabetic retinopathy: Ophthalmology versus fundus photography. *Ophthalmol.*, 100:1504–1512, 1993.
- [20] C. E. Metz. ROC methodology in radiologic imaging. *Invest. Radiol.*, 21:720–733, 1986.
- [21] National Health and Medical Research Council. *Management of Diabetic Retinopathy Clinical Practice Guidelines*. Australian Government, Canberra, 1997.
- [22] E. Sussman, W. Tsiaris, and K. Soper. Diagnosis of diabetic eye disease. *J. Am. Med. Assoc.*, 247:3231–3234, 1982.

H.2 *Automated segmentation of retinal blood vessels and identification of proliferative diabetic retinopathy*

Anexo o trabalho *Automated segmentation of retinal blood vessels and identification of proliferative diabetic retinopathy*, publicado no *Journal of the Optical Society of America*, 2007. Este trabalho compara os resultados de segmentações manuais e automáticas de imagens de árvores venosas da retina, usando análise de formas, mediante avaliação de um conjunto proposto com 8 características. Além disso, também avalia o poder discriminante do conjunto de características para diferenciar formas de segmentações automáticas de árvores venosas com e sem Retinopatia Diabética Progressiva.

Automated segmentation of retinal blood vessels and identification of proliferative diabetic retinopathy

Herbert F. Jelinek

School of Community Health, Charles Sturt University, Wilson Street, Albury 2640, Australia

Michael J. Cree

Department of Engineering, University of Waikato, Hillcrest Road, Hamilton 3240, New Zealand

Jorge J. G. Leandro, João V. B. Soares, and Roberto M. Cesar, Jr.

Department of Computer Science, Creative Vision Research Group, Instituto de Matemática e Estatística—University of São Paulo, Brazil

A. Luckie

Albury Eye Clinic, Swift Street, Albury 2640, Australia

Received August 15, 2006; revised November 21, 2006; accepted December 20, 2006;
posted January 5, 2007 (Doc. ID 73724); published April 11, 2007

Proliferative diabetic retinopathy can lead to blindness. However, early recognition allows appropriate, timely intervention. Fluorescein-labeled retinal blood vessels of 27 digital images were automatically segmented using the Gabor wavelet transform and classified using traditional features such as area, perimeter, and an additional five morphological features based on the derivatives-of-Gaussian wavelet-derived data. Discriminant analysis indicated that traditional features do not detect early proliferative retinopathy. The best single feature for discrimination was the wavelet curvature with an area under the curve (AUC) of 0.76. Linear discriminant analysis with a selection of six features achieved an AUC of 0.90 (0.73–0.97, 95% confidence interval). The wavelet method was able to segment retinal blood vessels and classify the images according to the presence or absence of proliferative retinopathy. © 2007 Optical Society of America

OCIS codes: 100.5010, 070.5010.

1. INTRODUCTION

Diabetes and its associated complications, including diabetic retinopathy (DR), have been identified as a significant growing global public health problem.¹ Between 22% and 36% of people with diabetes have retinopathy, and of those one-third have vision-threatening retinopathy.² Those with retinopathy will lose vision, leading to potential blindness and loss of quality of life if not treated.³ In proliferative retinopathy, new blood vessels are formed that emerge from the area of the optic disk and spread toward the macula or emerge from peripheral vessels.⁴ Timely intervention for diabetic retinopathy lessens the possibility of blindness.^{5,6}

A. Retinal Blood Vessel Segmentation

A necessary initial step in applying shape analysis to retinal blood vessels is to segment the blood vessels from the background.⁷ Segmentation algorithms for retinal blood vessels are numerous and have been described in detail elsewhere. Different approaches have explored image processing techniques such as matched filters, mathematical morphology, threshold probing, supervised classification,

deformable models, and tracking.^{8–14} We have developed a supervised approach for vessel segmentation based on the continuous wavelet transform (CWT), which has been shown to outperform other state-of-the-art methods with respect to receiver-operating-characteristic (ROC) analysis on public retinal image databases.^{15–17} This segmentation approach is applied in the present paper for the detection of proliferative retinopathy using fluorescein-labeled images of the posterior pole. There is no difference in principle between using fluorescein-labeled images versus images obtained using a color nonmydriatic camera, as the segmentation algorithm is applied to the green channel only. The CWT is a powerful and versatile tool that has been applied in many different image processing problems, from image coding to shape analysis.¹⁸ This success is due largely to the fact that wavelets are especially suitable for detecting edges in signals such as blood vessel borders and for performing fractal and multifractal analysis.^{19–21} Fluorescein angiograms generally require parameter reconfiguration for most methods in the literature. One of the advantages of our approach is that it aims at minimizing the need for reconfiguring the automated segmentation parameters for every fluorescein im-

age by applying a supervised classification procedure that depends only on manually segmented images to compose a training set for segmentation.

B. Classification of Proliferative Diabetic Retinopathy

Research into automated assessment of optic fundus images has concentrated mainly on the identification of features associated with nonproliferative diabetic retinopathy.^{22–25} Interpreting changes in the branching pattern of retinal blood vessels remains a challenge in the assessment of proliferative diabetic retinopathy (PDR). As there is no *a priori* way of identifying appropriate shape features for studying pathological changes in retinal blood vessels, the task remains to search for new shape features that add new explanatory power and diagnostic accuracy. Vessel morphology has been quantified with geometrical/metric and topological parameters such as length, angles, branching order, tree asymmetry, bending energy, fractal dimension, Sholl diagrams, and area of influence.^{8,26–28} Early studies concentrated on the analysis of optic fundus blood vessel patterns using hand-drawn vessel patterns to analyze morphological features. These studies demonstrated that vessel occlusion and neovascularization could be identified using a feature parameter such as fractal dimension.²⁸ The Gabor wavelet used here to segment the blood vessels is very efficient through its wavelet characteristics, reducing overall computation time. The resulting vessel patterns are then characterized by several feature parameters based on the wavelet-derived information.

In the present paper we explore our previously described method for the segmentation of the retinal blood vessels.¹⁷ We then show that the wavelet approach may also be successfully used for describing structural attributes possibly associated with PDR. As PDR is defined by a change in vessel pattern by either loss or addition of vessels within the retina, a skeletonized vessel pattern is sufficient for classification.

The shape features adopted here are based on calculating wavelet gradients from skeleton images, followed by the extraction of meaningful measures, including second wavelet moment, entropy of orientation, curvature, and fractal dimension. Such wavelet shape features have already been successfully employed for the characterization of another type of branching structure, i.e., neural cells.²⁹ Hence, the results reported here show that the wavelet approach is suitable for both segmentation and morphological characterization of retinal blood vessels in the context of proliferative retinopathy. PDR is characterized early by the presence of ischaemic areas, where a loss of blood vessels occurs, and later by new blood vessel formation caused by several angiogenic factors. Thus the blood vessel pattern obtained using the Gabor wavelet for retinae with PDR should be distinctly different from those retinae without PDR.

2. METHODS

A. Vessel Segmentation

1. Image Acquisition

Twenty-seven images (1024×1024 pixel) were obtained using a Topcon camera linked with IMAGE 2000 software.

Of these, 16 were of proliferative retinopathy, and the remaining 11 were without proliferative retinopathy but with background retinopathy and other pathology. These images were exported as TIFF images using the OBJECT-IMAGE imaging software (<http://rsb.info.nih.gov/nihimage/>). The vasculatures were then manually traced and automatically segmented for further analysis.

2. Wavelet Transform for Vessel Segmentation

The wavelet transform is used for two important, different purposes in the context of the present paper: vessel segmentation and morphological characterization. First, we applied the Gabor wavelet transform, which differentiates the blood vessels from the background in conjunction with a training set based on the manually traced training images. The complete process has been described elsewhere and is summarized here.^{17,27}

The property of being well localized both in the time and frequency domains makes wavelets adequate for local filtering and allows detection of localized singularities, such as blood vessels.³⁰ The Gabor wavelet was superimposed onto each pixel of the image at various angles and scales. To detect the blood vessels, for each scale value chosen, the transform was calculated over the range of 0 to 170 deg, at 10 deg steps, and the feature space was updated with the maximum value at each pixel position. Thus, the features used to identify vessel pixels were the maximum transform responses over all angles for different scale values (to span all possible vessel widths) and also the original pixel intensity value. The wavelet parameters associated with the Gabor wavelet (scale, frequency, elongation) were empirically determined in order to reach the best matching between wavelet and vessels. Once determined, the parameter configuration did not have to be changed from image to image and represented the final feature space for vessel segmentation.³¹

A normal transformation was applied to all features to obtain dimensionless values, which allows a comparison between features and avoids mistakes in classification steps, since different units would affect the distance in the feature space.¹⁸ The normal transformation is defined as

$$\hat{v}_i = \frac{v_i - \mu_i}{\sigma_i},$$

where \hat{v}_i is the i th transformed feature, v_i is the i th original feature, μ_i is the average value of the i th feature, and σ_i is the respective standard deviation. With the normal transformation, all features have mean equal to 0 and standard deviation equal to 1 with respect to the training set.

3. Supervised Classification for Vessel Segmentation

The final segmentation was obtained by classifying the original input image pixels into two classes, namely, vessel pixels and nonvessel pixels, according to the supervised classification approach.¹⁸ A Bayesian classifier was adopted in which class likelihoods were described using Gaussian mixture models, providing a fast classification while still allowing complex decision surfaces. The class priors were estimated by the fraction of each class's pixels

present in a training set composed of labeled samples, while the distribution parameters for each class's Gaussian mixture model were estimated from the training set through the application of the expectation-maximization algorithm.³²

The training set was obtained from fundus images manually segmented by an ophthalmologist, providing labeled pixels as training samples. The training samples' features were then normalized as described above and used to train the Bayesian classifier. The segmentation of new images was then performed by applying the classifier, without the need of tuning any additional parameters. The segmentation process can be implemented efficiently using the fast Fourier transform and the Fourier-domain definition of the CWT for pixel feature generation,¹⁷ which takes time $\Theta(N \log_2 N)$, where N is the total number of image pixels. The segmentation is obtained applying the Bayesian Gaussian mixture model classifier to all pixels, taking an additional $\Theta(N)$.

Some misclassified pixels appeared as undesirable noise in the output, and as only boundaries were classified for some vessels, it was necessary to perform postprocessing. The postprocessing operations applied were "area open" to eliminate small noisy components, "dilation" and "area close" to fill the vessels, followed by "skeletonization" to extract the vessel branching pattern. The multiscale skeletonization algorithm based on exact dilations has been applied in this last step.¹⁸ Morphological features were then extracted from the obtained vessel skeletons to distinguish such vessel patterns, as described in the next section.

B. Determination of Morphological Features

Automated classification into retinas with or without PDR requires high accuracy to be able to support screening programs currently carried out by ophthalmologists. Seven morphological features associated with the skeletonized vessel pattern were determined: area, perimeter, circularity, curvature, orientation entropy, second moment of the CWT, and correlation dimension, with physical interpretations described as below. With the exception of the area, perimeter, and circularity, all are based on data obtained from the application of the derivatives-of-Gaussian wavelets to the automated segmented vessel skeletons.

1. Area

The area A estimation of the binary skeletons is obtained by simply counting the number of object pixels, taking linear time, $\Theta(N)$ in asymptotic notation, where N is the number of pixels in the image. It provides an approximation of the shape space occupation.

2. Perimeter

The perimeter P is obtained by counting the number of pixels of the vessel skeleton contour, also in linear time $\Theta(N)$.

3. Circularity

The circularity C is just the ratio between the shape square perimeter P and its area A , i.e., P^2/A . It shows how similar the shape is to a circumference and is rota-

tion invariant, being often adopted as a shape complexity measure.¹⁸ This calculation takes time $\Theta(1)$.

4. Wavelet Transform for Morphological Analysis

Different differential morphological features were obtained, inspired by the shape analysis approach used for characterization of retinal ganglion cells.²⁹ In this paper, we utilized the first derivative of the Gaussian function in order to compose a wavelet gradient. Therefore, we define two analyzing wavelets $\psi_1(x,y)$ and $\psi_2(x,y)$ as partial derivatives of the Gaussian, i.e.,

$$\psi_1(x,y) = \frac{\partial g(x,y)}{\partial x}, \quad \psi_2(x,y) = \frac{\partial g(x,y)}{\partial y},$$

where $g(x,y)$ denotes the two-dimensional (2D) Gaussian.²⁰ By using $\psi_1(x,y)$ and $\psi_2(x,y)$ as wavelets, we can calculate the gradient wavelet as

$$\mathbf{T}_\psi[f](\mathbf{b},a) = \begin{pmatrix} T_{\psi_1}[f](\mathbf{b},a) \\ T_{\psi_2}[f](\mathbf{b},a) \end{pmatrix},$$

where ψ , f , \mathbf{b} , and a denote, respectively, the analyzing wavelet, the analyzed image, the displacement vector, and the scale factor. The wavelet transform $\mathbf{T}_\psi[f](\mathbf{b},a)$ for each pair (\mathbf{b},a) is actually a vector whose components are the respective coefficients of the wavelet transform using $\psi_1(x,y)$ and $\psi_2(x,y)$ as the analyzing wavelets. Since the wavelet transform has been implemented so as to be calculated in the Fourier domain, the calculation of these numerical derivatives take time $\Theta(N \log_2 N)$. The morphological features described below are calculated based on the derivatives-of-Gaussian wavelet.

5. Curvature

The skeleton outline curvature represents how the direction of a unit tangent vector varies along the shape contour. It is described by

$$k = \nabla \cdot \frac{\nabla f}{\|\nabla f\|} = \frac{f_{xx}f_y^2 - 2f_xf_yf_{xy} + f_{yy}f_x^2}{(f_x^2 + f_y^2)^{3/2}}, \quad (1)$$

where f_x , f_y , f_{xx} , f_{yy} , and f_{xy} denote the first partial derivatives of f with respect to x and y , the second partial derivatives of f with respect to x and y , and the partial derivative with respect to x and y , respectively.³³ These partial derivatives are estimated using the 2D wavelet transform in the same spirit described above for the gradients. Thus, given these partial derivatives, the curvature calculations take additional time $\Theta(N)$.

6. Orientation Entropy

The orientation entropy E indicates the orientation disorder degree encountered in a shape and is obtained from the angular distribution of the vector field defined by the wavelet gradient, as follows:

$$E = - \sum_{i \in K} p_i \ln(p_i), \quad (2)$$

where p_i is the occurrence frequency of some vector oriented toward the θ_i direction and K is the set of bins in the histogram. This calculation takes time proportional to

the number of bins in the histogram, that is, $\Theta(|K|)$.

7. Continuous Wavelet Transform Second Moment

The second moment of the CWT modulus is a statistical dispersion measure to indicate biases in the gradient vector field. If q is the CWT modulus histogram, with each bin q_i centered at i , then its CWT second moment is

$$m_2 = \sum_{i \in K} i^2 q_i. \quad (3)$$

Similarly, this feature is calculated in time $\Theta(|K|)$.

8. Correlation Dimension

According to chaos theory, the correlation dimension (CD) measures the dimension of the space occupied by a set of random points, or in other words, it gives the probability of finding two points closer than a certain distance.^{28,34} This probability $C(\varepsilon)$ varies along with the considered distance ε , which ranges between one pixel and the measure of the image diagonal. Usually, one might consider the mean, the median, or the global $C(\varepsilon)$ value. Mathematically, CD is obtained by the correlation integral:

$$C(\varepsilon) = \lim_{N \rightarrow \infty} \frac{2}{N^2} \sum_{i=1, j>i}^N \theta(\varepsilon - |X_i - X_j|), \quad (4)$$

where the Heaviside function is defined as

$$\theta(\varepsilon - |X_i - X_j|) = \begin{cases} 1 & \text{if } (\varepsilon - |X_i - X_j| \geq 0) \\ 0 & \text{if } (\varepsilon - |X_i - X_j| < 0), \end{cases} \quad (5)$$

and the CD itself is taken as the slope of the logarithm of the correlation integral:

$$C(\varepsilon) \propto \varepsilon^{CD}. \quad (6)$$

The CD curve itself is composed of several short straight segments, among which the first and the last should be disregarded, since in those regions the scales ε do not bring significant fractality information from the shape.³⁵

To determine the upper and lower bounds that are the singularities, we calculated the third derivative. The third derivative locates critical points found by the second derivative by searching for zero crossing points.³⁶ Hence, to find these region extremities, that is, singularities, we calculated the third derivative from $C(\varepsilon)$, determining its zero crossing points, where those abrupt changes in slope take place. The third derivative provides a vector of extremities for each segment of the curve. Thus by using the third-derivative information, we ignore the first and the last segments, sort the remaining ones, and take the median value, resulting in the median CD value. The global CD value is obtained by disregarding the first slope segment and the last one and then by considering the straight line between the points related to the second zero crossing and the penultimate point in order to evaluate the global slope. The correlation integral [Eq. (4)] takes time $\Theta(N^2)$, being the most time-consuming feature calculation, while its derivative calculations take an additional $\Theta(N \log_2 N)$.

C. Statistical Analysis

Results for the features associated with automated and manual segmentations were compared using a two-sample *t*-test with unequal variance to test the hypothesis that the means were not significantly different. Linear discriminant analysis (LDA) was performed on the matrix representing the data for each of the images (no retinopathy/background retinopathy versus PDR). Matrices were normalized so that each feature has zero mean and unit standard deviation followed by linear regression

A forward-backward feature-selection process using LDA as the classifier was also applied to select the best features. This process can be shown to be a monotonically increasing function of the signal-to-noise ratio and, using the area under the curve (AUC), indicates the effectiveness of the feature in discriminating between the two groups.

3. RESULTS

The classifier output obtained from the original image is a binary image with labeled pixels as vessel or nonvessel. This binary image is postprocessed and skeletonized, producing the results as illustrated in Fig. 1.

Figure 2 compares the outcome of the pattern analysis for orientation entropy and correlation integral for an image with and one without proliferative retinopathy.

Table 1 lists statistics of the seven features calculated between the manual and the automated segmented images. Manual segmented images resulted in larger feature parameters, except for circularity. A student *t*-test showed that the differences between the means of all features are statistically significant ($p < 0.001$), except for the entropy ($p = 0.38$). The differences arise because more vessels are identified by the observer than by the automated procedure.

Since we are interested primarily in automated detection of PDR, we proceed to further analyze the automated data only. These data were normalized so that each feature has zero mean and standard deviation of one. The ability to use each feature individually to discriminate between non-PDR and PDR is summarized in Table 2. Only the curvature feature achieves statistical significance ($p = 0.042$) for a two-sampled *t*-test for the difference in means between the non-PDR and the PDR classes. However we are more interested in classification ability, and the AUC of the parametric binormal receiver operating characteristic (ROC) was also calculated.³⁷ The traditional features (area, perimeter, and circularity) and the global CD are essentially random and thus have no classification ability (AUC = 0.49 in all cases). The wavelet-based features (second moment, entropy, curvature, and median CD) have some useful classification ability, with AUCs of about 0.7. This indicates that these latter features are indeed more sensitive to the blood vessel branching patterns throughout the retina.

The above analysis considered features in isolation. A combination of features with an appropriate classifier may provide better predictive ability. We first tried forward-backward selection of features using the linear discriminant and naïve-Bayes classifiers.³² The AUC calculated from the resultant classifier was used as the cri-

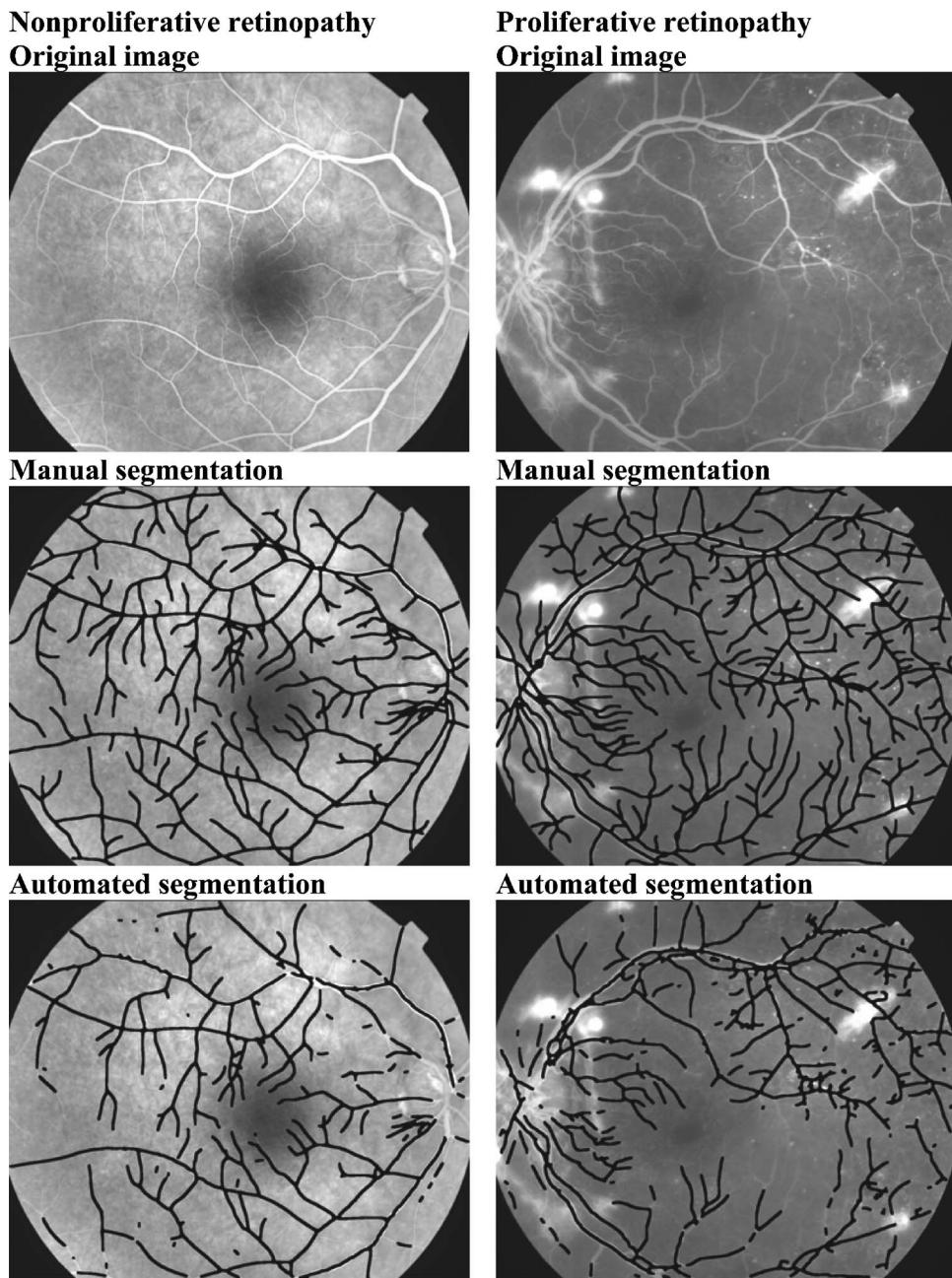


Fig. 1. Illustration of outcome for automated and manual segmentation compared with the original gray-scale image. Overlaid skeletonized vessel pattern, in black, has been dilated for easier viewing.

terion for selection or removal of features. We report AUCs with 95% confidence intervals (CIs) as calculated by the ROCKIT program.³⁷ The best features identified using forward-backward selection on the LDA classifier were median CD and wavelet second moment with an AUC of 0.77 (0.56–0.91). Starting with all features and removing any that improved classification led to a set of six features (area, perimeter, entropy, curvature, median CD, and global CD), with the LDA classifier achieving an AUC of 0.90 (0.73–0.97). As we are interested in the improvement provided by the wavelet-based features, we also ran LDA on the feature set comprising area, perimeter, and circularity only, which achieved an AUC of 0.61 (0.37–0.82). Note that LDA-based classification using traditional features (area, perimeter, and circularity) fails sta-

tistical significance for better than random classification at 95% confidence, whereas the wavelet-based feature achieves a statistically significant power to discriminate between PDR and non-PDR.

The naïve-Bayes classifier shows similar results. Using just area, perimeter, and circularity, a naïve-Bayes classifier has an AUC of 0.58 (0.37–0.78) and fails statistical significance for better than random discriminatory power. Using the best four features (area, second moment, entropy, and global CD) identified by forward-backward selection gives an AUC of 0.87 (0.67–0.96), which has statistically significant and useful discriminatory power. For the naïve-Bayes classifier, as with LDA, it is necessary to include some of the wavelet-based features to discriminate between PDR and non-PDR.

We show in Fig. 3 the full ROC curve (fitted to the data with the parametric binormal model due to Metz) for the LDA classifier operating on the best six features identified above and the ROC curve fitted to the LDA classifier operating on the traditional three features.

Using the best six features and the LDA classifier at the operating point marked with an asterisk on the ROC curve (Fig. 3), three images were incorrectly classified. Of these, one was opaque and of bad quality, while the second had a venous loop present. The third was of good quality with neovascularization present both near the disk and in the periphery. Thus the classifier achieved 15 true positives out of 16 images (sensitivity of 94%) and 2 false positives out of 11 negatives (false-positive rate of 18%), giving a specificity of 82% using automatically segmented images.

4. DISCUSSION

The importance of our work lies in combining automated segmentation procedures with automated recognition of proliferative retinopathy using features determined from the wavelet-derived results such as second wavelet moment, entropy, and median CD. In addition we have aimed to use a clinically relevant sample that included images where the quality was not optimal and those with panretinal laser surgery scars, diverse progression of retinopathy, and proliferation showing either as areas of ischaemia or as new blood vessels near the optic disk or in the periphery. The results obtained were produced using the same configuration parameters for all images; preprocessing and optic disc removal to improve the vessel segmentation were also not required owing to the nature of

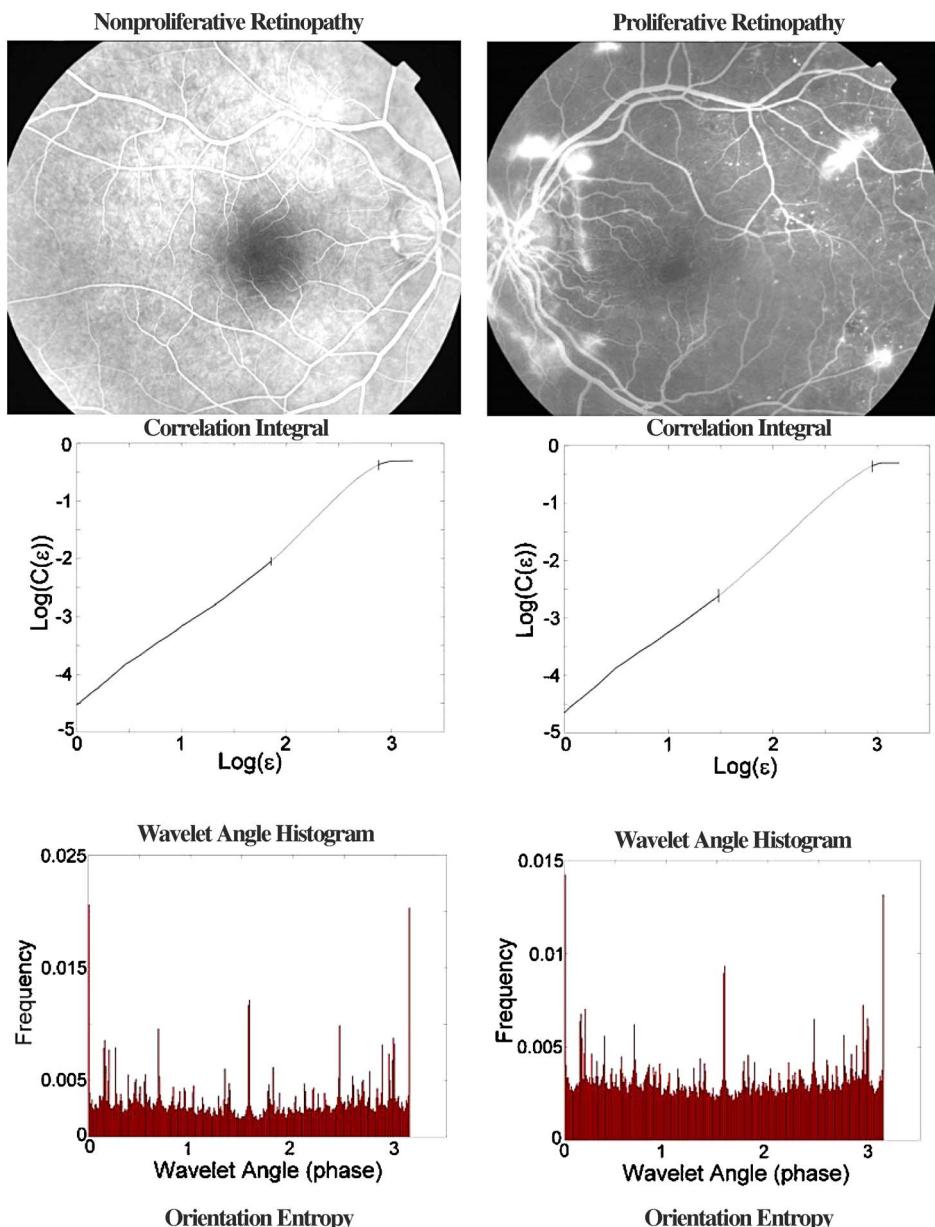


Fig. 2. (Color online) Correlation integral and wavelet histogram for the orientation entropy comparing an image with neovascularization to one with no neovascularization. Vertical lines on the log-log correlation integral indicate upper and lower bounds (see text).

Table 1. Feature Comparison for Manual and Automated Segmented Images

Feature	Manual Segmentation (Mean±SD) ^a	Automated Segmentation (Mean±SD)	t-Statistic	p-Value
Area	102,000±23,000	24,600±5200	-17.0	<0.0001
Perimeter	31,200±5200	24,400±8400	-6.74	<0.0001
Circularity	13,500±3000	24,300±5100	9.38	<0.0001
Second moment	4362±25	1080±24	-493	<0.0001
Entropy	5.63±0.01	5.62±0.02	-0.9	0.38 ^b
Curvature	178.0±3.9	133.6±4.3	-39.9	<0.0001
Median CD ^c	1.67±0.1	1.42±0.1	-11.8	<0.0001
Global CD	1.58±0.04	1.49±0.04	-8.2	<0.0001

^aMean ± standard deviation.^bNonsignificant.^cCorrelation dimension.**Table 2. Comparison between Nonproliferative and Proliferative Diabetic Retinopathy for the Eight Features Analyzed**

Feature	Non-PDR	PDR	t-Statistic	p-Value	AUC ^a
Area	0.038±0.82	-0.026±1.1	0.169	0.87	0.49
Perimeter	0.040±0.82	-0.027±1.1	0.178	0.86	0.49
Circularity	0.041±0.82	-0.029±1.1	0.186	0.85	0.49
Second moment	-0.35±0.63	0.24±1.1	-1.74	0.095	0.72
Entropy	-0.41±1.3	0.28±0.65	-1.64	0.12	0.70
Curvature	-0.46±0.85	0.31±1.0	-2.15	0.042	0.76
Median CD ^b	-0.44±1.1	0.30±0.87	-1.93	0.068	0.70
Global CD	-0.065±0.82	0.45±1.1	-0.292	0.77	0.49

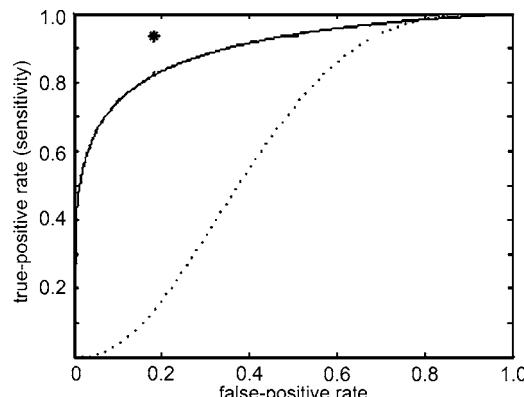
^aArea under curve.^bCorrelation dimension.

Fig. 3. ROC curve for automated segmented images using the set of six best features. * is operating point; solid curve, six best features; dashed curve, traditional features (area, perimeter, circularity).

the wavelet transform. This is a major step forward in automated segmentation and analysis, as it decreases the computation time and operator involvement.¹⁰ Nevertheless differences between manual and automated segmentation can be noted (Fig. 1). Quality of an image associated either with methodology or with pathology influences not only automated methods but also manual segmentation, as can be seen in the STARE database (<http://www.parl.clemson.edu/stare/>). These errors are,

however, randomly distributed across the nonproliferative and proliferative retinopathy images and should not have influenced the outcome of our classification.

Previous work comparing the CWT with other segmentation methods using the STARE and DRIVE database images as the gold standard showed that supervised learning and adaptive thresholding obtained a greater than 75% sensitivity for vessel segmentation.¹⁵ A comparison of six segmentation methods, including our CWT, indicated a similar outcome, with the sensitivity and specificity being greater than 80%.¹⁶

Ophthalmologists have an 80% to 95% success rate in identifying proliferative retinopathy. This success rate decreases with eye media opacification and for identifying earlier stages of proliferation. However nonspecialists perform no better than chance (50%).³⁸ Traditional features such as area or perimeter are not sensitive enough to discriminate between the nonproliferative retinopathy and the retinal fundi displaying neovascularization. This is because area and perimeter pick up only the total amount of vasculature, whereas PDR involves a reconfiguration of the vasculature. It is necessary therefore to use features that are sensitive to the branching and space-filling nature of the vessel pattern, such as the wavelet-based features that we employed.

Our feature analysis suggests that utilizing the six best features, which include median correlation dimension, curvature, and entropy, provides the most accurate clas-

sification, with 94% sensitivity and 82% specificity. The three images classified incorrectly included one with opacity, while another had advanced nonproliferative retinopathy, and the third had a venous loop and Hollenhorst plaques in addition to the presence of diabetic retinopathy. As such, including additional image analysis tools such as a microaneurysm detector may improve these findings. The use of digital nonmydriatic color images for routine community screening has been shown to be effective but will lead to a further challenge for detecting early-stage proliferative retinopathy using automated methods. In addition, classifiers other than the LDA or naïve-Bayes reported here may also improve the classification results, and this is currently being investigated on a larger set of images.

One of the advantages of our morphological feature set is that it identified images that contain neovascularization within the retina but not in the view of the posterior pole, suggesting that peripheral changes associated with neovascularization lead to a remodeling throughout the retina.

ACKNOWLEDGMENTS

Herbert Jelinek was in receipt of a Charles Sturt University grant (A514 713 9661). Roberto M. Cesar, Jr., and João V. B. Soares are grateful to State of São Paulo Research Foundation (FAPESP) (2005/00587-5 and 2006/56128-1) and to National Council of Technological and Scientific Development (CNPq) (300722/98-2, 474596/2004-4, 131403/2004-4, and 491323/2005-0). We thank Cherryl Kolbe and Bev deJong for technical assistance. Chris McQuellin was responsible for the retinal photography.

Corresponding author Herbert Jelinek can be reached by e-mail at hjelinek@csu.edu.au.

REFERENCES

- H. R. Taylor and J. E. Keeffe, "World blindness: a 21st century perspective," *Br. J. Ophthalmol.* **85**, 261–266 (2001).
- K. Viswanath and M. McGavin, "Diabetic retinopathy: clinical findings and management," *Community Eye Health* **16**, 21–24 (2003).
- K. S. Coyne, M. K. Margolis, T. Kennedy-Martin, T. M. Baker, R. Klein, M. D. Paul, and D. A. Revicki, "The impact of diabetic retinopathy: perspectives from patient focus groups," *Fam. Pract.* **21**, 447–453 (2004).
- J. Kanski, *Clinical Ophthalmology: A Systematic Approach* (Butterworth-Heinemann, 1989), pp. 465–479.
- U. Freudentsein and J. Verne, "A national screening programme for diabetic retinopathy," *BMJ* **323**, 4–5 (2001).
- S. J. Lee, C. Sicari, C. A. Harper, H. R. Taylor, and J. E. Keeffe, "Program for the early detection of diabetic retinopathy: a two-year follow-up," *Clin. Exp. Optom.* **29**, 12–25 (2001).
- T. Walter, J. C. Klein, P. Massin, and F. Zana, "Automated segmentation and registration of retinal fluorescein angiographies," presented at the First International Workshop on Computer Assisted Fundus Image Analysis (CAFIA), Herlev Hospital, Copenhagen, Denmark, May 29–30, 2000.
- X. W. Gao, A. Bharath, A. Stanton, A. Hughes, N. Chapman, and S. Thorn, "Quantification and characterisation of arteries in retinal images," *Comput. Methods Programs Biomed.* **63**, 133–146 (2000).
- A. Hoover, V. Kouznetsova, and M. Goldbaum, "Locating blood vessels in retinal images by piecewise threshold probing of a matched filter response," *IEEE Trans. Med. Imaging* **19**, 203–210 (2000).
- F. Zana and J.-C. Klein, "Segmentation of vessel-like patterns using mathematical morphology and curvature evaluation," *IEEE Trans. Image Process.* **10**, 1010–1019 (2000).
- C. Kirbas and F. Quek, "A review of vessel extraction techniques and algorithms," *ACM Comput. Surv.* **32**, 81–121 (2004).
- X. Jiang and D. Mojon, "Adaptive local thresholding by verification-based multithreshold probing with application to vessel detection in retinal images," *IEEE Trans. Pattern Anal. Mach. Intell.* **25**, 131–137 (2003).
- J. J. Staal, M. D. Abràmoff, M. Niemeijer, M. A. Viergever, and B. van Ginneken, "Ridge based vessel segmentation in color images of the retina," *IEEE Trans. Med. Imaging* **23**, 501–509 (2004).
- D. Nain, A. Yezzi, and G. Turk, "Vessel segmentation using a shape driven flow," presented at the Medical Image Computing and Computer-Assisted Intervention (MICCAI), Saint-Malo, France, September 26–29, 2004.
- D. Cornforth, H. F. Jelinek, J. J. G. Leandro, J. V. B. Soares, R. M. Cesar, Jr., M. J. Cree, P. Mitchell, and T. Bossomaier, "Evolution of retinal blood vessel segmentation using wavelet transform in assessment of diabetic retinopathy," *Complexity International* **11**, 50–60 (2005), <http://www.complexity.org.au/ci/vol11/>.
- M. J. Cree, J. J. G. Leandro, J. V. B. Soares, R. M. Cesar, Jr., H. F. Jelinek, and D. Cornforth, "Comparison of various methods to delineate blood vessels in retinal images," in *Proceedings of the 16th Australian Institute of Physics Congress* (AIP, 2005) pp. 1–4.
- J. V. B. Soares, J. J. G. Leandro, R. M. Cesar, Jr., H. F. F. Jelinek, and M. J. Cree, "Retinal vessel segmentation using the 2-D Gabor wavelet and supervised classification," *IEEE Trans. Med. Imaging* **25**, 1214–1222 (2006).
- L. F. Costa and R. M. Cesar, Jr., *Shape Analysis and Classification: Theory and Practice* (CRC, 2001).
- J. P. Antoine, D. Barache, R. M. Cesar, Jr., and L. F. Costa, "Shape characterization with the wavelet transform," *Signal Process.* **62**, 265–290 (1997).
- A. Arnéodo, N. Decoster, and S. G. Roux, "A wavelet-based method for multifractal image analysis. I. Methodology and test applications on isotropic and anisotropic random rough surfaces," *Eur. Phys. J. B* **15**, 567–600 (2000).
- A. Grossmann, "Wavelet transforms and edge detection," in *Stochastic Processes in Physics and Engineering*, S. Albeverio, P. Blanchard, M. Hazewinkel, and L. Streit, eds. (Reidel, 1988), pp. 149–157.
- M. J. Cree, J. A. Olson, K. McHardy, P. Sharp, and J. Forrester, "A fully automated comparative microaneurysm digital detection system," *Eye* **11**, 622–628 (1997).
- J. Hipwell, F. Strachan, J. Olson, K. McHardy, P. Sharp, and J. Forrester, "Automated detection of microaneurysms in digital red-free photographs: a diabetic retinopathy screening tool," *Diabetic Med.* **17**, 588–594 (2000).
- C. Sinthanayothin, J. F. Boyce, H. Cook, and T. Williamson, "Automated localisation of the optic disc, fovea and retinal blood vessels from digital colour fundus images" *Br. J. Ophthalmol.* **83**, 902–912 (1999).
- A. Osareh, M. Mirmehdi, B. Thomas, and R. Markham, "Automated identification of diabetic retinal exudates in digital colour images," *Br. J. Ophthalmol.* **87**, 1220–1223 (2003).
- M. E. Martinez-Perez, A. D. Hughes, A. V. Stanton, S. A. Thom, N. Chapman, A. A. Bharath, and K. H. Parker, "Retinal vascular tree morphology: a semi-automatic quantification," *IEEE Trans. Biomed. Eng.* **49**, 912–917 (2002).
- R. M. Cesar, Jr., and H. F. Jelinek, "Segmentation of retinal fundus vasculature in nonmydriatic camera images using wavelets," in *Angiography and Plaque Imaging*, J. S. Suri and S. Laxminarayan, eds. (CRC, 2003), pp. 193–224.
- B. R. Masters, "Fractal analysis of the vascular tree in the

- human retina," *Annu. Rev. Biomed. Eng.* **6**, 427–452 (2004).
- 29. H. F. Jelinek, R. M. Cesar, Jr., and J. J. G. Leandro, "Exploring wavelet transforms for morphological differentiation between functionally different cat retinal ganglion cells," *Brain and Mind* **4**, 67–90 (2003).
 - 30. J. P. Antoine, P. Carette, R. Murenzi, and B. Piette, "Image analysis with two-dimensional wavelet transform," *Signal Process.* **31**, 241–272 (1993).
 - 31. J. J. G. Leandro, J. V. B. Soares, R. M. Cesar, Jr., and H. F. Jelinek, "Blood vessel segmentation of non-mydriatic images using wavelets and statistical classifiers," in *Proceedings of the Brazilian Conference on Computer Graphics, Image Processing and Vision* (Sibgrapi, 2003), pp. 262–269.
 - 32. R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2nd ed. (Wiley Interscience, 2001).
 - 33. L. F. Estrozi, L. G. Rios, A. G. Campos, R. M. Cesar, Jr., and L. d. F. Costa, "1D and 2D Fourier-based approaches to numeric curvature estimation and their comparative performance assessment," *Digit. Signal Process.* **13**, 172–197 (2003).
 - 34. P. Asvestas, G. K. Matsopoulos, and K. S. Nikita, "Estimating of fractal dimension of images using fixed mass approach," *Pattern Recogn. Lett.* **20**, 347–354 (1999).
 - 35. J. P. Rigaut, "An empirical formulation relating boundary lengths to resolution in specimens showing 'non-ideally fractal' dimensions," *J. Microsc.* **133**, 41–54 (1984).
 - 36. R. C. Gonzalez, R. E. Woods, and S. L. Eddins, *Digital Image Processing Using Matlab* (Pearson Prentice Hall, 2004).
 - 37. C. E. Metz, "ROC methodology in radiologic imaging," *Invest. Radiol.* **21**, 720–733 (1986).
 - 38. E. Sussman, W. Tsiaras, and K. Soper, "Diagnosis of diabetic eye disease," *J. Am. Med. Assoc.* **247**, 3231–3234 (1982).

H.3 *Determining the branching structure of 3D structures from respective 2D projections*

Anexo o trabalho *Determining the branching structure of 3D structures from respective 2D projections* apresentado no *XIX SIBGRAPI - Brazilian Symposium on Computer Graphics and Image Processing*. Este trabalho descreve um sistema para extração automática de estruturas com padrões ramificados em imagens bidimensionais, obtidas de formas tridimensionais. É apresentado um algoritmo para rastreamento e rotulação de ramos, capaz de transpor regiões de cruzamentos e sobreposições, provendo pleno acesso às regiões mais internas das estruturas ramificadas, consequentemente uma caracterização completa de formas ramificadas.

Determining the branchings of 3D structures from respective 2D projections

Jorge J. G. Leandro, Roberto M. Cesar Jr.
Institute of Mathematics and Statistics - USP
Department of Computer Science
Rua do Matão, 1010 São Paulo, SP
05508-090 Brazil
{jleandro,cesar}@vision.ime.usp.br

Luciano da Fontoura Costa
Instituto de Física de São Carlos - USP
Department of Physics and Informatics
Av. Trabalhador Säocarlense, 400 São Carlos, SP
13560-970 Brazil
luciano@ifsc.usp.br

Abstract

This work describes a new framework for automatic extraction of 2D branching structures images obtained from 3D shapes, such as neurons and retinopathy images. The majority of methods for neuronal cell shape analysis that are based on the 2D contours of cells fall short of properly characterizing such cells because crossings among neuronal processes constrain the access of contour following algorithms to the innermost regions of the cell. The framework presented in this article addresses, possibly for the first time, the problem of determining the continuity along crossings, therefore granting to the contour following algorithm full access to all processes of the neuronal cell under analysis. First, the raw image is preprocessed so as to obtain an 8-connected, one-pixel wide skeleton as well as a set of seed pixels for each subtree and all the branching/crossing regions. Then, for each seed pixel, the algorithm labels all valid neighbors, until a branching/crossing region is reached, when a decision about the proper continuation is taken based on the tangent continuity. The algorithm has shown robustness for images with parallel segments and low densities of branching/crossing densities. The problem of too high densities of branching/crossing regions can be addressed by using a suitable data structure. Successful experimental results using real data (neural cell images) are presented.

1. Introduction

One of the most important implications of the intensive investigations in genetics and biomolecular sciences along the last decades has been the realization that more complete understanding and control of phenotypic features of individuals can not be fully achieved without effective characterization of such features as well as the consideration of influences of the external and internal environment where cells

and tissues develop (e.g. [4]). Whole new areas, including *post-genomics*, *neuroinformatics*, and *systems biology* have ultimately derived from the intensifying efforts in investigating such fundamental questions.

Few biological systems provide such a challenging and rich laboratory for relating phenotypic characteristics to genetics and animal development as the nervous system. Indeed, the ever changing shapes of neurons are closely related not only to the genetic cell content, but also to external stimuli and internal biochemical and anatomical changes. Already important as a subsidy for diagnosis, the characterization of *neuronal shape* has progressively established itself as a key resource for several investigations in biology and neuroscience. Indeed, it is only by obtaining a precise and comprehensive representation and characterization of the shapes of the neuronal cells under analysis that more objective and quantitative efforts can be made so as to relate neuronal phenotype with genetics, phylogenetics, comparative neurology and animal development. Another important problem related to neuronal shape concerns the shape-function paradigm (e.g. [5]), which addresses how the structure of neuronal cells would be related to their respective function. By being prototypically complex, neuronal cell shapes also constitute a particularly interesting type of data for shape analysis.

Despite its key role in so many areas, it was only more recently that the endeavor of neuronal cell shape analysis started to establish itself as an important research area on itself. The intensification of related research efforts along the last years has significantly contributed to formalizing and developing a large number of measurements and models of neuronal shape, to the point that it becomes difficult to provide a comprehensive review of this area in the current work. Among the several approaches aimed at characterizing the geometry and connectivity of neuronal cells [2, 8, 9], those based on the contours of the cells provide particularly effective means for measuring and characterizing the respective shape as a consequence of the mapping from the

2D (or 3D) spaces where the cells were imaged into 1D parametric curves describing the outline of the cells (Figure 1). Among other possibilities, such curves can be used in order to infer the normal and/or tangent orientation fields along the cell contours, as well as the estimation of the respective curvature (e.g. [3]), which can provide particularly valuable information about the local degree of bending of the curve as well as its concavity. However, such approach is often limited by the presence of crossings between the neuronal processes, implying some regions of the cell to become inaccessible for contour extraction.

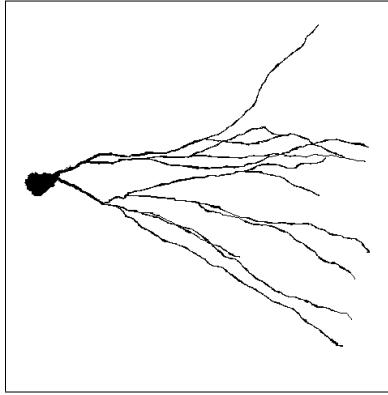


Figure 1. Example of neuron image considered in this work.

Usually, the contour [7] obtained from a *shape* is understood as a parameterized curve, from which several measurements, such as curvature, can be obtained. A typical shortcoming in extracting contours from branching structures consists in the fact that the chain-code-based contour following algorithm (Figure 2) can not traverse regions delimited by crossings (due to the 3D to 2D projection). As a result, only the outer contour of the cell is obtained, while the innermost structures remain unaccessible (fig 3). In other words, the contour following algorithm based in the chain-code can not deal with non-Jordan curves [1].

The current work is aimed precisely at solving such a problem, so that more complete parametric representations of the cell shape can be obtained and analyzed. This is achieved through the incorporation of several criteria, with particular emphasis given to ensuring the continuity of the tangent orientation as the means to identify the proper continuation of the neuronal processes as they go through crossing-points. The introduced algorithm represents the main original contribution of the present paper. To circumvent the aforementioned shortcoming in branching structures contour following, we propose an algorithm devised

1	8	7
2	C	6
3	4	5

Figure 2. Chain-code illustration. The neighborhood of the pixel c is scanned according to the increasing sequence 1..8.

to separate crossing branches within a branching structure, namely the *Branch Tracking Algorithm (BTA)*, so allowing the chain-code contour following algorithm to yield a proper contour for most neuronal shapes.

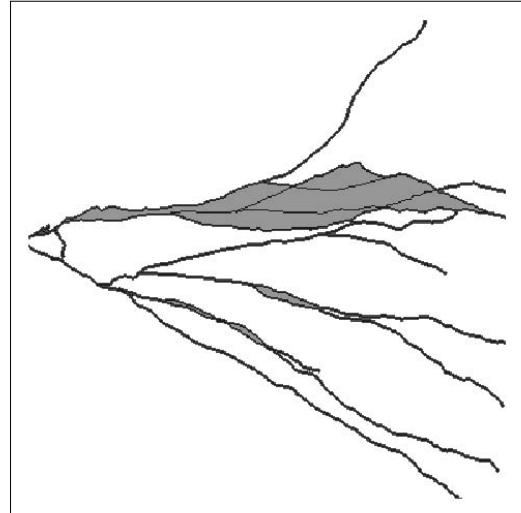


Figure 3. Neuron skeleton (black) and respective contour (dark gray). The light gray shaded areas are unreachable for the contour following algorithm based on the chain-code.

This paper is organized as follows. Section 2 presents an overview of the proposed framework, which is subsequently detailed in Section 3. Experimental results obtained from the application of the proposed algorithm to images of real neural cell are presented in Section 4. The paper is concluded with some discussion on the obtained results and our ongoing work in Section 5.

2. Concepts and Overview

In general, an image obtained by means of an imaging device such as a photograph camera is a two-dimensional projection from a three-dimensional shape, hence lacking information regarding depth. Particularly, when dealing with *complex shape images*, such as neurons and retinal images, depth information becomes crucial to set apart branches that seem to intersect one another. This is the main problem addressed in the present paper.

A **branching structure** is a binary image of the 8-connected one-wide pixel skeleton, obtained from a *shape image*, where *bifurcation regions* and *crossing regions* are present. A **branch** is the binary image comprised of a set of pixels starting from either the soma up to its termination or a bifurcation/crossing region up to its termination, whose object pixel coordinates may be parameterized as a smooth curve. A **bifurcation region** is the set of pixels where an inward branch splits into two outward branches, one of them in quite a distinct orientation. A **crossing region** is the set of pixels where an inward branch splits into $N > 2$ outward branches, with $N - 1$ outward branches in quite distinct orientations.

The proposed approach involves two main steps:

- Preprocessing the original image through mathematical morphology transformations, yielding its 8-connected one-wide pixel skeleton image, its crossing regions image and a queue containing the origins of the branches, to be taken as the seeds for the *BTA*;
- Calling the *BTA* for each source point (i.e. each seed). Starting by the current seed, the algorithm labels iteratively its valid neighboring pixels, i.e still non-labeled pixels, whose vicinity equals two pixels until a *crossing/bifurcation* region is reached. The algorithm takes a decision to continue with the tracking procedure for the current branch. The *branching structure* topology itself provides the algorithm with proper information to decide among several possible outward branchings.

These steps are detailed in next section.

3. General Framework

3.1. Preprocessing

Firstly, the following preprocessing algorithm is performed in order to achieve the desired *branching structure* to be labeled:

```
BTA:Preprocessing
begin
```

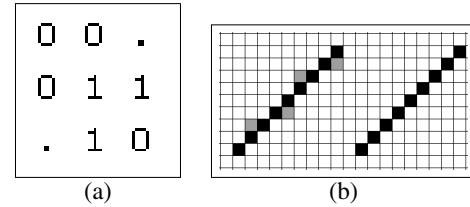


Figure 4. (a) Structuring Element for Hit-or-Miss filtering the prunned skeleton to yield an 8-connected one-pixel wide skeleton
(b) Light gray pixels at the left hand should be removed yielding the non-redundant structure at the right hand.

- Thresholding the original gray level image to binarize it;
- Eroding the binary image by a disk structuring element to get rid of dendrites;
- Area Filtering the eroded image to segment only the neuron soma;
- Dilating the filtered neuron soma to retrieve its original size;
- Dilating the binary image in order to merge branch portions which are too close and almost parallel;
- Skeletonizing the binary image by morphological thinning [6];
- Pruning the skeleton (Figure 5-a) to get rid of noise structures [6];
- Building a 3×3 mask template for the Hit-or-Miss operation (Figure 4-a) to eliminate redundant information (Figure 4-b) and achieve an 8-connected one-pixel wide skeleton structure (Figure 5-b);
- Subtracting the soma from the 8-connected one-pixel wide skeleton;
- Detecting end points twice. First in the 8-connected one-pixel wide skeleton structure *along* the soma segmented previously and then in the 8-connected one-pixel wide skeleton structure *without* the soma segmented previously. The difference between these two images should ultimately yields only the **origin** points to be used as source or seeds to feed the *BTA*, as depicted in (Figure 5-c).
- Creating a neighborhood image, where each pixel value represents the number of object pixels in its 8 neighborhood. Henceforth this image will be referred to as *Crossing regions* image. (Figure 5-d).

end

It is worth mentioning that structure elements size for all morphological operations were empirically found for the set of processed images. Different sized images may require different structure element sizes.

3.2. Branches Tracking

For the branch tracking step, it is introduced the *Branch Tracking Algorithm (BTA)* to segment, that is to label, each neuron branch as a distinct object on its own. Roughly speaking, the *BTA* is comprised of two great loops, the outermost one associated to a Queue for seeds (source points), while the innermost one manages a Stack for valid neighbors, neighbors of neighbors and so forth. Objects, i.e. non-labeled and non-crossing pixels, are considered as being valid. Thus, for each dequeued seed, the *BTA* is called for a partial branch tree, by stacking all valid pixels pertaining to the same branch. These stacked pixels will be subsequently labeled until either the current branch termination pixel or a branching/crossing region is reached. Every time a branching/crossing region is reached, the *BTA* identifies the best branch to continue the labeling process, while enqueueing seeds for partial branch trees to be labeled in a recursive-like fashion, until no more branches stemming from a branch within the tree rooted at the current source seed is found.

In order to determine the branch to continue the labeling past a branching/crossing region, a *breadth-first search* has been adopted, which works by enqueueing all the pixels into an auxiliary queue while getting across the just detected branching/crossing region. This *breadth-first search* takes place until a stability condition is achieved, i.e. there are only non-crossing pixels enqueued, for a consecutive number C of times. This process is illustrated in figure 6 and table 1.

Notice that scanning the neighborhood of the pixel **a** (state 0 in table 1) according to the chain-code defined in figure 2, we obtained the state 1. By setting the stability condition parameter C to 8, this procedure is repeated until state 22, when Σ equals C . At this point, the remaining pixels in the auxiliary queue, **y**, **w** and **z**, provide us with the terminations of the desired outward direction vectors. Strictly speaking, each corresponding direction vector origin should fulfill two requirements simultaneously: (*i*) be neighbor of the nearest crossing region pixel and (*ii*) have an existing path of valid pixels between it and the respective termination. For images presenting low branching/crossing region densities, the second requirement might be ignored, but not in the case of the figure 6-a. Figure 6-b shows the vectors obtained by considering both conditions (*i*) and (*ii*). All the vectors are then normalized and dot products between the

state	current	queue	B	Σ
0	\emptyset	a	1	1
1	a	b	0	0
2	b	c d \check{a}	0	0
3	c	d e $\check{d}b$	0	0
4	d	e $\check{c}f\check{b}$	1	1
5	e	f g \check{c}	1	2
6	f	g $\check{d}h$	1	3
7	g	h i \check{e}	0	0
8	h	i j \check{f}	0	0
9	i	j l m \check{g}	0	0
10	j	l m n \check{h}	0	0
11	l	m n o p $\check{m}\check{i}$	0	0
12	m	n o p $\check{l}\check{p}\check{i}$	0	0
13	n	o p q \check{j}	0	0
14	o	p q r \check{l}	0	0
15	p	q r \check{l} s \check{m}	1	1
16	q	r s \check{n} t	1	2
17	r	s t u \check{o}	1	3
18	s	t u \check{p} v	1	4
19	t	u v \check{q} x	1	5
20	u	v x y \check{r}	1	6
21	v	x y \check{s} w	1	7
22	x	y w \check{t} z	1	8

Table 1. States of the auxiliary queue. B is 1 if all pixels in a given state are non-crossing and 0 otherwise. Σ is increased by one if the respective B is on, and zeroed otherwise. Notice the check marks on some enqueued pixels; these checked pixels will be ignored, thus removed from the queue, as they have already been considered before.

inward direction vector v_0 and each outward direction vector are computed. Obviously, the outward vector for which the dot product result is maximum gives the proper direction to continue with the labeling, therefore the next pixel to be stacked. The remaining vectors are taken as side branches seeds to be enqueued for further processing. In the example described in table 1 and figure 6, the vector v_3 would give the direction toward which to continue the labeling, while vectors v_1 and v_2 would give directions for future consideration.

In brief, the *BTA* might be described as follows:

BTA:LabelTree

begin

- Initialize a stack for valid branch pixels and a queue for seeds.
- While queue is not empty

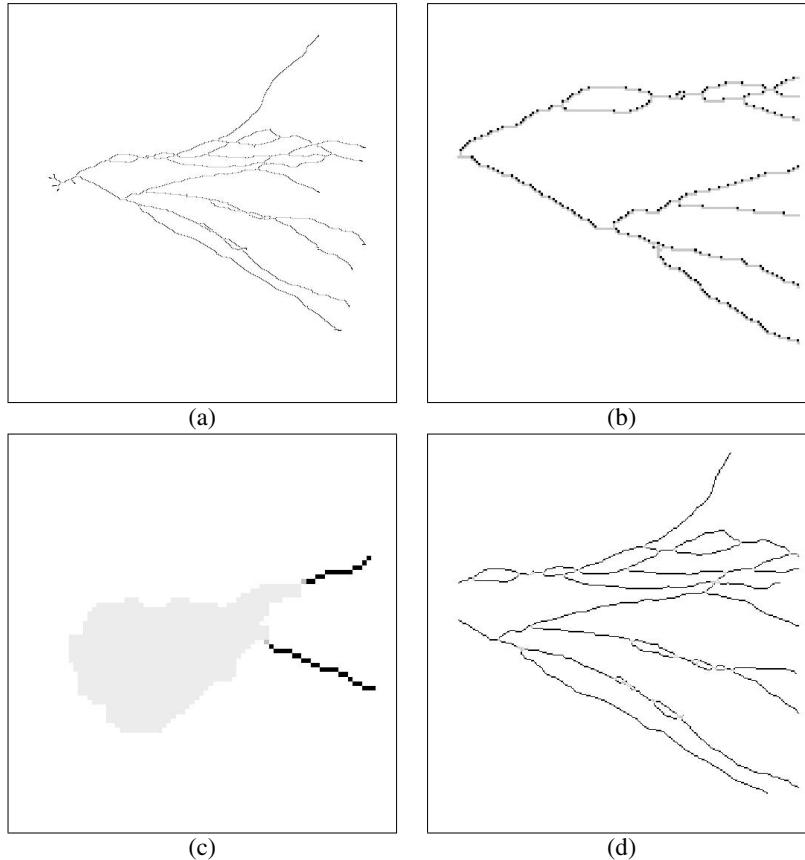


Figure 5. Preprocessing results: (a)Pruned skeleton. (b)The darkest pixels were removed by the Hit-or-Miss filtering yielding the 8-connected one-pixel wide skeleton shown in lighter gray level. (c) Soma (light gray), seed (dark gray) and skeleton(black). (d)Crossing regions (light gray) and skeleton (black).

```

    – Call BTA:LabelBranch.
• Check conditions to stack only valid neighbor pixels,
  that is object and non-labeled pixels.
end
```

The **BTA:LabelTree** algorithm makes a call to the **BTA:LabelBranch** procedure, which is defined below:

```

BTA:LabelBranch
begin
  • Pop the current pixel.
  • Keep track of the  $j^{th}$  preceding pixel to compute
    the inward direction vector when arriving at a cross-
```

ing/bifurcation region, where j may be a parameter. Usually, it suffices to take the 4th pixel, that is $j = 4$.

- If the current pixel is valid, i.e. it is an object, non-labeled and outside a crossing region, then label it.
- Probe the current pixel neighborhood to check if there is a *crossing/bifurcation* region.
- In case there is any inward crossing region pixel in the neighborhood, compute the inward normal direction vector.
- In case there is any inward crossing region pixel in the neighborhood, enqueue all neighbors iteratively, until reaching a number of non-crossing pixels outward the branches for a consecutive number c of times, thus providing a stability condition.

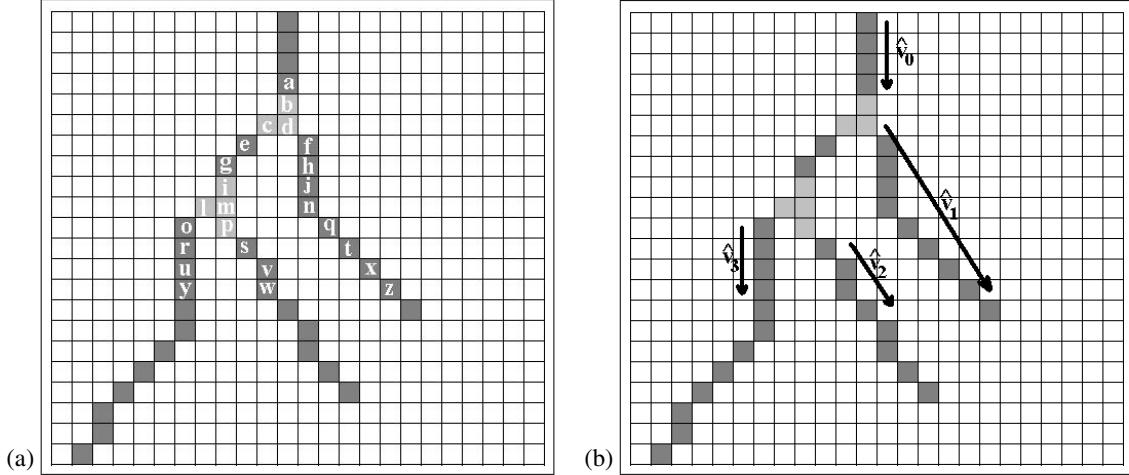


Figure 6. A typical critical region and the corresponding direction vectors obtained from it. (a) Close critical regions (light gray). The *breadth-first search* starts as the pixel a is reached and stops when only pixels y , w and z remain in the auxiliary queue. (b) the obtained direction vectors.

- These just reached non-crossing pixels will be the terminations of our outward direction vectors.
- For each termination pixel, take its nearest pixel inside the *crossing/bifurcation* region and compute the respective normalized direction vector.
- Now calculate the dot product between each outward direction v_i vector and the inward direction vector v_0 and take the k_{th} index related to the vector v_k that gives rise to the largest dot product result, i.e:

$$k = \text{argmax}_i (\langle v_0, v_i \rangle) \quad (1)$$

- If there are only two outward branches, push the pixel related to the k_{th} outward normal direction vector onto the stack and enqueue the other.
- Reinitialize the auxiliary variables in order to retrieve the ante-penultimate valid pixel coordinates for the next inward direction vector.
- If there are more than two outward branches, push the pixel related to the k_{th} outward normal direction vector onto the stack and enqueue all the remaining non-crossing vectors that are not in opposite orientations.

end

4. Results

The algorithms described so far have been implemented as *Matlab* scripts. The mathematical morphology operations have been applied using

the *Mathematical Morphology Toolbox* by SDC (<http://www.mmorph.com/>). The method has been evaluated and results obtained by labeling rat hippocampal cells from public available *Southampton Archive* (http://www.compneuro.org/CDROM/nmorph/index/topindex_tn.html). Some results are presented in figure 7. The left column shows the original images while the right column presents the respective labeling results, shown as connected components with the same gray-level as label. New labels are assigned to dendrite segments born from branches. The algorithm is able to distinguish branches from crossing, which is reflected by the correct assigned labels for the outward segments from such structures. Notice how parallelism and high densities of branching/crossing regions imply the *BTA* to label the same branch with different labels.

For the 3 test images shown in figure 7, it has been visually identified 40 bifurcations and 3 clear-cut crossings. All the bifurcations were correctly labeled and so were the 3 crossings without parallelism occurrences.

5. Concluding Remarks

The proposed approach starts by extracting the skeleton of the 2D projection of a 3D branching structure. Any skeletonization method may be applied to obtain the skeleton and incorporated (e.g. morphological thinning, exact dilations, medial axis transform). The important issue is to

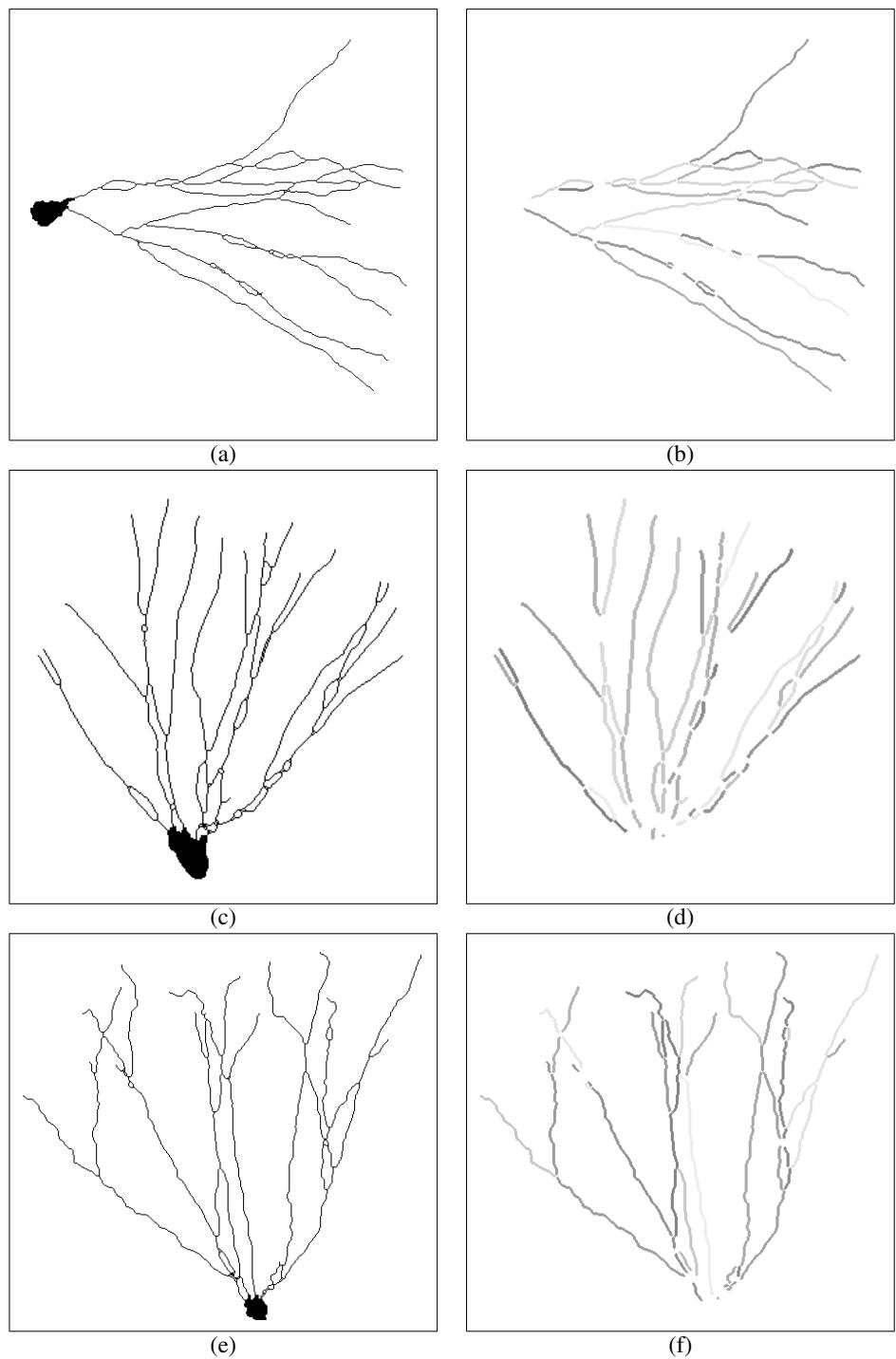


Figure 7. Results. Original images (a), (c) and (e). Labeled images (b),(d) and (f). A different gray level value has been used to label each different segment in (b),(d) and (f). The set of used gray level values was chosen so as to enhance the contrast among segments. Also, these images have been dilated by a 1-sized cross structuring element in order to improve their visualization.

have a suitable skeleton as input to the method. The preprocessing parameters have been empirically chosen and depend on the adopted skeletonization algorithm.

Despite the good results presented in the previous section, there are still some pending cases deserving special attention in future research. In general, the major pitfalls encountered during *BTA* running were related to different particular structure topologies, leading to high crossing/bifurcation regions densities, superposition of branches and/or branches parallelism.

As pointed out in the *BTA* description, in case of images with high crossing/bifurcation regions density, both conditions should be accomplished in order to get the direction vectors origins, otherwise shortcomings may occur, such as missing some branches. In the example presented in figure (6), notice that for all the termination pixels y , w and z the nearest branching/crossing region pixel is p . Hence, by considering only the proximity condition, instead of the obtained vectors v_1 , v_2 and v_3 , all vectors origins would lie in s , i.e. the nearest non-crossing pixel which is neighbor of p . It is plain to see that, as a consequence, only pixel s would be stacked for continuing the labeling process. However there are not valid paths between s and y and between s and z . For this reason, the incoming branch would have its continuation from pixel s on, instead of pixel f as expected. As a result, the *BTA* would miss two branches. This is a straightforward consequence of the *breadth-first search agglutinating effect*, since the *breadth-first search* has been implemented by using a queue data structure, which is memory-less regarding the shape topology. In an effort to accomplish the stability condition, the algorithm has clumped both bifurcation/crossing regions (Figure 8-(a)) into only one (Figure 8-(b)). A possible solution for this problem is to implement the *breadth-first search* by using a tree data structure, so as to keep memory of the topology at the critical region. As soon as the stability condition would be accomplished, the *BTA* would just get the tree leaves, which would host the *direction vectors terminations*, and ask the tree for the *direction vectors origins* by climbing the tree from each termination up its nearest non-crossing bifurcation/crossing neighbor. The tree itself would provide us properly with the valid paths between terminations and origins.

Particularly, it has been observed in rat hippocampal cells that too close branches parallelism may occur with high frequencies. This might create short cycles along the structure leading to the *BTA* incorrectly labeling the branches. As a solution, the image has been dilated just before it has been skeletonized. By doing so, short cycles would be shrunk into a branching/crossing region. In future works, we intend to proceed a multiscale analysis to circumvent this shortcoming.

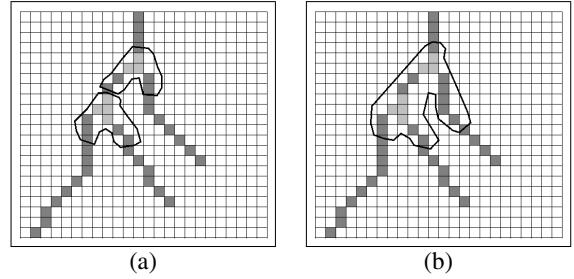


Figure 8. (a) Two separate regions (b) Two regions clumped into one.

Acknowledgements

RMC is grateful to FAPESP (2005/00587-5), CNPq (300722/98-2, 474596/2004-4, 491323/2005-0) and CAPES for financial support. LDPC is grateful to CNPq(308231/03-1) and FAPESP for sponsorship. The authors are grateful to Southampton Archive for the rat hippocampal cells images hosted at: http://www.compneuro.org/CDROM/nmorph/index/topindex_tn.html.

References

- [1] M. F. Carmo. *Differential geometry of curves and surface*. Prentice-Hall, Englewood Cliffs, N.J., 1976.
- [2] F. Caserta, H. Stanley, W. Eldred, G. Daccord, R. Haussman, and J. Nittmann. Physical mechanisms underlying neurite outgrowth: A quantitative analysis of neuronal shape. *Physical Review Letters*, 64(1):95–98, 1990.
- [3] R. M. Cesar-Jr. and L. F. Costa. Neural cell classification using wavelets and multiscale curvature. *Biological Cybernetics*, 79(4):347–360, 1998.
- [4] L. da F. Costa. Bioinformatics: Perspectives for the future. *Genetic and Molecular Biology*, 3(4):564–574, 2004.
- [5] L. da F. Costa. Morphological complex networks: Can individual morphology determine the general connectivity and dynamics of networks? 2005. <http://xxx.lanl.gov/abs/q-bio.MN/0503041>.
- [6] E. R. Dougherty and R. A. Lotufo. *Hands-On Morphological Image Processing*. SPIE-International Society for Optical Engine, 2003.
- [7] G. T. Herman. *Geometry of Digital Spaces*. Birkhauser Boston, 1998.
- [8] H. Jelinek and E. Fernandez. Neurones and fractals: how reliable and useful are calculations of fractal dimensions? *Journal of Neuroscience Methods*, 81(1-2):9–18, 1998.
- [9] K. Morigiwa, M. Tauci, and Y. Fukuda. Fractal analysis of ganglion cell dendritic branching patterns of the rat and cat retinae. *Neuroscience Research Suppl.*, 10:S131–S140, 1989.

H.4 *Automatic Contour Extraction from 2D Neuron Images*

Anexo o preprint *Automatic Contour Extraction from 2D Neuron Images* submetido e aceito em 29/10/2008 pelo *Journal of Neuroscience Methods - Elsevier* [61] e também disponível como *Technical Report* no servidor aberto *ArXiv* [63]¹. Este trabalho descreve um sistema automático para extração do contorno paramétrico de estruturas ramificadas bidimensionais, obtidas de imagens de neurônios tridimensionais, sendo uma evolução do método de rastreamento de ramos apresentado anteriormente. Algoritmos de extração de contorno tradicionais falham ao tentar caracterizar tais estruturas, por não conseguirem entrar nas regiões mais internas dos esqueletos.

¹<http://arxiv.org/abs/0804.3234v1>

Automatic Contour Extraction from 2D Neuron Images

J. J. G. Leandro^{a,*}, R. M. Cesar-Jr^a, L. da F. Costa^b

^a*Institute of Mathematics and Statistics - USP, Department of Computer Science
Rua do Matão, 1010 - São Paulo - SP, 05508-900 Brazil*

^b*Instituto de Física de São Carlos - USP, Department of Physics and Informatics
Av. Trabalhador Säocarlense, 400 - São Carlos - SP, 13560-970 Brazil*

Abstract

This work describes a novel methodology for automatic contour extraction from 2D images of 3D neurons (e.g. camera lucida images and other types of 2D microscopy). Most contour-based shape analysis methods can not be used to characterize such cells because of overlaps between neuronal processes. The proposed framework is specifically aimed at the problem of contour following even in presence of multiple overlaps. First, the input image is preprocessed in order to obtain an 8-connected skeleton with one-pixel-wide branches, as well as a set of *critical regions* (i.e., bifurcations and crossings). Next, for each subtree, the tracking stage iteratively labels all valid pixel of *branches*, up to a *critical region*, where it determines the suitable direction to proceed. Finally, the labeled *skeleton segments* are followed in order to yield the *parametric contour* of the neuronal shape under analysis. The reported system was successfully tested with respect to several images and the results from a set of three neuron images are presented here, each pertaining to a different class, i.e. alpha, delta and epsilon ganglion cells, containing a total of 34 crossings. The algorithms successfully got across all these overlaps. The method has also been found to exhibit robustness even for images with close parallel segments. The proposed method is robust and may be implemented in an efficient manner. The introduction of this approach should pave the way for more systematic application of contour-based shape analysis methods in neuronal morphology.

Key words: Contour extraction, branching structures, neuron, shape analysis, image processing, neuronal morphology, pattern recognition

* Corresponding author.

Email addresses: jleandro@vision.ime.usp.br (J. J. G. Leandro), cesar@vision.ime.usp.br (R. M. Cesar-Jr), luciano@ifsc.usp.br (L. da F. Costa).

1 Introduction

Neurons can be understood as cells specialized in interconnections, which are implemented through synapses extending from axonal to dendritic arborizations. Though the connectivity of mature neuronal systems may seem to be stable, it is actually subjected to continuing re-organizations influenced by stimuli presentation and biological changes. The number of connections which a neuron may receive is to a large extent defined by the shape of its dendritic tree, which serves as a target for growing axons. As a consequence, the full understanding of the functionality of neuronal circuits requires the proper characterization of the neuronal morphology (e.g. [6, 8, 19]). Among the several approaches based on the characterization of the geometry and connectivity of neuronal cells [3, 8, 13, 18, 19], a particularly important and broad set of shape analysis algorithms relies on a parametric representation of the neuronal shape [7], i.e. in the form $c(t) = (x(t), y(t))$.¹ The proper contour extraction of 2D neuron images yields parametric signals, from which features can be calculated and used to characterize differences between neuronal shapes. Despite the availability of algorithms to extract parametric contours from digital images, they can not be applied directly in neuroscience because of the intense overlap (crossings) which is frequently observed among the neuronal processes. In order to better appreciate this limitation, please refer to Fig. 1.

Often, the 3D neurons are projected into the 2D space (e.g. camera lucida and several types of 2D microscopy), so that the contour of the cells can, in principle, be represented as 1D parametric curves [11] (Fig. 1(a)). Important information, such as the normal and/or tangent orientation fields along such contours, as well as the arc length of each segment, can then be obtained, allowing the estimation of important geometrical properties such as the contour curvature and wavelets (e.g. [4, 8]), which are known to provide particularly valuable information about the shape of the neuron, including its bending and concavity. Another possible application of contours in neuroscience is as a means to automatically obtain neuronal dendograms [5]. However, such an approach is often complicated by the presence of crossings between the neuronal processes in the 2D image, implying some regions of the cell to become inaccessible for traditional contour extraction algorithms (see Fig. 1(b)). Henceforth, we refer to such inaccessible contour portions as the *innermost regions* of the shape. In brief, most contour following algorithms work as follows. Firstly, the algorithm detects an initial contour pixel. Then, the algorithm searches the next contour pixel, by probing the current contour pixel vicinity. The algorithm travels around the whole object, until revisiting the first pixel, once the task has been completed. Further details can be found in the full description of such methods in [7].

In contour following algorithms, it is usually impossible to traverse the innermost

¹ For instance, a circle may be represented as $c(t) = (x(t), y(t)) = (\cos(t), \sin(t))$

regions delimited by crossings (due to the 3D to 2D projection). Consequently, only the outer contour of the cell is represented, thus missing the innermost structures. This fact is illustrated in Fig. 1(b), where the light gray shaded innermost regions represent areas inaccessible to traditional contour following algorithms, thus yielding just the red curve as the respective contour.

It should be observed that, in principle, 3D imaging is naturally better than 2D imaging because it contains more information about the acquired structure. However, 3D imaging intrinsically demands additional computational resources, especially for the enhancement and interpretation of the structures. In addition, in cases where the original neurons are mostly planar, such as the ganglion cells in the retina, the 3D capture often contributes little to the separation between thin crossing segments, which are mostly contained within a narrow range of z-coordinates. Another reason why 2D imaging remains valuable in neuroscience concerns the fact that not all types of microscopy can be performed in 3D. Finally, many 2D neuron imaging systems and images are already available (e.g. camera lucida images from previous experiments and published literature), so that research using such images would also benefit from the introduction of the methodology proposed in the present paper.

Despite the importance of the problem of contour extraction from 2D neuron images, we were not able to find any related approaches in the literature. To the best of our knowledge, the only similar work in the literature was described in [20, 21], which presents a semi-automatic method to separate veins and arteries in the vascular trees of fundus images. In order to solve such a problem, it is necessary to label the vessels, which are also branching structures. Firstly, a skeleton image is obtained from the vessel trees. The skeleton is then represented as a vascular graph named G , which comprises all the information regarding the connectivity among critical regions and branches segments. Subsequently, that method propagates the labels manually assigned to other vessels throughout the vascular graph G . In this case, it is well-known that crossings only take place between a vein and an artery, thus such a priori knowledge on the vessels structure is used to simplify the approach, determining that opposite branches segments in a crossing should be necessarily equally labeled. Such assumptions make this approach specific to the vein/artery tracing problem. Alternatively, neighbouring branches segments in a crossing should be labeled with distinct tags. Although there are some similarities between the approaches reported in [20, 21] and in the current article (mainly concerning the use of skeletons and case analysis of bifurcations and crossings), the latter methodology adopts a different and potentially more general approach. Since the aforementioned system has been specially designed for vascular trees, the number of labels available to be assigned is bounded to two: veins and arteries. Conversely, our algorithms yield assignments of a distinct label for each existing branch and also a distinct label for each dendritic tree, regardless of the number of existing dendritic trees in the neuron. Also, by segmenting each branch within the image, our method allows the branches to be counted as well as the lengths of the

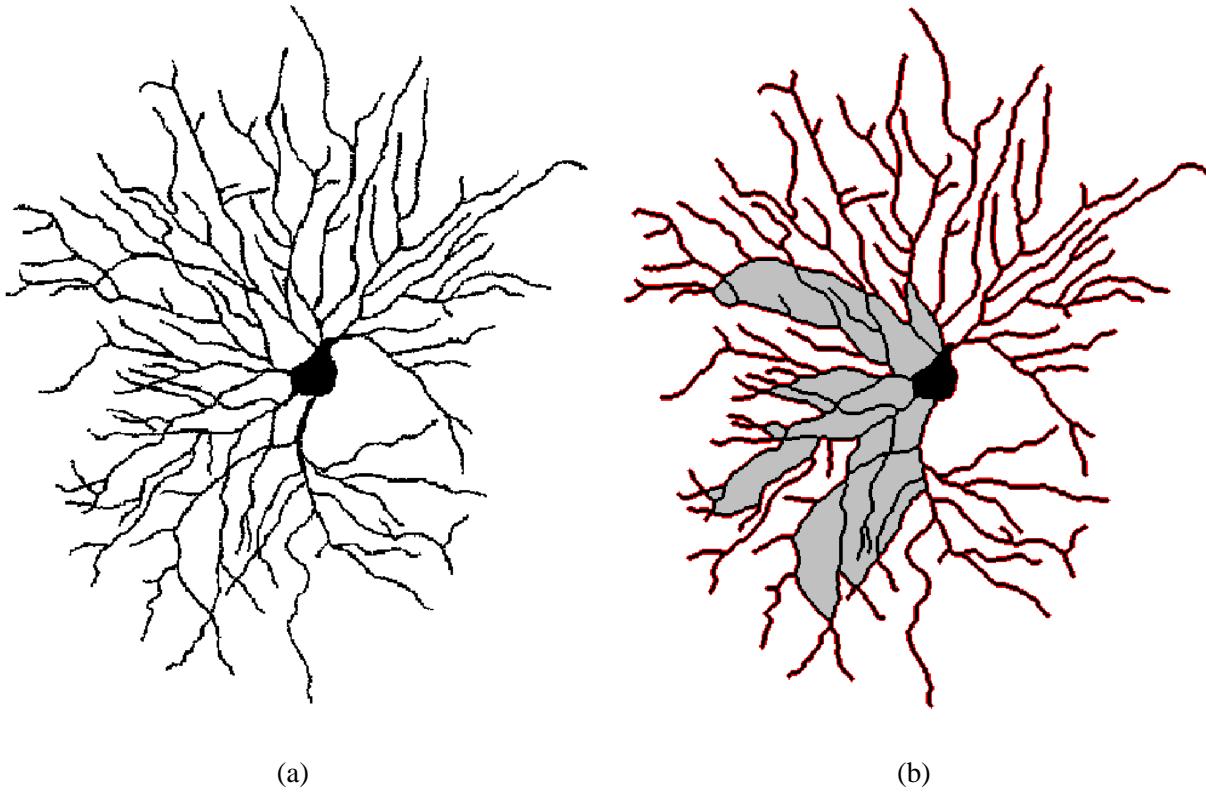


Fig. 1. (a) Example of 2D neuron image considered in this work. (b) Neuron image (black) and respective contour (red) as provided by a traditional contour following algorithm. The light gray shaded areas represent the innermost regions that remain inaccessible for such algorithms.

segments to be measured. Furthermore, our system provides the parametric contour of the whole structure, being thus possible to extract several geometrical features to be fed to a classifier.

The rationale of the present work is to deal with neuronal overlaps by incorporating several criteria such as the use of similarities along the tangent orientation as a means to identify the proper continuation of the neuronal processes at crossing-points. The proposed methodology is composed by the subsequent application of the following three algorithms:

- (1) Preprocessing
- (2) *Branches Tracking Algorithm (BTA)*
- (3) *Branching Structures Contour Extraction Algorithm (BSCEA)*

In short, the *BTA* is an algorithm aimed at the segmentation of each distinct branch within a 2D neuron image other than the soma and intercepting regions. The *BSCEA* is an algorithm intended to the extraction of the parametric contour from a 2D neuron image, based on the *BTA*.

Table 1
Summary of concepts.

TERM	DESCRIPTION	
Points		
seed	primary	origin of a segment stemming from the soma
	secondary	origin of a segment stemming from a critical region
termination	end point of a branch	
Lines		
segment	line of pixels delimited by other structures	
inward segment	incoming segment at a critical region	
outward segment	outgoing segment from a critical region	
branch	string of segments	
Critical Regions (<i>CR</i>)		
bifurcation	cluster of pixels where an inward segment splits into two; one of which in other direction	
crossing	cluster of pixels where an inward segment splits into three; two of which in other direction	
superposition	cluster of bifurcations where an inward segment splits into three; two of which in other direction	
Collections		
dendritic arbour	collection of branches growing out of soma	
periphery	collection of dendritic arbours (excluding soma)	
skeleton	one-pixel wide skeleton from the periphery	

The current article is organized as follows. Section 2 contains an overview of the proposed framework, which is further detailed in Section 3. Experimental results considering real neuronal cells are presented in Section 4. The paper concludes in Section 5 ².

2 Preamble

2.1 Terminology

Initially, our approach considered the existence of only two types of structures among branches, namely bifurcations and crossings. However the number of ad-

² A more detailed description of the methodology is available in a technical report [15].

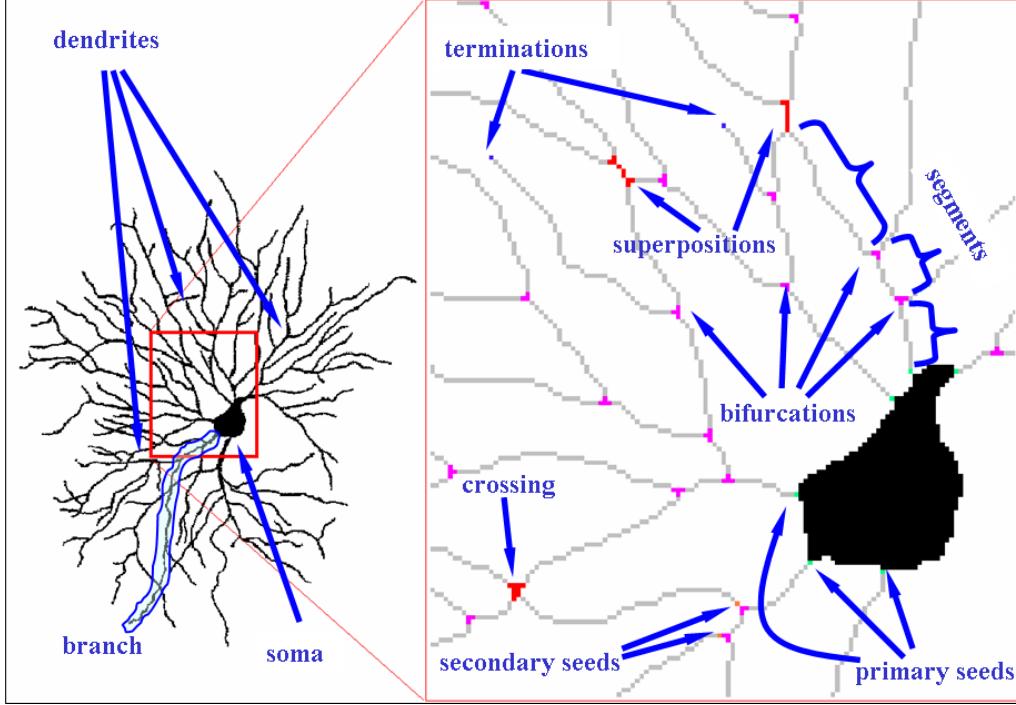


Fig. 2. Extended terminology adopted in this work: dendrites, soma, branches, segments, seeds, terminations and critical regions.

jacent segments at each critical region proved not to be enough to properly classify them, leading to misclassifications. Only through the incorporation of additional information, namely the identification of several geometrical features along the neuronal shape, it has been possible to achieve correct classification of the critical regions. The neuronal shape can be described as the union of its constituent parts, i.e. soma, dendrites or dendritic arbours and axon. In order to elaborate on the explanation of our methods, we extend such a terminology. Notice that it does not refer to additional functional parts in neurons, but rather to morphological building blocks which compose those functional parts in their respective 2D images. As a matter of fact, some structures such as crossings and superpositions do not even occur in a real 3D neuron, being just an immediate consequence of the projection from the 3D space onto the plane or their close proximity even in 3D spaces. Hence, the arbours present in 2D skeletons obtained from neuron images are subdivided into their morphological constituent parts as follows (see Fig. 2 and Tab. 1):

- *Points*
- *Lines*
- *Critical Regions*
- *Collections*

The aforementioned categories of structures encompass the typical structures which usually appear in 2D neuron images. Such structures must be distinguished so as to provide additional information about the original neuronal shape, therefore allow-

ing more general and effective performance.

The category *Points* comprises three classes of extremity points: primary seeds, secondary seeds and terminations. Each extremity point is classified regarding its location, i.e. a *primary seed* corresponds to a junction point between a dendritic tree and the soma, while a *secondary seed* refers to a junction point between a critical region and a dendritic subtree. Basically, the difference between a primary seed and a secondary seed is that a primary seed is necessarily adjacent to the soma, while a secondary seed is not. *Terminations* are end points of branches. The reason for distinguishing between points is that the tracking starts from the primary seeds and finishes at terminations, occasionally repeating itself in a recursive-like fashion from secondary seeds.

The category *Lines* encompasses two cases: segments and branches. Each line is classified with respect to its extremity points, i.e. a segment may grow out from either a primary or a secondary seed, but does not necessarily end at a termination. *Segments* are lines of pixels delimited by a pair of minor structures, for instance a seed and a critical region, or two critical regions, or a critical region and a termination. Conversely, a branch may stem from either a primary or a secondary seed, ending necessarily at a termination. It follows from such a definition that a *branch* is a ramification made up of a string of segments, growing out of a seed up to a termination, as shown in Fig.2. In addition, segments may be further subclassified depending on their relationship to an adjacent critical region. By analyzing a neuron shape from inside out, that is, from the soma towards its terminations, an incoming adjacent segment to a critical region is said an *inward segment*, while an outgoing adjacent segment to a critical region is said an *outward segment*. The reason for distinguishing between lines is the need to recognize the constituent parts (segments) of branches every time the tracking algorithm reaches a critical region. Adjacent segments which present tangent similarity should be regarded as part of the same branch. Segments and branches play different roles in the tracking algorithm, thus deserving distinct names.

Critical Regions are clusters of pixels where branches meet each other. This category includes three classes of regions: bifurcations, crossings and superpositions. Each Critical Region is classified by considering its shape, the number of segments adjacent to it and their mutual orientation relationship, as well as the proximity relationship between the current critical region and other regions nearby. On a *bifurcation*, an inward segment often separates into two outward segments with different orientations, as depicted in Fig.4-(a-b). Occasionally, bifurcations may occur very close one another. Viewing such regions from a larger scale would suggest just one critical region, where an inward segment splits into three outward segments, as shown in Fig.4-(c-d). Similarly, on a *superposition*, an inward segment splits into three outward segments, two of them in normally distinct and opposite orientations, as can be seen in Fig. 4-(e). If superpositions are not considered, they could be locally misunderstood as two very close bifurcations attached by a tiny segment.

Finally, a *Crossing Region* is a cluster of pixels where an inward segment splits into three outward segments, two of them in quite distinct and necessarily opposite orientations, as shown in Fig.4-(f).

Though all critical regions share the property of being formed by pixels with neighborhood greater than two, their shape structure are quite different. The reason for distinguishing between critical regions is to assure that both the tracking and the contour extraction algorithms behave as expected whenever such structures are found. The algorithms undergo different processings for each kind of critical region.

At this point, it is worth emphasizing the difference between a crossing and a superposition: although both share the property of having an inward segment splitting into three outward segments, their shapes are slightly different. Notice that a crossing appears as just a cluster of pixels, while a superposition is apparently made up of two clusters of pixels (bifurcations) attached by a short line. In spite of the fact that both structures have been originated from overlapping processes, the angle of inclination between these processes plays a central role, in that the steeper the slope between them, the greater the chance of obtaining a crossing, while the smoother the slope between them, the greater the chance of obtaining a superposition, as illustrated in Figure 3.

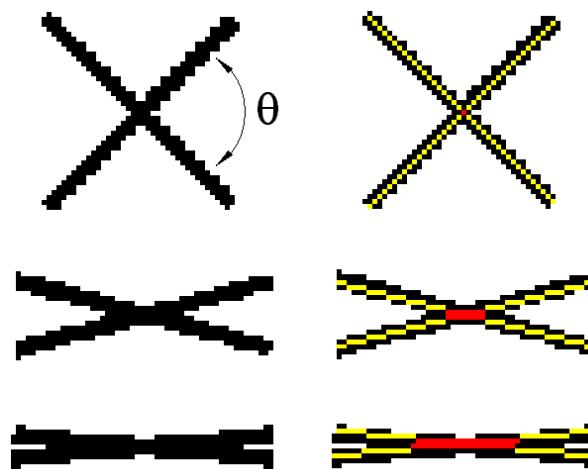


Fig. 3. The dependence between the angle θ of inclination between intercepting branches (yellow) and the shape of the respective critical region (red). Observe that as θ decreases, more elongated the critical region becomes.

The category *Collections* simply represents groups of the aforedefined objects. A *Dendritic Arbour* is a collection of branches having roots in the soma. Henceforth the collection of Dendritic Arbours, that is, the neuron without the soma, is simply referred as the *Periphery*.

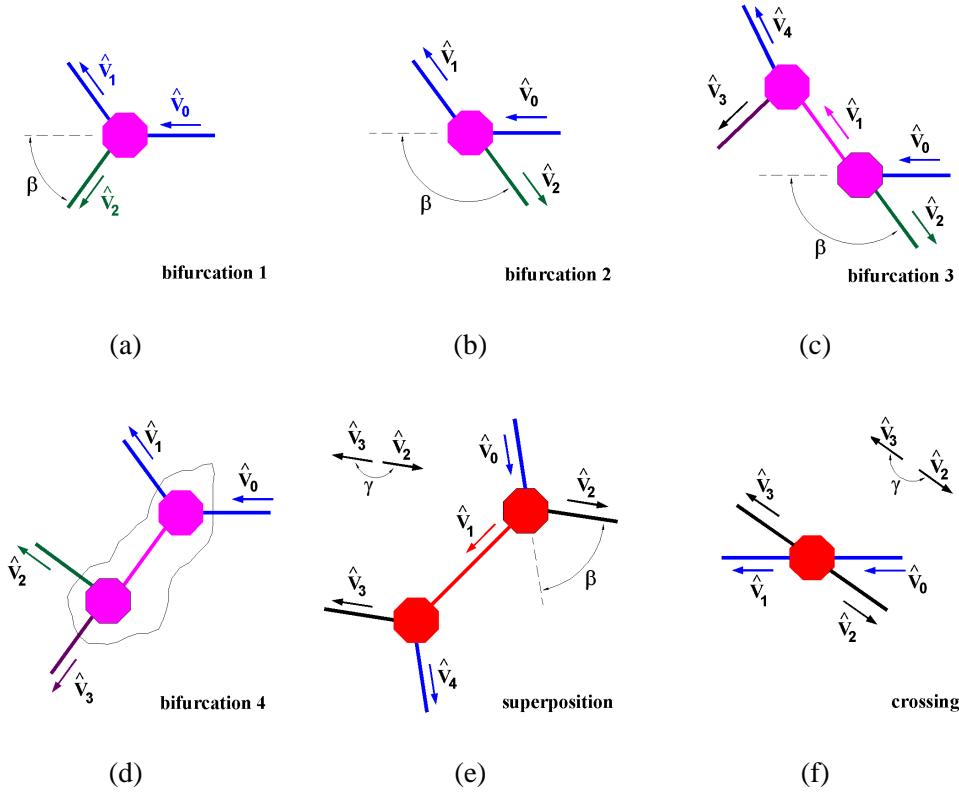


Fig. 4. The critical regions classification rules take into account the angle β between the inwards direction vector - \hat{v}_0 - and the outwards direction vector - \hat{v}_2 , the angle γ between any pair of outwards direction vectors and the cardinality $|E_1|$ of the set of outwards direction vectors E_1 related to the current critical region s_1 . (a) If a critical region presents $|E_1| = 2$ and $\beta < 90^\circ$ it is immediately classified as a bifurcation 1; (b) if $\beta \geq 90^\circ$ it may be a bifurcation 2 or (c) even a bifurcation 3 as long as there is another critical region nearby. (d) A bifurcation 4 presents $|E_1| = 3$, but there is no $\gamma \approx 180^\circ$. (e) A superposition appears with $|E_1| = 2$ and another critical region s_2 nearby, with $\gamma \approx 180^\circ$. (f) Finally, a crossing has $|E_1| = 3$ and $\gamma \approx 180^\circ$. Notice that bifurcation 1 could be mistaken for either bifurcation 4 or superposition and bifurcation 2 could be mistaken for bifurcation 3 in case the adopted rules had not been considered.

2.2 System overview

Briefly, the proposed approach for the parametric contour extraction of branching structures involves the following three steps:

- **Input Preprocessing Algorithm.** In summary, this algorithm is aimed at preprocessing the input 2D image, unfolding it into additional images containing its required structural building blocks for the subsequent steps. The input image is preprocessed by means of mathematical morphology operations [9, 23], yielding its separate components, namely:

- periphery skeleton image, henceforth referred to as skeleton
- critical regions image
- terminations image
- soma image
- queue of primary seeds
- **Branch Tracking Algorithm.** The **BTA** has two main goals: to label each branch and to classify each critical region. It is applied for every primary seed present in the queue. The labelling procedure starts at the segment adjacent to the primary seed. After reaching a critical region, the current segment will have been entirely labeled, so a decision concerning the next segment to continue with the tracking must be taken. In addition to finding the optimal segment to move ahead, the algorithm also identifies the current critical region as either a bifurcation, a superposition or a crossing. If the current critical region is a bifurcation, the *BTA* stores the related secondary seed in an auxiliary queue, otherwise the *BTA* stores the addresses of the current segment end point and the next segment starting point. By doing so, the *BTA* labels all the segments comprising each dendritic branch in a recursive-like fashion, until reaching a termination.
- **Branching Structure Contour Extraction Algorithm.** The **BSCEA** main role is to extract the parametric contour $c(t) = (x(t), y(t))$ along the segments comprising a 2D neuron image by using the labeled branches and classified critical regions obtained in the previous step. Basically, the *BSCEA* follows the segments defining branching structures (resulting from the union between the labeled skeleton and the soma) by entering all the shape innermost regions. During the contouring process, whenever a branching region is found, the *BSCEA* contours the shape outwards, as the traditional algorithm would. On the other hand, whenever a crossing or a superposition is found, the *BSCEA* contours the shape inwards, by traversing the current critical region through the addresses stored in pointers by the *BTA*. Finally the *BTA* gives as a result the contour parametric functions $x(t)$ and $y(t)$ as well as a contour image (Fig.12(b)).

These procedures are detailed in Sections 3.1, 3.2 and 3.3, respectively.

3 General Framework

3.1 Preprocessing

Some important shape parts are detected by taking into account specific features, such as the number of each pixel's neighbors and the size of the shape. For example, pixels of branches are expected to have only 2 neighbors each, while critical regions and the soma have more. Moreover, the soma area is greater than the areas of the critical regions.

Initially, a preprocessing pipeline involving mathematical morphology transformations ³ is carried out on the input image, so as to obtain the separate components of the neuron image, that is the skeleton comprised of 8-connected one-pixel-wide branches, the critical regions, the terminations, the soma and the queue of primary seeds. The referred separate components on different images are obtained as described in the flowchart diagram depicted in the Fig. 5.

In order to isolate the soma, the image is eroded by a disk with radius 3, followed by a dilation with a disk with radius 1. It is known that soma shapes do not follow a clear pattern, making their segmentation critical. Herein, the soma segmentation is attained through erosion, noise filtering by area opening, followed by a dilation. Casual noisy pixels surrounding the soma image are wiped out through the skeleton area opening. Then, additional processing is applied in order to obtain an 8-connected skeleton with one-pixel wide branches [15].

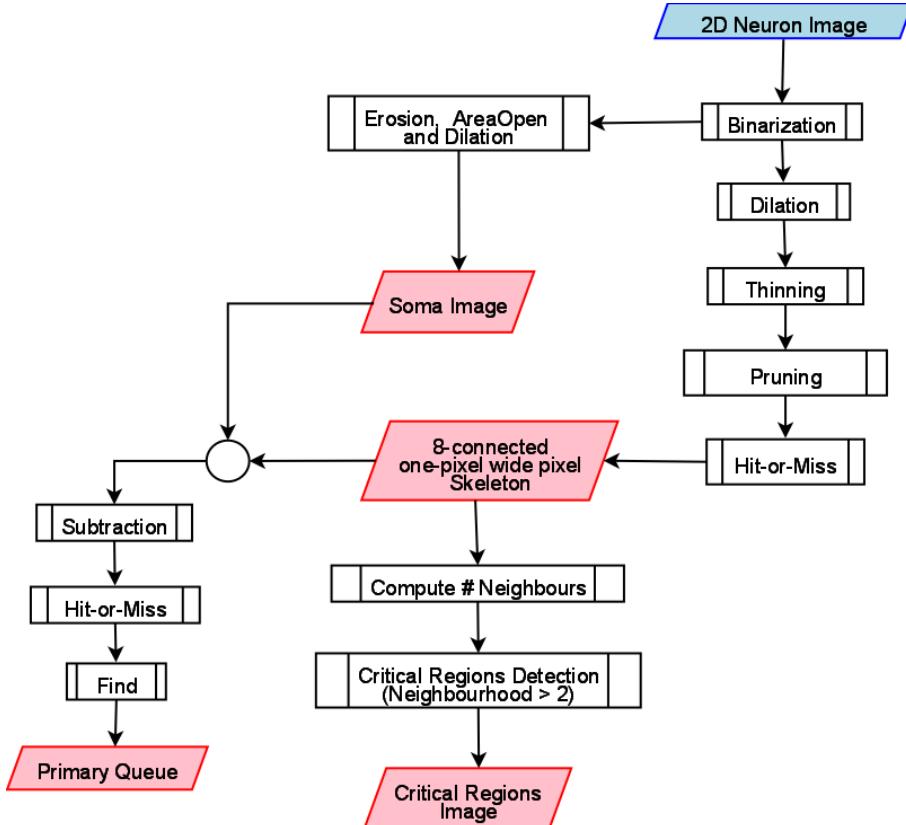


Fig. 5. Flowchart of the preprocessing pipeline. The red polygons represent the outcomes.

³ The reader is referred to [9, 23] for details on the mathematical morphology operations.

3.2 Tracking of Branches

One of the main goals at this stage is to label each dendritic branch as a whole object on its own. This is achieved by pixel-by-pixel labeling of each branch. Considering the sequential nature of such a processing, this problem may be described as estimating the spatial coordinates (x, y) of each subsequent branch pixel. Because this is analogous to tracking problems [1] in the computer vision literature, this algorithm is called *Branches Tracking Algorithm (BTA)*.

Tracking is usually divided into *Prediction*, *Measure* and *Update* stages [1]. During the Prediction stage, the algorithm estimates the next state of the system. On the Measure stage, the algorithm probes the system by looking for plausible states nearby, in this case valid pixels, through some measures, herein the spatial coordinates (x, y) of pixels. During the Update stage, the algorithm merges both pieces of information gathered on the previous two stages, through a linear combination, giving as a result the optimal estimation for the next state. So, in terms of Tracking, the *BTA* Prediction and Measure stages are carried out in a single step, through the 8-neighborhood scanning by using the chain-code [7].

The *BTA* Update stage is related to the pixel labeling. This stage labels each dendritic subtree growing out of the soma in the same way, i.e. by starting from the related primary seed and labeling the entire branch adjacent to it, up to its termination. Meanwhile, its branches are marked to be labeled afterwards. Thereafter, every branch is labeled as the first branch was, and the respective encountered branches are similarly marked to be labeled afterwards in a recursive-like fashion until the whole dendritic subtree is labeled.

On arriving at a critical region, the *BTA* may perform one or two of the following tasks, *Continuity of the Tangent Orientation Assessment* and *Critical Regions Classification*. The former (detailed in the Section 3.2.1) is always carried out, while the latter (described in the Section 3.2.2) is performed only if the current critical region has not been classified yet. Notice that though the critical regions are now available from the previous preprocessing step, they are not classified yet, i.e. we do not know which is a bifurcation, a crossing or a superposition. This classification is important for the contour extraction step.

3.2.1 Continuity of the Tangent Orientation Assessment

Analogously to the tracking process during branches labeling as described in 3.2, this step also comprises Prediction, Measure and Update stages, however in a slightly different fashion. Coming to a critical region in this step is similar to approaching the occlusion case in tracking problems [10], where different objects follow trajectories which apparently overlap.

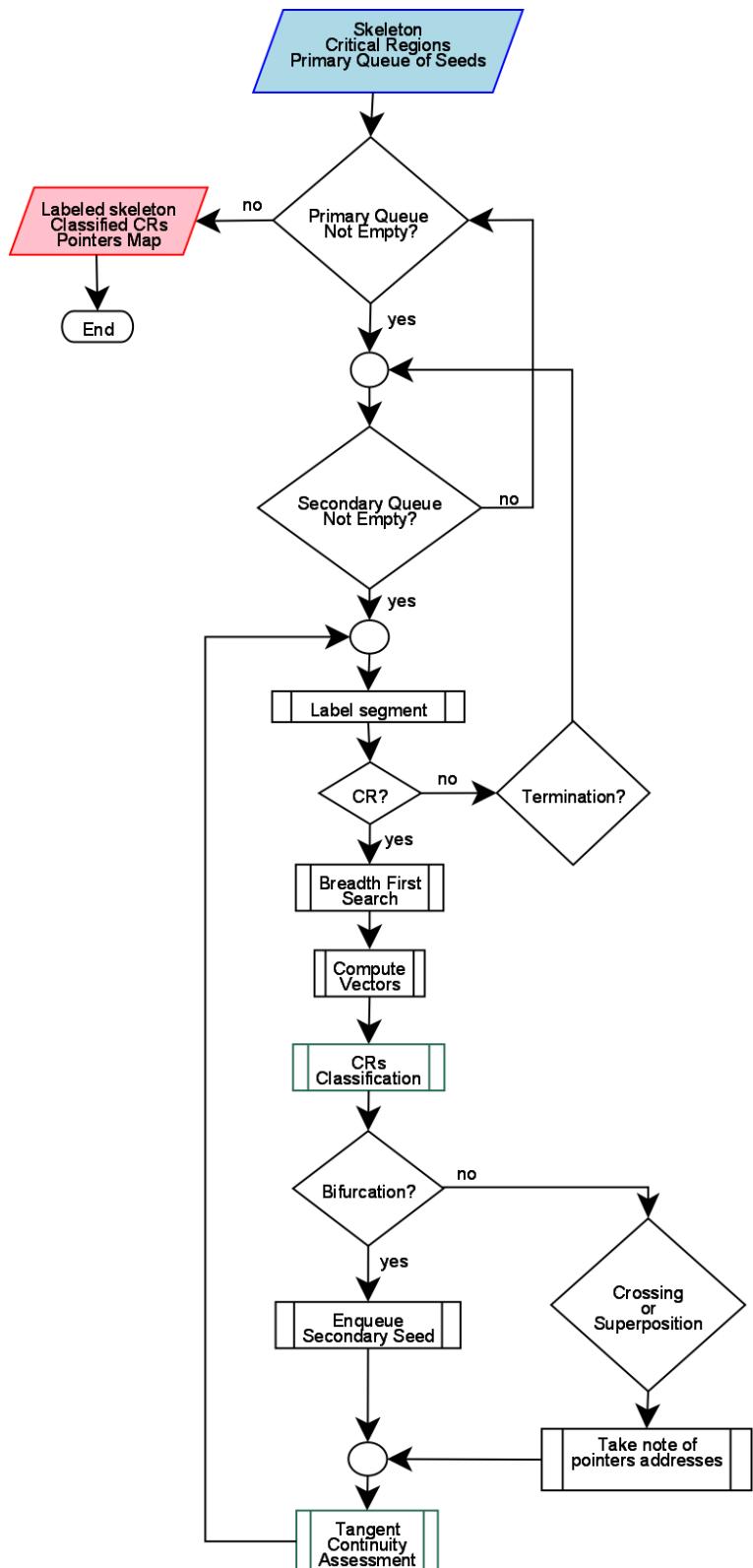


Fig. 6. Flowchart of BTA.

So, after arriving at a critical region, the Prediction stage is performed by computing the *inwards direction vector* \hat{v}_0 . In the Measure stage the algorithm calculates all *outwards direction vectors* \hat{v}_i . Finally, in the Update stage the outcomes from the Prediction and Measure stages are merged. The reason to do that is to estimate the best candidate segment among all the alternatives so as to carry forward the tracking procedure. This merging is achieved by the calculation of inner products (projections) between the inwards direction vector \hat{v}_0 and each outwards direction vector \hat{v}_i (observe that these two vectors have unitary magnitude), according to Eq. 1:

$$k = \arg \max_i (\langle \hat{v}_0, \hat{v}_i \rangle) \quad (1)$$

where the index k is assigned to the direction vector \hat{v}_k among \hat{v}_i , for which the inner product result is the largest. The extremity points for properly computing each vector are determined through a *Breadth-First Search* approach ⁴.

Every time a critical region is encountered, the Breadth-First Search is triggered and all the forward neighboring pixels are iteratively enqueued into an auxiliary queue, while passing across the just detected critical region. At each Breadth-First Search iteration, the auxiliary queue is run through in search of critical pixels. The stop condition for the Breadth-First Search is set beforehand as a number C of consecutive executions through the auxiliary queue without finding any critical pixel.

The starting pixel of the optimum segment to proceed is lastly stacked and labeled. Also, the alternative path origin is considered as a secondary seed, that is a side branch seed to be enqueued in case a bifurcation is detected. Conversely, in case either a superposition or a crossing is detected, the next segment starting point V_{n+1} and the current segment last point V_n (Fig. 10(b)) addresses are stored into the Pointers Map.

All side branches are recursively labeled, until no unlabeled branch growing out of any secondary branch within the dendritic subtree originated from the current primary seed remains.

3.2.2 Critical Regions Classification

While assessing the orientation of the tangent direction vectors at each critical region, as described in section 3.2.1, the BTA also gathers enough information to classify the current critical region into one of the 6 different classes (see Fig. 4), which have been identified as being critical for the skeletons. *Critical Regions Classification* is a crucial concept for the proper functioning of our *Contour Extraction*

⁴ The Breadth-First Search is a suitable method to find the shortest path between two nodes in a graph, implemented by using a queue of nodes as a data structure [22].

algorithm presented in section 3.3. Although *Critical Regions Classification* is not an algorithm on its own, it is an important part of the *BTA*. Therefore, each critical region is classified according to some special rules. The decision tree depicted in Fig. 7 details both the classification rules themselves and the order in which they should be considered, i.e. it illustrates the flow of decisions required to properly classify a critical region into one of the 6 classes showed in Fig 4. These classes have been abstracted from the analysis of several images, during the development of our methodology. We started from the assumption that 2D branching structures are comprised of only bifurcations 1 and crossings. So the number of adjacent segments (3 or 4) at every critical region should be enough to classify them. However, misclassifications during the system development implied a more complete description. The system became more and more robust, as we moved further by taking into account new pieces of information, such as orientation between incoming and outgoing direction vectors, proximity relation between neighbor crossing regions, besides the basic and first criterion of number of adjacent segments to each crossing region.

In brief, the critical regions classification rules take into account the angle β between the inwards direction vector - \hat{v}_0 - at the current critical region s_1 and its outwards direction vector - \hat{v}_2 ; the angle γ between any pair of outwards direction vectors \hat{v}_i, \hat{v}_j ; and the cardinality $|E_1|$ of the set of outwards direction vectors E_1 related to the current critical region s_1 .

3.3 Contour Extraction

The main steps composing the *BSCEA* are depicted on the respective flowchart in Fig. 8. The contour following algorithm explained in [7] has been adopted in this work.

Since the input for the *BSCEA* corresponds to a union of the labeled skeleton and the soma images, it is necessary to adopt a policy to properly find the next pixel in each case. Hence, the *BSCEA* considers contouring branches as the default case, taking the first background pixel which is also neighbor of a foreground pixel in the neighborhood defined by the chain-code. Conversely, the *BSCEA* considers contouring the soma as a particular case, taking the last pixel, instead of the first one, to be included as contour. By so doing, the *BSCEA* is able to contour branches, while preserving the ability of more traditional approaches to circumvent the problem of contouring occasional one-pixel wide entrances into the soma, consequently allowing the contour to be closed [7].

It is also necessary to devise a strategy for critical regions processing, according to their classes, as described in section 3.2.2. Regions classified as Bifurcation should be contoured outwards, while those ones classified as either Superposition or Cross-

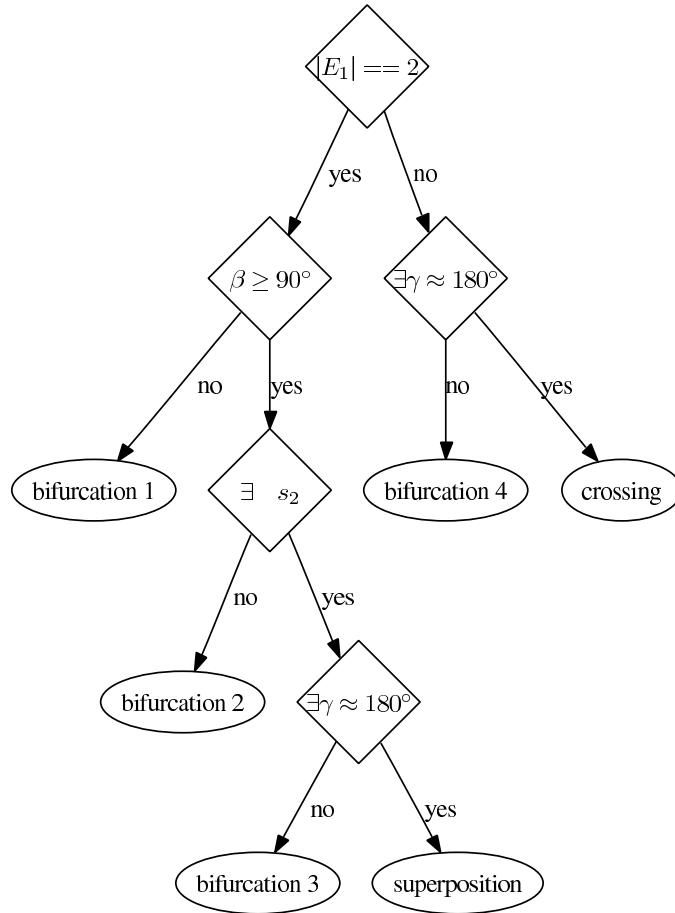


Fig. 7. Decision tree representing the sequence of rules applied for Critical Regions Classification by the **BTA**. The variables s_1 and s_2 represent respectively the current critical region and other nearby in the set S of critical regions (see Fig. 4). The variables E_1 and E_2 stand for the sets of unitary **outwards vectors** related to s_1 and s_2 respectively. Since a critical region is individually classified as it is found during the labeling process, it is clear that the current critical region s_1 exists. The variable β measures the angle between the inwards direction vector and the outwards direction vector, while the variable γ measures the angle between any two outwards direction vectors

ing should be contoured inwards, through pointer addresses written to the *Pointers Map* data structure during the tracking stage. The integration between soma and labeled skeleton is critical for the successful contour extraction, since it guarantees the contour closing.

The *BSCEA* can deal with both cases by taking into account the labels of previous and current pixels, which convey valuable information concerning particular situations, i.e. if the critical region is a bifurcation, "contour it outwards" (see Fig. 8 and Fig. 9), as well as the traditional contour extraction algorithm would [7]. In case it is a superposition or a crossing, "contour it inwards", (see Fig. 8 and Fig. 10), which

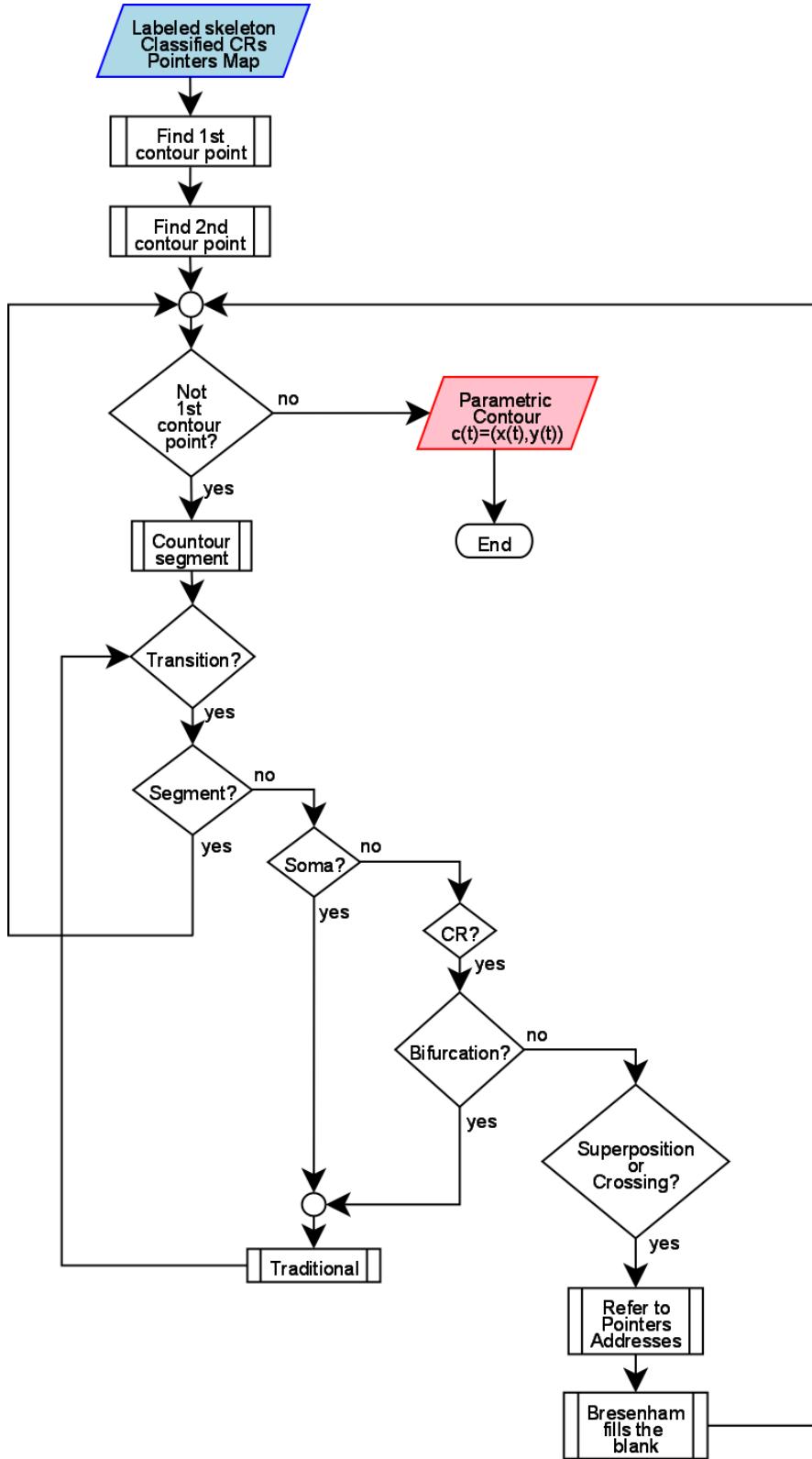


Fig. 8. Flowchart of the BSCEA.

means to trace a line between the current segment end point and the next segment starting point. Both points are known from the pointers marked by the *BTA*. The line is traced by using the Bresenham algorithm [2] for tracing a digital straight line segment.

Notice that the *BSCEA* cannot tell which pixels of a superposition or crossing are related one another or to a branch, since the projection from the 3D neuron onto the 2D plane suppresses this information. Such a problem is circumvented by replacing the shared pixels in the critical region by two short intercepting segments given by the Bresenham's algorithm, as illustrated in Fig.10.

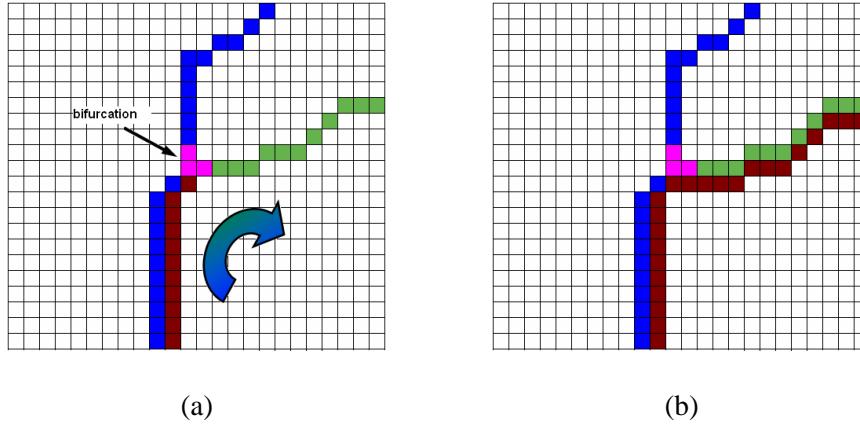


Fig. 9. Contouring a bifurcation. Branches appear labeled in blue and green, while the critical region previously classified as a bifurcation appears in magenta. The contour is shown in brown. (a) By detecting labels transition, the *BSCEA* identifies that it has arrived at a bifurcation, thus deciding to contour the shape outwards. (b) Having left the critical region behind, it proceeds until reaching another critical region.

4 Results

All the methods described in this work have been implemented as *Matlab*[®] scripts, using the *SDC Morphology Toolbox for MatLab*TM [12]. The overall method has been evaluated with respect to a data set containing several images of ganglion cells in the retina of cats, acquired by camera lucida. In order to reflect an important biological investigation, we have chosen 2D neuron images from [17].

Results obtained for three images were chosen to be presented in this work. Figure 11 shows labeled neuronal images (right column) obtained from alpha, delta and epsilon (left column) types of neurons. New labels were assigned to dendrite segments originating from branches. The algorithm is able to distinguish between critical region classes, reflected by the correct assigned labels for the outwards segments from such structures. Notice how the cases of close parallelism imply the

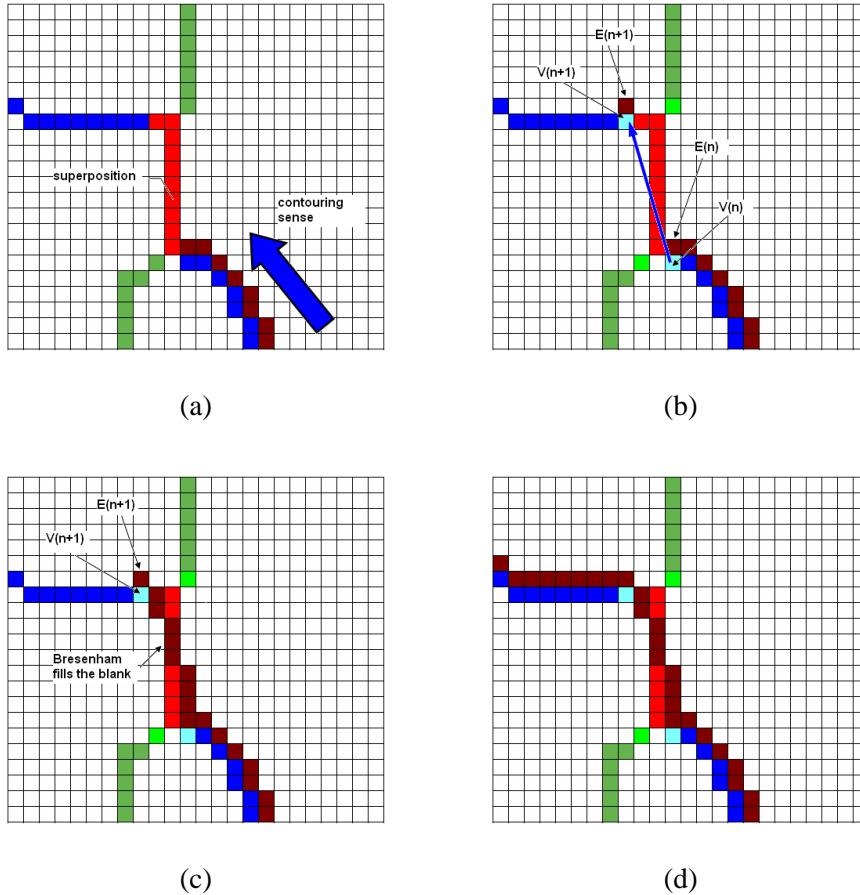


Fig. 10. Contouring an overlap. Branches appear labeled in blue and green, while the critical region previously classified as a superposition appears in red. The contour is shown in brown. (a) By detecting labels transition, the *BSCEA* identifies that it has arrived at an overlap, thus deciding to contour the shape inwards. (b) Firstly the *BSCEA* looks in the vicinity for the pointer P_n , related to the current segment end point pixel V_n . The pointer P_n stores the address of the next segment pixel beyond the critical region, namely V_{n+1} . Subsequently the algorithm determines the direction relationship $d_{E(n)} \leftrightarrow V_n$. Assuming $d_{E(n+1)} \leftrightarrow V_{n+1} = d_{E(n)} \leftrightarrow V_n$, it finds the next segment starting pixel $E(n+1)$. (c) The Bresenham's algorithm is applied to trace a digital line between $E(n)$ and $E(n+1)$, filling the blank in the parametric contour signal over the *critical region*. (d) Having left the critical region behind, it proceeds until reaching another critical region.

BTA to label the clumped segments as superposition regions. Moreover, long overlaps are also labeled as superposition regions, as long as such an overlap is smaller than D_{max} , which is the minimum path length allowed between two critical regions. Extremely close bifurcations of type 1 can be labeled as a bifurcation of type 4. All the bifurcations, superpositions and crossings have been correctly labeled.

Results for the *Branching Structures Contour Extraction Algorithm* are presented in Figure 12, where one can see the parametric contour trace for the shape and a comparison between the results obtained by using both the traditional and the *BSCEA*

approaches. Observe from Figures 12(a), 12(c) and 12(e) how the traditional algorithm did not afford access to the innermost neuron contour portions, while the *BSCEA* conversely ensured full access to all neuronal processes, as shown in Figures 12(b), 12(d) and 12(f).

Additional experiments have been carried out in order to validate the algorithms and to assess their performance with respect to noisy neuronal images. A validation test was performed with a synthetic neuron image, by labeling it manually and automatically through the *BTA* (Fig. 13). Despite differences between manual and automatic labelings, due to distinct assessments of tangent continuity for some bifurcations in both approaches, notice that both labelings are consistent, providing suitable input for the *BSCEA* which yielded identical parametric contours for both labelings. The robustness of the proposed methodology for noisy versions of the same synthetic neuron image has been tested by convolving the original synthetic image with six different 2D Gaussians, using bandwidths parametrized by values of standard deviations spanning from 10^{-8} up to $2 \cdot 10^{-6}$ in the Fourier domain. Figure 14 shows results for extreme cases, namely the original synthetic image and the smoothest version of it. Notice that the *BTA* provided consistent labelings for the original and the smoothed versions, and the *BSCEA* yielded identical parametric contours for both cases.

5 Concluding Remarks

The proper shape characterization of branching structures is a particularly important problem, as it plays a central role in several areas of medicine and biology, especially in neuroscience. Indeed, the current understanding of the physiological dynamics in biological neuronal networks can be reinforced through the proper characterization of neuronal cells shapes, since both the amount of synapses and the way in which neurons organize in networks are strongly related to the cells shapes.

Accurate contour following in neuronal shapes has not been possible so far because of the presence of overlappings among neuronal processes implied by the projection from 3D neuronal shapes onto 2D images. Whenever a crossing takes place in these images, traditional contour following algorithms, based on the chain-code, fail to enter their innermost regions. The present work described an original methodology capable of properly tackling the problem of following the contour of branching structures, even in the presence of intercepting branches and noise. The main original contributions of our method⁵ include both the tracking of the branching structures, such as neurons, as well as the extraction of the respective parametric contours. In addition, the features adopted in this work to classify critical regions

⁵ Preliminary results of the proposed approach have been described in conferences ([14, 16]).

– such as shape, number of adjacent segments and angles among segments, have intrinsic potential for providing additional information to be used in neuronal characterization and classification.

Future possible developments include the extension of the methodology to separate cells in images containing multiple cells. In addition, several applications of the methodology proposed in this work can be performed regarding neuronal networks images as well as other types of biological structures such as retinal vessel trees.

Acknowledgement

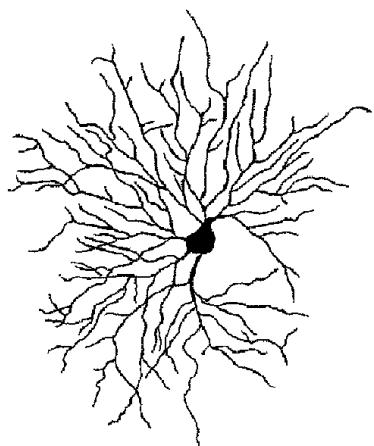
The authors are grateful to Fapesp, CNPq, Capes and Finep for financial support, as well as to the *Nature Publishing Group* for the license to use the images in [17].

References

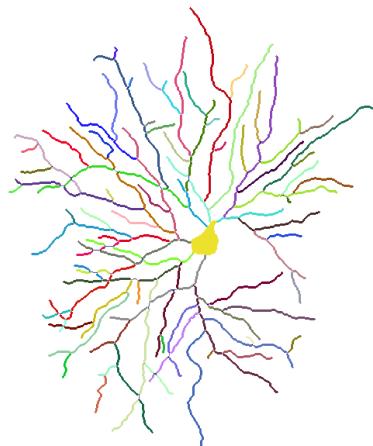
- [1] S. Arulampalam, S. Maskell, N. Gordon, T. Clapp, A tutorial on particle filters for on-line non-linear/non-gaussian bayesian tracking, *IEEE Transactions on Signal Processing* 50 (2) (2002) 174–188.
- [2] J. Bresenham, Algorithm for computer control of a digital plotter, *IBM Systems Journal* 4 (1) (1965) 25–30.
- [3] F. Caserta, H. Stanley, W. Eldred, G. Daccord, R. Haussman, J. Nittmann, Physical mechanisms underlying neurite outgrowth: A quantitative analysis of neuronal shape, *Physical Review Letters* 64 (1) (1990) 95–98.
- [4] R. M. Cesar-Jr., L. da F. Costa, Neural cell classification using wavelets and multiscale curvature, *Biological Cybernetics* 79 (4) (1998) 347–360.
- [5] R. M. Cesar-Jr., L. da F. Costa, Dendrogram generation for neural shape analysis, *The Journal of Neuroscience Methods* 93 (1999) 121–131.
- [6] L. da F. Costa, Morphological complex networks: Can individual morphology determine the general connectivity and dynamics of networks?, in: COSIN 2005 Final meeting, Conference on Complex Networks, Salou, Spain, 2005, <http://xxx.lanl.gov/abs/q-bio.MN/0503041>.
- [7] L. da F. Costa, R. M. Cesar-Jr., *Shape Analysis and Classification: Theory and Practice*, CRC Press, 2001.
- [8] L. da F. Costa, E. Manoel, F. Faucereau, J. Chelly, J. van Pelt, G. Ramakers, A shape analysis framework for neuromorphometry, *Network: Computation in Neural Systems* 13 (August 2002) 283–310(28), <http://www.ingentaconnect.com/content/apl/network/2002/00000013/00000003/art00303>.

- [9] E. R. Dougherty, R. A. Lotufo, Hands-On Morphological Image Processing, SPIE-International Society for Optical Engine, 2003.
- [10] P. F. Gabriel, J. G. Verly, J. H. Piater, A. Genon, The state of the art in multiple object tracking under occlusion in video sequences, in: In Advanced Concepts for Intelligent Vision Systems (ACIVS), 2003, 2003.
- [11] G. T. Herman, Geometry of Digital Spaces, Birkhauser Boston, 1998.
- [12] S. Information Systems, SDC Morphology Toolbox for Matlab, <http://www.mmorph.com/>, version 1.5 (October 2006).
- [13] H. Jelinek, E. Fernandez, Neurons and fractals: How reliable and useful are calculations of fractal dimensions?, *Journal of Neuroscience Methods* 81 (1-2) (1998) 9–18.
- [14] J. J. G. Leandro, R. M. Cesar-Jr., L. da F. Costa, Determining the branching structure of 3D structures from respective 2D projections, in: Proc. 19th SIBGRAPI - Brazilian Symposium on Computer Graphics and Image Processing, IEEE Computer Society Press, 2006.
- [15] J. J. G. Leandro, R. M. Cesar Jr, L. da F. Costa, Automatic contour extraction from 2d neuron images, <http://aps.arxiv.org/abs/0804.3234> (2008).
- [16] J. J. G. Leandro, R. M. Cesar-Jr., L. da F. Costa, Automatic contour extraction of neurons in presence of overlap, in: Workshop on Bio-Image Informatics: Biological Imaging, Computer Vision and Data Mining, 2008, Santa Barbara, CA, USA, 2008, <http://www.ece.ucsb.edu/bioimage/workshop2008/index.html>.
- [17] R. H. Masland, The fundamental plan of the retina, *Nature Neuroscience* 4 (9) (2001) 877–886.
- [18] K. Morigiwa, M. Tauci, Y. Fukuda, Fractal analysis of ganglion cell dendritic branching patterns of the rat and cat retinae, *Neuroscience Research Suppl.* 10 (1989) S131–S140.
- [19] M. B. L. Rocchi, D. Sisti, M. C. Albertini, L. Teodori, Current trends in shape and texture analysis in neurology: Aspects of the morphological substrate of volume and wiring transmission, *Brain Research Reviews* 55 (2007) 97–107.
- [20] K. Rothaus, P. Rhiem, X. Jiang, Separation of the retinal vascular graph in arteries and veins, in: F. Escolano, M. Vento (eds.), GbRPR 2007, Graph-Based Representations in Pattern Recognition, 6th IAPR-TC-15 International Workshop, Alicante, Spain, Proceedings, vol. 4538 of Lecture Notes in Computer Science, Springer Verlag, 2007, <http://dblp.uni-trier.de/db/conf/gbrpr/gbrpr2007.html#RothausRJ07>.
- [21] K. Rothaus, P. Rhiem, X. Jiang, Separation of the retinal vascular graph based upon structural knowledge, doi:10.1016/j.imavis.2008.02.013, in press, *Image and Vision Computing* (2008).
- [22] R. Sedgewick, Algorithms, Addison-Wesley Publishing Company, Reading, MA, USA, 1983.

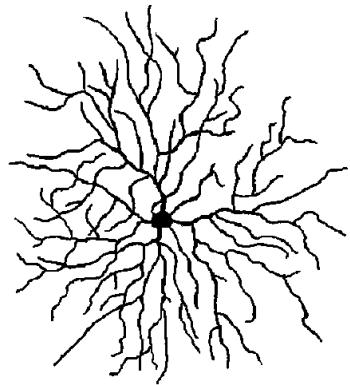
[23] P. Soille, Morphological Image Analysis: Principles and Applications, Springer Verlag, 1999.



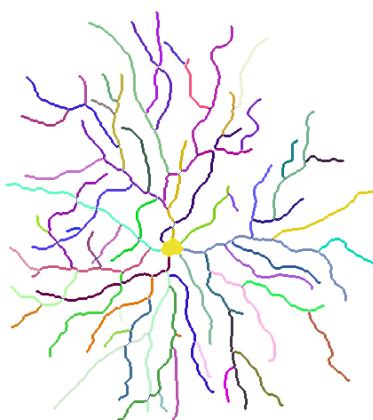
(a)



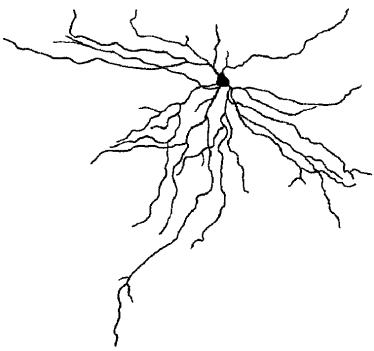
(b)



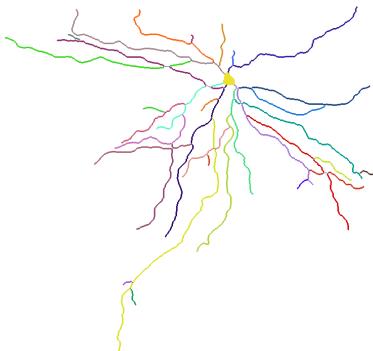
(c)



(d)



(e)



(f)

Fig. 11. Results obtained by the *Branches Tracking Algorithm* when applied to alpha, delta and epsilon neuron images. (a) Alpha (c) Delta and (e) Epsilon neuron images and respective labeled images in (b), (d) and (f). Distinct branches appear in different colours.

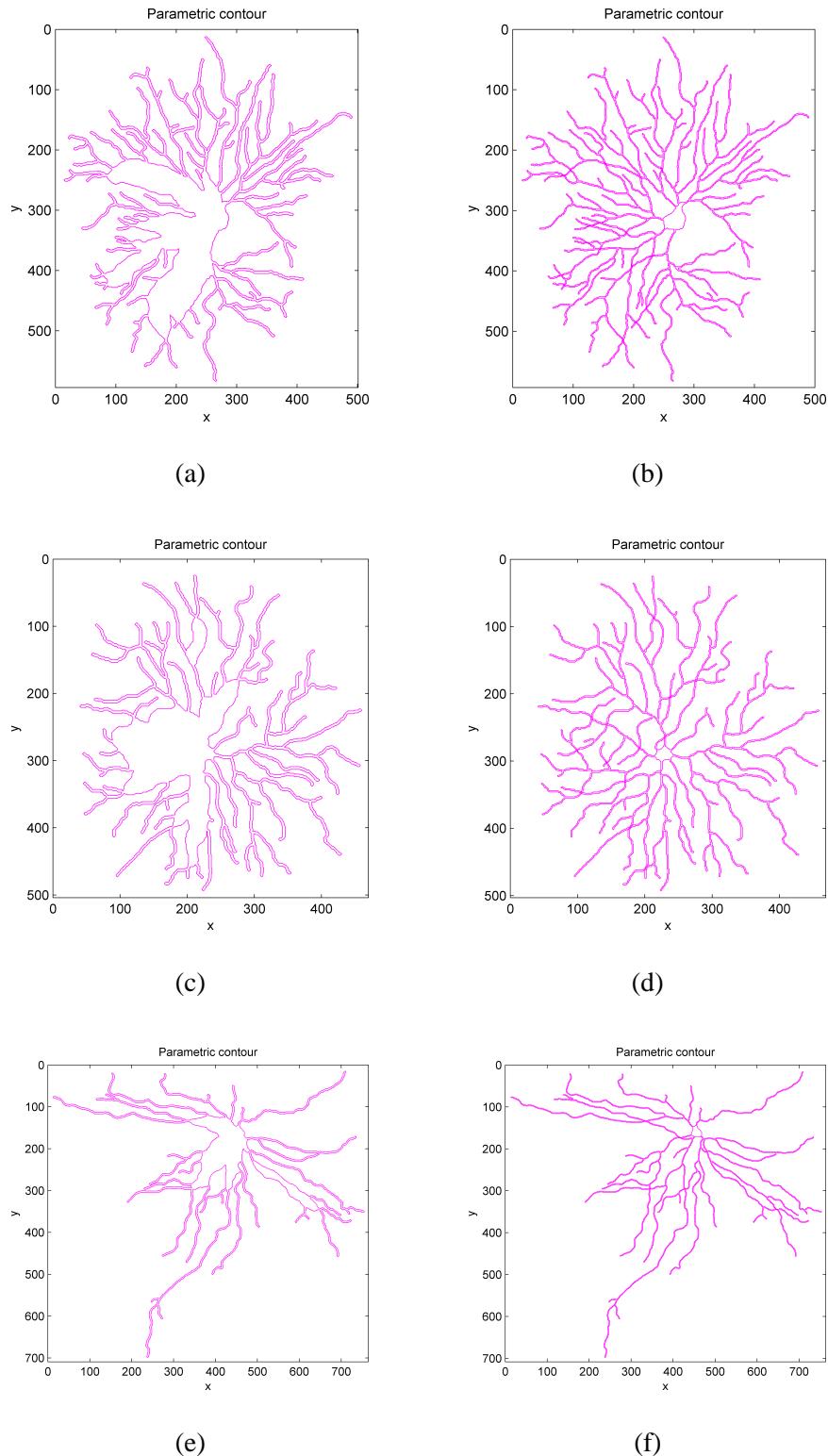


Fig. 12. Comparison for alpha, delta and epsilon neuron images between results yielded by the traditional (a-c-e) and (b-d-f) BSCEA algorithms. The BSCEA enters all the regions, surpassing the traditional algorithm.

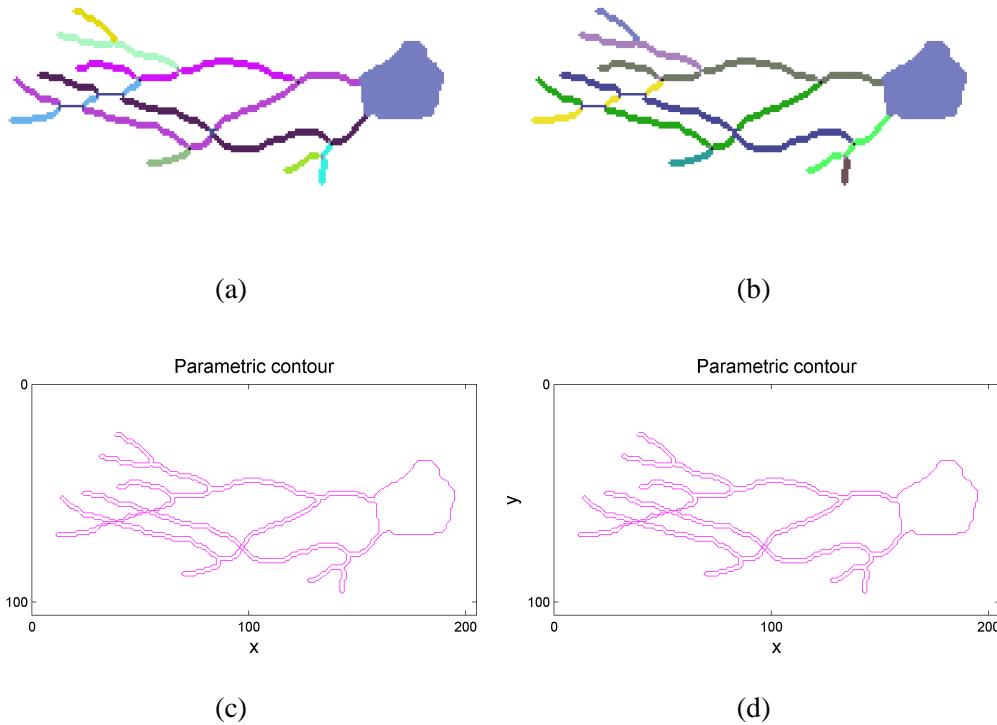


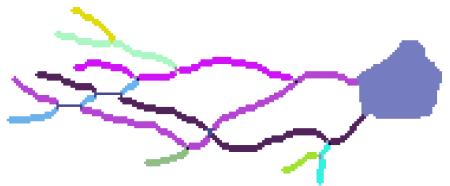
Fig. 13. Comparison between a synthetic neuron shape labeled automatically (a) and manually (b). Parametric contours provided by the BSCEA for the automatic (c) and manual (d) labelings. Despite minor differences between automatic and manual labelings, notice that both contours are in complete accordance.



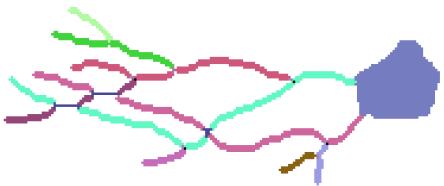
(a)



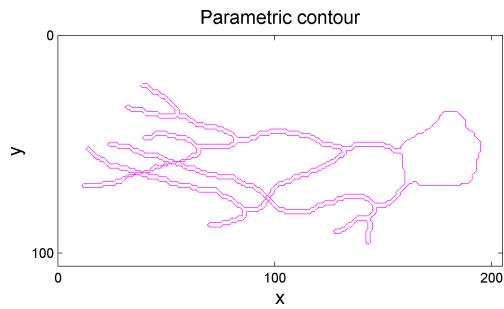
(b)



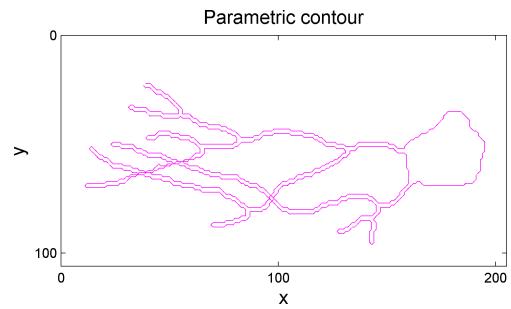
(c)



(d)



(e)



(f)

Fig. 14. Assessment of noise effects on neuron shape labeling and respective contour extraction. A synthetic neuron shape has been corrupted by 2D Gaussians with different bandwidths. The Gaussian scale parameters σ in Fourier domain varied from 10^{-8} up to $2 \cdot 10^{-6}$. Both BTA and BSCEA ensured robustness within this range of smoothing bandwidths.

Referências Bibliográficas

- [1] K. T. Alligood, T. D. Sauer, and J. A. Yorke. *Chaos: An Introduction to Dynamical Systems*. Springer-Verlag New York, Incorporated, New York, USA, first edition, 1996. [15](#), [17](#)
- [2] J.-P. Antoine. Wavelet analysis: A new tool in signal processing. *Physicalia Magazine*, 16:17–42, 1994. [10](#), [12](#)
- [3] J.-P. Antoine, P. Carette, R. Murenzi, and B. Piette. Image analysis with two-dimensional continuous wavelet transform. *Signal Processing*, 31:241–272, 1993. [9](#), [10](#), [11](#)
- [4] A. Arnéodo, N. Decoster, and S. Roux. A wavelet-based method for multifractal image analysis. i. methodology and test applications on isotropic and anisotropic random rough surfaces. *The European Physical Journal B*, 15:567–600, 2000. [9](#), [11](#), [12](#)
- [5] S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for on-line non-linear/non-gaussian bayesian tracking. *IEEE Transactions on Signal Processing*, 50(2):174–188, feb 2002. [63](#)
- [6] J. O. Berger. *Statistical Decision Theory and Bayesian Analysis*. Springer-Verlag New York, Inc, New York, USA, 2nd edition, 1985. [30](#)
- [7] S. Bingham and M. Kot. Multidimensional trees, range searching, and a correlation dimension algorithm of reduced complexity. *Physics Letters A*, 140:327–330, 1989. [23](#)
- [8] J. L. Boldrini, S. I. R. Costa, V. L. Figueiredo, and H. G. Wetzler. *Álgebra Linear*. Harper & Row do Brasil, São Paulo, 3rd edition, 1980. [14](#)
- [9] S. T. Bow. *Pattern Recognition and Image Processing*. Marcel Dekker, Inc., New York, NY, USA, 2nd edition, 2002. [43](#), [44](#)
- [10] B. Boycott and H. Wassle. The morphological types of ganglion cells of the domestic cat's retina. *Journal of Physiology*, 240(2):397–419, July 1974. [53](#), [54](#)
- [11] A. P. Bradley. The use of the area under the roc curve in the evaluation of machine learning algorithms. *Pattern Recognition*, 30(7):1145–1159, 1997. [41](#)
- [12] D. Bray. Branching patterns of individual sympathetic neurons in culture. *The Journal of Cell Biology*, 56:702–712, 1973. [1](#)

- [13] J. Bresenham. Algorithm for computer control of a digital plotter. *IBM Systems Journal*, 4(1):25–30, 1965. [69](#)
- [14] M. F. Carmo. *Differential geometry of curves and surface*. Prentice-Hall, Englewood Cliffs, N.J., 1976. [50](#)
- [15] E. L. G. Casella. *Theory of point estimation*. Springer-Verlag New York Inc., New York, USA, 2nd edition, 1998. [30](#)
- [16] F. Caserta, H. Stanley, W. Eldred, G. Daccord, R. Haussman, and J. Nittmann. Physical mechanisms underlying neurite outgrowth: A quantitative analysis of neuronal shape. *Physical Review Letters*, 64(1):95–98, 1990. [47](#)
- [17] K. R. Castleman. *Digital Image Processing*. Prentice-Hall, Englewood Cliffs, NJ, 1979. [10](#), [48](#)
- [18] R. M. Cesar-Jr. *Análise Multi-Escala de Formas Bidimensionais*. PhD thesis, IFSC - USP, São Carlos, 1997. [50](#), [54](#)
- [19] R. M. Cesar-Jr. and L. da F. Costa. Neural cell classification using wavelets and multiscale curvature. *Biological Cybernetics*, 79(4):347–360, 1998. [9](#), [49](#)
- [20] R. M. Cesar-Jr. and L. da F. Costa. Dendrogram generation for neural shape analysis. *The Journal of Neuroscience Methods*, 93:121–131, 1999. [49](#)
- [21] Q. Chen, T. J. Chen, P. C. Letourneau, L. da Fontoura Costa, and D. Schubert. Modifier of cell adhesion regulates n-cadherin-mediated cell-cell adhesion and neurite outgrowth. *The Journal of Neuroscience*, 25(2):281–290, 2005. [60](#)
- [22] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*. The MIT Press, 2001. [9](#), [64](#)
- [23] C. da C. Freitas, C. D. Rennó, and M. A. Sousa-Jr. Estatística - curso i. Material de referência para o curso 1 de Estatística, Março 2003. [33](#), [39](#)
- [24] L. da F. Costa. Morphological complex networks: Can individual morphology determine the general connectivity and dynamics of networks? In *COSIN 2005 Final meeting, Conference on Complex Networks*, Salou, Spain, March 2005. <http://xxx.lanl.gov/abs/q-bio.MN/0503041>. [47](#)
- [25] L. da F. Costa and R. M. Cesar-Jr. *Shape Analysis and Classification: Theory and Practice*. CRC Press, 2001. [4](#), [8](#), [9](#), [10](#), [12](#), [13](#), [14](#), [15](#), [16](#), [20](#), [21](#), [28](#), [35](#), [49](#), [51](#), [67](#), [68](#), [83](#)
- [26] L. da F. Costa, E. Manoel, F. Faucereau, J. Chelly, J. van Pelt, and G. Ramakers. A shape analysis framework for neuromorphometry. *Network: Computation in Neural Systems*, 13:283–310(28), August 2002. <http://www.ingentaconnect.com/content/apl/network/2002/00000013/00000003/art00303>. [8](#), [9](#), [47](#), [49](#)
- [27] A. C. da S. Braga. *Curvas ROC: Aspectos Funcionais e Aplicações*. PhD thesis, Universidade do Minho, Braga, Portugal, 2000. In Portuguese. [40](#)

- [28] P. S. Doods and D. H. Rothman. Unified view of scaling laws for river networks. *Phys. Rev. E*, 59(5):4865–4877, 1999. [1](#), [8](#)
- [29] R. O. Duda, P. E. Hart, and D. Stork. *Pattern Classification*. Wiley-Interscience, NY, 2000. [4](#), [8](#), [40](#), [44](#)
- [30] K. Falconer. *Fractal Geometry: Mathematical Foundations and Applications*. John Wiley & Sons, Chichester, England, 1990. [14](#), [16](#)
- [31] F. Family, B. Masters, and D. Platt. Fractal pattern formation in human retinal vessels. *Physica D*, 38:98–103, 1989. [15](#)
- [32] J. D. Farmer, E. Ott, and J. A. Yorke. The dimension of chaotic attractors. *Physica D*, 7(1):153–180, 1983. [15](#), [17](#), [18](#), [19](#), [21](#), [23](#)
- [33] T. Fawcett. An introduction to roc analysis. *ptl*, 27(8):861–874, 2006. [28](#), [39](#), [40](#), [41](#)
- [34] E. T. Federighi. Extended tables of the percentage points of student's t-distribution. *Journal of the American Statistical Association*, 54(287):683–688, 1959. [33](#)
- [35] V. Fleury, J. F. Gouyet, and M. Léonetti. *Branching in Nature - Dynamics and Morphogenesis of Branching Structures*. Springer-Verlag, Berlin Heidelberg, 2001. [4](#)
- [36] J. G. Fox. *Biology and Diseases of the Ferret*. Blackwell Publishing, 2nd edition, 1998. [54](#)
- [37] H. Freeman. On the encoding of arbitrary geometric configurations. *IEEE Eletr. Comp.*, EC-10:260–268, 1961. [50](#)
- [38] H. Froehling, J. P. Crutchfield, D. Farmer, N. H. Packard, and R. Shaw. On determining the dimension of chaotic flows. *Physica D Nonlinear Phenomena*, 3:605–617, 1981. [18](#)
- [39] K. Fukunaga. *Introduction to statistical pattern recognition (2nd ed.)*. Academic Press Professional, Inc., San Diego, CA, USA, 1990. [8](#), [41](#), [42](#)
- [40] M. C. Gandur. *Comportamento Dinâmico Complexo em Despelamento de Fitas Adesivas*. PhD thesis, Instituto de Química - Universidade Estadual de Campinas, São Paulo, Brasil, 2001. In Portuguese. [17](#)
- [41] R. Gentleman and R. Ihaka. The R Project for Statistical Computing. <http://www.r-project.org/>, 2008. GNU - GPL. [34](#)
- [42] R. C. Gonzalez and R. E. Woods. *Digital Image Processing*. Addison-Wesley Publishing Company, 1992. [21](#)
- [43] I. Grassberger, P. Procaccia. Characterization of strange attractors. *Physical Review Letters*, 50(5):346–349, 1983. [17](#), [18](#), [19](#), [20](#), [21](#)
- [44] I. Grassberger, P. Procaccia. Measuring the strangeness of strange attractors. *Physica D*, 9:189–208, 1983a. [18](#), [19](#), [20](#), [21](#)
- [45] H. S. Greenside, A. Wolf, J. Swift, and T. Pignataro. Impracticality of a box-counting algorithm for calculating the dimensionality of strange attractors. *Physical Review A*, 25:3453–3456, 1982. [18](#)

- [46] A. Grossmann. Wavelet transforms and edge detection. In S. Albeverio, P. Blanchard, M. Hazeinkel, and L. Streit, editors, *Stochastic Processes in Physics and Engineering*. Reidel Publishing Company, Dordrecht, 1988. [10](#)
- [47] F. Hausdorff. Dimension und äüeres maß. *Mathematische Annalen*, 79(157):157–179, 1919. [15](#)
- [48] G. T. Herman. *Geometry of Digital Spaces*. Birkhauser Boston, 1998. [49](#)
- [49] H. Honda and K. Yoshizato. Formation of the branching pattern of blood vessels in the wall of the avian yolk sac studied by a computer simulation. *Development, Growth and Differentiation*, 39(5):581–589, 1997. [1](#)
- [50] S. Information Systems. SDC Morphology Toolbox for Matlab. <http://www.mmorph.com/>, October 2006. version 1.5. [71](#)
- [51] H. Jelinek and R. M. Cesar-Jr. Segmentation of retinal fundus vasculature in non-mydriatic camera images using wavelets. In J. Suri and T. Laxminarayan, editors, *Angiography Imaging: State-of-the-Art Acquisition, Image Processing and Applications Using Magnetic Resonance, Computer Tomography, Ultrasound and X-rays*, chapter 5. CRC Press, 2003. [9, 11, 20, 21, 25, 28, 29, 30](#)
- [52] H. Jelinek, R. M. Cesar-Jr., and J. J. G. Leandro. Exploring the structure-function relationship of cat retinal ganglion cells using wavelets. *Brain and Mind*, 4(1):67–90, 2003. [10, 54](#)
- [53] H. Jelinek, R. M. Cesar-Jr., J. J. G. Leandro, and I. Spence. Automated computational morphometric analysis of the cat retinal α/y , β/x and δ ganglion cells using clustering algorithms. *Journal of Integrative Neuroscience*, 3(4):415–432, 2004. [9, 10, 14, 48, 54](#)
- [54] H. Jelinek and E. Fernandez. Neurons and fractals: How reliable and useful are calculations of fractal dimensions? *Journal of Neuroscience Methods*, 81(1-2):9–18, 1998. [16, 47](#)
- [55] H. Jelinek, J. J. G. Leandro, R. M. Cesar-Jr., and M. Cree. Classification of pathology in diabetic eye disease. In *Proc. APRS Workshop on Digital Image Computing - Pattern Recognition and Imaging for Medical Applications*, Brisbane, Griffith University, 2005. [4, 81](#)
- [56] H. F. Jelinek, M. J. Cree, J. J. G. Leandro, J. V. B. Soares, R. M. Cesar-Jr., and A. Luckie. Automated segmentation of retinal blood vessels and identification of proliferative diabetic retinopathy. *Journal of the Optical Society of America A*, 24(5):1448–1456, 2007. [4, 7, 8, 9, 11, 25, 34, 44, 81](#)
- [57] E. R. Kandel, J. H. Schwartz, and T. M. Jessell. *Principles of Neural Science*. McGraw-Hill Publishing Companies, Inc., 4th edition, 2000. [47](#)
- [58] A. N. Kolmogorov. A new invariant for transitive dynamical systems. *Dokl. An. SSR*, 119:861–864, 1958. [15](#)
- [59] G. A. Korn and T. M. Korn. *Mathematical Handbook for Scientists and Engineers: Definitions, Theorems, and Formulas for Reference and Review*. Dover Publications, New York, 2nd edition, June 2000. [12, 13](#)

- [60] J. J. G. Leandro, R. M. Cesar-Jr., and L. da F. Costa. Determining the branching structure of 3D structures from respective 2D projections. In *Proc. 19th SIBGRAPI - Brazilian Symposium on Computer Graphics and Image Processing*. IEEE Computer Society Press, 2006. 4, 5, 82, 84
- [61] J. J. G. Leandro, R. M. Cesar Jr, and L. da F. Costa. Automatic contour extraction from 2d neuron images. <http://dx.doi.org/10.1016/j.jneumeth.2008.10.037>, 2008. In Press, Journal of Neuroscience Methods - Elsevier. 5, 56, 82, 141
- [62] J. J. G. Leandro, R. M. Cesar-Jr., and L. da F. Costa. Automatic contour extraction of neurons in presence of overlap. In *Workshop on Bio-Image Informatics: Biological Imaging, Computer Vision and Data Mining, 2008*, Santa Barbara, CA, USA, 2008. <http://www.ece.ucsbg.edu/bioimage/workshop2008/index.html>. 5, 82, 84
- [63] J. J. G. Leandro, R. M. Cesar Jr, and L. da F. Costa. Technical Report - Automatic contour extraction from 2d neuron images, 2008. <http://aps.arxiv.org/abs/0804.3234>. 5, 56, 82, 141
- [64] J. J. G. Leandro, R. M. Cesar-Jr, and H. Jelinek. Blood vessels segmentation in retina: Preliminary assessment of the mathematical morphology and of the wavelet transform techniques. In *Proc. Brazilian Conference on Computer Graphics, Image Processing and Vision, (SIBGRAPI-01, Florianópolis - SC, Out 2001)*, pages 84–90. IEEE Computer Society Press, 2001. 9, 10, 11
- [65] J. J. G. Leandro, J. V. B. Soares, R. M. Cesar-Jr., and H. Jelinek. Blood vessels segmentation in non-mydriatic images using wavelets and statistical classifiers. In *Proc. 16th Brazilian Symposium on Computer Graphics, Image Processing - Sibgrapi03*, pages 262–269. IEEE Computer Society Press, 2003. 10, 11
- [66] J. Lewalle. Hamiltonian formulation for the diffusion equation. *Phys. Rev. E*, 55(2):1590–1599, Feb 1997. 12, 21
- [67] E. N. Lorenz. Deterministic non-periodic flow. *J. Atmos. Sci.*, 20:130–41, 1963. 17
- [68] A. Majumdar, A. M. Alencar, S. V. Bldyrev, Z. Hantos, H. E. Stanley, and B. Suki. Characterization of the branching structure of the lung from 'macroscopic' pressure-volume measurements. *phrl*, 87(5):058102–1–058102–4, 2001. 1
- [69] S. Mallat and W. Hwang. Singularity detection and processing with wavelets. *IEEE Transactions on Information Theory*, 38(2):617–643, 1992. 21
- [70] B. Mandelbrot. *The Fractal Geometry of Nature*. W.H. Freeman and Company, New York, 1982. 14
- [71] G. S. Mani. Modelling of lighting using fractals. In *Proceedings of the International Conference on Electromagnetic Interference and Compatibility '99.*, pages 493–496, 1999. 1
- [72] D. Marr and E. Hildreth. Theory of edge detection. *Proceedings of Royal Society of London B*, 207:187–217, 1980. 21

- [73] R. H. Masland. The fundamental plan of the retina. *Nature Neuroscience*, 4(9):877–886, 2001. [54](#)
- [74] B. R. Masters. Fractal analysis of the vascular tree in the human retina. *Annual Review of Biomedical Engineering*, 6:427–452, 2004. [15, 16](#)
- [75] X. Miao, W. He, H. Yang, and L. Fang. A speedy algorithm for estimating the correlation dimension. *International Journal of Modern Physics B*, 17(22-24):4284–4289, 2003. [23](#)
- [76] D. R. Montgomery and W. E. Dietrich. Channel initiation and the problem of landscape scale. *Science*, 255:826–830, 1992. [1](#)
- [77] K. Morigiwa, M. Tauci, and Y. Fukuda. Fractal analysis of ganglion cell dendritic branching patterns of the rat and cat retinae. *Neuroscience Research Suppl.*, 10:S131–S140, 1989. [47](#)
- [78] J. D. Murray. *Mathematical Biology - An introduction*. Springer-Verlag, New York, Incorporated, 3rd edition, 2002. [16](#)
- [79] E. Ott. *Chaos in Dynamical Systems*. Cambridge University Press, Cambridge, 1993. [17](#)
- [80] J. Panico and P. Sterling. Retinal neurons and vessels are not fractal but space-filling. *The Journal of Comparative Neurology*, 361:479–490, 1995. [16](#)
- [81] A. Papoulis. *Probability, Random Variables, and Stochastic Processes*. McGraw-Hill, New York, USA, 3rd edition, 1991. [32](#)
- [82] V. Prasad, D. Semwogerere, and E. R. Weeks. Confocal microscopy of colloids. *Journal of Physics: Condensed Matter*, 19(11), 2007. [51](#)
- [83] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical recipes in C (2nd ed.): the art of scientific computing*. Cambridge University Press, New York, NY, USA, 1992. [29, 30, 32, 33](#)
- [84] S. Ramon y Cajal. *Histologie Du Système Nerveux de l'Homme et Des Vertébrés*. Maloine, Paris, 1911. [55](#)
- [85] O. Rioul and M. Vetterli. Wavelets and signal processing. *IEEE Signal Processing Magazine*, pages 14–89, 1991. [10, 11](#)
- [86] M. B. L. Rocchi, D. Sisti, M. C. Albertini, and L. Teodori. Current trends in shape and texture analysis in neurology: Aspects of the morphological substrate of volume and wiring transmission. *Brain Research Reviews*, 55:97–107, 2007. [47](#)
- [87] K. Rothaus, P. Rhiem, and X. Jiang. Separation of the retinal vascular graph in arteries and veins. In F. Escolano and M. Vento, editors, *GbRPR 2007, Graph-Based Representations in Pattern Recognition, 6th IAPR-TC-15 International Workshop, Alicante, Spain, Proceedings*, volume 4538 of *Lecture Notes in Computer Science*, pages 251–262. Springer Verlag, June 2007. <http://www.springerlink.com/content/d573048432h4k13x/>. [48](#)
- [88] K. Rothaus, P. Rhiem, and X. Jiang. Separation of the retinal vascular graph based upon structural knowledge. doi:10.1016/j.imavis.2008.02.013, 2008. in press, *Image and Vision Computing*. [48](#)

- [89] J. A. Sanchez, W. Zeng, V. R. Coluci, and C. Simpson. How similar are branching networks in nature? a view from ocean: Caribbean gorgonian corals. *Journal of Theoretical Biology*, 222:135–138, 2003. [1](#)
- [90] C. E. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27:374–423 623–656, July and October 1948. [13](#)
- [91] M. Shelhamer. *Nonlinear Dynamics in Physiology: A State-space Approach*. World Scientific Publishing Company, Inc, Singapore, 2006. [14](#)
- [92] J. V. B. Soares. Segmentação de vasos sanguíneos em imagens de retina usando wavelets e classificadores estatísticos. Master's thesis, Institute of Mathematics and Statistics - University of São Paulo, November 2006. [iii](#), [v](#), [7](#), [9](#), [10](#), [28](#)
- [93] J. V. B. Soares, J. J. G. Leandro, R. M. Cesar-Jr., H. F. Jelinek, and M. J. Cree. Using the 2-D morlet wavelet with supervised classification for retinal vessel segmentation. In *IV Workshop de Teses e Dissertações em Computação Gráfica e Processamento de Imagens, CD-ROM - 18th Brazilian Symposium on Computer Graphics and Image Processing*, Natal, RN, 2005. [9](#), [10](#), [28](#)
- [94] J. V. B. Soares, J. J. G. Leandro, R. M. Cesar-Jr., H. F. Jelinek, and M. J. Cree. Retinal vessel segmentation using the 2-D Gabor wavelet and supervised classification. *IEEE Transactions on Medical Imaging*, 25:1214–1222, 2006. [9](#), [10](#), [28](#), [52](#)
- [95] J. C. Sprott and G. Rowlands. Improved correlation dimension calculation. *International Journal of Bifurcation and Chaos*, 11(7):1865–1880, 2001. [19](#)
- [96] K. R. Symon. *Mechanics*. Addison-Wesley Publishing Company, Inc., Massachusetts, USA, 2nd edition, 1961. [14](#)
- [97] D. G. Tarboton. Fractal river networks, Horton's laws and Tokunaga cyclicity. *Journal of Hydrology*, 187:105–117, December 1996. [1](#)
- [98] Y. Termonia and Z. Alexandrowicz. Fractal dimension of strange attractors from radius versus size of arbitrary clusters. *Phys. Rev. Lett.*, 51(14):1265–1268, Oct 1983. [18](#)
- [99] J. Theiler. Efficient algorithm for estimating the correlation dimension from a set of discrete points. *pra*, 36:4456–4462, 1987. [17](#), [19](#), [21](#)
- [100] S. Theodoridis and K. Koutroumbas. *Pattern Recognition*. Academic Press, San Diego, USA, 2nd edition, 2003. [32](#), [34](#), [36](#), [40](#), [41](#), [42](#)
- [101] D. L. Turcotte and J. D. Pelletier. Shapes of river networks and leaves: are they statistically similar? *Philos. Trans. R. Soc. London B*, 355:307–311, 2000. [1](#)
- [102] D. L. Turcotte, J. D. Pelletier, and W. I. Newman. Networks with side branching in biology. *Journal of Theoretical Biology*, 193:577–592, 1998. [1](#)
- [103] A. R. Webb. *Statistical Pattern Recognition*. John Wiley and Sons Ltd., England, 2nd edition, 2002. [40](#), [43](#), [44](#)

- [104] D. J. Wright. Dynamical systems and fractals lecture notes. <http://www.math.okstate.edu/mathdept/dynamics/lecnotes/lecnotes.html>, May 1996. Department of Mathematics - Oklahoma State University. [17](#)
- [105] H. Zhang. The optimality of naive bayes. In *Proceedings of the Seventeenth International Florida Artificial Intelligence Research Society Conference, Miami Beach, Florida, USA - FLAIRS Conference*, Miami Beach, Florida - USA, 2004. [44](#)

Índice Remissivo

- abstract, [v](#)
- algoritmos, [60](#)
- análise de formas
 - árvores vasculares
 - conclusão, [81](#)
 - tabela de resultados, [87](#)
 - caracterização, [4, 7](#)
 - classificação, [4](#)
 - complexidade de formas, [4](#)
 - neurônios
 - conclusão, [82](#)
 - ordem de canal, [8](#)
 - segmentação, [5, 11](#)
 - validação, [9](#)
 - vetor de características, [8](#)
 - Análise Discriminante Linear, [43](#)
 - aquisição
 - camera lucida, [49](#)
 - atrator, [17](#)
 - estranho, [17](#)
 - AUC, *veja* Análise de Performance de Classificadores
 - bacias hidrográficas
 - lei de Tokunaga, [8](#)
 - ordem de canal, [8](#)
 - Bayes ingênuo, [44](#)
 - bidimensional, [48](#)
 - bifurcação, [55](#)
 - Branches Tracking Algorithm, [58, 63](#)
 - Branching Structure Contour Extraction Algorithm, [59, 66](#)
 - BSCEA, *veja* Branching Structure Contour Extraction Algorithm
 - BTA, *veja* Branches Tracking Algorithm
 - Caos, [17](#)
 - características, [8](#)
 - 2º momento estatístico, [14](#)
 - área, [9](#)
 - circularidade, [10](#)
 - combinação, [41](#)
 - curvatura, [12](#)
 - degenerescência, [9](#)
 - dimensão de correlação, [14](#)
 - entropia, [13](#)
 - lei de tokunaga, [8](#)
 - ordem de canal, [8](#)
 - perímetro, [10](#)
 - resultados, [25](#)
 - seleção, [8](#)
 - wavelet, [10](#)
 - CFT, *veja* transformada contínua de Fourier
 - Classificação
 - Análise Discriminante Linear, [43](#)
 - Bayes Ingênuo, *veja* Regra de Bayes
 - Bayes ingênuo, [44](#)
 - classificador *Bayesiano*, [38](#)
 - LDA, *veja* Análise Discriminante Linear
 - ROC, [28, 36](#)
 - supervisionada, [39](#)
 - coeficiente de correlação, [28](#)
 - contorno
 - introdução, [49](#)
 - paramétrico, [49](#)

- resultados, 71
- cruzamento, 56
- CWT, *veja* transformada contínua Wavelet
- detector de bordas, 10
- DFT, *veja* transformada discreta de Fourier
- dimensão, 14
 - base, 14
 - capacidade, 15
 - coordenadas, 14
 - dimensão de correlação, 16
 - algoritmos, 21
 - integral de correlação, 19
 - dimensão de medida natural, 18
 - dimensão métrica, 18
 - euclidiana, 14
 - fractal, 14
 - graus de liberdade, 14
 - Hausdorff-Besicovitch, 14
 - inteira, 14
 - topológica, 14
- DLA, *veja* Estruturas de Ramificação
- DSP, *veja* processamento digital de sinais
- entropia
 - mecânica estatística, 13
 - teoria da informação, 13
- escala
 - auto-similaridade, 14
 - invariância, 14
- espaço de fase, 17
- estruturas de ramificação
 - árvores vasculares de retina, 1, 7, 48, 51, 84
 - Agregação por Difusão Limitada, 1
 - agregados moleculares, 1
 - algas, 1
 - bacias geográficas, 1
 - bacias hidrográficas, 8
 - descargas elétricas atmosféricas, 1
- neurônios, 1, 47
 - plantas, 1
 - pulmões, 1
 - similaridade, 4
 - ubiqüidade, 1, 51
- f.d.p., *veja* Função Densidade de Probabilidade Fourier
- derivada, 12
- transformada, *veja* transformada de Fourier
- FPR, *veja* Taxa de Falsos Positivos
- fractal, 14
 - aleatório, 15
 - fractalidade, 15
 - regular, 15
 - retina, 15
- função risco, 37
- LDA, *veja* Análise Discriminante Linear
- linha, 55
 - ramo, 55
 - segmento, 55
- MSE, 28
- neurônio
 - axônio, 55
 - conceitos, 53
 - dendritos, 55
 - doutrina, 55
 - introdução, 47
 - linha, 55
 - ramo, 55
 - segmento, 55
 - periferia, 56
 - ponto
 - semente, 55
 - terminação, 55
 - região crítica, 55
 - bifurcação, 55
 - cruzamento, 56
 - sobreposição, 56
 - segmento
 - emergente, 55
 - incidente, 55
 - soma, 56
 - terminologia, 55
- PDR, *veja* Retinopatia Diabética Proliferativa
- periferia, 56
- perspectivas, 84
- pré-processamento, 58, 60
- probabilidade
 - a posteriori, 37
 - a priori, 38

- razão entre verossimilhanças, 38
- verossimilhança, 38
- publicações, 115
- ramo, 55
- rastreamento, 63
 - atualização, 63
 - medição, 63
 - previsão, 63
- reconhecimento de padrões, 36
- região crítica, 68
 - classificação, 65, 113
- regra de Bayes, 37
- regra de decisão, 37
 - Bayes, 38
- resumo, iii
- retinopatia diabética proliferativa, 9
- ROC, 28, 36, *veja* Análise de Performance de Classificadores
 - AUC, *veja* Análise de Performance de Classificadores
 - FPR, *veja* Taxa de Falsos Positivos
 - TPR, *veja* Taxa de Verdadeiros Positivos
- segmento, 55
- Seleção de Características
 - SBS, *veja* Seleção Seqüencial Regressiva
 - SFS, *veja* Seleção Seqüencial Progressiva
- Seleção Seqüencial Progressiva, 42
- Seleção Seqüencial Regressiva, 42
- semente, 55
- singularidade, 10
- Sistemas Dinâmicos, 17
- soma, 56
- STFT, *veja* transformada de Fourier de tempo reduzido
- tangente
 - continuidade, 64
 - orientação, 63
- teoria de decisão, 36
- teoria de detecção de sinais, 40
- terminação, 55
- teste de hipóteses, 31
- TPR, *veja* Taxa de Verdadeiros Positivos
- transformada
 - Fourier, 10
- Wavelet, 10
 - tridimensional, 48
- v.a., *veja* variável aleatória
- visão computacional, 4
- wavelet, 10
 - convolução, 12
 - gabor, 11
 - gradiente da gaussiana, 11
 - hermitiana, 11, 52
 - morlet, 11