

Universidade Estadual de Campinas
Faculdade de Engenharia Elétrica e de Computação

Karinne Saraiva da Silva

**Uma proposta imuno-inspirada para segmentação de
imagens com texturas usando transformada *wavelet packet***

Dissertação de Mestrado apresentada à Faculdade de Engenharia Elétrica e de Computação como parte dos requisitos para obtenção do título de Mestre em Engenharia Elétrica. Área de concentração: Telecomunicações e Telemática.

Orientador: Yuzo Iano

Campinas, SP
2010

FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DA ÁREA DE ENGENHARIA E ARQUITETURA - BAE - UNICAMP

Si38p Silva, Karinne Saraiva
Uma proposta imuno-inspirada para segmentação de
imagens com texturas usando transformada wavelet
packet / Karinne Saraiva da Silva. --Campinas, SP: [s.n.],
2010.

Orientador: Yuzo Iano.
Dissertação de Mestrado - Universidade Estadual de
Campinas, Faculdade de Engenharia Elétrica e de
Computação.

1. Processamento de imagens - Técnicas digitais. 2.
Análise de textura. 3. Wavelets (Matemática). 4.
Métodos bio-inspirados. I. Iano, Yuzo. II. Universidade
Estadual de Campinas. Faculdade de Engenharia Elétrica
e de Computação. III. Título.

Título em Inglês: An immune-inspired proposal for textured image segmentation using
wavelet packet transform

Palavras-chave em Inglês: Digital image processing, Texture analysis, Wavelets
(Mathematics), Bio-inspired methods

Área de concentração: Telecomunicações e Telemática

Titulação: Mestre em Engenharia Elétrica

Banca examinadora: Luiz César Martini, Nelson Delfino d'Ávila Mascarenhas

Data da defesa: 14/04/2010

Programa de Pós Graduação: Faculdade de Engenharia Elétrica

COMISSÃO JULGADORA - TESE DE MESTRADO

Candidata: Karinne Saraiva da Silva

Data da Defesa: 14 de abril de 2010

Título da Tese: "Uma Proposta Imuno-inspirada para Segmentação de Imagens com Texturas usando Transformada "Wavelet Packet""

Prof. Dr. Yuzo Iano (Presidente): _____

Prof. Dr. Nelson Delfino d'Ávila Mascarenhas: _____

Prof. Dr. Luiz César Martini: _____

Yuzo Iano

Nelson Delfino d'Ávila Mascarenhas

Luiz César Martini

Resumo

Segmentação de texturas é um ponto crucial em muitas aplicações da área de visão computacional e processamento digital de imagens. Muitas são as aplicações que utilizam imagens com texturas, como: sensoriamento remoto, análise de imagens médicas, inspeção industrial, etc.

Para análise de texturas, é essencial o uso de um extrator de características capaz de representar bem cada textura presente na imagem. A transformada *wavelet packet* fornece a caracterização necessária para discriminação de texturas, oferecendo também uma representação multi-escala, ferramenta muito importante na análise de texturas.

Outro ponto importante neste trabalho, é o fato da metodologia aqui proposta ser não supervisionada. Para tal, é utilizado o algoritmo de clusterização ARIA, que determina automaticamente o número de *clusters* presentes no conjunto de dados.

A eficiência do método desenvolvido é comprovada aplicando-o em diversas imagens, como: mosaicos de Brodatz, imagens naturais, imagens médicas e outras aplicações.

Palavras-chave: análise de texturas, discriminação de texturas, transformada *wavelet packet*, ARIA.

Abstract

Texture segmentation is a crucial aspect in many computer vision and digital image processing applications. Several of these applications use texture images, such as remote sensing, medical image analysis, industrial inspection, etc.

For texture analysis, it is essential to use a feature-extractor that can represent precisely each of the textures present in the picture. The wavelet packet transform provides the characteristics required for discrimination of the textures, as well as offering a multi-scale representation, which is a very important tool in texture analysis.

Another important aspect in this work is that the proposed methodology is unsupervised. To achieve that, the clustering algorithm ARIA is used, which automatically determines the number of clusters present in the data set.

The efficiency of the developed method is clear in the application of the method on several types of images, such as mosaics of Brodatz, natural images, medical images and other applications.

Keywords: texture analysis, texture discrimination, wavelet packet transform, ARIA.

Agradecimentos

Primeiramente, a Deus, que sempre coloca as pessoas certas, nos momentos certos, no nosso caminho.

Ao meu orientador, Prof. Yuzo Iano, sou grata pela orientação durante esses dois anos.

À toda minha família, especialmente minha mãe, meu pai e minha irmã, pelo apoio durante esta jornada.

Ao meu namorado Fellipe, que sempre me apoiou e foi meu alicerce durante toda minha estadia em Campinas. E à toda sua família que me adotou como filha.

Aos colegas do Laboratório de Comunicações Visuais, pela amizade, companheirismo e sugestões para este trabalho.

Aos professores e funcionários do Programa de Pós Graduação em Engenharia Elétrica da Universidade Estadual de Campinas, por sua ajuda sempre necessária.

À CAPES, pelo apoio financeiro.

A todos que, direta ou indiretamente, colaboraram para a realização deste trabalho.

Um agradecimento especial ao Programa CAPES de Formação de Recursos Humanos em Televisão Digital (RH-TVD) pelo apoio financeiro fundamental a todos no Laboratório de Comunicações Visuais (LCV).

Dedico este trabalho à minha família, meu namorado e aos amigos de verdade que me ajudaram nessa jornada.

Sumário

Lista de Figuras	xv
Lista de Tabelas	xix
Glossário	xxi
Trabalhos Publicados Pelo Autor	xxiii
1 Introdução	1
1.1 Considerações Gerais	1
1.2 Objetivos e Contribuições	2
1.3 Organização do Trabalho	3
2 Análise Textural	5
2.1 Métodos para Análise de Textura	7
2.1.1 Técnicas de Extração de Características	8
2.2 Aplicações da Análise de Texturas	14
2.3 Comentários	16
3 Transformada <i>Wavelet</i> e Transformada <i>Wavelet Packet</i>	17
3.1 Introdução	17
3.2 Transformada <i>Wavelet</i> Contínua	18
3.3 Transformada <i>Wavelet</i> Discreta	20
3.4 Análise Multiresolução	21
3.5 Transformada <i>Wavelet</i> Bidimensional	26
3.6 Transformada <i>Wavelet Packet</i>	30
3.6.1 Transformada <i>Wavelet Packet</i> 2D	31
3.6.2 Algoritmo <i>Best Basis</i>	34
3.7 Comentários	37

4	Reconhecimento de Padrões	39
4.1	Extração de Características	39
4.2	Seleção de Características	41
4.2.1	Análise de Componentes Principais	41
4.3	<i>Clusterização</i>	46
4.3.1	O Sistema Imunológico	48
4.3.2	O Algoritmo	48
4.4	Comentários	52
5	Metodologia Proposta	53
5.1	Transformada <i>Wavelet Packet</i>	54
5.2	Características	57
5.3	<i>Minimal Spanning Tree</i>	58
5.4	Finalização	60
5.5	Comentários	61
6	Resultados	63
6.1	Trabalhos Anteriores	63
6.2	Segmentação de Texturas de Brodatz	65
6.2.1	Imagens com 2 Texturas	65
6.2.2	Imagens com mais de 2 Texturas	73
6.3	Imagens Naturais	75
6.4	Imagens Médicas	80
6.5	Outras Aplicações	83
6.6	Comentários	85
7	Considerações Finais	87
7.1	Comentários Sobre as Simulações e Resultados	89
7.2	Trabalhos Futuros	91
	Referências Bibliográficas	93

Lista de Figuras

2.1	Exemplos de texturas. (a) D2. (b) D23. (c) D56. (d) D62. (e) D81. (f) D94.	7
2.2	Imagem <i>Trees</i>	9
2.3	Histograma da imagem <i>Trees</i>	10
2.4	Exemplos de filtros de Gabor.	13
2.5	Exemplo de sensoriamento remoto.	15
3.1	Espaço vetorial W_{2^j}	24
3.2	Esquema da decomposição <i>wavelet</i> 1D.	25
3.3	Decomposição <i>wavelet</i> bidimensional na forma piramidal.	28
3.4	Reconstrução <i>wavelet</i> bidimensional.	29
3.5	Imagem formada pela textura D56 do álbum de Brodatz.	29
3.6	Decomposição <i>wavelet</i> da Fig. 3.5.	30
3.7	Esquema da transformada <i>wavelet</i> discreta.	32
3.8	Esquema da transformada <i>wavelet packet</i>	32
3.9	Esquema da transformada <i>wavelet packet</i> 2D.	33
3.10	Transformada <i>wavelet packet</i> 2D na forma piramidal.	33
3.11	Transformada <i>wavelet packet</i> 2D da Fig. 3.5.	34
3.12	Representação de uma base <i>wavelet packet</i>	35
3.13	Representação de nó da árvore da TWP.	36
3.14	Representação da árvore da TWP podada.	37
4.1	Representação das etapas do Reconhecimento de Padrões.	40
4.2	Esquema simplificado da redução da dimensionalidade das características.	44
4.3	(a) Objeto binário. (b) Eixos principais (auto-vetores). (c) Objeto rotacionado através do uso de (4.17).	44
4.4	Transformada <i>wavelet packet</i> da imagem.	45
4.5	Componentes principais da decomposição por transformada <i>wavelet packet</i>	46
4.6	Exemplo do uso do critério escolhido para a eliminação das arestas da MST. Ao examinar a aresta \overline{CD} , tem-se que a menor de suas arestas vizinhas, \overline{AC} , \overline{BC} , \overline{DE} e \overline{DF} , tem comprimento 1,39. Para $n=2$, \overline{CD} deve ser removida. Obtém-se então dois <i>clusters</i> : (A, B, C) e (D, E, F).	52

5.1	Imagem exemplo.	53
5.2	Fluxograma da metodologia proposta	54
5.3	<i>Wavelet packet Tree</i>	55
5.4	<i>Best tree</i>	56
5.5	Decomposição <i>wavelet packet</i> de nível 3.	57
5.6	<i>Best Tree</i> da decomposição da Fig. 5.5.	57
5.7	(a) Primeira iteração. (b) Iteração 5. (c) Convergência.	59
5.8	(a) <i>Minimal spanning tree</i> . (b) <i>Minimal spanning tree</i> após eliminação de uma aresta.	60
5.9	Segmentação obtida após rotulamento.	60
5.10	Filtragem da segmentação obtida na Fig. 5.9.	61
5.11	(a) Segmentação obtida na 2ª escala de decomposição. (b) Segmentação obtida na 1ª escala de decomposição. (a) Segmentação obtida a partir da imagem original.	61
6.1	(a) Imagem original. (b) Resultado obtido pela metodologia proposta. (c) Resultado obtido pela metodologia proposta sobreposto à imagem original. (d) Resultado obtido em [1] (contorno desejado em branco). (e) Resultado obtido em [2] (colorido) sobreposto ao contorno desejado (branco). (f) Imagem de referência.	65
6.2	(a) Imagem original. (b) Resultado obtido pela metodologia proposta. (c) Resultado obtido pela metodologia proposta sobreposto à imagem original. (d) Resultado obtido em [3]. (e) Imagem de referência.	66
6.3	(a) Imagem original. (b) Resultado obtido pela metodologia proposta. (c) Resultado obtido pela metodologia proposta sobreposto à imagem original. (d) Resultado obtido em [3]. (e) Imagem de referência.	67
6.4	Imagem de referência para as Figs. 6.5.(a), 6.6.(a), 6.7.(a)	68
6.5	(a) Imagem original. (b) Resultado obtido pela metodologia proposta. (c) Resultado obtido pela metodologia proposta sobreposto à imagem original. (d) Resultado obtido em [4].	70
6.6	(a) Imagem original. (b) Resultado obtido pela metodologia proposta. (c) Resultado obtido pela metodologia proposta sobreposto à imagem original. (d) Resultado obtido em [4].	71
6.7	(a) Imagem original. (b) Resultado obtido pela metodologia proposta. (c) Resultado obtido pela metodologia proposta sobreposto à imagem original. (d) Resultado obtido em [4].	72
6.8	(a) Imagem original. (b) Resultado obtido pela metodologia proposta. (c) Resultado obtido pela metodologia proposta sobreposto à imagem original. (d) Resultado obtido em [5]. (e) Resultado obtido em [2]. (f) Imagem de referência.	73

6.9	(a) Imagem original. (b) Resultado obtido pela metodologia proposta. (c) Resultado obtido pela metodologia proposta sobreposto à imagem original. (d) Imagem de referência.	74
6.10	(a) Imagem original. (b) Resultado obtido pela metodologia proposta. (c) Resultado obtido pela metodologia proposta sobreposto à imagem original. (d) Imagem de referência.	75
6.11	(a) Imagem original. (b) Resultado obtido pela metodologia proposta. (c) Resultado obtido pela metodologia proposta sobreposto à imagem original. (d) Resultado obtido em [3]. (e) Imagem de referência.	76
6.12	(a) Imagem original. (b) Resultado obtido pela metodologia proposta. (c) Resultado obtido pela metodologia proposta sobreposto à imagem original. (d) Resultado obtido em [3]. (e) Imagem de referência.	77
6.13	(a) Imagem original. (b) Resultado obtido pela metodologia proposta. (c) Resultado obtido pela metodologia proposta sobreposto à imagem original. (d) Imagem de referência.	78
6.14	(a) Imagem original. (b) Resultado obtido pela metodologia proposta. (c) Resultado obtido pela metodologia proposta sobreposto à imagem original. (d) Imagem de referência.	79
6.15	(a) Imagem original. (b) Resultado obtido pela metodologia proposta. (c) Resultado obtido pela metodologia proposta sobreposto à imagem original. (d) Imagem de referência.	81
6.16	(a) Imagem original. (b) Resultado obtido pela metodologia proposta. (c) Resultado obtido pela metodologia proposta sobreposto à imagem original.	82
6.17	Imagem Envisat's ASAR adquirida em 17 de novembro 2002.	84
6.18	(a) Imagem original. (b) Resultado obtido pela metodologia proposta. (c) Resultado obtido pela metodologia proposta sobreposto à imagem original. (d) Imagem de referência.	85
6.19	(a) Imagem original. (b) Resultado obtido pela metodologia proposta. (c) Resultado obtido pela metodologia proposta sobreposto à imagem original. (d) Imagem de referência.	85

Lista de Tabelas

3.1	Coeficientes da Transformada <i>Wavelet</i>	26
4.1	Exemplo de aplicações de reconhecimento de padrões e métodos de extração de características.	40
4.2	Autovalores da decomposição por transformada <i>wavelet packet</i> da Fig. 4.4.	46
4.3	Descrição dos símbolos usados no pseudo-código.	50
5.1	Quantidade de subimagens geradas pela decomposição <i>wavelet packet</i>	55
6.1	Percentual de erro para imagens com 2 texturas de Brodatz.	68
6.2	Porcentagem de erro para diversos métodos de segmentação.	69
6.3	Percentual de erro para imagens com mais de 2 texturas de Brodatz.	74
6.4	Percentual de erro para imagens naturais.	78
6.5	Percentual de erro para imagens médicas.	80
6.6	Percentual de erro para outras aplicações.	84
7.1	Dimensão das imagens utilizadas nos testes.	90

Glossário

ARIA *Adaptive Radius Immune Algorithm for Data Clustering*

CVE Coeficiente de Variação Espacial

ICA *Independent Component Analysis*

MST *Minimum Spanning Tree*

TF Transformada de Fourier

TFJ Transformada de Fourier Janelada

TW Transformada *Wavelet*

TWC Transformada *Wavelet* Contínua

TWD Transformada *Wavelet* Discreta

TWP Transformada *Wavelet Packet*

Trabalhos Publicados Pelo Autor

1. Karinne S. Silva, Yuzo Iano. “An Immune-Inspired Approach for Unsupervised Texture Segmentation using Wavelet Packet Transform”. *22th Brazilian Symposium on Computer Graphics and Image Processing (Sibgrapi 2009)*, Rio de Janeiro, Rio de Janeiro, Brasil, Outubro 2009.
2. Karinne S. Silva, Yuzo Iano. “Segmentação de Texturas Monocromáticas usando Transformada Wavelet Packet”. *XXVII Simpósio Brasileiro de Telecomunicações (SBrT 2009)*, Blumenau, Santa Catarina, Brasil, Setembro 2009.

Capítulo 1

Introdução

1.1 Considerações Gerais

Observando os recentes avanços da tecnologia digital, como impressoras, *scanners*, câmeras fotográficas, aparelhos de TV e conversores para a tecnologia de TV digital, percebe-se que, cada vez mais, as imagens analógicas dão lugar às imagens digitais. Isso exige o desenvolvimento de métodos mais eficientes para processamento de imagens e visão computacional.

Várias são as características de uma imagem que são utilizadas na execução dos algoritmos de processamento de imagens digitais. Dentre elas estão: valor dos *pixels*, histograma, luminosidade e **textura**. Para os sentidos visuais humanos, é relativamente simples diferenciar duas texturas, entretanto, para que o computador possa executar esta mesma tarefa são necessários algoritmos de processamento digital de imagens.

Em geral, a primeira etapa desses algoritmos é a segmentação, que pode ser definida, de forma simplificada, como o processo que efetua a distinção entre o que é considerado objeto e o fundo da imagem. Desse modo, se configura como uma fase essencial para as mais diversas aplicações envolvendo imagens digitais.

Apesar de diversas técnicas terem sido utilizadas para segmentação de texturas, os métodos baseados em transformada *wavelet* têm ganho muito destaque entre os pesquisadores por apresentarem uma melhor representação das características texturais.

Neste trabalho, optou-se por não utilizar a transformada *wavelet* tradicional, e sim a transformada *wavelet packet*, que pode ser considerada uma generalização da transfor-

mada *wavelet*. Tem-se como vantagem a peculiaridade de decompor a imagens em médias frequências. Em geral, é nessa faixa de frequência que são encontradas muitas das características texturais. Fornece assim, uma melhor análise para posterior segmentação.

Entretanto, sabe-se que para efetuar a segmentação de uma imagem, além da extração das características, é necessário também que haja um processo capaz de distinguir quais dessas características pertencem ao objeto e quais são inerentes ao fundo da imagem. Para tal, são utilizados algoritmos de clusterização.

A opção aqui escolhida para realizar tal tarefa é um algoritmo imuno-inspirado denominado ARIA (*Adaptive Radius Immune Algorithm for Data Clustering*). Estes têm como principais aplicações a otimização de soluções e a clusterização. Eles fazem parte de um grupo de algoritmos inspirados na natureza, assim como os algoritmos genéticos e as redes neurais.

Os algoritmos imuno-inspirados têm como base o sistema imunológico humano. Eles realizam suas funções tratando os dados como antígenos (invasores do corpo humano, como vírus e bactérias) e os protótipos dos *clusters* como as células de defesa do corpo, os anticorpos. Apesar de existirem inúmeros algoritmos de clusterização, o que diferencia os algoritmos imuno-inspirados é que eles são capazes de determinar automaticamente o número de *clusters* presentes nos dados.

A metodologia aqui proposta tem como base as duas ferramentas acima descritas: a transformada *wavelet packet* e o ARIA. A junção delas permite que seja realizada a segmentação de diversos tipos de imagens, tendo como base a informação textural nelas contida. Possibilita assim, um ponto de partida para os demais processos utilizados em imagens digitais.

1.2 Objetivos e Contribuições

Em áreas como o sensoriamento remoto, a medicina, a indústria e a botânica, é possível encontrar problemas que podem ser resolvidos com processamentos de imagens digitais.

Em muitas dessas aplicações, o atributo **textura** é o mais relevante, pois os objetos de estudo dessas áreas, como manchas de óleo, nuvens, órgãos, tecidos, folhas e animais têm texturas específicas que os caracterizam. Dessa forma é possível, utilizando as ferramentas adequadas, separar o objeto de interesse do fundo da imagem.

Apesar de a segmentação ser apenas a primeira etapa de um algoritmo de processamento digital de imagens, ela é fundamental para que as outras etapas sejam executadas com sucesso. Assim, permite que técnicas desejadas sejam aplicadas apenas na região de interesse.

O objetivo deste trabalho é desenvolver uma metodologia para segmentação de imagens com informações texturais através de uma análise multiresolução utilizando a transformada *wavelet packet*, que fornece um maior detalhamento da imagem, pois a decomposição em baixas, médias e altas frequências.

Ainda, foi utilizado ao final um algoritmo imuno-inspirado para realizar a clusterização das características obtidas pela transformada *wavelet packet*. Esse algoritmo foi escolhido pela capacidade de obter automaticamente, sem interferência direta do usuário, o número de *clusters* presentes nos dados.

Por fim, foram realizados diversos testes em imagens de diferentes áreas para validação do método aqui apresentado.

Desse modo, as principais contribuições deste trabalho estão aqui resumidas:

- a- Revisão bibliográfica sobre análise textural e extração de características de textura;
- b- Aplicação de um algoritmo imuno-inspirado na clusterização de características de imagens de textura;
- c- Desenvolvimento de uma nova metodologia para segmentação não supervisionada de imagens com informação textural. Isso foi feito sem a necessidade do usuário para determinar, a priori, o número de texturas presentes na imagem, tornando o processo automatizado e adaptável à vários tipos de imagens, podendo ser aplicadas nas mais diversas áreas do conhecimento.

1.3 Organização do Trabalho

Capítulo 1 - Introdução: Neste capítulo é feita a apresentação do tema descrito neste trabalho, além dos objetivos, motivação e contribuições.

Capítulo 2 - Análise Textural: Aqui são introduzidos os conceitos básicos sobre texturas, análise textural e métodos utilizados na extração de características de textura, bem como são apresentadas algumas aplicações que usam como ferramenta a análise

textural, como: análise de imagens médicas, sensoriamento remoto, inspeção industrial e biologia.

Capítulo 3 - Transformada *Wavelet* e Transformada *Wavelet Packet*: Esse capítulo destina-se a mostrar a teoria das transformadas *wavelet* e *wavelet packet*. Primeiramente, é realizado um breve estudo sobre as transformadas *wavelet* contínua e discreta e sua relação com análise multiresolução. Em seguida, são introduzidos os conceitos da transformada *wavelet packet*, utilizada neste trabalho, e do algoritmo *Best Basis*.

Capítulo 4 - Reconhecimento de Padrões: Concluindo a base teórica deste trabalho, esse capítulo apresenta conceitos da teoria de reconhecimento de padrões, como extração de características, seleção de características, e clusterização.

Capítulo 5 - Metodologia Proposta: Esse capítulo apresenta a junção de toda a base teórica anteriormente apresentada. Cada etapa da metodologia proposta é detalhada, utilizando diagramas de blocos e diversas imagens que demonstram, passo-a-passo, como a segmentação é obtida.

Capítulo 6 - Resultados: Nesse capítulo são apresentados os resultados obtidos pela metodologia proposta em imagens de várias aplicações da análise textural, incluindo imagens sintéticas, localização de animais, mamografias, melanomas, sensoriamento remoto e tecidos. Ainda, são realizadas diversas comparações com resultados obtidos e àqueles disponíveis na literatura.

Capítulo 7 - Considerações Finais: Aqui são expostas as conclusões obtidas a partir da análise dos resultados apresentados no capítulo 6 e as perspectivas para melhorias da metodologia apresentada.

Capítulo 2

Análise Textural

Não há nenhuma definição estrita para textura, entretanto essa propriedade rica em informação visual é facilmente percebida pelos seres humanos. Abaixo seguem alguns exemplos de definições de textura encontradas na literatura:

- "Podemos definir textura como aquilo que constitui uma região macroscópica. Sua estrutura é simplesmente atribuída aos padrões repetitivos em que os elementos ou primitivas são arranjados de acordo com uma "regra de organização". "[6].

- "Uma região em uma imagem tem uma textura constante se um conjunto de estatísticas locais ou outras propriedades locais da função da imagem são constantes, variam lentamente ou são aproximadamente periódicas." [7].

- "A imagem de textura que consideramos é não figurativa e celular... Uma imagem de textura é descrita pelo número e tipos das suas primitivas e da organização espacial ou leiaute de suas primitivas... Uma característica fundamental da textura: não pode ser analisada sem um quadro de referência da primitiva tonal seja explícita ou implícita. Para toda superfície lisa em tons de cinza, existe uma escala tal que quando a superfície é examinada, ela não tem textura. Então, quando a resolução cresce, ela ganha uma textura suave e então uma textura mais grosseira." [8].

- "Textura é, aparentemente, um conceito paradoxal. Por um lado, ela é comumente usada no pré-processamento da informação visual, especialmente para propósitos de classificação prática. Por outro lado, ninguém conseguiu produzir uma definição de textura aceita por todos. A resolução deste paradoxo, sentimos, dependerá de um modelo rico, mais desenvolvido para o processamento da informação visual, um aspecto central do

que serão os sistemas de representação nos diferentes níveis de abstração. Estes níveis provavelmente incluirão uma intensidade real na parte inferior e irá progredir através das bordas e descritores de orientação para superfície, e talvez descritores volumétricos. Dadas estas estruturas multi-níveis, parece claro que elas devem ser incluídas na definição e na computação de descritores de textura."[9].

-"Textura é definida, para os nossos propósitos como atributos de um campo que não tem componentes que aparecem enumeráveis. A relação de fase entre os componentes não é, então, aparente. Ou o campo não deve conter um gradiente óbvio. O objetivo desta definição é direcionar a atenção do observador para as propriedades globais do *display*, em geral, aspereza, profundidade, ou suavidade. Fisicamente, padrões não-numeráveis (aperiódicos) são gerados por processos estocásticos em oposição aos determinísticos. Perceptivelmente, entretanto, o conjunto de todos os padrões sem componentes numeráveis óbvios incluirá muitas texturas determinísticas (e também periódicas)." [10].

Podemos verificar, através dessa coleção de definições, que não há consenso entre os pesquisadores, e sim uma influência da aplicação de cada autor. Neste trabalho, faremos uso da seguinte definição: "texturas são padrões complexos compostos por subpadrões, que têm características de brilho, cor, direção, tamanho, etc. As propriedades dos subpadrões geram várias percepções, como luminosidade, uniformidade, densidade, aspereza, granularidade, linearidade, frequência, fase, direcionalidade, randomicidade, suavidade, etc, da textura como um todo"[11].

Para dar uma idéia visual do que é denominado como textura, pode-se visualizar na Fig. 2.1 alguns exemplos de texturas retiradas do album de Brodatz [12].

Pode-se dividir a análise textural em várias etapas, cada uma delas possuindo sua função específica. Materka e Strzelecki [13] dividiram-na em quatro tarefas principais:

1. Extração de características: retirar dados numéricos de uma imagem digital de forma a representar as propriedades da textura;
2. Discriminação de texturas: separar uma imagem de texturas em regiões homogêneas, correspondente à segmentação de imagens;
3. Classificação de texturas: determinar a que classe pertence uma determinada textura, dentro de um número de classes pré-definida;

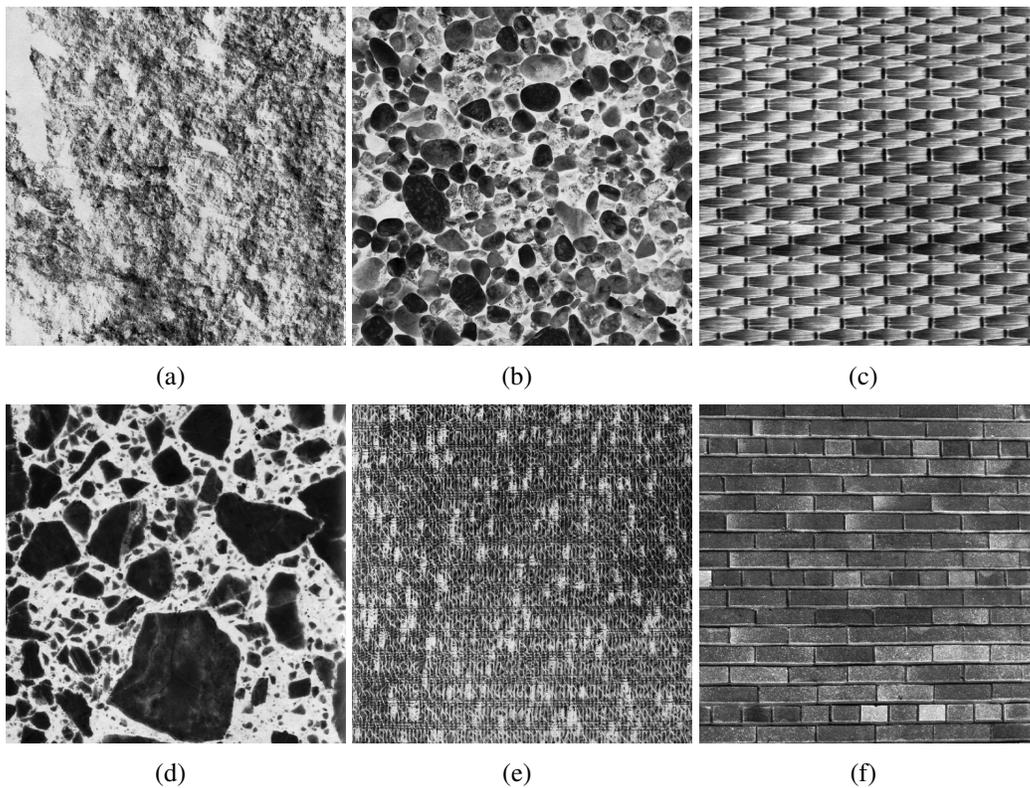


Fig. 2.1: Exemplos de texturas. (a) D2. (b) D23. (c) D56. (d) D62. (e) D81. (f) D94.

4. Reconstrução de texturas: reconstruir uma geometria de superfície 3D da informação textural.

No primeiro estágio da análise de texturas, extração de características, são obtidas as informações necessárias para realizar os outros processos, como segmentação, classificação ou determinação de forma de objetos. Este trabalho está restrito aos dois primeiros passos descritos acima, tendo como objetivo segmentar texturas. Desse modo, neste capítulo, serão apresentadas algumas técnicas utilizadas para a realização do primeiro passo.

2.1 Métodos para Análise de Textura

Usualmente, os métodos para análise de textura são divididos em quatro grupos [13]:

Estruturais: esses métodos representam uma textura como uma composição de texturas primitivas dispostas de acordo com algumas regras sintáticas. A vantagem dessas

abordagens é que elas permitem uma boa descrição simbólica da imagem, entretanto, esses métodos são mais úteis para sínteses do que para análise de imagens. Um exemplo de ferramenta para análise estrutural é a morfologia matemática [14].

Estatísticos: representam a textura indiretamente por propriedades não determinísticas que regem a distribuição e as relações entre os níveis de cinza da imagem. Métodos baseados em estatísticas de segunda ordem têm se mostrado com alto poder de discriminação. Imagens de textura em tons de cinza são discriminadas espontaneamente somente se elas diferem no seu momento de segunda ordem. Momentos de segunda ordem iguais, mas diferentes momentos de terceira ordem requerem esforço cognitivo mais alto. Exemplos desse tipo de método são a análise do histograma [15], matriz de co-ocorrência [16] e o espectro de textura.

Modelados: realiza a análise utilizando modelos matemáticos sofisticados (fractais [17] e estocásticos [18]). Os parâmetros do modelo são usados para análise da imagem. Métodos estocásticos têm alto custo computacional. A modelagem fractal, por sua vez, tem se mostrado útil em texturas naturais, porém a falta de orientação seletiva pode ser um empecilho em determinadas aplicações.

Transformadas: esses métodos têm sido muito utilizados na análise textural, pois representam a imagem em um novo espaço, como por exemplo frequência e escala, onde as características texturais se tornam mais acessíveis. Exemplos de transformadas utilizadas são as de Fourier, Gabor [19] e *wavelet* [20]. Métodos baseados na transformada de Fourier não oferecem bons resultados na prática pela falta de informação espacial. Os filtros de Gabor solucionam esse problema, entretanto não existem filtros de localização simples que possam localizar uma estrutura espacial em texturas naturais. A transformada *wavelet* possui algumas vantagens, como a variação da resolução espacial, que permite a representação da textura na melhor escala e um grande conjunto de funções *wavelets*, podendo assim, ser escolhida a mais adequada para cada aplicação

2.1.1 Técnicas de Extração de Características

Dentre as metodologias citadas na seção anterior, algumas das técnicas mais empregadas atualmente estão descritas a seguir.

Histograma

Assumindo que uma imagem é uma função $f(x, y)$ de duas variáveis, $x = 0, 1, \dots, N - 1$ e $y = 0, 1, \dots, M - 1$, onde N e M são as dimensões da imagem, então $f(x, y)$ pode assumir os valores $i = 0, 1, \dots, G - 1$, onde G é o número total de tons de cinza na imagem. O histograma é definido como sendo o número de *pixels* na imagem com um determinado valor de tom de cinza:

$$h(i) = \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} \delta(f(x, y), i), \quad (2.1)$$

onde $\delta(j, i)$ é a função de Kronecker:

$$\delta(j, i) = \begin{cases} 1, & j = i; \\ 0, & j \neq i; \end{cases} \quad (2.2)$$

Pode-se observar que o cálculo do histograma utiliza apenas o valor de cinza dos *pixels* isoladamente, nos dando uma informação estatística de primeira ordem. Dividindo $h(i)$ pelo número de *pixels* na imagem, $N \times M$, obtém-se a densidade de probabilidade dos níveis de cinza da imagem, como em (2.3). A Fig. 2.2 mostra a imagem *Trees* em tons de cinza, e a Fig. 2.3 o histograma correspondente.



Fig. 2.2: Imagem *Trees*

$$p(i) = h(i)/(NxM), \quad i = 0, 1, \dots, G - 1 \quad (2.3)$$

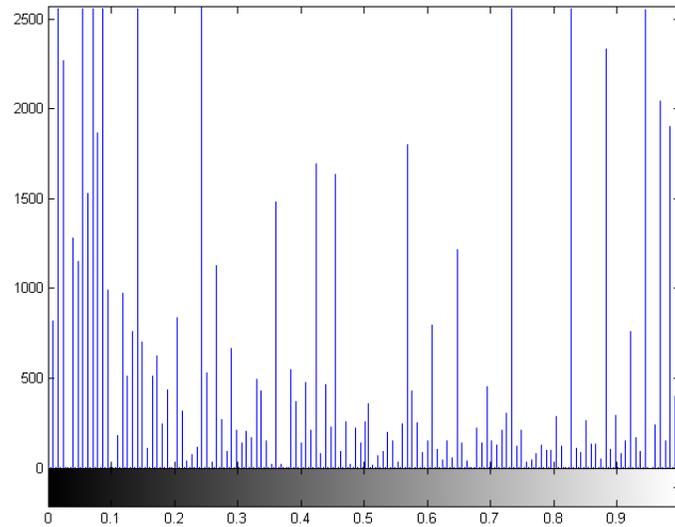


Fig. 2.3: Histograma da imagem *Trees*.

Observando o exemplo de histograma dado na Fig. 2.3, pode-se concluir que ele pode nos oferecer algumas dicas sobre a imagem, por exemplo um histograma estreito sugere uma imagem com baixo contraste. Diferentes características de primeira ordem da imagem podem ser obtidas a partir de seu histograma, como média, variância, assimetria e curtose, definidos em (2.4-2.7).

Média:

$$\mu = \sum_{i=0}^{G-1} ip(i) \quad (2.4)$$

Variância:

$$\sigma^2 = \sum_{i=0}^{G-1} (i - \mu)^2 p(i) \quad (2.5)$$

Assimetria:

$$\mu_3 = \sigma^{-3} \sum_{i=0}^{G-1} (i - \mu)^3 p(i) \quad (2.6)$$

Curtose:

$$\mu_4 = \sigma^{-4} \sum_{i=0}^{G-1} (i - \mu)^4 p(i) - 3 \quad (2.7)$$

Também são utilizados outros dois parâmetros, energia e entropia, descritos matematicamente em (2.8) e (2.9), respectivamente.

$$E = \sum_{i=0}^{G-1} p(i)^2 \quad (2.8)$$

$$H = - \sum_{i=0}^{G-1} p(i) \log_2[p(i)] \quad (2.9)$$

A média nos fornece o valor médio dos níveis de cinza da imagem, enquanto que a variância descreve a variação do tom de cinza em torno da média. A assimetria tem valor zero se o histograma é simétrico em relação à média, podendo também assumir valores positivos ou negativos dependendo da posição dos valores em torno da média. Curtose é uma medida de dispersão que caracteriza o "achatamento" da curva em relação à distribuição normal. A entropia mede a uniformidade do histograma, Outras possibilidades de características derivadas do histograma são: mínimo, máximo, intervalo e mediana.

Os valores de média e variância podem não conter informações de textura, pois são influenciados por fatores como iluminação, sombras, etc. Nesses casos pode-se utilizar o processo de normalização dos valores, utilizando-se frequentemente $\mu = 0$ e $\sigma = 1$.

Matriz de co-ocorrência

Apesar de sua simplicidade, as características de primeira ordem não são capazes de discriminar texturas completamente. Assim sendo, os métodos estatísticos mais utilizados para análise de texturas são de segunda ordem.

O histograma de segunda ordem é definido como matriz de co-ocorrência $h_{d\theta}(i, j)$. Quando é dividida pelo número total de *pixels* vizinhos $R(d, \theta)$ na imagem, essa matriz passa a estimar a probabilidade conjunta $p_{d\theta}(i, j)$ de dois *pixels* distantes de d a um ângulo θ nos pontos de coordenada com valores i e j .

Isso leva à uma matriz quadrada de dimensão igual ao número de tons de cinza na imagem, para cada distância d e ângulo θ . Dado o esforço computacional envolvido, em geral, apenas as distâncias $d = 1, 2 \text{ pixels}$ e os ângulos $\theta = 0, 45, 90, 135$ são considerados.

A classificação de texturas pequenas requer valores pequenos de d , enquanto texturas maiores exigem altos valores para d . A redução do número de tons de cinza da imagem diminui o custo computacional, mas resulta em perda de informação textural. A matriz a seguir é a matriz de co-ocorrência da imagem *Trees* na Fig. 2.2 calculada para oito tons de cinza.

$$\begin{bmatrix} 18392 & 4758 & 191 & 43 & 20 & 7 & 4 & 0 \\ 4601 & 11001 & 1884 & 263 & 95 & 60 & 34 & 0 \\ 245 & 1761 & 4537 & 1648 & 228 & 81 & 57 & 9 \\ 74 & 253 & 1600 & 4846 & 1035 & 207 & 95 & 33 \\ 45 & 66 & 240 & 1087 & 2444 & 1251 & 197 & 36 \\ 25 & 51 & 79 & 165 & 1328 & 3651 & 1086 & 70 \\ 11 & 40 & 55 & 84 & 203 & 1138 & 5057 & 1290 \\ 0 & 6 & 9 & 19 & 31 & 87 & 1336 & 10793 \end{bmatrix}$$

Características como energia, correlação, contraste, valor absoluto, diferença inversa e entropia podem ser calculadas a partir da matriz de co-ocorrência. Informações mais detalhadas podem ser encontradas em [21].

Filtros de Gabor

Apesar dos filtros de Gabor serem criticados pela falta de ortogonalidade, resultando em informações redundantes nas diferentes escalas, eles são muito utilizados tanto na segmentação [22], [19], [23] como na classificação [24], [25] de texturas.

Filtros de Gabor são, em sua essência, funções complexas seno e cosseno orientadas com uma determinada frequência W e em orientação θ moduladas por uma gaussiana. Sua forma bidimensional é dada por (2.10) no domínio do espaço, e (2.11) no domínio da frequência.

$$g(x, y) = \left(\frac{1}{2\pi\sigma_x\sigma_y} \right) e^{-\frac{1}{2}\left(\frac{x^2}{\sigma_x^2} + \frac{y^2}{\sigma_y^2}\right) + 2\pi j W x} \quad (2.10)$$

$$G(u, v) = e^{-\frac{1}{2}\left[\frac{(u-W)^2}{\sigma_u^2} + \frac{v^2}{\sigma_v^2}\right]} \quad (2.11)$$

onde $\sigma_u = 1/2\pi\sigma_x$, $\sigma_v = 1/2\pi\sigma_y$ e W são determinados empiricamente com valores no intervalo $[0.01, 0.4]$. A Fig. 2.4 mostra exemplos de filtros de Gabor no domínio espacial.

Os parâmetros a serem determinados para a geração de um filtro de Gabor são: a frequência W e a orientação θ , possibilitando a geração de inúmeros filtros diferentes. As funções de Gabor formam um conjunto de filtros completo, porém não-ortogonal, o que

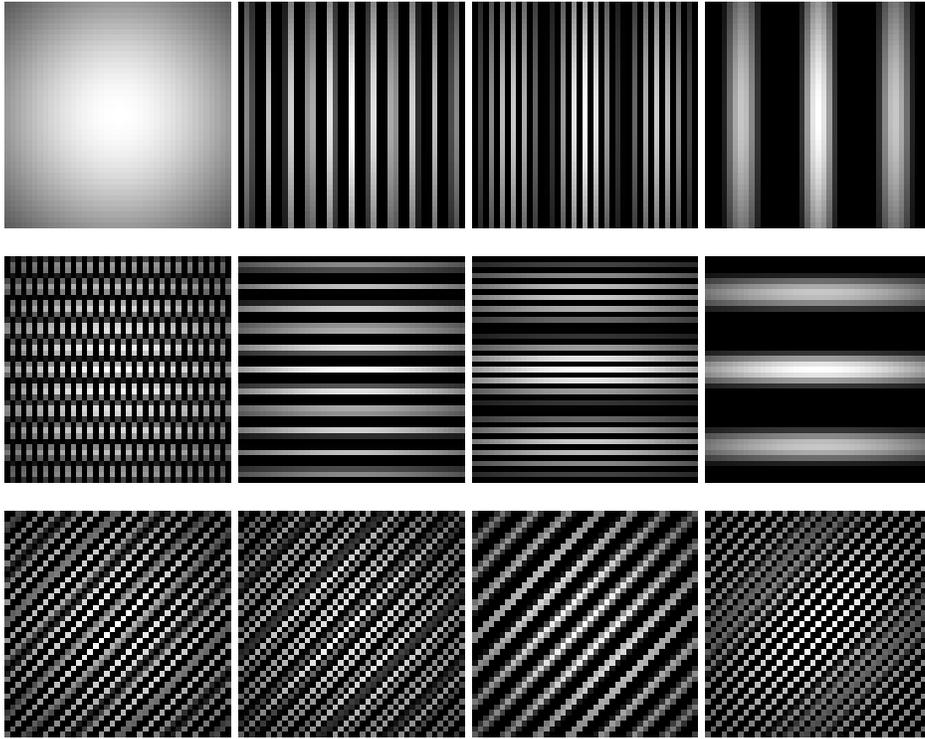


Fig. 2.4: Exemplos de filtros de Gabor.

gera redundância de informação. O procedimento realizado para extração de características consiste basicamente na convolução da imagem de entrada por uma família de filtros de Gabor que apresentam em sua composição várias escalas e orientações. Existem diversas formas para a escolha do conjunto de filtros adequados.

Na literatura são mostrados diversos métodos para a escolha dos parâmetros adequados. Por exemplo, os autores em [26] utilizam apenas quatro orientações (0 , $\pi/4$, $\pi/2$, $3\pi/4$) com o objetivo de reduzir o custo computacional e estudos psicofísicos para se determinar as frequências. Uma discussão mais abrangente sobre as diversas metodologias pode ser encontrada em [27].

Wavelets

Atualmente as metodologias mais utilizadas para análise de textura têm como base as funções *wavelets*, sejam elas na forma de transformada *wavelet* ou transformada *wavelet packet*. A transformada *wavelet packet* pode ser considerada uma generalização da trans-

formada *wavelet*. O capítulo 3 deste trabalho abordará mais detalhadamente as duas transformadas.

A extração das características é, em geral, realizada através dos componentes de detalhes obtidos através da decomposição *wavelet*. O trabalho [28] utiliza essa técnica para obtenção das características texturais em conjunto com conceitos de lógica *fuzzy* para realizar *tracking* de objetos, baseados nas texturas.

Os trabalhos [29] e [30] são outros exemplos de estudos que utilizam a energia dos coeficientes de detalhes, entretanto, nesse caso, foi escolhida a decomposição por transformada *wavelet packet* para realizar a tarefa de classificação de texturas.

Chang, em [31] optou por um outro método de extração de características utilizando a transformada *wavelet packet* para classificação de textura. Após a decomposição da textura, é observado em quais canais a energia é melhor conservada e partir desses dados é realizada a classificação.

2.2 Aplicações da Análise de Texturas

Essa seção apresenta algumas aplicações que utilizam sistemas de reconhecimento de textura.

Análise de Imagens Médicas

Inúmeras técnicas não-invasivas têm sido utilizadas na medicina para obtenção de imagens, objetivando a obtenção de diagnósticos, por exemplo fotografias, raios x, ultra-som, tomografia, etc. Outras técnicas são invasivas, como fotografias microscópicas ou biópsias.

A análise dessas imagens depende de algumas propriedades, tais como cor, morfologia e textura. Em vários desses tipos de imagens, as propriedades texturais são determinantes para fins de diagnóstico. O câncer, por exemplo, é frequentemente caracterizado por apresentar texturas peculiares, podendo ser identificado fazendo se uso de diferentes técnicas de processamento de imagens médicas. Técnicas relacionadas à textura têm sido aplicadas com sucesso em mamografias [32], exames de retina [33], lesões hepáticas [34], dentre outras.

Sensoriamento Remoto

É possível definir sensoriamento remoto como sendo a tecnologia que permite a aquisição de dados sobre objetos sem contato físico com os mesmos. Tem-se como principais objetivos distinguir e identificar as composições de diferentes materiais superficiais, como vegetação, padrões de uso de solo, rochas, etc. Técnicas de sensoriamento remoto incluem fotografias e medidas multiespectrais feitas por satélite, inspeções feitas por sismógrafos ou sonares, entre outros. A Fig. 2.5¹ mostra um exemplo de imagem obtida por sensoriamento remoto.



Fig. 2.5: Exemplo de sensoriamento remoto.

Geralmente, na interpretação de imagens de sensoriamento remoto, são utilizadas três propriedades: tonalidade (refere-se à cor e ao brilho), textura (magnitude e frequência da variação dos tons) e contexto (localização de tons, textura e padrões). O estudo desses atributos juntamente com técnicas de processamento digital de imagens, indicam formas de localizar e caracterizar alvos, como desmatamentos, queimadas, desertificação e derramamento de óleo [35], entre outros.

Inspeção Industrial

O monitoramento da qualidade dos produtos manufaturados industriais é um aspecto de fundamental importância. Existem vários métodos de inspeção, incluindo a inspeção visual humana, que é muito utilizada e muito útil. Porém, a inspeção visual está sujeita a

¹Imagem obtida em <http://www.mgaminerao.com.br/areas/mineracao/pesquisa/fotogeologia/index.html>

falhas humanas, principalmente se for realizada de forma repetitiva. O auxílio do processamento digital de imagens reduz o risco de erros decorrente do cansaço visual.

Os defeitos em superfícies costumam aparecer de modo claro no arranjo textural, possibilitando sua localização através da segmentação das texturas, por exemplo. Vários trabalhos têm proposto abordagens para inspeção industrial através de texturas [36], [37].

Biologia

O processamento digital de imagens, em especial a análise de texturas, tem sido utilizada em vários processos biológicos, tais como: identificação de espécies vegetais [27], objetivando a automatização da extração de dados de morfologia, anatomia e fisiologia vegetal; classificação de células [38]; cristalização de macromoléculas biológicas, tendo como objetivo da cristalização o estudo estrutural dessas macromoléculas, pelo método de difração de raios X [39].

2.3 Comentários

Neste capítulo, foram apresentados vários métodos para análise de texturas, como o histograma, a matriz de co-ocorrência, os filtros de Gabor e as *wavelets*. Esse último foi o método utilizado neste trabalho, através da transformada *wavelet packet*, que será melhor detalhada no capítulo 3.

Também foram enumeradas algumas áreas do conhecimento onde a análise textural é aplicada, como análise de imagens médicas, sensoriamento remoto, inspeção industrial e biologia.

Capítulo 3

Transformada *Wavelet* e Transformada *Wavelet Packet*

Wavelets têm sido utilizadas nas mais diversas áreas, como matemática, física, estatística, computação gráfica e engenharia. Na linha de processamento de imagens, as aplicações incluem: edição, compressão, análise de textura, dentre outras. A transformada *wavelet* (TW) expande um sinal dentro de um conjunto completo de funções de base (geralmente ortogonal). Diferentemente das funções de Fourier, as *wavelets* 2D representam uma imagem no domínio espaço/frequência que, contendo informações espaciais, é capaz de revelar aspectos importantes como limites, pontos de inflexão, descontinuidades em grandes derivadas e similaridade.

Neste capítulo serão introduzidos os conceitos básicos da análise *wavelet*, TW e transformada *wavelet packet* (TWP), fornecendo uma base teórica necessária para a aplicação dessa teoria nesta dissertação.

3.1 Introdução

Em nossa interação com universo, somos capazes de perceber diversas funções, ou sinais, que são interpretados pelos nossos sentidos e analisados pelo cérebro, como cores e sons. Eletrocardiograma, eletroencefalograma, velocidade e imagens são outros exemplos de sinais bastante conhecidos.

Os sinais são caracterizados por sua frequência. Para que a análise de sinais seja pos-

sível através de suas frequências são utilizadas transformadas matemáticas, como a transformada de Fourier (TF).

A TF é uma ferramenta clássica na análise de funções. A TF contínua de um sinal $f(x)$ é dada por:

$$F(s) = \int_{-\infty}^{\infty} f(x)e^{-j2\pi sx} dx \quad (3.1)$$

Na análise de Fourier não há interesse em saber em que instante de tempo cada frequência está presente no sinal. Entretanto, essa informação pode ser muito importante em algumas aplicações. Para resolver essa limitação, a TF foi adaptada para analisar apenas uma pequena seção do sinal, passando a se chamar transformada de Fourier janelada (TFJ), a qual considera um sinal não-estacionário como sendo composto por porções de sinais estacionários, conforme a equação 3.2.

$$\bar{F}(\tau, s) = \int_{-\infty}^{\infty} f(x)g(x - \tau)e^{-j2\pi sx} dx \quad (3.2)$$

onde $\bar{F}(\tau, s)$ representa a TFJ.

Essa transformada oferece as informações de frequência e tempo, mas a precisão desses dados está limitada pelo tamanho da janela $g(t)$, que é fixo para todo o sinal. Se o sinal possui detalhes importantes muito menores que a largura da janela, tem-se um problema semelhante ao da TF, ou seja, os detalhes não serão localizados. Entretanto se as características forem muito maiores que a janela, ocorre o problema inverso [40].

Esse foi um dos motivos que levou os pesquisadores a utilizarem a TW, em detrimento a TFJ, pois a TW oferece uma resolução variável, eliminando o problema da escolha do tamanho da janela.

3.2 Transformada *Wavelet* Contínua

A transformada *wavelet* contínua (TWC) foi desenvolvida por Grossman e Morlet [41] em 1984. *Wavelets* são funções geradas a partir de dilatações (ou compressões) e translações de uma única função $\psi(x)$, chamada *wavelet*-mãe. Assim, sendo $a > 0$ e $b \in \mathfrak{R}$:

$$\psi_{a,b}(x) = \frac{1}{\sqrt{a}}\psi\left(\frac{x-b}{a}\right) \quad (3.3)$$

onde a é responsável pela dilatação ou compressão e b pela translação de $\psi(x)$ ao longo do eixo x . Variando-se esses parâmetros pode-se gerar todo o espaço $L^2(\mathfrak{R})$. A constante $a^{-1/2}$ em (3.3) é o termo de normalização da energia da função com relação ao parâmetro a , assegurando que as normas das funções-base sejam iguais, pois [17]:

$$\left\|f\left(\frac{x-b}{a}\right)\right\| = \sqrt{\int_{-\infty}^{\infty} \left|f\left(\frac{x-b}{a}\right)\right|^2 dx} = \sqrt{a}\|f(x)\| \quad (3.4)$$

A transformada *wavelet* contínua é definida como o produto interno na forma,

$$W_{a,b} = \langle f, \psi_{a,b} \rangle \triangleq \int_{-\infty}^{\infty} f(x)\psi_{a,b}(x)dx \quad (3.5)$$

onde $\psi_{a,b}(x)$ é dado em (3.3).

A TWC inversa é obtida através da equação:

$$f(x) = \frac{1}{c_h} \int_0^{\infty} \int_{-\infty}^{\infty} W(a,b)\psi_{a,b}(t) \frac{da.db}{a^2} \quad (3.6)$$

Em (3.6), c_h é uma constante que depende de $\psi(t)$. A equação (3.7) é chamada de critério de admissibilidade que restringe a classe de funções que podem ser utilizadas como *wavelets*-mãe,

$$c_h = \int \frac{|\Psi(w)^2|}{|w|} dw < +\infty \quad (3.7)$$

onde $H(w)$ é a transformada de Fourier da *wavelet*-mãe $\psi(x)$. Como w é denominador da integral, é necessário que:

$$\Psi(0) = 0 = \int_{-\infty}^{\infty} \psi(x)dx = 0 \quad (3.8)$$

Como $\Psi(\infty) = 0$, pode-se afirmar que a transformada de Fourier da *wavelet*-mãe

admissível é similar à função de transferência de um filtro passa-faixas [17].

As *wavelets*-mãe podem assumir diversas formas, sendo que para cada aplicação existem funções mais adequadas do que outras. Algumas das *wavelets* mais importantes são de Meyer [42], Battle-Lemarié [43, 44] e Daubechies [45].

3.3 Transformada *Wavelet* Discreta

Calcular os coeficientes *wavelets* para todas as escalas gera muita informação redundante e requer alto esforço computacional. A transformada *wavelet* discreta (TWD) aparece como alternativa à TWC, utilizando parâmetros de escalonamento e translação discretos [46].

A TWD do sinal $f(x)$ é dada por,

$$TWD(m, p) = \frac{1}{\sqrt{a_0^m}} \sum_n f(x) \psi_{m,p}^d \left(\frac{p - na_0^m b_0}{a_0^m} \right) \quad (3.9)$$

onde $\psi_{m,p}^d$ é a família de *wavelets* gerada utilizando os coeficientes m e p discretos, a_0 e b_0 são constantes e m e p , são, respectivamente, o parâmetro de dilatação e translação, e pertencem ao conjunto dos números inteiros. Escolhendo $a_0 = 2$ e $b_0 = 1$, tem-se a *wavelet* diádica, amostrada a uma taxa de 2^m .

Tem-se, então, que:

$$\psi_{m,p}^d = \frac{1}{\sqrt{2^m}} \psi^d \left(\frac{t - 2^m p}{2^m} \right), \quad m, p \in Z. \quad (3.10)$$

Desse modo, a escala de dilatação é uma potência de dois ($a_m = 2^m$) e os passos de translação iguais a um passo da escala de dilatação ($b_m = 2^m p = a_m p$).

As componentes $\psi_{m,p}^d$ do conjunto de funções *wavelets* devem ser ortogonais e reais. Essa condição será satisfeita se:

$$\langle \psi_{m,k}^d, \psi_{m,q}^d \rangle = \delta(k - q) = \begin{cases} 1, & k = q; \\ 0, & k \neq q; \end{cases} \quad (3.11)$$

Daubechies [47] mostrou que a condição necessária e suficiente para realizar a reconstrução do sinal $f(x)$ a partir dos coeficientes *wavelets* é que a energia do somatório do módulo ao quadrado de $\psi_{m,p}^d$ deve estar entre dois limites positivos:

$$A\|f\|^2 \leq \sum_{m,p} |\langle f, \psi_{m,p}^d \rangle|^2 \leq B\|f\|^2 \quad (3.12)$$

onde $\|f\|^2$ é a energia de $f(t)$, $A > 0$ e $B < \infty$. Se $A = B$, a energia da TW é proporcional a energia do sinal e as *wavelets* se comportam como bases ortogonais. A transformada *wavelet* discreta inversa é dada por:

$$f(x) = \frac{2}{A+B} \sum_m \sum_p c_{m,p} \psi_{m,p}^d(t) \quad (3.13)$$

3.4 Análise Multiresolução

Uma outra forma de aplicar a TWD, através de banco de filtros, foi desenvolvida por Mallat [48], utilizando o conceito de multiresolução. A análise multiresolução é uma técnica que permite analisar o sinal através da decomposição do seu conteúdo em um conjunto de detalhes em diferentes resoluções. Um sinal $f(x) \in L^2(\mathfrak{R})$ definido no intervalo $0 < x < x_0$ tem resolução r_j quando é definido para r_j pontos do intervalo, ou seja, resolução é o número de pontos em que o sinal está definido no intervalo. Uma função contínua $f(x)$ tem resolução infinita, já que tem seu valor conhecido para infinitos pontos [17].

Uma aproximação do sinal original $f(x)$ é definida quando sua resolução é alterada. Dada uma sequência de resoluções $r_j, j \in Z$ adotadas para o sinal $f(x)$, tem-se que os detalhes em uma dada resolução r_j do sinal são definidos como a diferença da informação entre a aproximação com resolução r_j e a aproximação com resolução r_{j-1} [48].

Seja o sinal original $f(x) \in L^2(\mathfrak{R})$, o operador linear A_{2^j} é responsável pela projeção de $f(x)$, gerando sua aproximação com resolução 2^j . Como essa aproximação é decorrente de uma mudança na resolução de $f(x)$, pode-se entendê-la como o resultado de uma mudança na taxa de amostragem do sinal original e, dessa forma, encarar A_{2^j} como um pente de Dirac com 2^j amostras no intervalo $]0, x_0[$ [17].

Temos que o operador A_{2^j} é um operador de projeção no subespaço vetorial $V_{2^j} \subset L^2(\mathfrak{R})$, o qual pode ser interpretado como o conjunto de todas as possíveis aproximações de todas as funções $L^2(\mathfrak{R})$ no nível de resolução 2^j . O conjunto dos subespaços vetoriais V_{2^j} formam o espaço vetorial $\{V_{2^j}, j \in Z\}$.

A aproximação de $f(x)$ no nível de resolução 2^{j+1} contém toda a informação necessária para calcular a sua aproximação no nível de resolução 2^j . Uma vez que A_{2^j} é um operador de projeção em V_{2^j} , tem-se que:

$$\forall j \in \mathbb{Z}, V_{2^j} \subset V_{2^{j+1}} \quad (3.14)$$

Variando o valor de j , obtém-se:

$$\dots \subset V_{2^{-1}} \subset V_{2^0} \subset V_{2^1} \subset V_{2^2} \subset \dots \quad (3.15)$$

A partir de (3.15), pode-se concluir que os subespaços das aproximações podem ser obtidos através dos outros subespaços a partir de dilatações e contrações das aproximações.

Teorema 1 *Seja $\{V_{2^j}, j \in \mathbb{Z}\}$ uma aproximação por multiresolução em $L^2(\mathbb{R})$. Dado j , há um único operador A_{2^j} que faz a projeção ortogonal de $f(x)$ na base ortonormal de V_{2^j} . Existe uma única função $\phi(x) \in L^2(\mathbb{R})$ responsável pela formação de todas as bases ortonormais de todos os subespaços. Essa função $\phi(x)$ é denominada função de escalonamento que é definida como [48]:*

$$\phi_{j,n}(x) = 2^{j/2} \phi(2^j x - n), \quad n \in \mathbb{Z}. \quad (3.16)$$

E essa família de funções, geradas a partir da variação de n , é uma base ortonormal para cada V_{2^j} .

O Teorema 1 mostra que a aproximação por multiresolução $\{V_{2^j}, j \in \mathbb{Z}\}$ é completamente caracterizada pela função de escalonamento $\phi(x)$, que deve ser continuamente diferenciável. O decaimento assintótico de $\phi(x)$ e de sua derivada $\phi'(x)$ no infinito deve satisfazer as seguintes relações:

$$\phi(x) = O(x^{-2}) \quad (3.17)$$

$$|\phi'(x)| = O(x^{-2}) \quad (3.18)$$

Para obter a expressão que determina a função de escalonamento $\phi(x)$ a seguinte expressão é utilizada [45]:

$$\phi(x) = \sqrt{2} \sum_k h(k) \phi(2x - k) \quad (3.19)$$

Para determinar $\phi(x)$, de tal forma que ela seja bem localizada em x deve-se impor um conjunto de restrições a $h(k)$. Sendo $H(w)$ a TF de $h(n)$ [17]:

$$H(w) = \sum_{n=-\infty}^{\infty} h(n) e^{-inw} \quad (3.20)$$

As seguintes restrições deve ser satisfeitas por $H(w)$ [49]:

$$H(0) = \sum_{k=-\infty}^{\infty} h(k) = \sqrt{2} \quad (3.21)$$

$$h(n) = O(x^{-2}), \quad \text{quando } x \rightarrow \infty \quad (3.22)$$

$$|H(w)|^2 + |H(w + \pi)|^2 = 2, \quad \forall w \quad (3.23)$$

$$|H(w)| \neq 0, \quad \text{para } w \in [0, \frac{\pi}{2}] \quad (3.24)$$

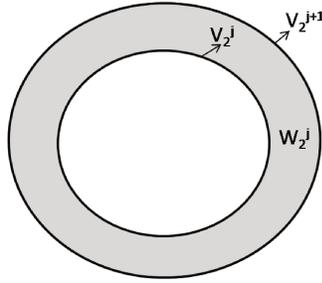
Nota-se através das equações (3.22) a (3.24) que $H(w)$ é um filtro passa-baixas.

O sinal formado a partir da diferença de informação entre os níveis de resolução 2^j e 2^{j+1} é chamado de sinal de detalhamento no nível de resolução 2^j . Foi visto que as aproximações de $f(x)$ nos níveis de resolução 2^j e 2^{j+1} são, respectivamente iguais às suas projeções ortogonais nos subespaços vetoriais V_{2^j} e $V_{2^{j+1}}$. Assim, o sinal de detalhamento em 2^j é obtido através da projeção ortogonal de $f(x)$ no subespaço vetorial que é o complemento ortogonal de V_{2^j} , W_{2^j} , ou seja:

$$V_{2^j} \oplus W_{2^j} = V_{2^{j+1}} \quad (3.25)$$

A Fig. 3.1 representa o subespaço vetorial W_{2^j} .

Para calcular a projeção de $f(x)$ em W_{2^j} , é preciso encontrar uma base ortonormal que gere o subespaço. Dessa forma, precisa-se encontrar uma função $\psi(x)$ capaz de gerar a base ortogonal em questão. Assumindo a existência de $\psi(x)$, para qualquer $n \in \mathbb{Z}$, a função $\psi_{j,n}(x) = 2^{j/2} \psi(2^j x - n)$ pertence a $W_{2^j} \subset V_{2^{j+1}}$. Pode-se então expandir $\psi_{j,n}$ na base ortonormal de $V_{2^{j+1}}$ [17]:

Fig. 3.1: Espaço vetorial W_{2^j} .

$$\psi_{j,n}(x) = \sum_{k=-\infty}^{\infty} \langle \psi_{j,n}, \phi_{j+1,k} \rangle \phi_{j+1,k}(x) \quad (3.26)$$

Calculando-se cada coeficiente resultante da projeção de $f(x)$ através da aplicação do produto interno de $f(x)$ com ambos os lados de (3.26), tem-se:

$$\langle f, \psi_{j,n} \rangle = \sum_{k=-\infty}^{\infty} \langle \psi_{-1,0}(u), \phi_{0,k-2n}(u) \rangle \cdot \langle f(u), \phi_{j+1,k}(u) \rangle \quad (3.27)$$

Pode-se encarar (3.27) como um processo de filtragem, definindo o filtro discreto $g(n)$ e reescrevendo-a:

$$\langle f, \psi_{j,n} \rangle = \sum_{k=-\infty}^{\infty} g(k - 2n) \cdot \langle f, \phi_{j+1,k}(u) \rangle \quad (3.28)$$

Deve-se notar que, diferentemente de $\phi(w)$, não há recursão para $\psi(w)$. $G(w)$ deve satisfazer a propriedade (3.29):

$$|G(w)|^2 + |G(w + \pi)|^2 = 2, \quad \forall w \quad (3.29)$$

Testando-se a admissibilidade para $\psi(w)$, substituindo 3.22 em 3.24, tem-se:

$$G(0) = 0 \Rightarrow \psi(0) = \frac{\phi(0)}{2^{1/2}} G(0) \Rightarrow \psi(0) = 0 \quad (3.30)$$

Logo, $\psi(w)$ satisfaz a condição de admissibilidade. Assim, a família $\{\psi_{j,n}\}$, é uma base *wavelet*.

É possível estabelecer uma relação entre os filtros $h(n)$ e $g(n)$. Em [17] demonstra-se

que a magnitude do espectro de $g(n)$ é igual ao espectro de $h(n)$ deslocado de π . Então, $g(n)$ é um filtro passa-altas.

Teorema 2 *Seja $\{V_{2^j}\}$ o conjunto de subespaços vetoriais de uma análise de multiresolução, $\phi(x)$ a função de escalonamento e $h(x)$ o filtro correspondente. Seja $\psi(x)$ a função cuja TF é dada por:*

$$\Psi(w) = G(0, 5w)\phi(0, 5w)2^{-1/2} \quad (3.31)$$

onde $G(w) = e^{-iw} \cdot H^*(w + \pi) e$, $\psi_{j,n}(x) = 2^{j/2}\psi(2^jx - n)$, tem-se que

$$\psi_{j,n}(x) = 2^{j/2}\psi(2^jx - n), \quad n \in Z. \quad (3.32)$$

é uma base ortonormal em W_{2^j} [50].

O Teorema 2 estabelece uma relação entre a *wavelet* $\psi_{j,n}$ e a função de escalonamento $\phi(x)$. Tem-se que, para construir uma família de *wavelets* é usualmente mais fácil começar determinado $h(x)$. Se $h(x)$ tem somente um número finito de entradas não-nulas, então $\phi(x)$, $\psi(x)$ e as *wavelets*-filhas resultantes terão todas suporte compacto, isto é, serão não-nulas somente em um pequeno intervalo em x [17].

Diversos pesquisadores estabeleceram valores para $h(x)$, a tabela 3.1 mostra os valores para $h(x)$ das *wavelet* de Daubechies (10-tap e 16-tap)[45] e de Battle-Lemarié [43].

A Fig. 3.2 mostra o esquema da decomposição do sinal $f(x)$ em 3 níveis, onde h é o filtro passa-altas, g é o filtro passa-baixas, $2 \downarrow$ representa a decimação pelo fator dois a cada escala, D_i os coeficientes de detalhes e A_i os coeficientes de aproximação na escala i .

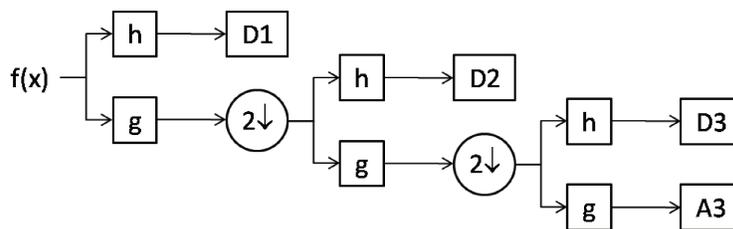


Fig. 3.2: Esquema da decomposição *wavelet* 1D.

Tab. 3.1: Coeficientes da Transformada *Wavelet*

	10-tap Daubechies	16-tap Daubechies	Battle-Lemarié
$h(0)$	0.160102	0.054416	0.766130
$h(1)$	0.603829	0.312872	0.433923
$h(2)$	0.724308	0.675631	-0.050202
$h(3)$	0.138428	0.585355	-0.110037
$h(4)$	-0.242294	-0.015829	0.032081
$h(5)$	-0.032245	-0.284016	0.042068
$h(6)$	0.077571	-0.000472	-0.017176
$h(7)$	-0.006241	0.128747	-0.017982
$h(8)$	-0.012580	-0.017369	0.008685
$h(9)$	0.003335	-0.044088	0.008201
$h(10)$		0.013981	-0.004354
$h(11)$		0.008746	-0.003882
$h(12)$		-0.004870	0.002187
$h(13)$		-0.000392	0.001882
$h(14)$		0.000675	-0.001104
$h(15)$		-0.000117	-0.000927

3.5 Transformada *Wavelet* Bidimensional

Esta seção tem como objetivo estender a análise multiresolução para o caso bidimensional (2D). O subespaço vetorial $V_{2^j}^2$ é formado por:

$$V_{2^j}^2 = V_{2^j}^1 \otimes V_{2^j}^1 \quad (3.33)$$

Tem-se que a aproximação de uma imagem $f(x, y)$ no nível de resolução 2^j é a sua projeção no subespaço vetorial $V_{2^j}^2$.

Estendendo o Teorema 1 para o caso bidimensional, pode-se mostrar que há uma única função de escalonamento $\phi(x, y) \in L^2(\mathbb{R}^2)$ responsável pela formação de todas as bases ortogonais dos subespaços em questão [48]. Assim, a família $\{\phi_{j;n_x n_y}(x, y)\}$ dada por

$$\phi_{j;n_x n_y}(x, y) = 2^j \phi(2^j x - n_x, 2^j y - n_y), \quad (n_x, n_y) \in Z^2 \quad (3.34)$$

forma uma base ortonormal para cada $V_{2^j}^2$.

Sendo $\phi(x)$ a função de escalonamento unidimensional referente a $\{V_{2^j}^1\}$, Mallat [51] mostra o caso em que a função de escalonamento bidimensional $\phi(x, y)$ é ortogonal e

separável, podendo ser expressa por:

$$\phi(x, y) = \phi(x) \cdot \phi(y) \quad (3.35)$$

O mesmo ocorre para a base ortogonal em $V_{2^j}^2$:

$$\phi_{j;n_x,n_y}(x, y) = \phi_{j;n_x}(x) \cdot \phi_{j;n_y}(y), \quad (n_x, n_y) \in Z^2 \quad (3.36)$$

Analogamente ao caso unidimensional, a aproximação de uma imagem $f(x, y)$ no subespaço vetorial $V_{2^j}^2$ é obtida por [17]:

$$A_{2^j} \cdot f(x, y) = \sum_{n_x=-\infty}^{\infty} \sum_{n_y=-\infty}^{\infty} \langle f(u, v), \phi_{j;n_x,n_y} \rangle \cdot \phi_{j;n_x,n_y} \quad (3.37)$$

Definindo o filtro discreto $h(l, m)$ de resposta impulsiva em (3.38), pode-se encarar o produto interno de $f(x, y)$ e $\phi_{j;n_x,n_y}(x, y)$ como o processo de filtragem descrito em (3.39).

$$\begin{aligned} h(l, m) &= \langle \phi_{-1;0,0}(u, v), \phi_{0;l,m}(u, v) \rangle = \\ &= \langle \phi_{-1;0}(u), \phi_{0;l}(u) \rangle \cdot \langle \phi_{-1;0}(v), \phi_{0;m}(v) \rangle = h(l) \cdot h(m) \end{aligned} \quad (3.38)$$

Deve-se notar que o filtro $h(l, m)$ é separável nas direções x e y , podendo ser decomposto no produto de dois filtros digitais unidimensionais. A equação (3.39) é a expressão que, para (n_x, n_y) fixo, obtém cada coeficiente da projeção de $f(x, y)$ na resolução 2^j a partir de todos os coeficientes da projeção de $f(x, y)$ na resolução 2^{j+1} .

$$\begin{aligned} \langle f(x, y) \cdot \phi_{j;n_x,n_y}(x, y) \rangle &= \\ &= \sum_{k_y=-\infty}^{\infty} h(k_y - 2n_y) \cdot \sum_{k_x=-\infty}^{\infty} h(k_x - 2n_x) \cdot \langle f(x, y) \cdot \phi_{j+1;k_x,k_y}(x, y) \rangle \end{aligned} \quad (3.39)$$

Essa equação mostra que $A_{2^j}^d \cdot f$ pode ser calculada através de filtrações sucessivas de $A_{2^{j+1}}^d \cdot f$ em ambas as direções. Primeiramente, $A_{2^j}^d \cdot f$ é filtrado na direção horizontal (direção das linhas, x) com o filtro $h(-n)$ 1D, dizimando-se o resultado por um fator de 2 nessa direção (ou seja, descartando-se uma linha sim e outra não). A seguir, o resultado

é filtrado na direção vertical (direção das colunas, y) com o mesmo $h(-n)$, seguido novamente pela decimação por um fator de 2 na direção vertical (ou seja, descartando-se uma coluna sim e outra não). Esse processo encontra-se esquematizado no ramo superior da Fig. 6.1.

Todas as aproximações $A_{2^j}^d \cdot f$ para $j < 0$ podem ser obtidas recursivamente tendo como ponto de partida $A_1^d \cdot f$, utilizando esse procedimento. Alternativamente, pode-se realizar primeiramente o processamento por colunas e depois por linhas, à semelhança do que ocorre com as demais transformadas bidimensionais. Essa opção também é válida para o sinal de detalhamento e para a transformação inversa.

Sendo $\{V_{2^j}^2\}$ a aproximação por multiresolução separável em $L^2(\mathbb{R}^2)$, o detalhe no nível de resolução 2^j é a projeção de $f(x, y)$ no subespaço vetorial $\{W_{2^j}^2\}$ que, analogamente ao caso 1D, é o complemento ortogonal de $\{V_{2^j}^2\}$ em $\{V_{2^{j+1}}^2\}$ [17].

Diferentes imagens de detalhes $D_{2^j}^D \cdot f$ podem ser calculadas através de filtrações sucessivas de $D_{2^{j+1}}^D \cdot f$ em ambas as direções. Primeiramente $D_{2^{j+1}}^D \cdot f$ é filtrado na direção horizontal (linhas, x) com um filtro 1D, dizimando-se o resultado por um fator de 2 nesta direção. A seguir, o resultado é filtrado na direção vertical (colunas, y) com outro filtro 1D, seguido novamente pela decimação por um fator de 2 na direção vertical. O filtro utilizado em cada direção depende da *wavelet* em questão [17]. A Fig. 3.3 ilustra a representação *wavelet* ortogonal 2D para três estágios ($J = 3$).

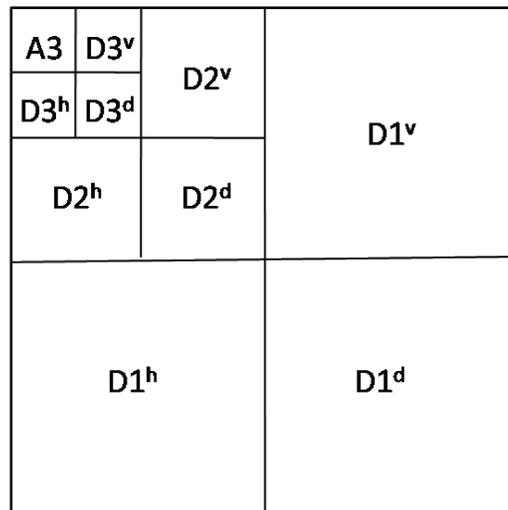


Fig. 3.3: Decomposição *wavelet* bidimensional na forma piramidal.

Para efetuar a reconstrução do sinal $f(x)$ a partir dos coeficientes da decomposição *wavelet*, deve-se notar que cada coeficiente de $A_{2^{j+1}} \cdot f$ é formado por todos os coeficientes de $A_{2^j} \cdot f$, $D_{2^j}^v \cdot f$, $D_{2^j}^h \cdot f$ e $D_{2^j}^d \cdot f$. O diagrama em blocos de um estágio da reconstrução da imagem encontra-se na Fig. 3.4. A imagem $f(x, y)$ é reconstruída a partir da representação *wavelet* repetindo-se este processo para $-1 \leq j \leq -J$.

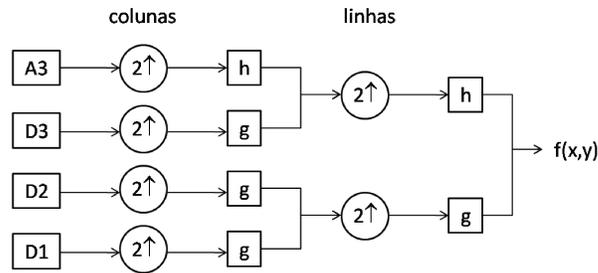


Fig. 3.4: Reconstrução *wavelet* bidimensional.

A Fig. 3.6 mostra um exemplo de decomposição por transformada *wavelet* da Fig. 3.5 formada pela textura D49 do álbum de Brodatz [12]. Apesar da imagem ser composta por apenas uma textura, pode-se observar que o quadrado central foi girado em 90 graus, para que se possa verificar o efeito das imagens de detalhes nas direções verticais, horizontais e diagonais.

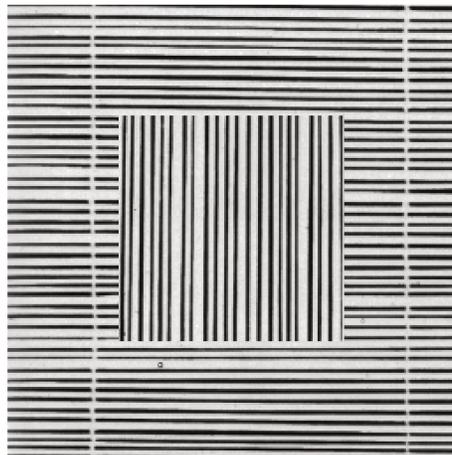


Fig. 3.5: Imagem formada pela textura D56 do álbum de Brodatz.

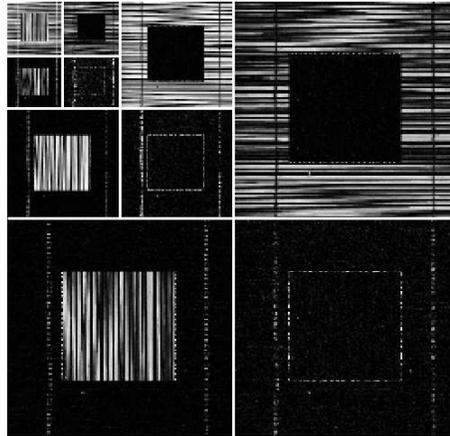


Fig. 3.6: Decomposição *wavelet* da Fig. 3.5.

3.6 Transformada *Wavelet Packet*

Foi visto na seções 3.4 e 3.5 que na construção da decomposição *wavelet* de um sinal qualquer $f(x)$ ou $f(x, y)$, esse pode ser analisado como uma sequência de filtragem, que utilizam filtros passa-baixas e passa-altas. Apenas os coeficientes de aproximação sofrem novas decomposições. Esse processo pode ser feito n vezes, onde 2^n é o tamanho do sinal amostrado.

Na transformada *wavelet packet* (TWP) além dos coeficientes de aproximação, os de detalhes também são subdivididos em sinais de aproximação e de detalhes, decompondo o sinal em uma árvore binária. Dessa forma, tem-se o sinal representado em diversas escalas.

Pode-se definir a biblioteca de funções *wavelet packet* como o conjunto de funções na forma $W_n(2^l x - k)$, onde l e $k \in \mathbb{Z}$ e $n \in \mathbb{N}$. Cada função é determinada por três parâmetros: l , o parâmetro de escala, k , o parâmetro de localização e n o parâmetro de oscilação [31]. Esses parâmetros também podem receber a seguinte notação: s , p e f , respectivamente [46].

A biblioteca de funções *wavelet packet* é construída segundo (3.40) e (3.41),

$$W_{2n}(x) = \sqrt{2} \sum_k h(k) W_n(2x - k) \quad (3.40)$$

$$W_{2n+1}(x) = \sqrt{2} \sum_k g(k) W_n(2x - k) \quad (3.41)$$

onde $W_0(x)$ e $W_1(x)$ correspondem, respectivamente a função de escalonamento $\phi(x)$ e a *wavelet*-mãe $\psi(x)$.

Os coeficientes da decomposição *wavelet packet* do sinal $f(x) \in L^2(\mathfrak{R})$ são obtidos através do produto interno entre o sinal e as funções *wavelet packet* $w_{lkn}(x)$:

$$\lambda_{2k}(n) = \langle f(x), w_{lkn} \rangle = \int_{-\infty}^{\infty} f(x) 2^{-l/2} w_f(2^{-l}x - k) dx \quad (3.42)$$

A decomposição em $L^2(\mathfrak{R})$ em múltiplas escalas também é possível utilizando (3.44) para obter versões trasladadas e dilatadas das funções *wavelets packets*:

$$w_{lkn}(x) = 2^{-l/2} w_f(2^{-l}x - k) \quad (3.43)$$

Como na TWD, a TWP pode ser implementada como uma sequência de filtros, satisfazendo a seguinte relação recursiva [52]:

$$\lambda_{l+1,2n}(k) = \sum_{j \in \mathbb{Z}} h(j) \lambda_{ln}(2k - j) = H \lambda_{ln}(p) \quad (3.44)$$

$$\lambda_{l+1,2n+1}(k) = \sum_{j \in \mathbb{Z}} g(j) \lambda_{ln}(2k - j) = G \lambda_{ln}(p) \quad (3.45)$$

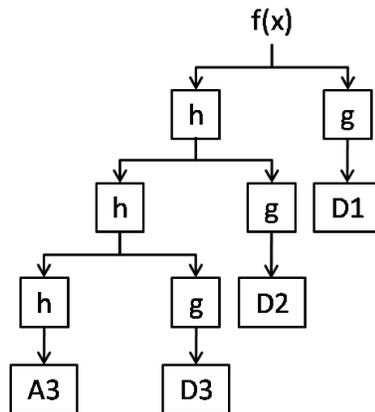
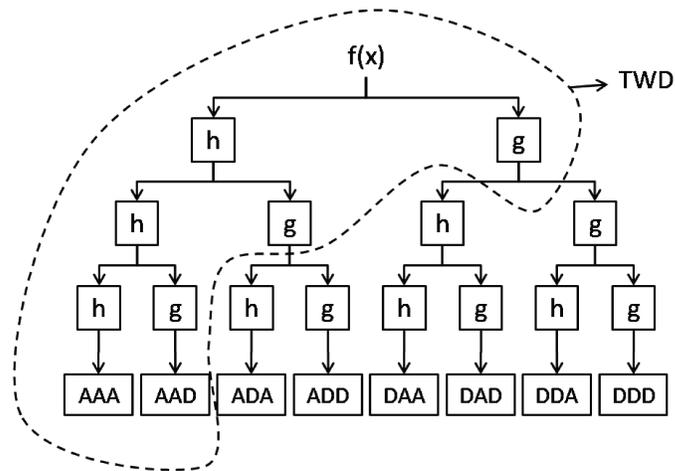
Com as equações (3.44) e (3.45) não é necessário recalculer o produto interno de todo o sinal, apenas aplicar os filtros $h(n)$ e $g(n)$ nos coeficientes do nível de dilatação anterior [46].

As Figs. 3.7 e 3.8 mostram os esquemas das TWD e da TWP, respectivamente. Pode-se verificar que a TWD é apenas constituída de apenas uma parte da TWP, sendo possível afirmar que a TWP é uma generalização da TWD [29].

3.6.1 Transformada *Wavelet Packet* 2D

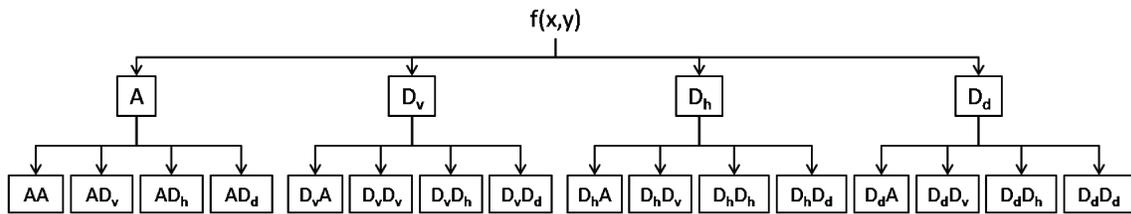
Em duas dimensões, a TWP, assim como a TWD, também é aplicada nos subespaços de detalhes verticais, horizontais e diagonais, gerando assim uma quantidade de 4^j subespaços ao invés de $1 + 3j$ da TWD. A Fig. 3.9 representa a decomposição *wavelet packet* 2D no nível 2.

Na Fig. 3.10 pode-se visualizar a disposição dos componentes da decomposição

Fig. 3.7: Esquema da transformada *wavelet* discreta.Fig. 3.8: Esquema da transformada *wavelet packet*.

wavelet packet de nível 2 na forma piramidal. E a Fig. 3.11 mostra um exemplo da decomposição *wavelet* da imagem formada pela textura D49 do álbum de Brodatz [12] em 3 níveis de decomposição.

Em relação a transformada *wavelet* padrão, a TWP fornece um *zoom* na faixa de frequência desejada. É utilizada como ferramenta para extrair características relevantes de uma ou mais dimensões, melhorando o desempenho (performance) de classificadores ao extrair características relativas através da decomposição em diferentes bandas de frequência [53]. Diversos autores fizeram uso dessa transformada para extrair informações sobre texturas [31], [29].

Fig. 3.9: Esquema da transformada *wavelet packet* 2D.

AA	D_vA	AD_v	D_vD_v
D_hA	D_dA	D_hD_v	D_dD_v
AD_h	D_vD_h	AD_d	D_vD_d
D_hD_h	D_dD_h	D_hD_d	D_dD_d

Fig. 3.10: Transformada *wavelet packet* 2D na forma piramidal.

Uma única decomposição *wavelet packet* fornece muitas bases admissíveis, das quais pode-se escolher a melhor, de acordo com o objetivo final. Assim, existem mais de um conjunto de funções possíveis que formam uma base. As áreas em cinza na Fig. 3.12 constituem um conjunto de bases para uma representação particular do sinal $f(x)$. Esse conjunto é uma representação mínima na qual o sinal pode ser reconstruído. No entanto, existem vários outros subconjuntos de bases disponíveis que também representam a decomposição completa do sinal $f(x)$.

Dado que existem várias possíveis bases para a representação do sinal (ou imagem), pode-se escolher um subconjunto que melhor se adequa a um problema em particular. O método para a seleção da melhor base, também chamada de representação ótima, é chamado *best basis*.

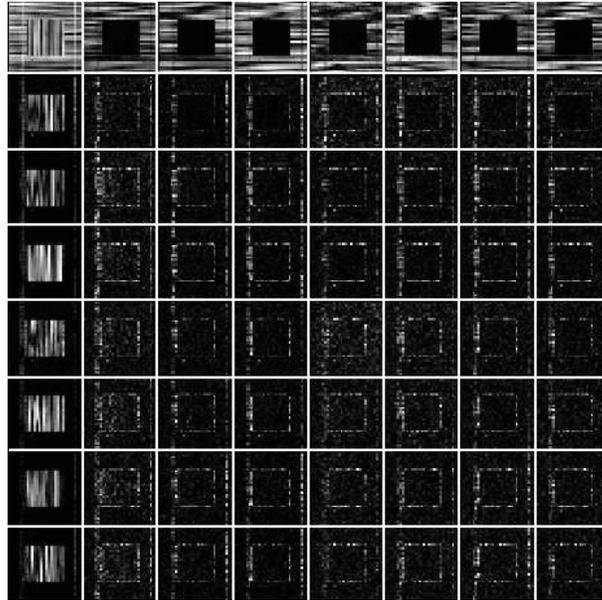


Fig. 3.11: Transformada *wavelet packet* 2D da Fig. 3.5.

3.6.2 Algoritmo *Best Basis*

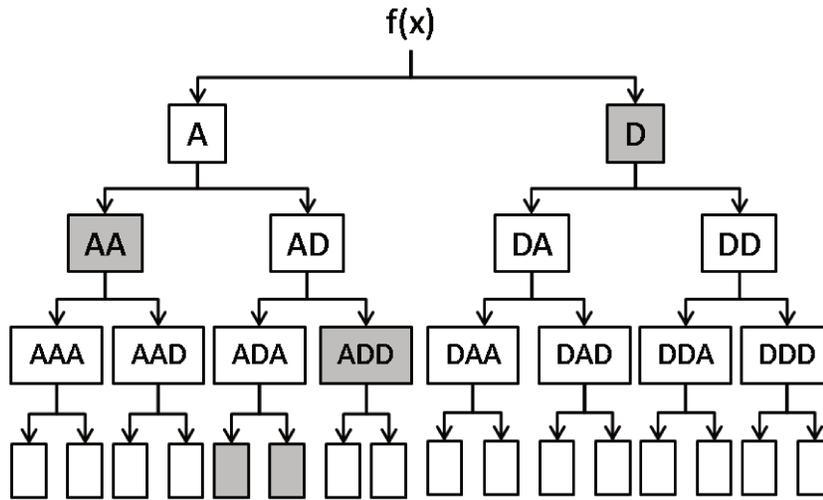
O algoritmo *best basis* utiliza o conceito de função custo. Seja uma função custo de informação (ou custo de representar a informação escolhida) em sequências de números reais e complexos, podendo assumir qualquer valor real funcional M satisfazendo a seguinte condição de aditividade [46]:

$$M(u) = \sum_{k \in Z} \mu(|u(k)|) \mu(0) = 0 \quad (3.46)$$

Seja β uma coleção de bases e B uma base qualquer, tal que $B \in \beta$, $u(k)$ uma sequência de coeficientes de $u(k) = \langle b_k, x(k) \rangle = B.x$, onde $b_k \in \beta$ é o k -ésimo vetor da base B , μ uma função definida no intervalo $[0, \infty)$, então o custo de informação de representar um vetor x na base B é $M_x(B.x)$ [46].

A *Best Basis* do vetor x é a base B na qual $M_x(B.x)$ seja mínimo. Alguns exemplos da função custo de informação são: *threshold*, concentração em l^p , entropia e logaritmo da energia. A seguir, os critérios serão melhor detalhados [52].

Threshold: essa função custo consiste em dado um limiar ε , fornecer o número de coeficientes necessários para transmitir ou receber o sinal com precisão ε .

Fig. 3.12: Representação de uma base *wavelet packet*.

Concentração em l^p : um valor p é escolhido no intervalo $0 < p < 2$ [52] e determina-se $M(u) = (\|u\|_p)^p$, tal que u pertença ao espaço $l^p(Z)$. Se duas sequências possuem energia $\|u\| = \|v\|$ mas $M(u) < M(v)$, então u possui maior concentração de energia em menor número de elementos.

Entropia: o critério baseado na entropia, particularmente a entropia de Shannon é um dos mais utilizados e é expresso por:

$$\varepsilon^X = - \sum_i \frac{|x_i|^2}{|x|^2} \log_2 \frac{|x_i|^2}{|x|^2}, \quad \text{onde } |x|^2 = \sum_i |X_i|^2 \quad (3.47)$$

Logaritmo da Energia: seja $M(u) = \sum_{k=1}^N \log |u(k)|^2$. Isto pode ser interpretado como a entropia de um processo Gauss Markov, no qual que $u(k)$ produz N vetores cujas coordenadas têm variâncias $\sigma_1^2 = |u(1)|^2, \dots, \sigma_N^2 = |u(N)|^2$. É assumido que não há componentes no processo que não mudam, ou seja $\sigma_1^2 = |u(1)|^2 \neq 0$ para todos $k = 1, \dots, N$. Minimizando $M(u)$ sobre B encontra uma base Karhunen-Loève para o processo.

O algoritmo *best basis* está baseado na escolha da decomposição do sinal na qual se obtém o menor número de coeficientes necessários para se obter 99% da energia do

signal. Pode-se, resumidamente, assim descrever o algoritmo *best basis* utilizando a entropia como critério:

1. Calcula-se a entropia de todos os nós durante a decomposição, a árvore é analisada recursivamente de baixo para cima, os nós finais são identificados.
2. Então, compara-se a entropia de cada par adjacente com a entropia da união desses pares.
3. Escolhe-se a expansão de entropia mais baixa e continua-se o processo. Ou seja, quando a informação de custo do nó-pai é menor que a informação de custo dos nós-filhos, o pai é marcado.
4. Caso contrário, atribui-se a informação de custo total mais baixa dos nós-filhos ao nó-pai [53].

A Fig. 3.14 mostra um diagrama de um nó-pai e dois nós-filhos para o cálculo da função custo.

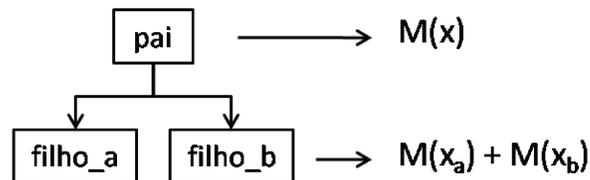


Fig. 3.13: Representação de nó da árvore da TWP.

Aqueles nós que possuem maior custo são podados da árvore. O processo é feito até atingir o topo da árvore, formando então uma base completa de acordo com a função custo escolhida. A árvore resultante é denominada *best basis*. Um exemplo de árvore podada pode ser visto na Fig. 3.14.

A complexidade da busca pelas melhores bases é proporcional ao número de nós da árvore. Essa complexidade é determinada pelo cálculo do custo de todos os coeficientes para todas as bases da biblioteca, o que leva para $O(N \log N)$ para *wavelets packets*.

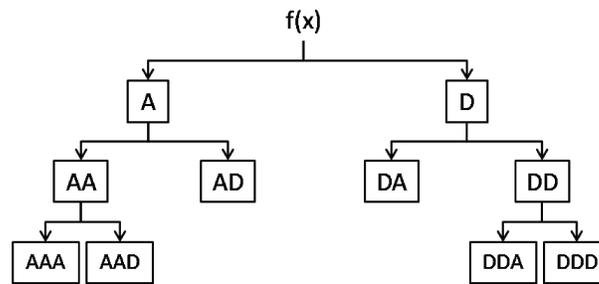


Fig. 3.14: Representação da árvore da TWP podada.

3.7 Comentários

Este capítulo apresentou a teoria das transformada *wavelet* e *wavelet packet*. Foi feita uma introdução à TW desde a TF, passando pela TWD. Foi mostrado o conceito de análise multidimensional para a decomposição dos sinais em diversas escalas, permitindo realizar a extração de características do sinal para as mais diversas aplicações. Por fim, foi apresentada a teoria da TWP, que será utilizada nesta dissertação para extrair as características texturais.

Capítulo 4

Reconhecimento de Padrões

Reconhecimento de padrões é uma sub-área da área de pesquisa de aprendizagem de máquina, que tem o objetivo de classificar ou descrever padrões ou objetos, baseado em características conhecidas *a priori* ou através de informações obtidas dos próprios padrões.

Padrão é qualquer objeto ou entidade. Alguns exemplos são: imagem, textura, música, rosto humano, sinal de voz, texto, etc.

Pode-se dividir o reconhecimento de padrões em dois tipos: classificação e *clustering*. No primeiro, há conhecimento sobre o conjunto de classes. No segundo, deseja-se agrupar os padrões em classes desconhecidas.

A Fig. 4.1 mostra as etapas básicas de um processo geral de reconhecimento de padrões. Nas seções seguintes cada etapa aqui apresentada será detalhada.

São inúmeras as atuais aplicações de reconhecimento de padrões, dentre elas pode-se citar: reconhecimento de caracteres, análise e reconhecimento da fala, detecção de defeitos na indústria, visão por computador e identificação de retina.

4.1 Extração de Características

Em geral, para se ter a capacidade de identificar um objeto, obtem-se dele algumas características, como forma, tamanho, cor, etc. Pode-se aplicar a mesma condição para a visão por computador, pois, para que os algoritmos sejam capazes de classificar um objeto como pertencente a uma classe são necessárias algumas características.

Existem diversas formas de obter as características desejadas, dependendo de cada

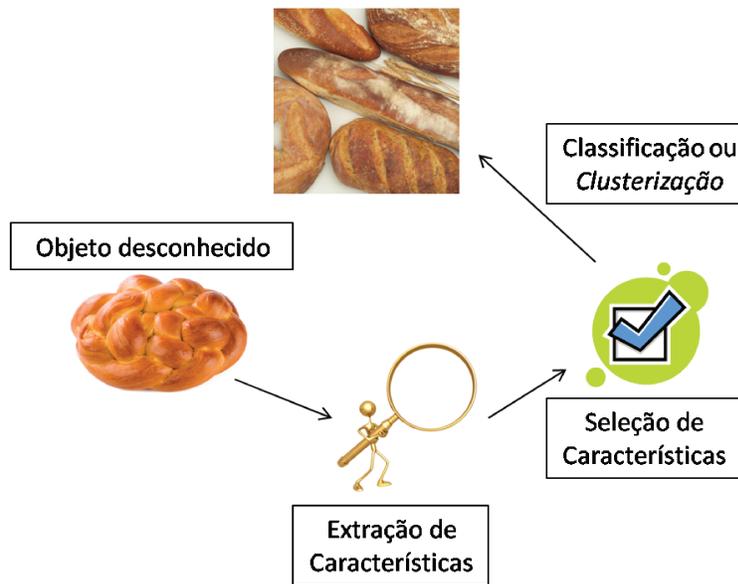


Fig. 4.1: Representação das etapas do Reconhecimento de Padrões.

Aplicação	Metodologia
Reconhecimento de caracteres	Filtros de Gabor [54]
Identificação de retina	Transformada de Hough [55]
Reconhecimento da fala	Redes neurais [56]
Reconhecimento de microcalcificações em mamografias	Modelo de Markov [57]

Tab. 4.1: Exemplo de aplicações de reconhecimento de padrões e métodos de extração de características.

aplicação.

A Tab. 4.1 mostra alguns exemplos de trabalhos encontrados na literatura.

Como pode-se verificar, muitas vezes as transformadas são utilizadas pois essas têm a capacidade de representar a imagem em outro domínio que, em geral, torna as características desejadas mais acessíveis. A transformada *wavelet*, apresentada no capítulo 3, tem sido muito utilizada nas mais diversas aplicações.

4.2 Seleção de Características

A segunda etapa do processo de reconhecimento de padrões é a seleção de características. Esse passo é fundamental, pois tem a função de selecionar aquelas características que têm a maior capacidade de discriminação entre os objetos (no caso da clusterização) ou entre as classes.

Existem na literatura vários algoritmos que executam esta tarefa, mas o mais utilizado é a análise de componentes principais, também conhecido como transformada de Hotelling ou transformada de Karhunen-Loève.

4.2.1 Análise de Componentes Principais

Como foi dito anteriormente, o objetivo dessa análise ou transformada é reduzir a dimensionalidade das características e também torná-las capazes de diferenciar os padrões em questão. Para entender o funcionamento da transformada, são necessários alguns conhecimentos de álgebra linear e processos estocásticos.

Dado X um vetor aleatório de dimensão m com média zero:

$$E[X] = 0 \quad (4.1)$$

onde E é o operador estatístico do valor esperado. Se A é a projeção de X em q , um vetor unitário também de dimensão m , tem-se que:

$$A = X^T q = q^T X \quad (4.2)$$

Isto é, A é o produto interno entre q e X , tendo a restrição: $\|q\| = (q^T q)^{1/2} = 1$.

A projeção A é uma variável aleatória com média e variância iguais ao vetor X . Como X , tem média zero, A também a terá, ou seja:

$$E[A] = q^T E[X] = 0 \quad (4.3)$$

E a variância de A , pode ser assim obtida:

$$\sigma^2 = E[A^2] \quad (4.4)$$

$$= E[(q^T X)(X^T q)] \quad (4.5)$$

$$= q^T E[XX^T]q \quad (4.6)$$

$$= q^T Rq \quad (4.7)$$

Seja R a matriz $m \times m$ de correlação do vetor X :

$$R = E[XX^T] \quad (4.8)$$

a matriz de correlação é sempre simétrica, então:

$$R^T = R \quad (4.9)$$

Segue que se a e b são vetores de tamanho $m \times 1$:

$$a^T Rb = b^T Ra \quad (4.10)$$

De (4.5) tem-se que a variância da projeção A é função de q , pode-se então escrever:

$$\Psi(q) = \sigma^2 = q^T Rq \quad (4.11)$$

Ou seja, através de $\Psi(q)$ ve-se que a variância depende exclusivamente de q . Tem-se que a variância da projeção do vetor X será função exclusivamente da base ortogonal q .

Para encontrar q , tal que seja um vetor unitário, faz-se o cálculo da matriz dos autovetores da matriz de correlação R . Dados os autovalores $\lambda_1, \lambda_2, \lambda_3, \dots, \lambda_m$, tal que $\lambda_1 > \lambda_2 > \lambda_3 > \dots \lambda_m$, e seus autovetores associados $q_1, q_2, q_3, \dots, q_m$, pode-se escrever:

$$Rq_j = \lambda_j q_j \quad j = 1, 2, 3, \dots, m \quad (4.12)$$

$$Q = [q_1, q_2, q_3, \dots, q_m] RQ = Q\Lambda \quad (4.13)$$

onde Λ é uma matriz diagonal tendo $\lambda_1, \lambda_2, \lambda_3, \dots, \lambda_m$ como elementos da matriz principal.

A partir dessa definição pode-se definir uma fórmula de síntese, dada por [58]:

$$x = Qa = \sum_{j=1}^m a_j q_j \quad (4.14)$$

onde:

$$a = [a_1, a_2, a_3, \dots, a_m]^T = [x^T q_1, x^T q_2, x^T q_3, \dots, x^T q_m] = Q^T x \quad (4.15)$$

Para a realização do objetivo em questão, que é diminuir a dimensionalidade do vetor X , escolhendo características que melhor o representem, é possível descartar combinações lineares em (4.14), conservando aquelas com variâncias elevadas. Tem-se, então, que:

$$\hat{x} = \sum_{j=1}^l a_j q_j = [q_1, q_2, \dots, q_l] \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_l \end{bmatrix}, \quad l \leq m \quad (4.16)$$

O conjunto de componentes a_l podem ser calculados a partir da equação de análise:

$$a_j = q_j^T x = x^T q_j, \quad j = 1, 2, \dots, m \quad (4.17)$$

onde a_j são as projeções de x sobre as direções principais representadas pelos autovetores ortogonais e unitários q_j , e, por isso são chamados de componentes principais de x .

Pode-se representar (4.17) na forma reduzida [58]:

$$\begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_l \end{bmatrix} = \begin{bmatrix} q_1^T \\ q_2^T \\ \vdots \\ q_l^T \end{bmatrix} x, \quad l \leq m \quad (4.18)$$

A Equação 4.18 sugere que existe um mapeamento do espaço de dados para o espaço de características, assim definido como o espaço formado pelo autovalores da matriz de correlação R . O mapeamento é realizado de um espaço R_m para um espaço R_l , representando assim um "codificador" para essa representação aproximada. A Fig. 4.2 representa, resumidamente, a redução da dimensionalidade dos dados.

A seguir, serão apresentados dois exemplos, retirados de [59], que ilustrarão a teoria

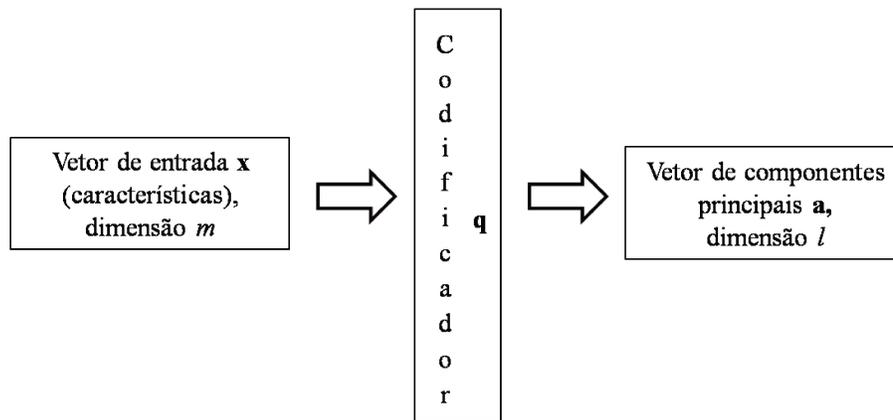


Fig. 4.2: Esquema simplificado da redução da dimensionalidade das características.

acima descrita.

A Fig. 4.3.(a) representa um objeto binário tratado como um conjunto bidimensional. Cada *pixel* do objeto é um vetor bidimensional $x = (a, b)^T$, em que a e b são os valores das coordenadas de cada *pixel* com respeito aos eixos x_1 e x_2 . Esses vetores são utilizados para computar o vetor médio e a matriz de covariância do objeto.

O efeito resultante de (4.17) é o de estabelecer um novo sistema de coordenadas cuja origem está no centróide do conjunto e cujos eixos estão na direção dos autovetores, como mostrado na Fig. 4.3.(b). Esse sistema de coordenadas mostra que a transformação dada em (4.17) é uma transformação de rotação que alinha os dados com os autovetores, como mostrado na Fig. 4.3.(c). De fato, esse alinhamento é exatamente o mecanismo que des-correlaciona os dados.

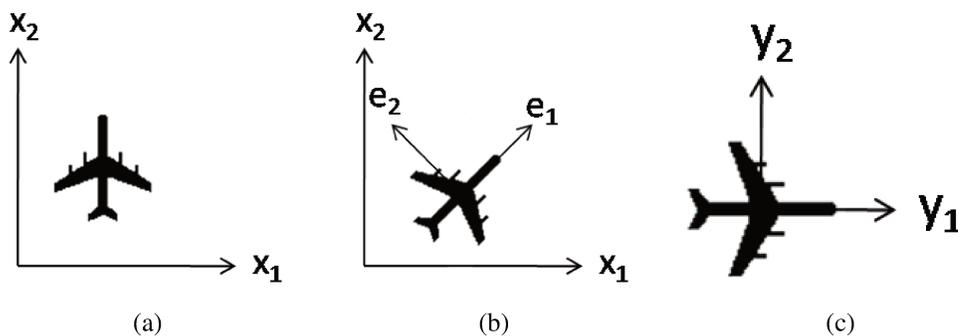


Fig. 4.3: (a) Objeto binário. (b) Eixos principais (auto-vetores). (c) Objeto rotacionado através do uso de (4.17).

Essa propriedade desempenha um papel importante na metodologia aqui proposta, pois descorrelaciona as características extraídas das imagens de textura, facilitando o processo de clusterização.

A outra propriedade a ser exemplificada diz respeito à reconstrução do vetor X a partir de a , como é mostrado por (4.16). Utilizando um número menor de autovetores é possível diminuir o número de características em questão.

Será utilizada como exemplo a transformada *wavelet packet* de nível 2, mostrada na Fig. 4.4.

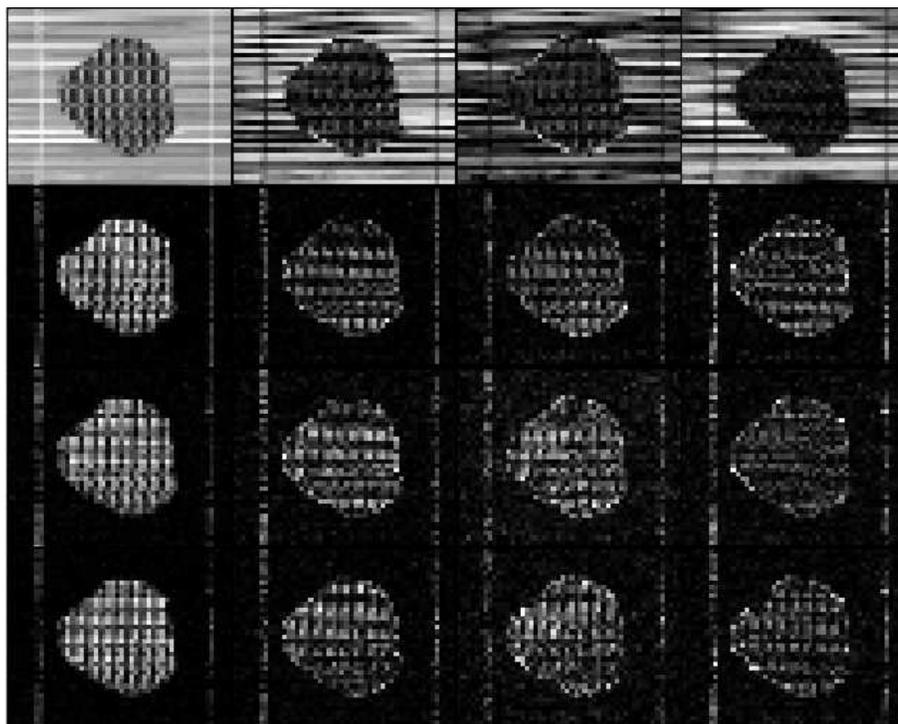


Fig. 4.4: Transformada *wavelet packet* da imagem.

A Tab. 4.2 mostra os autovalores obtidos para cada decomposição em ordem decrescente.

O uso de (4.17) gerou um conjunto de vetores transformados a partir de X . A partir deles, dezesseis componentes principais das imagens foram montados através do processo de síntese mostrado em (4.14). A Fig. 4.5 mostra os resultados obtidos. A primeira imagem (de cima para baixo e da esquerda para a direita) tem o maior autovalor associado, mostrado na Tab. 4.2, e conseqüentemente, o maior contraste.

λ_1	λ_2	λ_3	λ_4	λ_5	λ_6	λ_7	λ_8
0,0830	0,0287	0,0151	0,0083	0,0070	0,0058	0,0040	0,0026
λ_9	λ_{10}	λ_{11}	λ_{12}	λ_{13}	λ_{14}	λ_{15}	λ_{16}
0,0024	0,0015	0,0012	0,0009	0,0006	0,0004	0,0003	0,0002

Tab. 4.2: Autovalores da decomposição por transformada *wavelet packet* da Fig. 4.4.

As três primeiras imagens representam cerca de 78% da variância total, assim, pode-se utilizar apenas os três primeiros conjuntos de características para realizar o processo de clusterização.

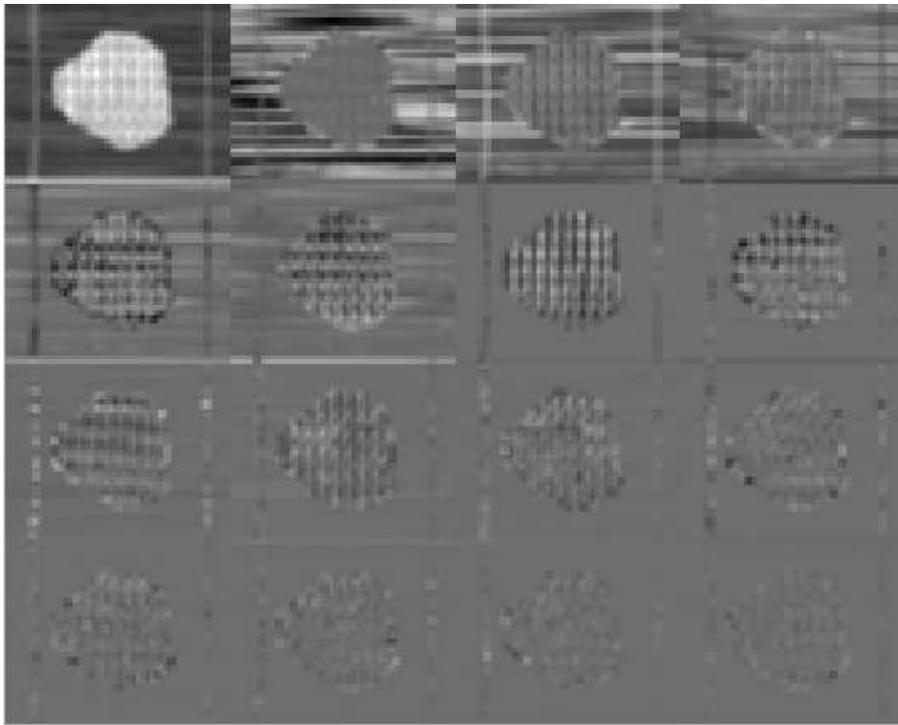


Fig. 4.5: Componentes principais da decomposição por transformada *wavelet packet*.

4.3 Clusterização

Após a extração e seleção das características apresentadas nas seções anteriores, resta ainda a última etapa do reconhecimento de padrões: a clusterização ou classificação, de-

pendendo da aplicação. Na metodologia proposta, deseja-se usar um processo de clusterização, pois trata-se de um processo não-supervisionado e não há conhecimento *a priori* sobre as texturas.

Diferentes abordagens para clusterização de dados têm sido descritas na literatura. Pode-se organizá-los em grupos, de acordo com a abordagem utilizada na definição dos clusters: particionamento, redes auto-organizáveis, baseados em densidade, hierárquico e baseado em grafos [60].

Na clusterização por particionamento, o conjunto de dados é dividido em clusters, podendo isso acontecer uma única vez ou hierarquicamente. Baseado numa função custo ou função objetivo. O *K-means* [61] é um dos algoritmos por particionamento mais famosos, em que os dados são divididos em *K clusters*, sendo *K* definido pelo usuário.

As redes neurais são utilizadas em geral para processos de classificação, onde dados são separados em classes previamente conhecidas. Entretanto, as redes auto-organizáveis (SOM) têm a capacidade de separar dados em grupos desconhecidos através de um processo aprendizagem, tendo a desvantagem de poder representar apenas *clusters* de formas esféricas [60].

Os métodos de clusterização por densidade utilizam a ideia de que os *clusters* localizam-se em regiões de maior densidade no espaço de métricas e são separados por regiões de baixa densidade. O algoritmo DBSCAN (*A Density Based Spatial Clustering of Applications with Noise*) [62] é um exemplo deste tipo de algoritmo.

Os algoritmos de clusterização hierárquica foram *clusters* a partir de diversas divisões no conjunto de dados de acordo com medidas de similaridade.

A clusterização baseada em grafos está fortemente relacionada com os algoritmos de particionamento e clusterização hierárquica. Representando os dados e sua proximidade através de um grafo $G(V, E)$, onde $V = v_1, \dots, v_n$ é o conjunto de vértices e E é o conjunto de arestas [60].

Para a metodologia proposta, deseja-se um algoritmo de clusterização não supervisionado, eliminando a necessidade de conhecimento prévio sobre o número de texturas presentes na imagem. O algoritmo escolhido para tal procedimento foi o ARIA (*Adaptive Radius Immune Algorithm for Data Clustering*), proposto por Bezerra et. al. [63].

4.3.1 O Sistema Imunológico

Antes de se falar sobre o algoritmo utilizado neste trabalho, é importante uma breve introdução sobre sua inspiração: o sistema imunológico humano.

O sistema imunológico ou sistema imune, tem como função proteger o nosso organismo de invasores (chamados de antígenos), como vírus, bactérias, protozoários ou qualquer corpo que seja estranho ao nosso organismo.

A resposta do sistema imunológico a um antígeno envolve duas etapas: reconhecimento e elaboração de uma reação dirigida ao elemento. Essas reações têm o objetivo de eliminar o corpo estranho do nosso organismo. Elas podem ser executadas de várias formas, que podem ser subdivididas em duas classes: resposta imune inata (não-adaptativa) e resposta adaptativa.

A resposta imune inata nasce com o indivíduo, tem como seus componentes principais as barreiras físicas e químicas, como pele, secreções, células fagocísticas e altas temperaturas. Essa é a linha inicial de defesa do organismo, tem ação rápida e sem memória. A resposta inata estimula as respostas imunes adquiridas.

A imunidade adquirida é desenvolvida quando o corpo é exposto a antígenos. Ela envolve basicamente dois tipos de linfócitos (glóbulos brancos): os linfócitos B ou células B e os linfócitos T ou células T, que são desenvolvidos de forma específica para cada tipo de antígeno.

As células B são responsáveis pela produção dos anticorpos, que nada mais são do que algumas proteínas capazes de defender o organismo de substâncias que foram identificadas como perigosas pelo sistema imunológico. Já as células T atacam os antígenos diretamente. Algumas dessas células se multiplicarão e terão o papel de memória do sistema imunológico, permitindo que quando ocorrer uma nova exposição ao mesmo antígeno, o organismo possa combatê-lo de forma mais rápida.

4.3.2 O Algoritmo

O ARIA tem como inspiração o sistema imunológico e tem como principais características, a capacidade de preservar a densidade dos dados através de seu raio adaptativo e determinar automaticamente o número de *clusters*. Essas características foram determinantes na escolha desse algoritmo para a metodologia proposta nesta dissertação.

No algoritmo os dados são representados por antígenos e os protótipos dos *clusters* pelos anticorpos. Pode-se dividir o ARIA em três partes principais:

- Maturação de afinidade: nessa etapa ocorrem as interações entre os antígenos e os anticorpos, que sofrem mutação para melhor adaptar-se ao antígenos.
- Expansão clonal: a rede cresce através da clonagem dos anticorpos mais estimulados.
- Supressão da rede: caso um anticorpo reconheça outro, um deles é eliminado da rede e a rede diminui.

O pseudo-código do ARIA é mostrado no Algoritmo 1. A Tab. 4.3 mostra a descrição dos parâmetros e símbolos utilizados no algoritmo. Os símbolos marcados com * são informados pelo usuário.

Algoritmo 1: Posicionamento dos protótipos do algoritmo ARIA.

1. Inicialização das variáveis;
 2. **para cada** *iteração* *l* **a gen faça**
 - 2.1. **para cada** *antígeno* **faça**
 - 2.1.1. Selecione o melhor anticorpo *Ab*;
 - 2.1.2. Mute *Ab* com taxa *mi*;
 - fim**
 - 2.2. Elimine os anticorpos que não foram estimulados;
 - 2.3. Clone os anticorpos que reconhecem antígenos localizados a uma distância maior que seu raio *R*;
 - 2.4. Calcule a densidade local para cada *Ab*;
 - 2.5. Calcule o limiar de supressão (raio) para cada *Ab*, fazendo $R = r(den_{max}/den(i))^{1/dim}$;
 - 2.6. Suprima os anticorpos que reconhecem outros anticorpos, dando prioridade de sobrevivência para aqueles com menor *R*;
 - 2.7. Faça $E = media(R)$;
 - 2.8. **se** *a geração atual é maior que gen/2* **então**
 - 2.8.1 Reduza *mi* ($min = mi * decay$);
 - fim**
 - fim**
-

No passo 1 do algoritmo, as variáveis são inicializadas e a população inicial de anticorpos e o valor de *R* são gerados aleatoriamente. Segundo os autores [63], o valor de *mi* deve ser iniciado com 1.

Símbolo	Descrição
R	Raio de cada anticorpo.
r^*	Menor raio possível.
mi	Taxa de mutação.
$decay^*$	Multiplicador usado para reduzir a taxa de mutação
E	Raio que define a vizinhança para estimar a densidade.
gen^*	Número de iterações.
dim	Dimensão dos dados de entrada.

Tab. 4.3: Descrição dos símbolos usados no pseudo-código.

A fase de maturação da afinidade começa no passo 2.1. Para cada interação entre um antígeno e os anticorpos, aquele com menor distância euclidiana é escolhido e sofre mutação, com taxa mi , na direção do antígeno. Aqueles anticorpos que não forem escolhidos são eliminados no passo 2.2.

A segunda fase, expansão clonal, está no passo 2.3 do algoritmo. Os anticorpos que foram escolhidos por antígenos que estão mais distantes que o raio R são clonados. Mesmo que essa condição seja satisfeita para mais de um antígeno, cada anticorpo deve sofrer apenas um processo de clonagem, mantendo assim a rede mais estável pelo processo de auto organização.

A densidade da vizinhança de cada anticorpo é calculada no passo 2.4. Esse é o valor de pontos de dados localizados na hiperesfera com centro no anticorpo e com raio E . A densidade é utilizada para atualizar o valor de R no passo 2.5. Aquele anticorpo que se localiza na região de maior densidade terá seu raio igual a r . Quando a densidade diminui, o raio aumenta. Tem-se também que o valor da densidade não é inversamente proporcional ao raio, e sim inversamente proporcional à raiz de ordem dim da densidade.

Após o crescimento controlado da rede de anticorpos pelo processo de clonagem, ela é reduzida na etapa de supressão da rede, no passo 2.6. Se a distância entre dois anticorpos for maior que o raio de um deles, aquele com maior raio deve ser eliminado. No passo 2.7 o raio da vizinhança E é atualizado.

Em seguida, a taxa de mutação também é atualizada, no passo 2.8. Ela permanece constante até $gen/2$ iterações, quando começa a decrescer, obrigando a rede a convergir. Após a convergência, a topologia da rede tem que ser definida. A técnica sugerida pelos autores é a *minimum spanning tree* (MST), em português, árvore geradora mínima, algo-

ritmo conhecido da teoria dos grafos. Retirando um arco da árvore geram-se subgrafos e essa é a estratégia utilizada para gerar *clusters*.

As seções a seguir oferecerão mais detalhes sobre os processos de mutação, clonagem e o critério de particionamento.

Mutação e Clonagem

Dado um anticorpo Ab e um antígeno Ag , a fórmula para efetuar a mutação pode ser vista em (4.19),

$$Ab' = Ab + mi * rand * (Ag - Ab) \quad (4.19)$$

onde Ab' é o anticorpo após a mutação, mi é a taxa de mutação e $rand$ é um número aleatório com distribuição uniforme entre 0 e 1.

O processo de clonagem usa a mesma equação. Cada clone é uma cópia do anticorpo original que sofreu mutação. Como muito antígenos podem escolher o mesmo anticorpo para ser clonado, este sofrerá mutação apenas na direção do primeiro antígeno que o selecionou.

Critério de Particionamento

Como foi citado anteriormente, a separação dos dados em *clusters* é realizada através da eliminação de um arco da MST construída a partir da rede de anticorpos. Cada subgrafo obtido corresponde a um *cluster*. Para escolher que arestas da árvore devem ser eliminadas, podem ser utilizadas inúmeras estratégias. A mais simples seria eliminar as maiores arestas, mas, segundo os autores, esta metodologia falha em relação a densidade dos dados.

O critério proposto por Zahn [64] mostrou-se muito mais efetivo na separação dos *clusters*: uma aresta é eliminada se ela for considerada grande em relação à sua vizinhança (calculando-se a média e o desvio padrão das arestas vizinhas). Os autores do ARIA [63] optaram por uma versão mais simplificada deste critério: Uma aresta é eliminada se ela é pelo menos n vezes maior que a menor de suas arestas vizinhas. A Fig. 4.6 ilustra como este critério é aplicado. Esse foi o critério de particionamento utilizado neste trabalho.

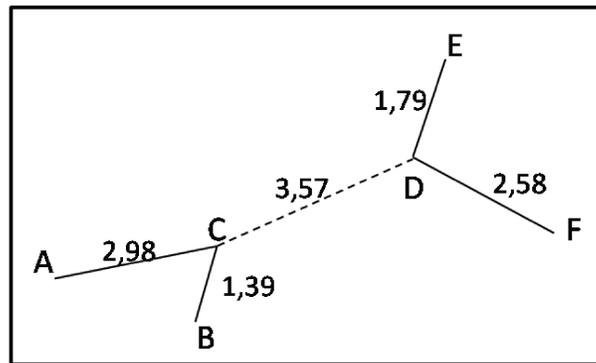


Fig. 4.6: Exemplo do uso do critério escolhido para a eliminação das arestas da MST. Ao examinar a aresta \overline{CD} , tem-se que a menor de suas arestas vizinhas, \overline{AC} , \overline{BC} , \overline{DE} e \overline{DF} , tem comprimento 1,39. Para $n=2$, \overline{CD} deve ser removida. Obtém-se então dois *clusters*: (A, B, C) e (D, E, F).

4.4 Comentários

Um pouco da teoria de reconhecimento de padrões foi apresentada neste capítulo, que foi dividido entre extração, seleção e *clusterização* de características. O algoritmo ARIA foi apresentado e foram exibidas também várias imagens das diversas etapas da metodologia apresentada neste trabalho.

Capítulo 5

Metodologia Proposta

O objetivo do presente trabalho é segmentar imagens com componentes texturais. Embora, a priori, pareça simples para o olho humano distinguir entre padrões de texturas diferentes, o reconhecimento de texturas de forma não-supervisionada ainda é uma tarefa difícil de ser executada na área de visão computacional.

A metodologia aqui apresentada busca segmentar imagens em tons de cinza, com características texturais, não se restringindo somente a um tipo de imagem ou textura.

Para um melhor entendimento da metodologia proposta neste trabalho, a Fig. 5.2 mostra um fluxograma, onde as várias etapas e técnicas podem ser visualizadas. As seções a seguir mostrarão detalhes de cada bloco usando como exemplo a imagem da Fig. 6.1.

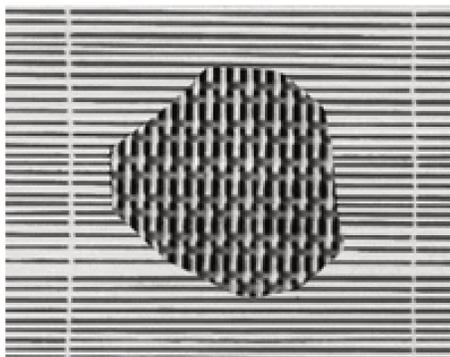


Fig. 5.1: Imagem exemplo.

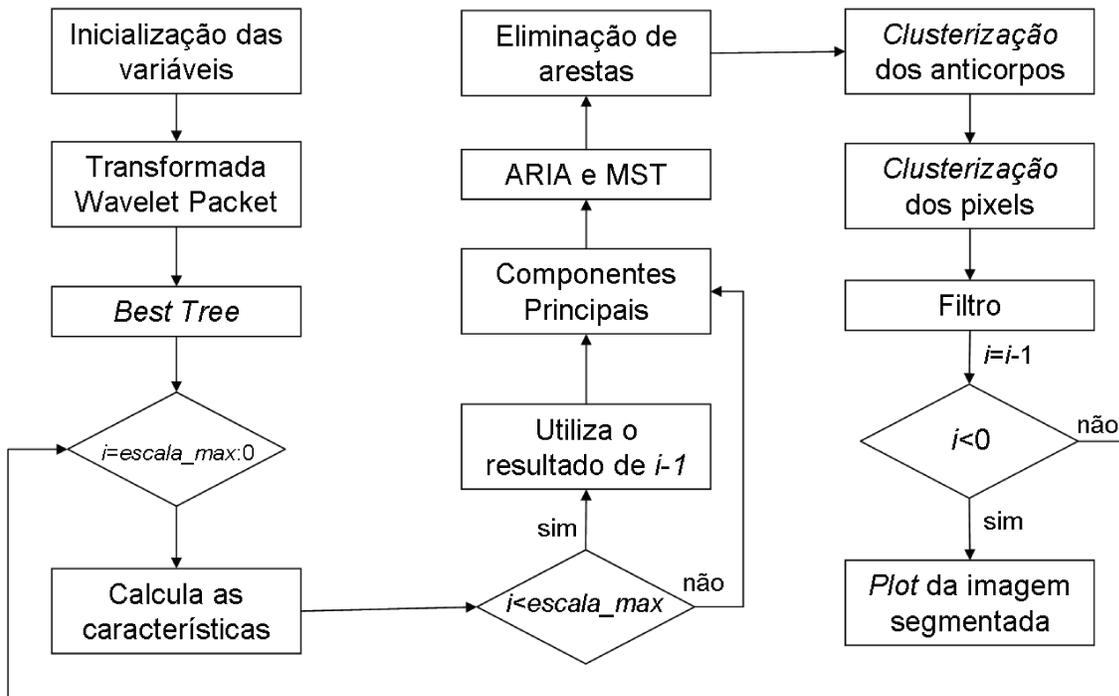


Fig. 5.2: Fluxograma da metodologia proposta

5.1 Transformada *Wavelet Packet*

Primeiramente, algumas variáveis precisam ser inicializadas:

O parâmetro '*media*' é utilizado para eliminar arestas inconsistentes da MST. Mais dois parâmetros podem variar: '*wavname*' e '*entropia*', mas apesar das várias opções disponíveis, optou-se por utilizar a *wavelet* de Haar e a entropia de Shannon, pela suas simplicidade e eficiência. Ainda nesta seção, ficará claro a função de cada um desses parâmetros.

O número de escalas utilizadas é determinado pelo próprio algoritmo, variando de 3 a 4, dependendo do tamanho da imagem a ser segmentada.

O número de características obtidas depende exclusivamente da escala de decomposição. A Tab. 5.1 mostra o número de subimagens geradas em cada escala. Nota-se que o número de subimagens é exponencial.

Após a inicialização das variáveis, são calculadas as decomposições *wavelet packet*, utilizando a função *wavelet* escolhida. Em nossos experimentos optou-se pela *wavelet* Haar. Tem-se que as decomposições *wavelet packet* podem ser representadas como uma

Nível de decomposição	Nº de subimagens
1	$4 = 4^1$
2	$16 = 4^2$
3	$32 = 4^3$
4	$64 = 4^4$
5	$256 = 4^5$
6	$1024 = 4^6$

Tab. 5.1: Quantidade de subimagens geradas pela decomposição *wavelet packet*.

árvore, como mostra a Fig. 5.3.

Nas Figs. 5.3 e 5.4, em cada nó há uma numeração do tipo (X, Y), onde X significa a escala de decomposição e o segundo número, Y, é o número do nó dentro da escala. As imagens com Y múltiplos de 4 são coeficientes de aproximação, as demais são coeficientes de detalhes.

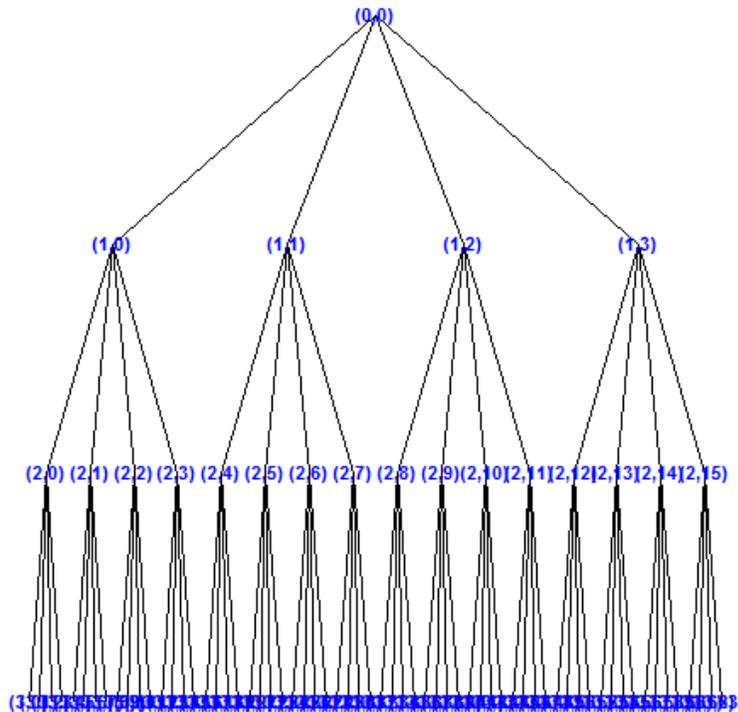


Fig. 5.3: *Wavelet packet Tree*.

Entretanto, como foi melhor detalhado no Cap. 3, algumas dessas decomposições não

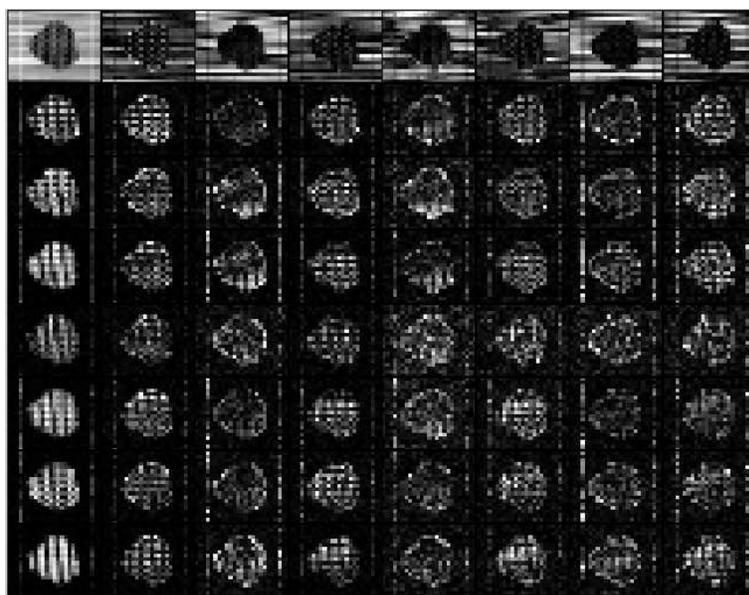


Fig. 5.5: Decomposição *wavelet packet* de nível 3.

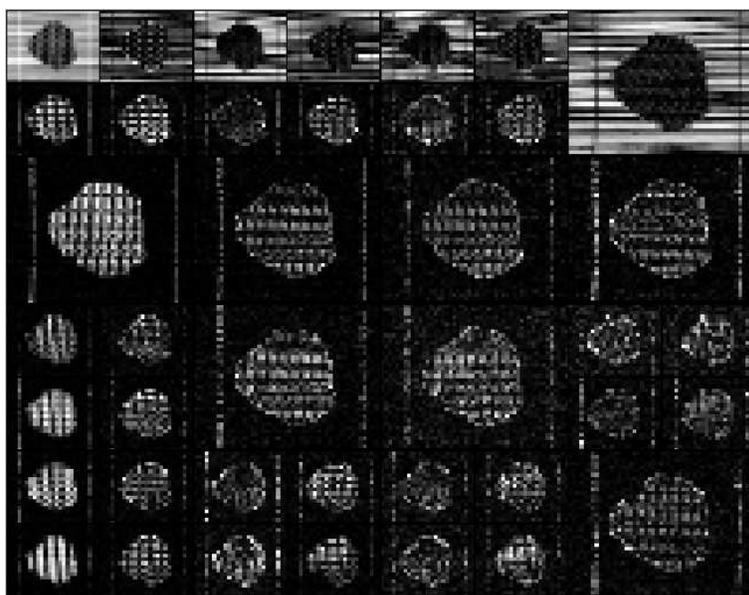


Fig. 5.6: *Best Tree* da decomposição da Fig. 5.5.

5.2 Características

Após a geração da *best tree*, o algoritmo inicia efetuando a segmentação do maior nível de decomposição da transformada *wavelet packet*.

Nas escalas maiores é mais fácil encontrar as diferentes texturas, entretanto a borda de separação entre elas fica muito comprometida. Por essa razão, há a transferência do resultado da segmentação de uma escala para outra. Quanto mais o algoritmo evolui, melhores serão as definições de bordas.

Após a construção da *best basis* são realizadas três operações simples para obtenção dos vetores de características da imagem em cada escala.

Cálculo da Energia Elevando-se o valor de cada *pixel* da decomposição ao quadrado, tem-se o valor da energia.

Normalização Apenas para manter um padrão nos valores da energia, independente da imagem, é realizada uma normalização, deslocando os valores obtidos no item anterior para o intervalo [0, 1].

Formação de regiões Sabe-se que as características texturais não estão presentes em apenas um *pixel*, e sim em uma vizinhança. Nesse ponto, a decomposição é dividida em várias pequenas regiões de tamanho variável de acordo com o tamanho da imagem.

Quando essas pequenas regiões são formadas, todos os *pixels* da região assumem o valor igual a média de todos eles.

Para poder aproveitar o resultado da segmentação da escala anterior na escala atual, mais uma característica é inserida nesse grupo. Para diminuir a forte interferência dessa segmentação nas características atuais, deve-se modificar o valor da média e da variância dos valores da segmentação para a mesma média e variância obtidas com as características de energia.

Por último, antes de gerar a MST, faz-se uso da análise por componentes principais, já explanada no capítulo 4. O objetivo dessa análise é diminuir a dimensionalidade das características e alterar o seu eixo, de forma que seja possível realizar a discriminação entre as texturas com um número reduzido de informação.

5.3 *Minimal Spanning Tree*

Após a seleção das características mais significativas, a *minimal spanning tree* é construída. Para estabelecer os nós da árvore, é utilizado o algoritmo imuno-inspirado ARIA,

explicado em detalhes no capítulo. 4. A Fig. 5.7 mostra diferentes etapas da convergência do algoritmo. Os eixos vertical e horizontal correspondem ao valor das características.

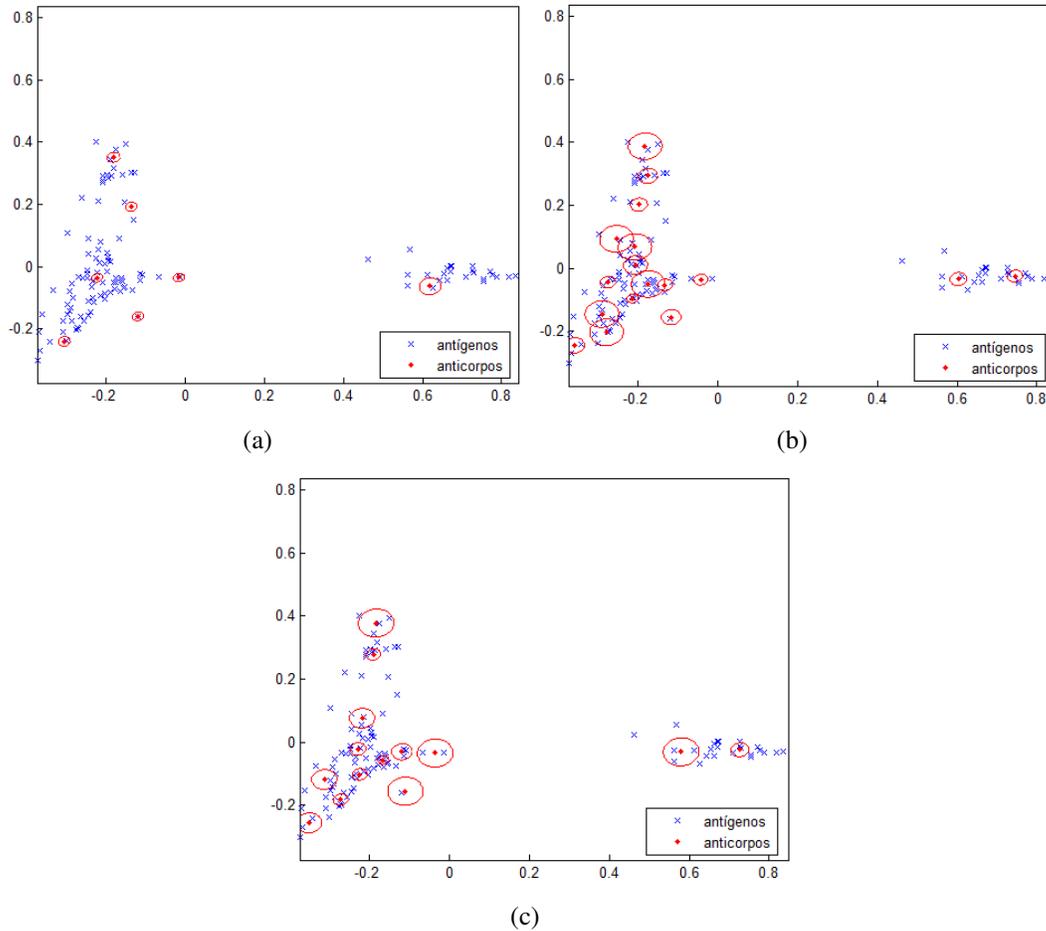


Fig. 5.7: (a) Primeira iteração. (b) Iteração 5. (c) Convergência.

Entre a primeira iteração, mostrada na Fig. 5.7.(a) e a convergência dos anticorpos, visualizada na Fig. 5.7.(c) pode-se observar a movimentação dos anticorpos e sua multiplicação para alcançar os antígenos (dados).

Após a geração da MST, mostrada na Fig. 5.8.(a), a variável '*media*' tem importância fundamental, pois é ela que define quais arestas da MST serão eliminadas, determinando os *clusters*. A MST "podada" pode ser vista da Fig. 5.8.(b).

Na escala de maior decomposição o número de *clusters* existentes é determinado, após essa primeira segmentação. O número de arestas a serem excluídas da árvore será igual ao

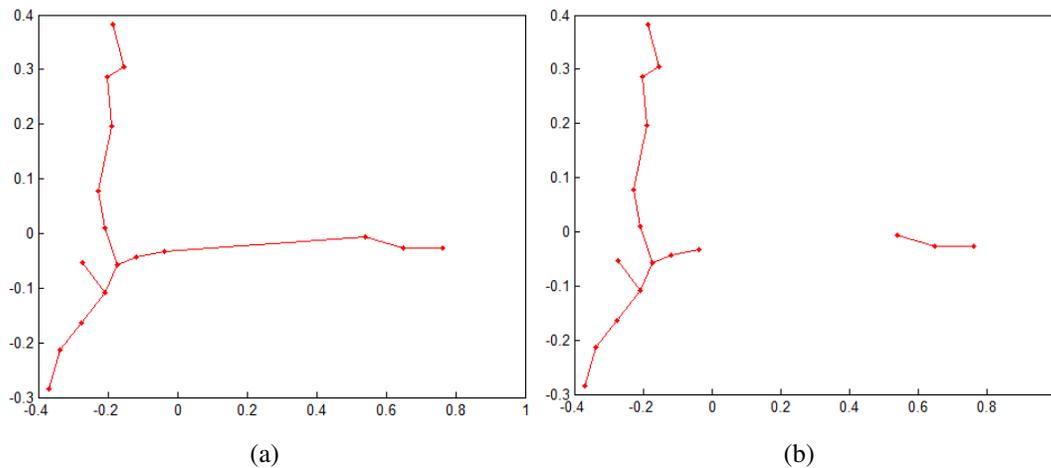


Fig. 5.8: (a) *Minimal spanning tree*. (b) *Minimal spanning tree* após eliminação de uma aresta.

número de *clusters* encontrados na primeira segmentação menos uma unidade.

Após o "podamento" da árvore, os anticorpos são separados em *clusters* e em seguida os *pixels* a eles ligados recebem os rótulos de segmentação. A Fig. 5.9 mostra um exemplo de rotulamento.

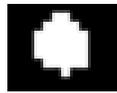


Fig. 5.9: Segmentação obtida após rotulamento.

5.4 Finalização

Após o rotulamento, é possível que alguns *pixels* sejam segmentados incorretamente. Para minimizar tal efeito utiliza-se um filtro da média de tamanho variável entre 5 e 9 *pixels*, dependendo do tamanho da imagem. Pode-se ver na Fig. 5.10 o efeito da aplicação do filtro na imagem da Fig. 5.9¹.

Terminado esse processo de segmentação para a maior escala de decomposição, o processo reinicia diminuindo a escala, ou seja, se aproximando da imagem original. A

¹As Figs. 5.9 e 5.10 estão em baixa resolução por se tratar de um resultado de segmentação obtido na escala 3 de decomposição. Resultando, no caso desta imagem, uma imagem de tamanho 20x25 *pixels*.



Fig. 5.10: Filtragem da segmentação obtida na Fig. 5.9.

Fig. 5.11 mostra o resultado da segmentação para cada escala.

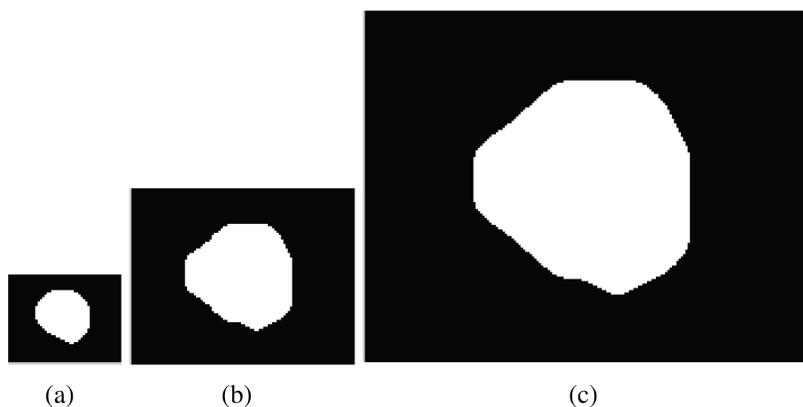


Fig. 5.11: (a) Segmentação obtida na 2ª escala de decomposição. (b) Segmentação obtida na 1ª escala de decomposição. (c) Segmentação obtida a partir da imagem original.

5.5 Comentários

Este capítulo descreveu, em detalhes, a metodologia aqui apresentada. Ela é composta basicamente de quatro etapas: a extração de características pela transformada *wavelet packet*, a redução da dimensionalidade dos dados pela análise de componentes principais, a clusterização, realizada pelo ARIA e, para finalizar, a filtragem do resultado obtido. Os resultados obtidos são exibidos no próximo capítulo.

Capítulo 6

Resultados

Neste capítulo, aplica-se a metodologia proposta apresentada do capítulo 5 em conjuntos de mosaicos de texturas extraídas do álbum de Brodatz [12], que são utilizados por vários trabalhos encontrados na literatura. Também serão analisados resultados obtidos em imagens naturais de animais e em diversas aplicações como em análise de mamografias.

As imagens que têm como base as texturas de Brodatz foram retiradas do álbum "*Photographic Album for Artists and Designers*" de Phil Brodatz, lançado em 1966. Esse álbum contém 112 fotos de texturas naturais, como cascas de árvores, pedras, grama, etc, e algumas texturas de materiais fabricados pelo homem, como tecidos e telas de aço. Várias combinações dessas texturas têm sido utilizadas por pesquisadores para avaliar algoritmos e comparar métodos.

A apresentação dos resultados está dividida em duas partes. Na seção 6.2 serão apresentados os resultados obtidos com mosaicos construídos a partir das texturas de Brodatz. Na seção 6.3 podem ser visualizados os resultados obtidos com imagens naturais de animais. A seção 6.4 mostra os resultados da aplicação da metodologia proposta em imagens médicas e na seção 6.5 em outras aplicações.

6.1 Trabalhos Anteriores

Na literatura é possível encontrar diversos trabalhos que utilizam as mais diversas técnicas com o objetivo de segmentação de texturas. Esta seção apresenta alguns dos trabalhos que serão utilizados para comparar os resultados obtidos pela metodologia proposta.

Em [1], os autores propõem um conjunto de filtros, denominado ICA (*Independent Component Analysis*). Esses filtros são similares aos filtros de Gabor, mas parecem ser mais precisos pois alguns filtros têm respostas complexas. Após a extração das características, os autores utilizam o algoritmo K-means [61] para efetuar a clusterização.

O *K-means* é muito utilizado para processos de clusterização, mas tem duas grandes desvantagens: pode convergir para um mínimo local da função custo, gerando um resultado não desejado e requer um conhecimento a priori do número de agrupamentos desejados.

Nunes e Conci, em seu trabalho [2], apresentam um novo coeficiente denominado CVE (Coeficiente de Variação Espacial), que considera a informação das bandas espectral, levando em conta qual cor a combinação de bandas gera e as posições espaciais dos *pixels*. Na estratégia proposta, é necessário que, para efetuar a etapa de treinamento, sejam selecionadas amostras de cada textura, tornando o processo supervisionado. Nesse caso, o *K-means* também foi utilizado como clusterizador.

Em [3], uma nova metodologia não supervisionada é proposta para segmentação de texturas. Esse trabalho utiliza métricas estatísticas nas vizinhanças da imagem. Apesar de vários parâmetros serem determinados automaticamente, o usuário precisa determinar o número de regiões a ser segmentado.

Em sua tese de doutorado [4], Pagamisse apresentou uma metodologia para discriminação de texturas baseada em transformada *wavelet*, onde foi desenvolvida uma representação *wavelet* invariante por translações, com quatro direções e, utilizando essa representação, desenvolve-se um método que agrega e combina as respostas da energia da transformada *wavelet* nas várias escalas e direções para obter a discriminação de texturas. Para efetuar o treinamento, é necessária uma amostra de cada textura.

Em 1999, foi publicado um estudo comparando abordagens de segmentação de texturas baseadas em filtragem no artigo de Randen e Husoy [65]. Os resultados por ele obtidos também serão utilizados para comparação neste capítulo.

Um das grandes dificuldades em realizar as comparações aqui feitas, está no fato de que muitos autores não disponibilizam em seus trabalhos as imagens originais utilizadas nem a porcentagem de erro/acerto obtidas. Portanto, alguns dos valores aqui apresentados são apenas estimativas dos erros dos resultados apresentados nos artigos acima descritos.

6.2 Segmentação de Texturas de Brodatz

As combinações de texturas aqui utilizadas foram extraídas de diversos artigos encontrados na literatura. Não houve qualquer pré-processamento nessas imagens.

6.2.1 Imagens com 2 Texturas

Os parâmetros utilizados para essas imagens foram: entre 1 e 3 características, com tamanho da região variando entre 2 e 4; e 'media' igual a 6. No caso de imagens com apenas 2 texturas, o valor da 'media' não é relevante, pois o algoritmo aqui descrito define 2 *clusters*, no mínimo. Esse parâmetro terá maior importância no caso de mais de 2 texturas.

A Fig. 6.1 mostra os resultados obtidos para a segmentação da imagem mostrada na Fig. 6.1.(a), formada pelas texturas D49 (fundo) e D53 (padrão interno). Além do resultado obtido pela metodologia proposta também podem ser visualizados os resultados obtidos em [1] e [2].

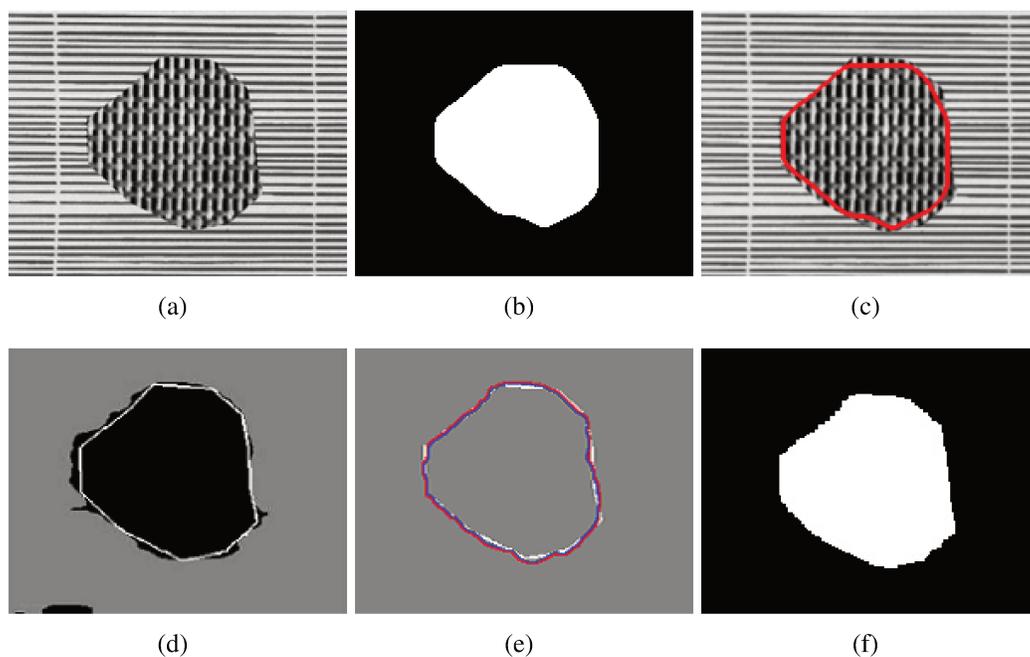


Fig. 6.1: (a) Imagem original. (b) Resultado obtido pela metodologia proposta. (c) Resultado obtido pela metodologia proposta sobreposto à imagem original. (d) Resultado obtido em [1] (contorno desejado em branco). (e) Resultado obtido em [2] (colorido) sobreposto ao contorno desejado (branco). (f) Imagem de referência.

As Figs. 6.2 e 6.3 mostram imagens utilizadas por [3]. Cada uma dessas combinações de texturas tem características peculiares. Não foi possível a obtenção das mesmas imagens utilizadas pelos autores. Foi realizada uma tentativa de reproduzi-las. Entretanto, como não são exatamente as mesmas imagens, os percentuais de erro aqui mostrados são apenas estimativas, portanto, devem ser analisados juntamente com a inspeção visual das imagens.

As texturas que formam a Fig. 6.2.(a), Grass (D9 - fundo) e Sand (D29 - padrão interno) são irregulares, isto é, apenas uma amostra não é capaz de caracterizar a textura como um todo (como ocorre na textura D53 presente na Fig. 6.1). Além disso, elas têm média e magnitude do gradiente similares, o que dificultaria a segmentação por métodos que utilizam esses dados estatísticos como parâmetros.

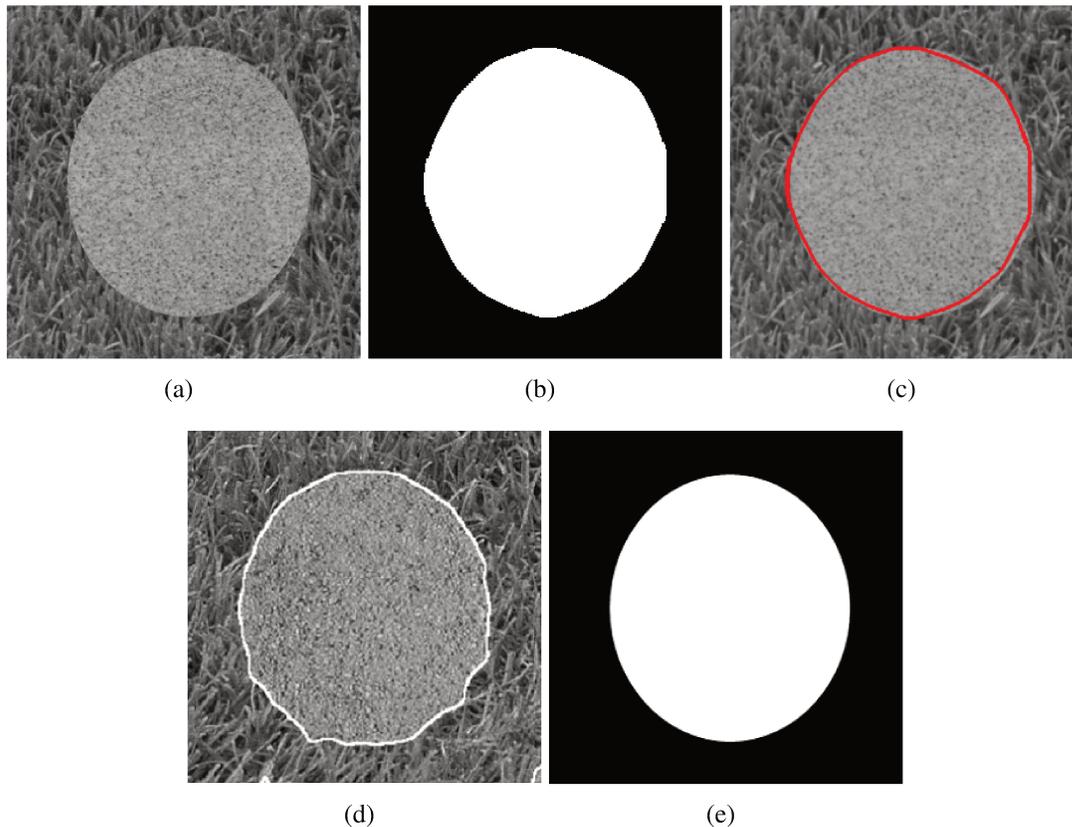


Fig. 6.2: (a) Imagem original. (b) Resultado obtido pela metodologia proposta. (c) Resultado obtido pela metodologia proposta sobreposto à imagem original. (d) Resultado obtido em [3]. (e) Imagem de referência.

Uma outra dificuldade inerente à segmentação de texturas ocorre quando uma mesma textura está em dois objetos distintos, como pode ser visto na Fig. 6.3.(a), formada pelas texturas Grass (D9 - fundo) e Straw (D15 - padrão interno). Métodos baseados em contorno são suscetíveis a separar esses dois objetos.

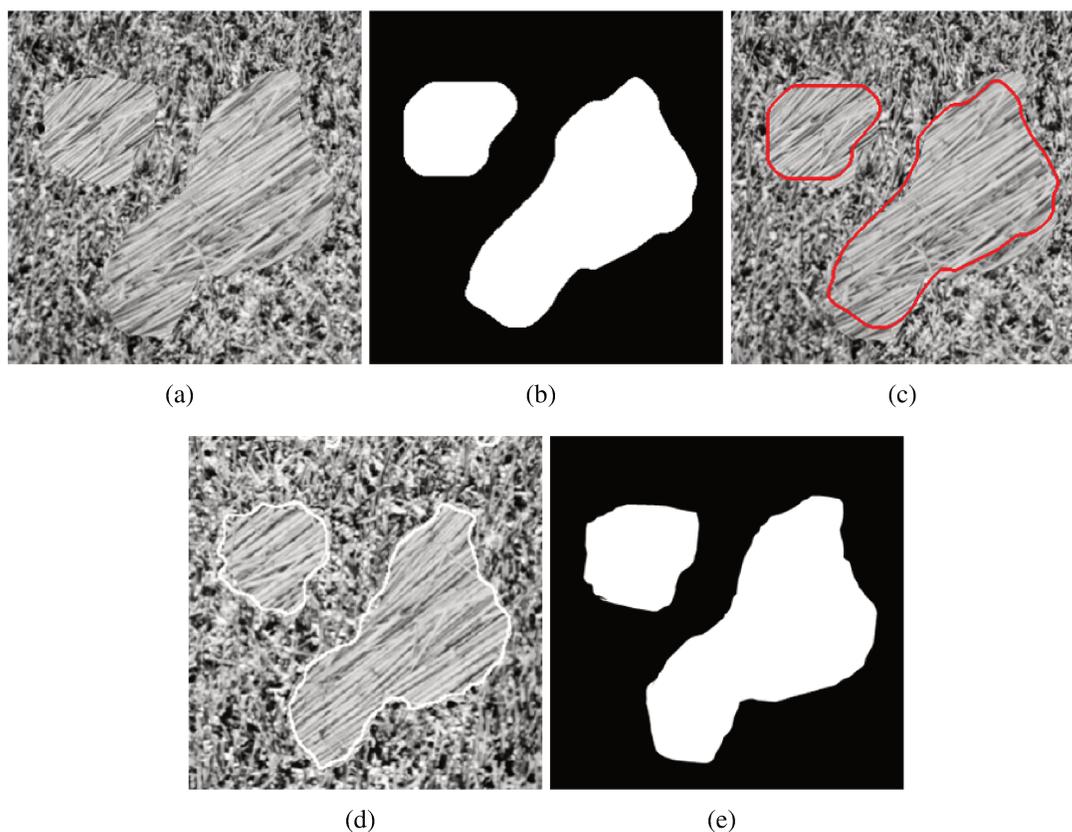


Fig. 6.3: (a) Imagem original. (b) Resultado obtido pela metodologia proposta. (c) Resultado obtido pela metodologia proposta sobreposto à imagem original. (d) Resultado obtido em [3]. (e) Imagem de referência.

A Tab. 6.1, a seguir, mostra a comparação dos percentuais de erro obtidos para imagens com 2 texturas de Brodatz. Embora a metodologia aqui apresentada não tenha tido o melhor resultado para a imagem D9D15, a diferença entre o percentual de erro é muito pequena. Os resultados obtidos nas outras imagens demonstram que o método aqui descrito é capaz de diferenciar pares de texturas homogêneas, irregulares e com características estatísticas semelhantes.

A imagens das Figs. 6.5.(a), 6.6.(a), 6.7.(a) foram utilizadas por [4] e por [65]. Elas

Método	D49D53	D9D29	D9D15
Metodologia Proposta	3.19	1.23	6.02

Tab. 6.1: Percentual de erro para imagens com 2 texturas de Brodatz.

apresentam uma junção entre 2 texturas de Brodatz. O primeiro par de texturas é caracterizado pela repetitividade, o segundo se diferencia principalmente pela escala e o terceiro pela orientação das texturas.

Pode-se observar que a metodologia proposta conseguiu acertar 100% da borda da imagem em 6.5.(c) e 6.7.(c), com erro de 1.4% para 6.6.(c). A imagem de referência pode ser vista na Fig. 6.4.

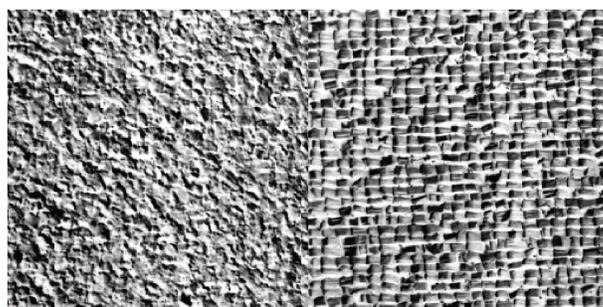
A Tab. 6.2, obtida em [4], mostra a porcentagem de erro para diversos métodos de segmentação, onde a metodologia proposta se destaca, alcançando a média de erro mais baixa. Esse fato deve-se principalmente à divisão da imagem em pequenas regiões para clusterização.



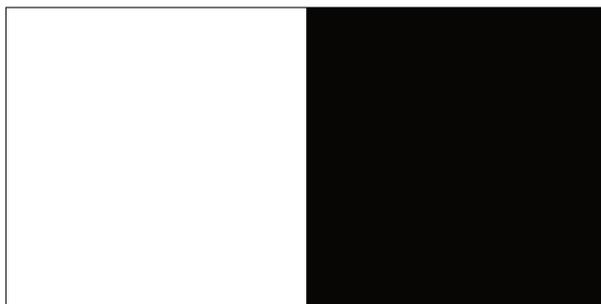
Fig. 6.4: Imagem de referência para as Figs. 6.5.(a), 6.6.(a), 6.7.(a)

Método	D04D84	D05D92	D12D17	Média
Filtros Anel/Wedge [66]	3.0	18.4	4.5	8.63
Banco de filtros diádicos de Gabor [67]	3.6	11.3	2.0	5.63
DCT [68]	6.4	2.5	2.2	3.7
Co-ocorrência [69]	1.9	3.3	4.8	3.33
AR [70]	11.3	3.0	1.9	5.4
Daubechies 10 <i>full rate</i>	4.4	5.1	1.2	3.57
f16b <i>full rate</i> [71]	8.1	8.2	0.8	5.7
F-2-1 <i>smpl. full rate</i> [71]	7.2	7.4	2.3	5.63
F-2-1-09. <i>full rate</i> [71]	6.7	6.0	2.1	4.93
<i>Eigenfilter</i> [72]	4.0	4.7	3.7	4.1
JU [73]	0.7	5.1	2.1	2.6
JF [73]	0.7	5.1	2.1	2.6
<i>Optimal Gabor filter</i> (melhor) [74]	5.3	10.8	3.3	6.5
4,6,10 <i>filter opt. Gabor filter bank</i> (melhor) [75]	3.8	12.8	3.8	6.8
Diádicas 2 direções [4]	0.8	2.6	2.4	1.9
Diádicas 4 direções [4]	0.6	1.1	1.1	0.9
Metodologia Proposta	0	1.4	0	0.46

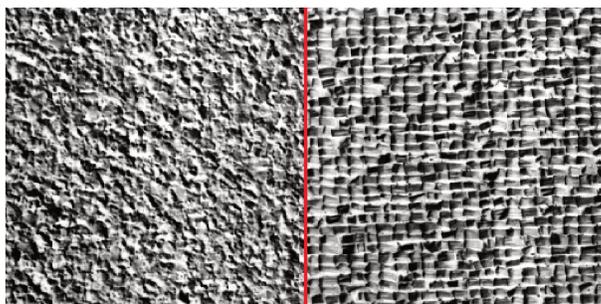
Tab. 6.2: Porcentagem de erro para diversos métodos de segmentação.



(a)



(b)

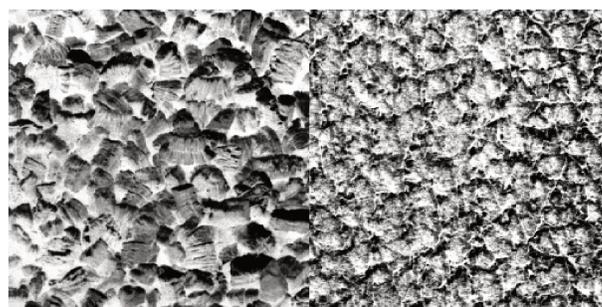


(c)

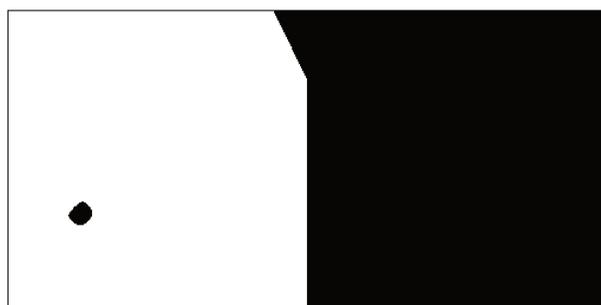


(d)

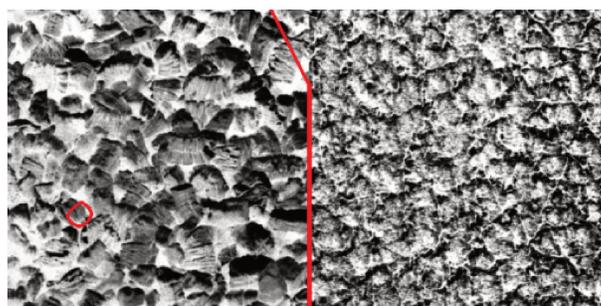
Fig. 6.5: (a) Imagem original. (b) Resultado obtido pela metodologia proposta. (c) Resultado obtido pela metodologia proposta sobreposto à imagem original. (d) Resultado obtido em [4].



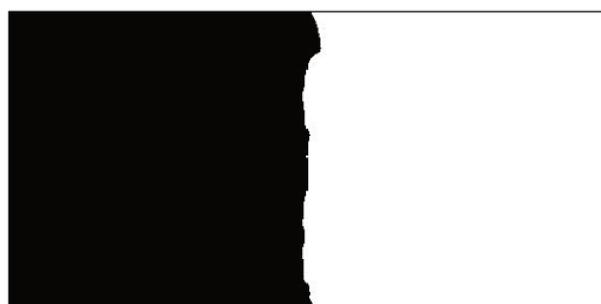
(a)



(b)

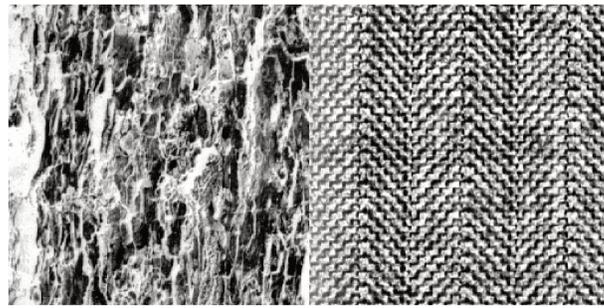


(c)

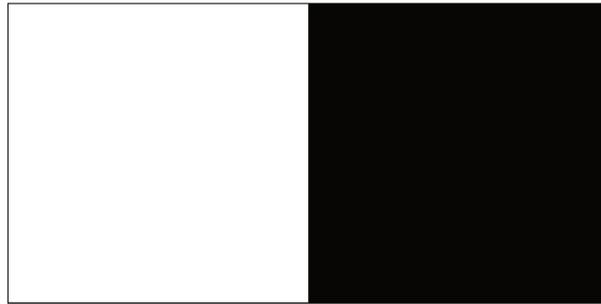


(d)

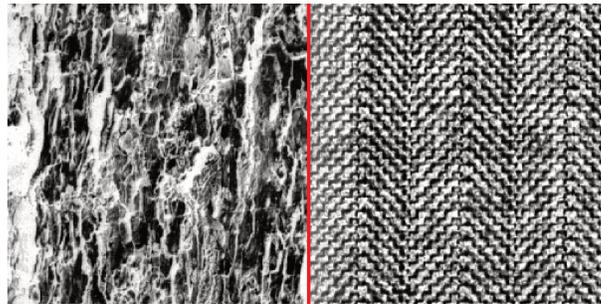
Fig. 6.6: (a) Imagem original. (b) Resultado obtido pela metodologia proposta. (c) Resultado obtido pela metodologia proposta sobreposto à imagem original. (d) Resultado obtido em [4].



(a)



(b)



(c)



(d)

Fig. 6.7: (a) Imagem original. (b) Resultado obtido pela metodologia proposta. (c) Resultado obtido pela metodologia proposta sobreposto à imagem original. (d) Resultado obtido em [4].

6.2.2 Imagens com mais de 2 Texturas

A seguir serão mostrados os resultados obtidos para segmentação de mais de 2 texturas de Brodatz. Em muitos artigos são apresentados resultados para mosaicos com 5 e até 10 texturas, mas como é muito improvável que ocorra uma situação real em que se faça necessário discriminar tantas texturas, preferiu-se, neste trabalho, dar maior ênfase à aplicações reais do que a um número maior de texturas numa mesma imagem.

Para as imagens desta seção, os parâmetros variam da seguinte forma: 2 a 3 características, tamanho da região entre 2 e 5 e 'media' entre 0.5 e 1.5.

A Fig. 6.8.(a) mostra uma imagem composta pela texturas D55 (fundo), D77 (polígono) e D84 (quadrado).

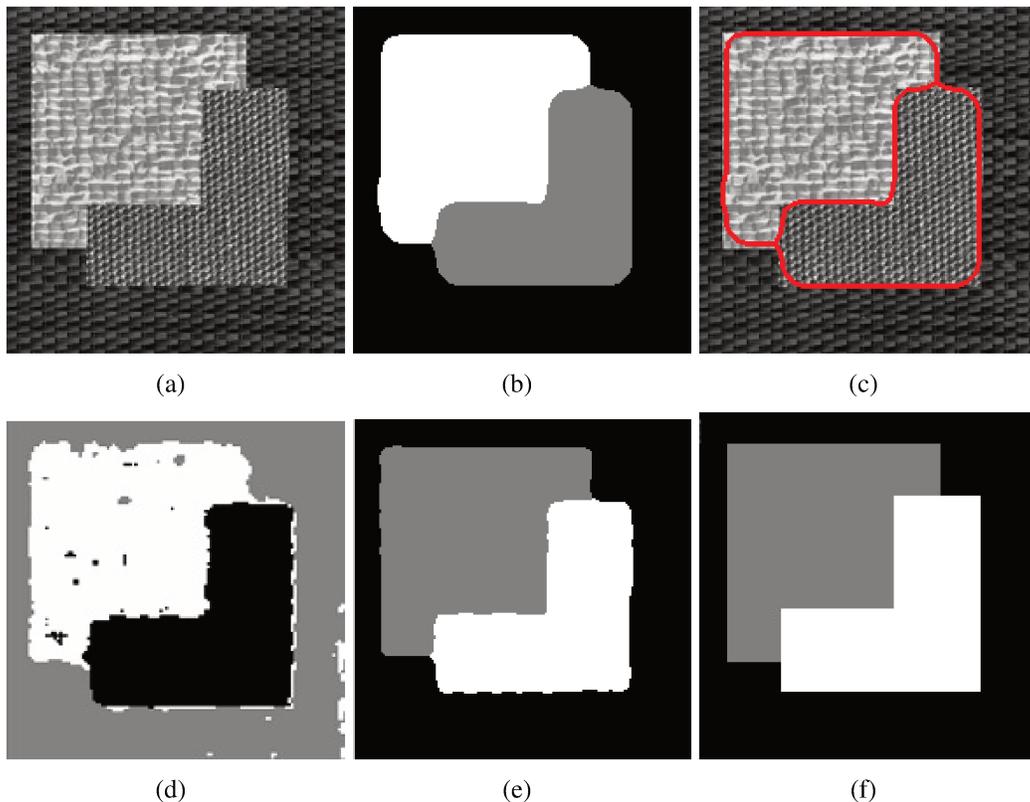


Fig. 6.8: (a) Imagem original. (b) Resultado obtido pela metodologia proposta. (c) Resultado obtido pela metodologia proposta sobreposto à imagem original. (d) Resultado obtido em [5]. (e) Resultado obtido em [2]. (f) Imagem de referência.

As duas imagens subsequentes, mostradas na Fig. 6.9 com 3 e 4 texturas distintas,

tiveram como base imagens mostradas em [3]. Como não foi possível reproduzi-las de forma razoável, optou-se por não comparar os resultados obtidos.

As imagens são formadas pelas texturas: D53 (fundo), D35 (semi-circunferência inferior), D93 (círculo) e D66 (semi-circunferência superior). A Tab. 6.3 mostra o percentual de erro obtido.

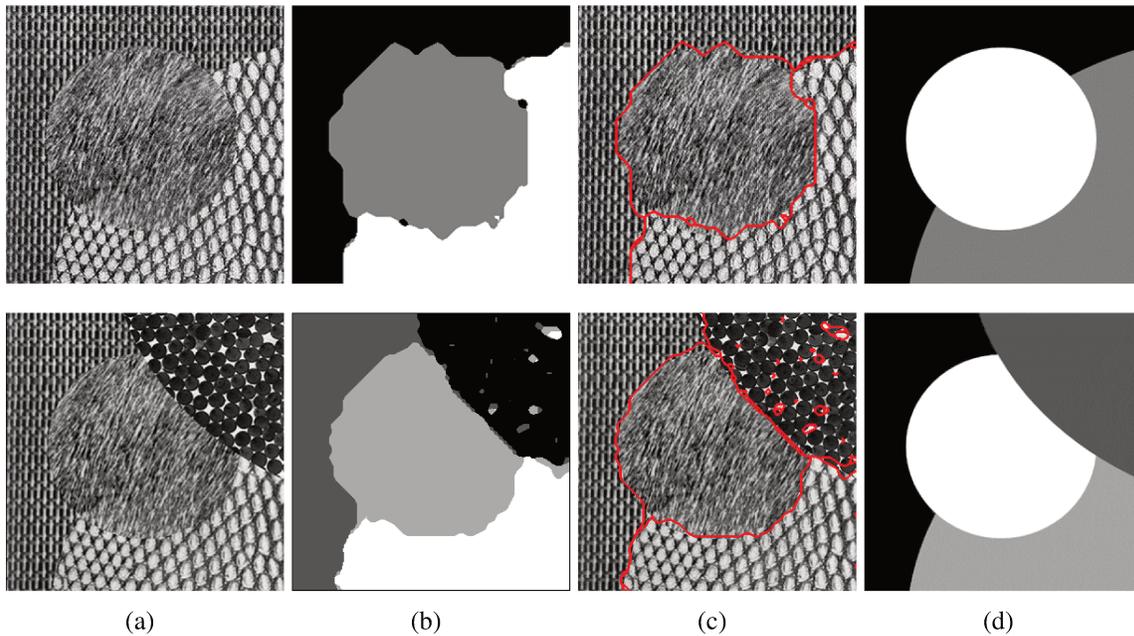


Fig. 6.9: (a) Imagem original. (b) Resultado obtido pela metodologia proposta. (c) Resultado obtido pela metodologia proposta sobreposto à imagem original. (d) Imagem de referência.

Método	D55D77D84	D53D35D93	D53D35D93D66
Metodologia Proposta	3.55	4.04	5.62

Tab. 6.3: Percentual de erro para imagens com mais de 2 texturas de Brodatz.

6.3 Imagens Naturais

É comum os trabalhos de segmentação de texturas mais atuais, mostrarem resultados de segmentação de animais em seu *habitat*. Muitos animais como: zebras, leopardos, tigres, girafas, apresentam em seu pêlo desenho que podem ser vistos como texturas.

A maior dificuldade em realizar essa tarefa consiste no fato de que as listras ou "manchas", têm tamanho, escala e direção diferenciadas.

A seguir são mostrados os resultados obtidos pela metodologia proposta para a segmentação de animais.

Para a imagem da Fig. 6.10.(a) foi feito um pré-processamento com filtro gaussiano de tamanho 25x25. O tamanho da região utilizado foi 5, sendo a '*media*' igual a 4.

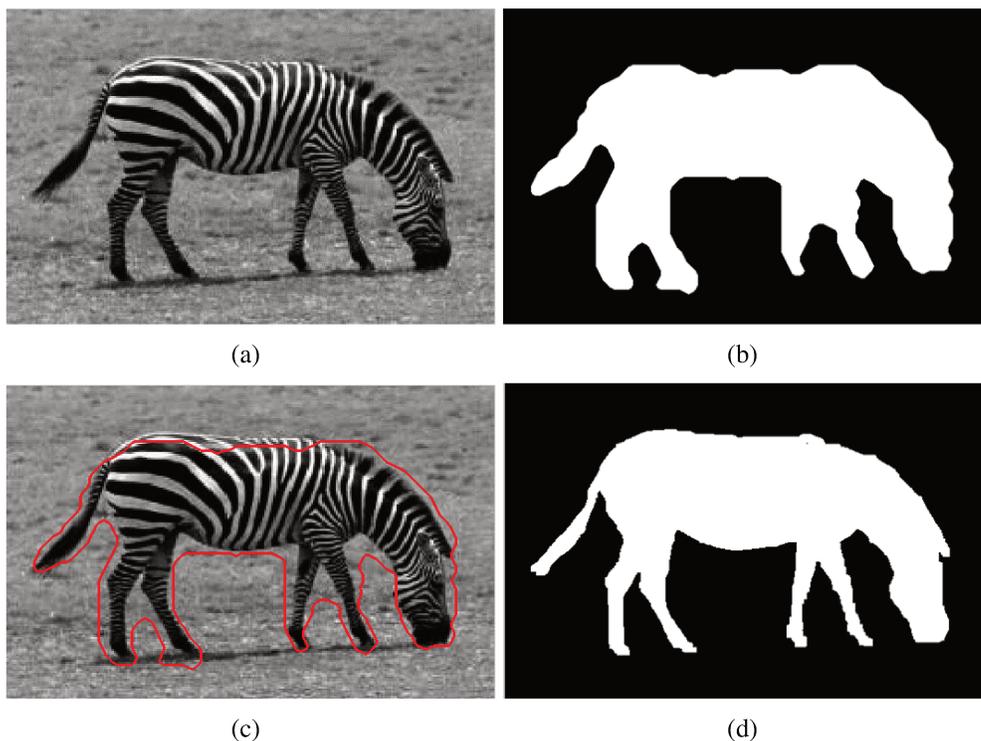


Fig. 6.10: (a) Imagem original. (b) Resultado obtido pela metodologia proposta. (c) Resultado obtido pela metodologia proposta sobreposto à imagem original. (d) Imagem de referência.

Já a imagem da Fig. 6.11.(a) não exigiu nenhuma espécie de pré-processamento. O tamanho da região utilizado foi 4, sendo a '*media*' igual a 4.

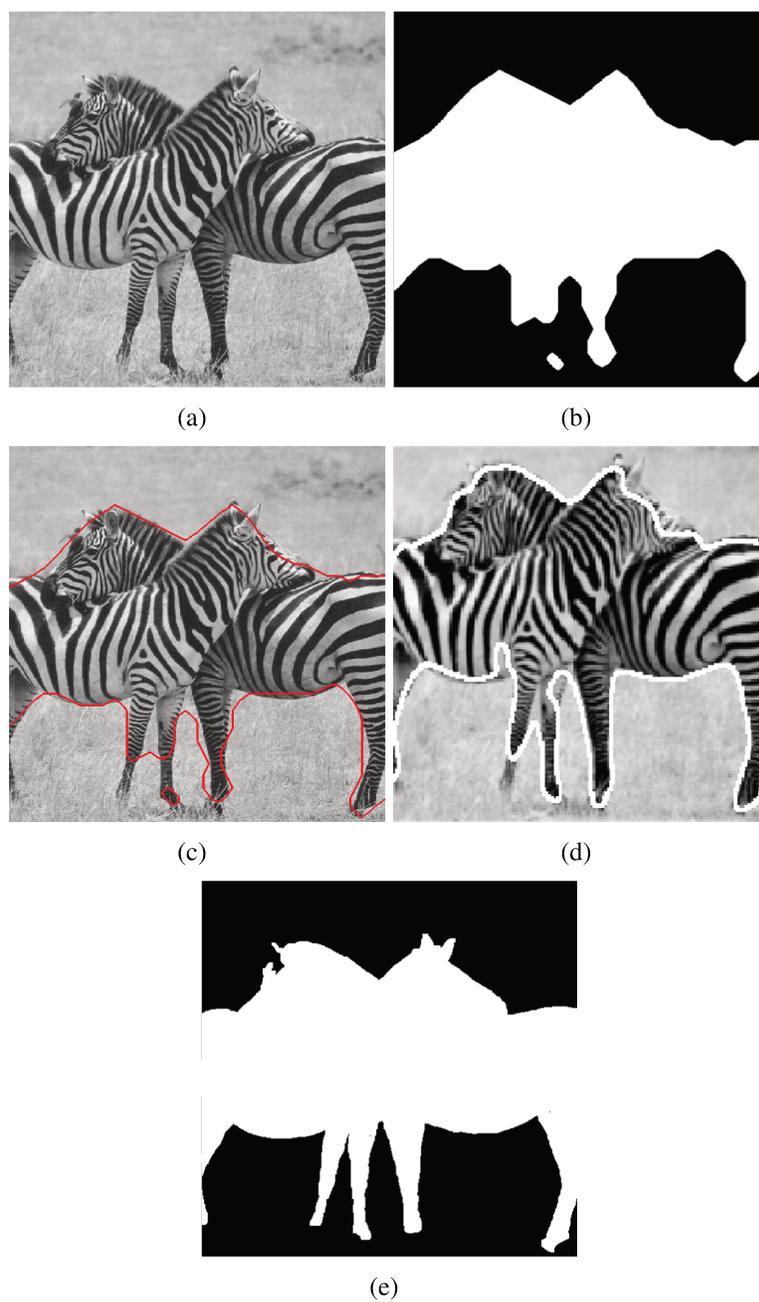


Fig. 6.11: (a) Imagem original. (b) Resultado obtido pela metodologia proposta. (c) Resultado obtido pela metodologia proposta sobreposto à imagem original. (d) Resultado obtido em [3]. (e) Imagem de referência.

Na Fig. 6.12.(a) pode-se ver mais uma imagem com uma zebra. O tamanho da região

utilizado foi 4, sendo a 'media' igual a 8.

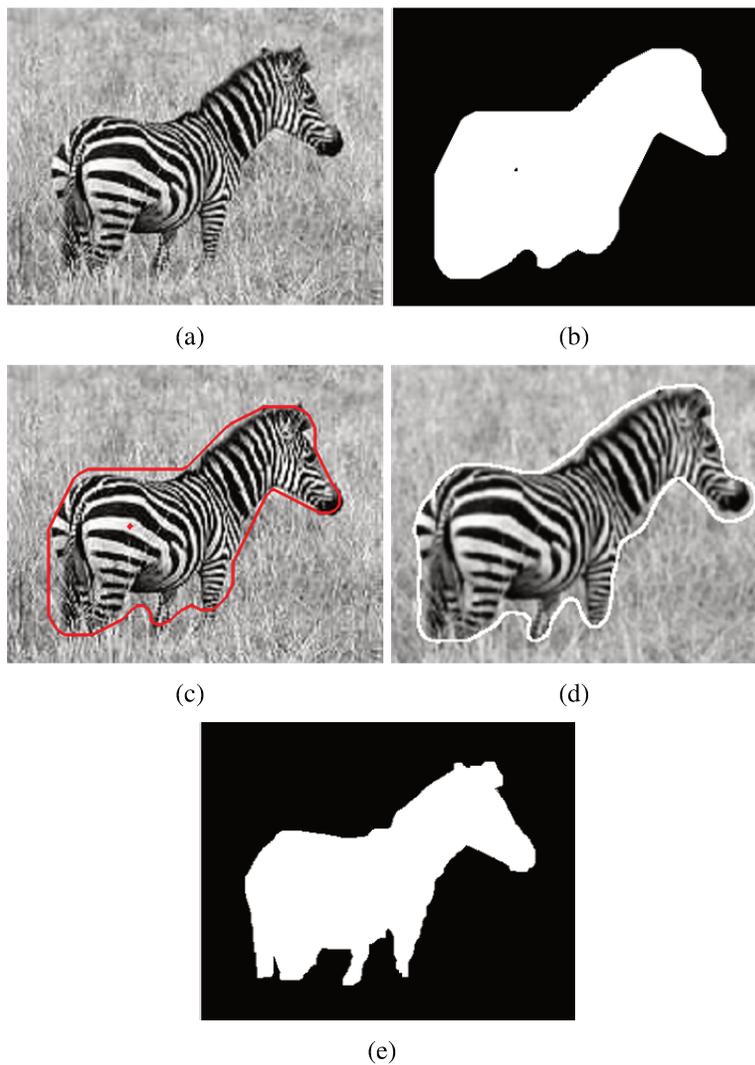


Fig. 6.12: (a) Imagem original. (b) Resultado obtido pela metodologia proposta. (c) Resultado obtido pela metodologia proposta sobreposto à imagem original. (d) Resultado obtido em [3]. (e) Imagem de referência.

Pode-se observar nas imagens com zebras, que a maior dificuldade em localizar as texturas, consiste no fato de que as listras mudam de orientação dependendo da parte do corpo do animal. Tem-se ainda que no rabo, orelhas e focinho a textura é totalmente diferente do restante do corpo. Entretanto, a metodologia proposta os classifica como parte das zebras, já que são mais similares ao animal do que ao fundo.

As Figs. 6.13.(a) e 6.14.(a) mostram imagens de um tigre e de um leopardo, respectivamente. Para segmentá-las foram utilizados tamanho da região 4 e 'media' igual a 8. Apesar da imagem do tigre estar colorida, ela foi transformada para tons de cinza antes de ser processada.

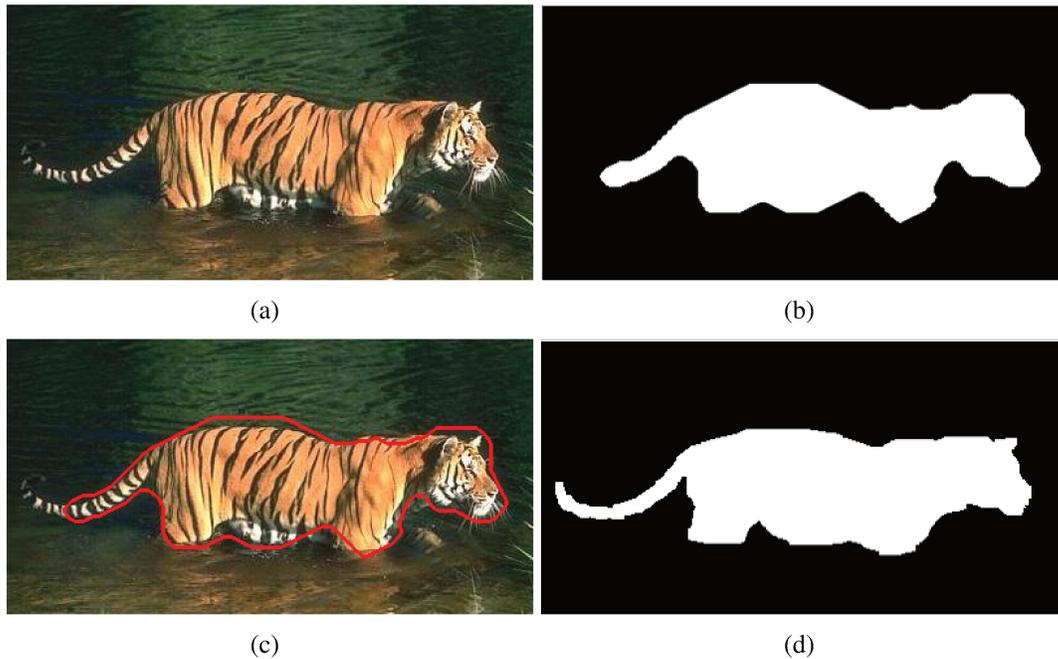


Fig. 6.13: (a) Imagem original. (b) Resultado obtido pela metodologia proposta. (c) Resultado obtido pela metodologia proposta sobreposto à imagem original. (d) Imagem de referência.

Na imagem do tigre encontram-se as mesmas dificuldades de mudança de orientação das listras, adicionando-se ainda as variações acentuadas nas cores, tanto do animal quanto do fundo. Já na imagem do leopardo, os problemas estão no focinho, nas orelhas e nas manchas, que mudam de tamanho. Nos dois casos, os maiores erros da metodologia proposta estão nas bordas. A Tab. 6.4 mostra o percentual de erro obtido.

Método	zebra1	zebra2	zebra3	Tiger2	leopard1
Metodologia Proposta	9.32	4.67	6.79	4.56	4.87

Tab. 6.4: Percentual de erro para imagens naturais.

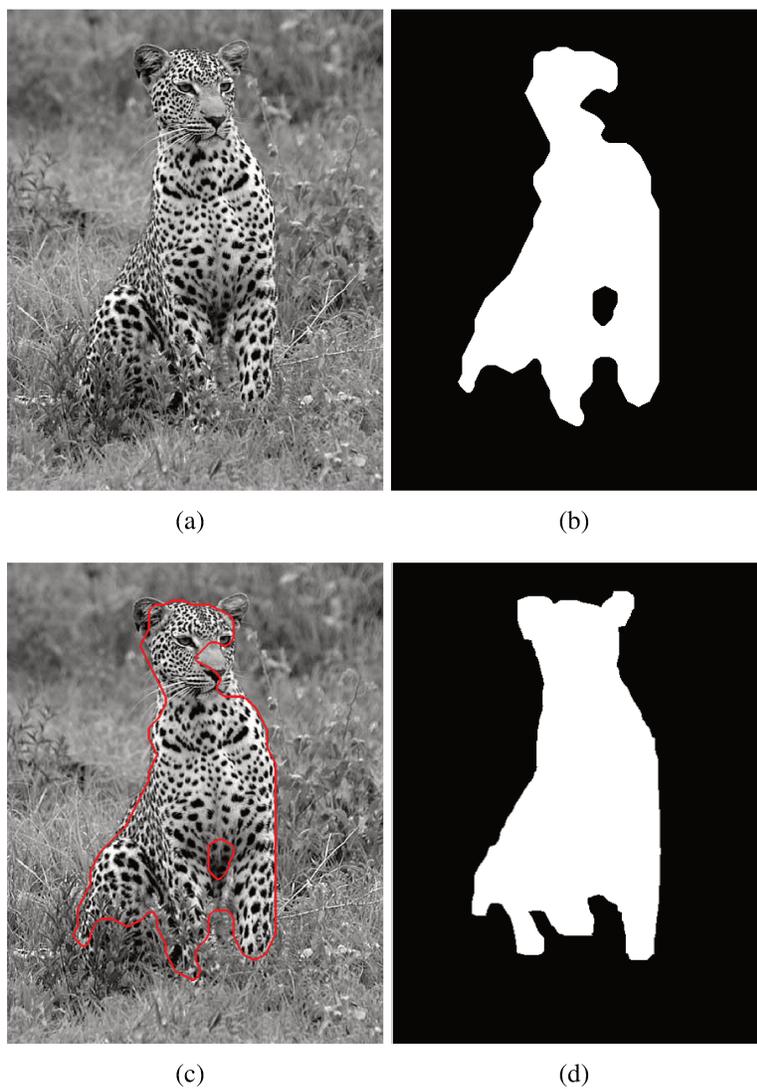


Fig. 6.14: (a) Imagem original. (b) Resultado obtido pela metodologia proposta. (c) Resultado obtido pela metodologia proposta sobreposto à imagem original. (d) Imagem de referência.

6.4 Imagens Médicas

Esta seção mostra a aplicação da metodologia proposta em imagens médicas que contenham informação de textura. O objetivo aqui não é competir com algoritmos especializados para realizar a segmentação de cada tipo de imagem, e sim mostrar que a metodologia desenvolvida neste trabalho tem ampla área de aplicação.

A Fig. 6.15 mostra imagens de mamografias. O objetivo é segmentar a área de interesse investigativo de doenças. O tamanho da região utilizado foi 7, sendo a '*media*' igual a 2, foram utilizadas 3 características.

Na primeira coluna, tem-se as imagens originais, conhecidas na literatura como 'mdb016' e 'mdb003'. Na segunda e na terceira colunas, pode ser visualizada a segmentação obtida pela metodologia proposta. A quarta coluna mostra a imagem de referência para a segmentação obtida em [32].

Pode-se visualizar que, nas mamografias, as partes do corpo mais próximas à pele são difíceis de ver, inclusive a olho nu. Existem alguns algoritmos que foram construídos para melhorar o contraste deste tipo de imagem [32], entretanto, neste trabalho não foi realizado nenhum pré-processamento nessas imagens. A Tab. 6.6 mostra o percentual de erro obtido pela metodologia proposta para as imagens da Fig. 6.15.

Método	mdb016	mdb003
Metodologia Proposta	6.63	12.34

Tab. 6.5: Percentual de erro para imagens médicas.

A Fig. 6.16 mostra imagens de melanomas (câncer de pele)¹. As segunda e terceira colunas mostram a segmentação obtida pela metodologia proposta. Os parâmetros utilizados foram: tamanho da região 4 e a '*media*' igual a 4, com 2 características.

Os estudos sobre detecção de melanomas via processamento digital de imagens ainda são recentes e não estão tão divulgados quanto os de mamografias, entretanto, os poucos estudos existentes contam com o apoio de especialistas para realizar a segmentação das imagens. A metodologia aqui apresentada se propõe também a ajudar nessa etapa do processamento. Não tendo a intenção, porém, de substituir o visor experiente do especialista, mas sim auxiliá-lo num possível diagnóstico.

¹As imagens originais foram obtidas em <http://bryanking.net/warning-signs-of-malignant-melanoma/> e de <http://www.daviddarling.info/encyclopedia/M/melanoma.html> em 02/12/2009

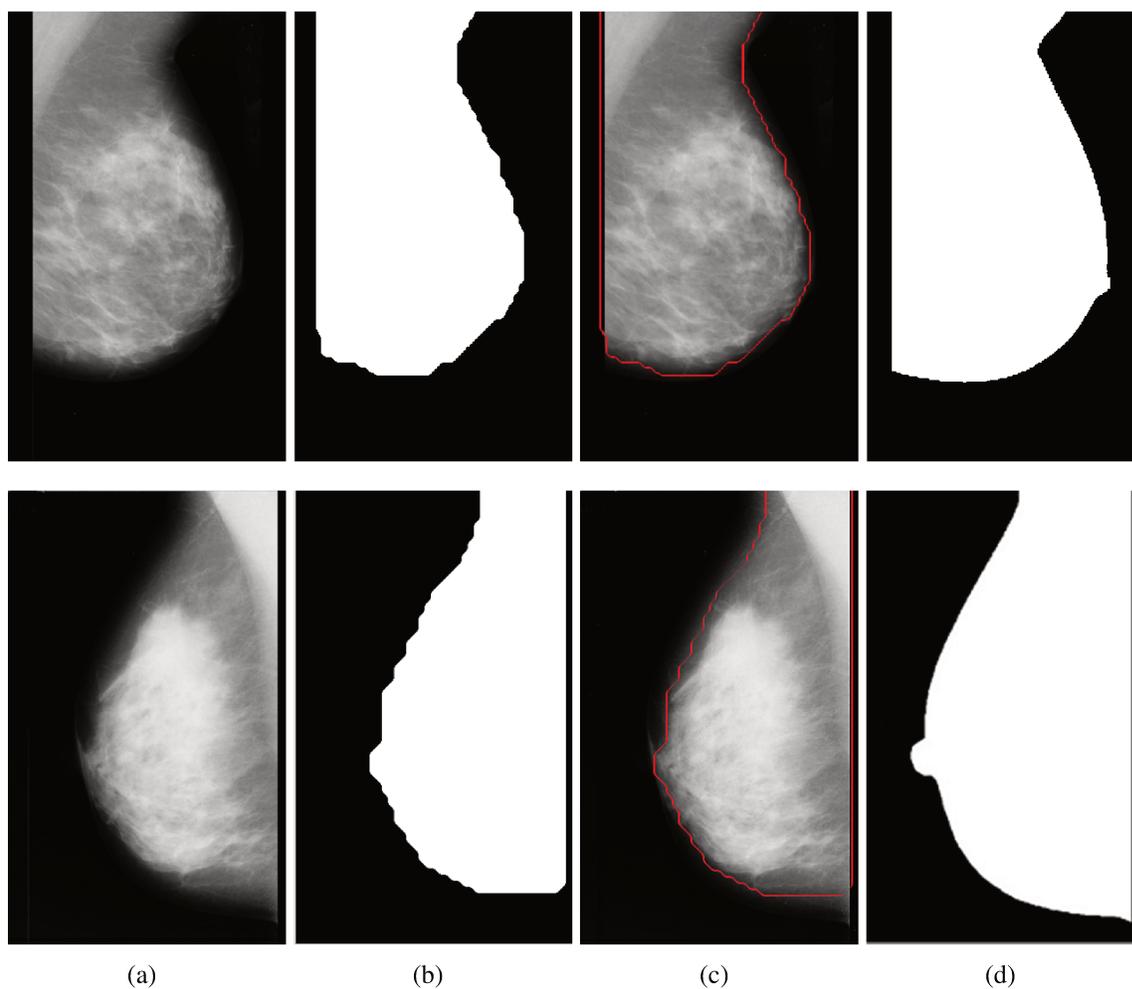


Fig. 6.15: (a) Imagem original. (b) Resultado obtido pela metodologia proposta. (c) Resultado obtido pela metodologia proposta sobreposto à imagem original. (d) Imagem de referência.

Existem muitas divergências entre os especialistas sobre a segmentação desse tipo de imagens, pois ainda não há um consenso sobre onde as bordas da lesão devem ser marcadas, por isso, nesse caso, não foi feita nenhuma comparação dos erros obtidos pela metodologia proposta.

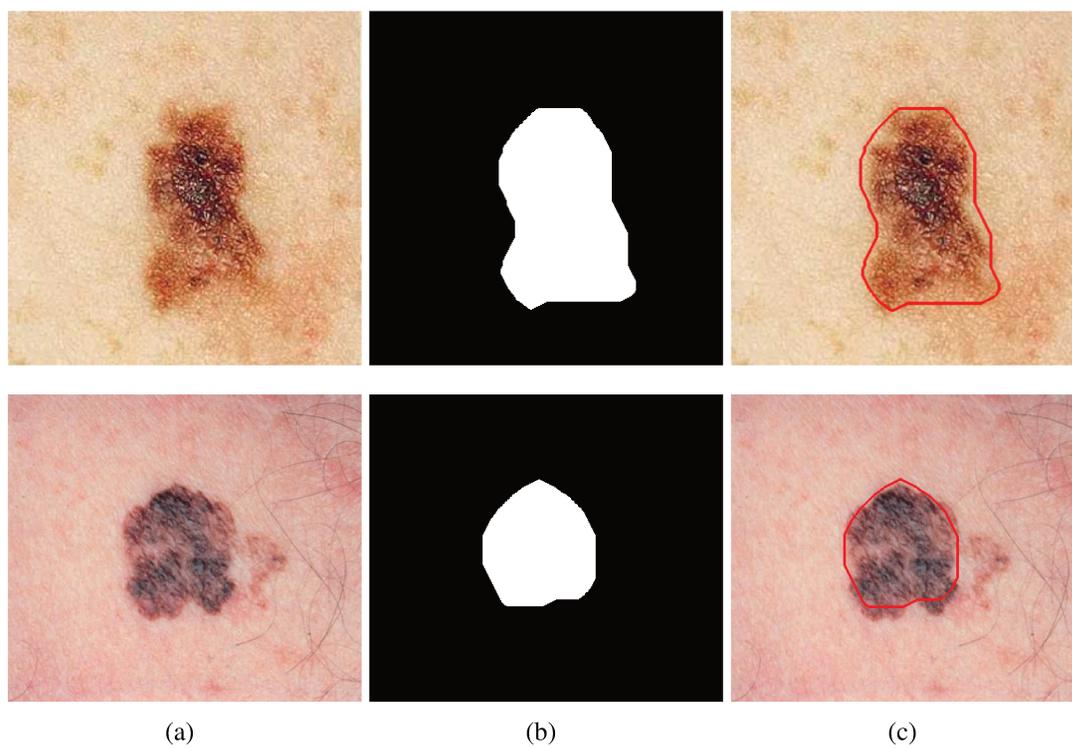


Fig. 6.16: (a) Imagem original. (b) Resultado obtido pela metodologia proposta. (c) Resultado obtido pela metodologia proposta sobreposto à imagem original.

6.5 Outras Aplicações

Dentre as outras aplicações que utilizam segmentação de imagens de textura, foram realizados teste em duas: imagens de satélite de derramamento de óleo no mar e amostras de tecidos.

A Fig. 6.17 mostra uma imagem² de derramamento de óleo de um navio petroleiro na costa espanhola. O processamento de imagens de satélite é bem comum, por se tratar de uma área de bastante interesse para monitoramento não só de derramamentos de óleo, mas também de níveis de desmatamento e de desertificação. Para aplicar a metodologia aqui proposta, foram utilizados apenas recortes da imagem original.

Grande parte das imagens de satélite utilizadas em processamento de imagens é obtida via Radares de Abertura Sintética (SAR). A qualidade radiométrica do dado SAR é afetada por fatores do instrumento e também pela geometria de iluminação. Sendo que o ruído *speckle* e o efeito do padrão da antena são as duas principais causas de distorções que prejudicam a interpretação das imagens de radar.

O *speckle* é um ruído multiplicativo proporcional a intensidade do sinal recebido. O efeito visual deste ruído gera uma textura granulosa que dificulta a interpretação das imagens de radar. Existem dois métodos para se diminuir o ruído *speckle*: a filtragem e o processamento *multi-look*. Os filtros devem manter o valor médio do retorno do radar, preservar as bordas presentes na imagem e as informações de textura. Mais informações podem ser obtidas em [76].

A primeira coluna da Fig. 6.18 mostra partes retiradas da Fig. 6.17. Para a segmentação dessas imagens não foi realizado nenhum pré-processamento, etapa comum nesse tipo de imagem pela grande presença de ruído. Os parâmetros utilizados foram: tamanho da região 2 e a '*media*' igual a 8, com 2 características.

Uma outra aplicação de segmentação de texturas é a inspeção industrial. Como exemplo, utiliza-se aqui imagens de tecidos que apresentaram alguma espécie de defeito de fabricação. Em geral, a indústria mantém pessoas que fazem a verificação da qualidade do produto. Quando essa tarefa é realizada visualmente, podem ocorrer falhas, decorrente principalmente de cansaço físico. O processamento digital de imagens, se propõe a auxiliar as indústrias nessa tarefa.

Entretanto, há uma dificuldade inerente a esse tipo de processamento: a necessidade

²Obtida em http://www.esa.int/esaCP/ESAGFD7708D_index_1.html em 02/12/2009

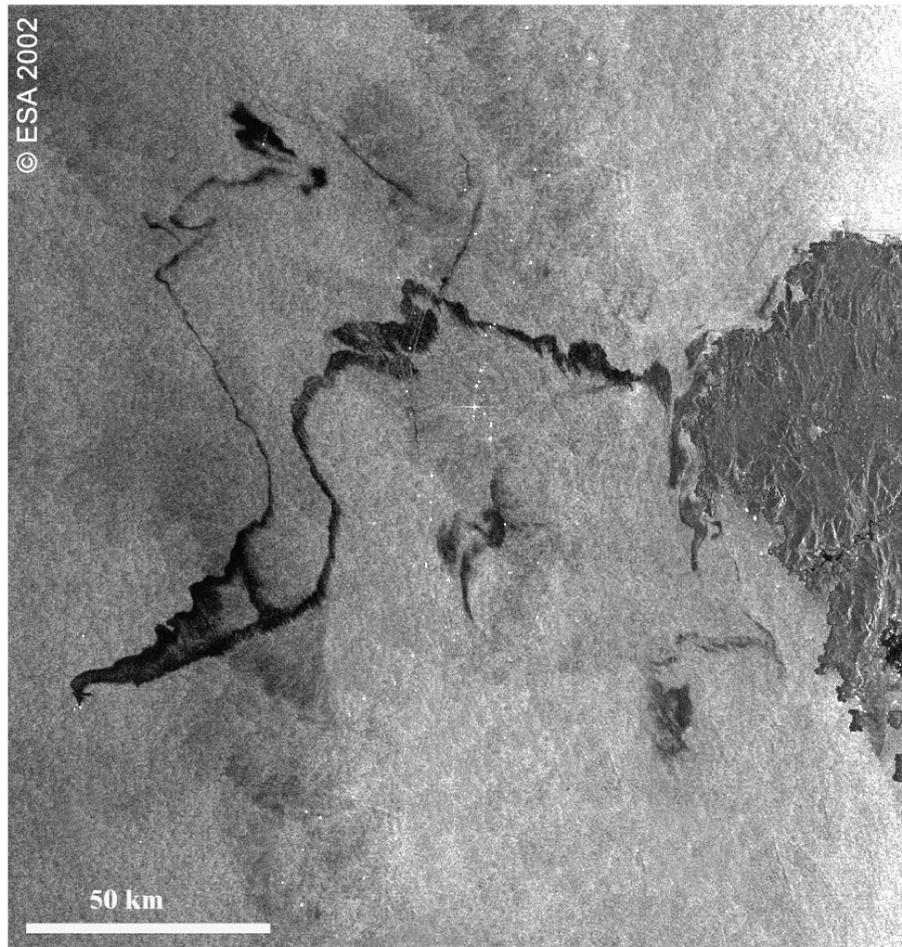


Fig. 6.17: Imagem Envisat's ASAR adquirida em 17 de novembro 2002.

de um algoritmo que realize a análise em tempo real. Neste trabalho, foram realizadas apenas simulações com imagens estáticas para mostrar mais uma aplicação possível para a segmentação de texturas.

As imagens da primeira coluna da Fig. 6.19 mostram amostras com defeitos em uma linha de produção de uma indústria têxtil³.

Método	oleo1	oleo2	tecido1	tecido2
Metodologia Proposta	3.82	2.39	2.33	5.94

Tab. 6.6: Percentual de erro para outras aplicações.

³Imagens obtidas em <http://www4.comp.polyu.edu.hk/~csajaykr/myhome/research/wavelets.html> em 02/12/2009

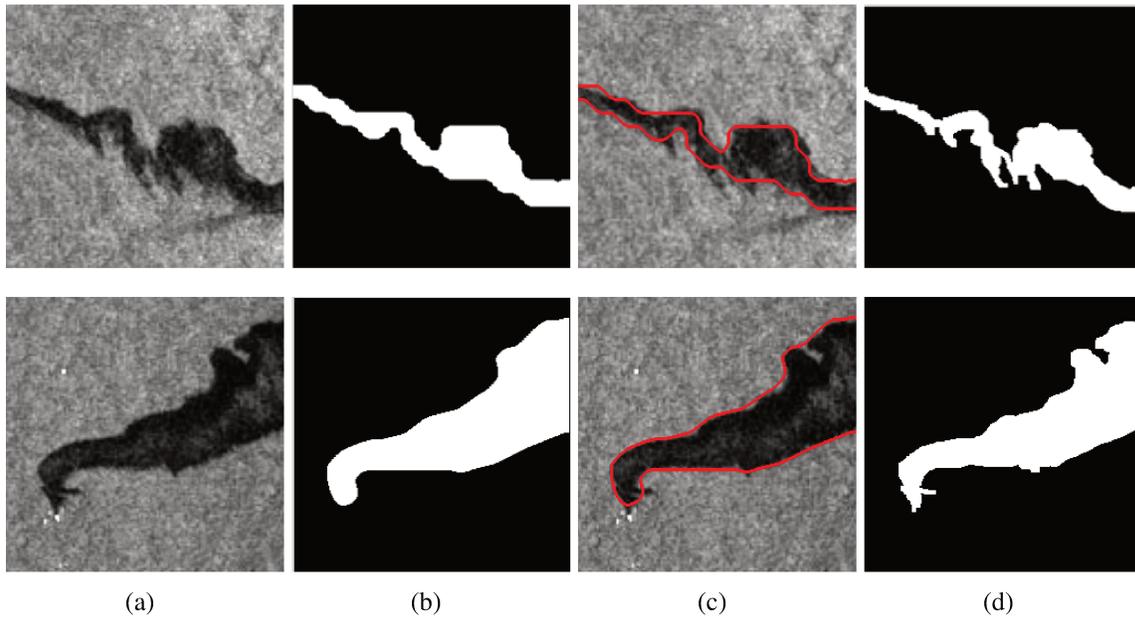


Fig. 6.18: (a) Imagem original. (b) Resultado obtido pela metodologia proposta. (c) Resultado obtido pela metodologia proposta sobreposto à imagem original. (d) Imagem de referência.

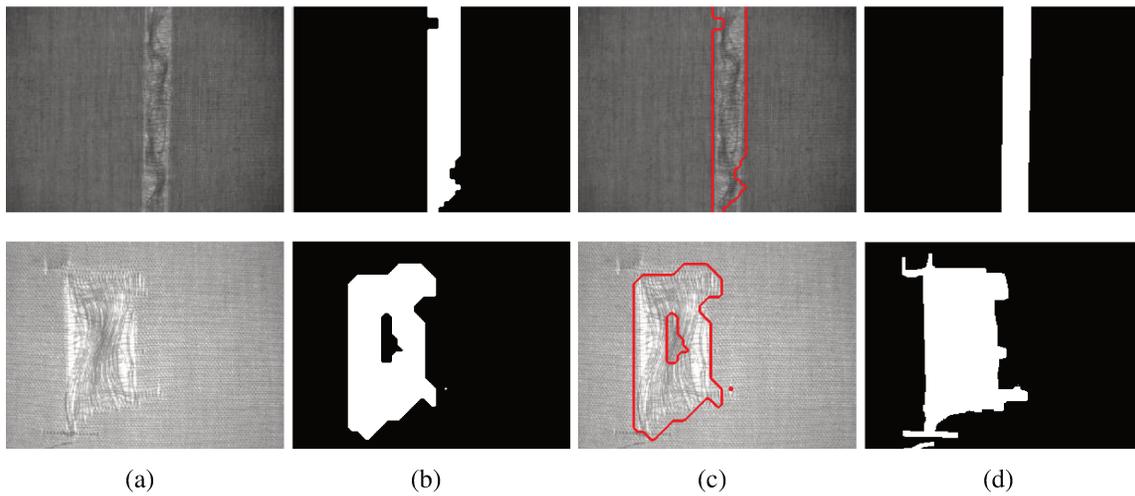


Fig. 6.19: (a) Imagem original. (b) Resultado obtido pela metodologia proposta. (c) Resultado obtido pela metodologia proposta sobreposto à imagem original. (d) Imagem de referência.

6.6 Comentários

Este capítulo apresentou os resultados obtidos pela metodologia proposta. As tabelas foram utilizadas para comparar as taxas de erro obtidas pelo algoritmo proposto e ou-

tras propostas existentes na literatura. Através das observações obtidas pode-se ver que a metodologia proposta se mostrou eficiente para diversos tipos de imagens.

Capítulo 7

Considerações Finais

A segmentação é o primeiro passo do processamento digital de imagens, tendo como função separar o objeto de interesse do restante da imagem. Trata-se de uma etapa fundamental, pois uma segmentação mal realizada implica em um processamento deficiente.

Dentre as diversas áreas de aplicação do processamento de imagens, a análise de textura tem ganhado espaço na caracterização dos objetos por ser uma característica inerente a praticamente todos os objetos existentes na natureza. Entretanto, apesar do olho humano ter uma extensa capacidade de diferenciar as mais diversas texturas, ainda existem muitas limitações para que essa mesma habilidade seja implementada em computadores.

Vários são os estudos já existentes sobre análise de texturas, alguns relacionados a segmentação, outros à geração de texturas sintéticas e, uma grande parte objetiva a classificação de texturas. Dentro desse contexto, este trabalho propõe uma metodologia para segmentação não supervisionada de texturas utilizando a transformada *wavelet packet* e o algoritmo imuno-inspirado ARIA.

Uma segmentação não supervisionada é fundamental para processamento em tempo real, por não implicar na necessidade de um usuário marcando regiões na imagem ou setando parâmetros. Essa é uma boa parte de contribuição deste trabalho.

A transformada *wavelet* tradicional tem sido extremamente explorada em diversas áreas da computação e da engenharia, inclusive na análise textural. Entretanto, optou-se por utilizar a transformada *wavelet packet* pelo destaque que ela tem recebido pela capacidade de realizar a análise da textura em médias frequências.

Vários estudos comprovaram que muita informação textural é encontrada na região de

médias frequências. Dado que a transformada *wavelet* tradicional decompõe a imagem apenas em altas e baixas frequências, a transformada *wavelet packet* se torna essencial para a discriminação de texturas.

O outro ponto diferencial deste trabalho está no algoritmo utilizado para a clusterização. Ao invés dos tradicionais algoritmos que necessitam um conhecimento a priori do número de *clusters* existentes nos dados extraídos das imagens optou-se por um algoritmo imuno-inspirado, que é capaz de determinar automaticamente quantos *clusters* são encontrados.

Os algoritmos imuno-inspirados foram propostos mais recentemente, e estão incluídos dentro do conjunto dos algoritmos inspirados na natureza. Eles têm se mostrado bem eficientes para executar funções como clusterização e otimização.

Neste trabalho foi utilizado o ARIA. Para realizar clusterização de dados de forma não supervisionada esse algoritmo utiliza os conceitos de clonagem e supressão retirados do funcionamento do sistema imunológico humano. A clonagem consiste em multiplicar aqueles anticorpos que estão combatendo muitos antígenos, aumentando assim o seu número. Já a supressão, tem o objetivo de diminuir o número de anticorpos, eliminando aqueles que não reconhecem nenhum antígeno.

Dessa forma, o ARIA é capaz de localizar onde os dados estão. E o seu raio adaptável permite que a informação de densidade dos dados também seja considerada. Aloca-se mais anticorpos apenas onde é necessário. Este torna-se o grande diferencial deste trabalho, diferentemente dos demais encontrados na literatura, na metodologia aqui apresentada, como já foi citado, não é necessário o conhecimento prévio do número de textura da imagem. Permite-se assim, um processo mais automatizado.

Além disso, os resultados aqui apresentados demonstram a versatilidade da metodologia proposta, que foi aplicada em diversas imagens com origens totalmente diferentes. Iniciando com imagens sintéticas de mosaicos já conhecidos e apresentados na literatura, passando por imagens médicas e chega-se até as aplicações industriais.

Em resumo, este trabalho foi assim exposto: primeiramente, o capítulo 1 apresentou uma breve introdução ao assunto aqui discutido, dando iniciação à metodologia aqui proposta. Expõem-se ainda os objetivos, a motivação e contribuições desta dissertação.

O capítulo 2 mostrou vários aspectos da análise textural. Inicialmente foram apresentadas diversas definições de textura. Em seguida, vários métodos utilizados para análise textural foram discutidos. E, para melhor compreensão do processo completo do processa-

mento de texturas, ele foi dividido em quatro etapas: extração de características, discriminação de texturas, classificação de texturas e reconstrução de texturas. A metodologia aqui proposta trata dos dois primeiros passos.

As teorias das transformadas *wavelet* e *wavelet packet* foram expostas no capítulo 3. Apesar da transformada *wavelet* tradicional não ter sido utilizada de forma direta neste trabalho, a sua teoria é de fundamental importância para o entendimento da transformada *wavelet packet*. Essa última mostrou-se eficiente na extração de características texturais, como pôde-se verificar.

Os conceitos de reconhecimento de padrões e o algoritmo imuno-inspirado ARIA foram apresentados no capítulo 4. A análise de componentes principais foi utilizada para reduzir a dimensionalidade das características obtidas a partir da imagem. Essa etapa é fundamental para diminuir uma possível redundância existente nos dados e diminuir o tempo de processamento.

Em seguida, foi introduzido o sistema imunológico humano, que é a base dos algoritmos imuno-inspirados. Esses algoritmos são bem recentes, tendo como principais aplicações a clusterização e a otimização. Neste trabalho foi escolhido o ARIA, que se mostrou eficiente para definição dos *clusters* existentes nos dados extraídos a partir da informação de textura da imagem.

O capítulo 5 expõe a metodologia proposta por este trabalho. Para tal, fez-se uso da teoria anteriormente explicitada. Espera-se que os diagramas de blocos e as imagens utilizadas como exemplo de cada etapa tornem mais fácil a compreensão da metodologia exposta neste trabalho.

Os resultados obtidos foram mostrados no capítulo 6, onde são comparados com outros resultados encontrados na literatura. Foram realizados testes em vários tipos de imagens: sintéticas retiradas do álbum de Brodatz, animais, mamogramas, melanomas, manchas de óleo e tecidos.

As conclusões obtidas e as sugestões para trabalhos futuros são apresentadas no presente capítulo, que finaliza este trabalho.

7.1 Comentários Sobre as Simulações e Resultados

O Matlab 7.9(R2006b) foi o ambiente utilizado para a implementação dos algoritmos aqui utilizados, inclusive para a visualização das imagens e geração dos gráficos aqui

apresentados.

As imagens utilizadas para os testes têm tamanhos variados, mostrados na Tab. 7.1. Pode-se ver que as dimensões são muito variadas, e isso deve principalmente ao fato de cada tipo de imagem ter sua origem distinta.

Imagem	Dimensão
D49D53	200x157
D09D29	300x300
D09D15	300x300
D04D84	512x256
D05D92	512x256
D12D17	512x256
D55D77D84	195x200
D53D35D93	300x300
D53D35D93D66	300x300
Zebra1	530x340
Zebra2	512x512
Zebra3	355x280
Tigre	405x210
Leopardo	450x575
mdb016	315x510
mdb003	315x510
Melanoma1	350x350
Melanoma2	400x350
Oil1	250x235
Oil2	250x250
Tecido1	300x225
Tecido2	300x225

Tab. 7.1: Dimensão das imagens utilizadas nos testes.

Segundo os resultados obtidos, pode-se observar a eficiência da metodologia proposta. Tem-se índices de erros que variam entre 0% e 12,34%. Os melhores resultados obtidos foram nas imagens de Brodatz com duas texturas. Isso era esperado, pois quanto maior o número de texturas mais difícil é a distinção das mesmas.

As imagens de animais colocam em prova a capacidade do algoritmo de unir em um único objeto a mesma textura mas com pequenas variações de orientação e tamanho. Nesse caso os bons resultados mostram uma aplicação do algoritmo proposto.

As mamografias se mostraram o maior desafio para a metodologia proposta, obtendo os piores resultados. Esse fato deve-se principalmente ao pouco contraste apresentado por esse tipo de imagem. Apesar disso, os erros apresentados são apenas nas regiões de borda, sendo a maior parte da imagem segmentada corretamente. Os resultados obtidos para as imagens de melanomas tiveram bons resultados, entretanto não foi possível comparar os resultados obtidos com outros da literatura.

Devido ao grande tamanho, as imagens de derramamento de óleo foram fracionadas, sendo esse procedimento comum para esse tipo de imagem. Isso se aplica principalmente quando se deseja apenas realizar testes de aplicabilidade de algoritmos, como é esse o caso. Entretanto, para um procedimento completo se faz necessária uma melhor otimização do algoritmo, para diminuir o custo computacional em grandes imagens.

Para mostrar um exemplo de aplicação industrial, amostras de tecido foram escolhidas. Esta aplicação mostra o que ocorre no dia-a-dia de uma fábrica de tecidos, onde é fundamental que haja inspeção da qualidade daquilo que é produzido. Nesse caso, a informação de borda não é de extrema relevância. Os resultados obtidos pela metodologia proposta mais uma vez mostram que ela pode ser aplicada nos mais diversos segmentos.

Conclui-se, portanto, que o objetivo de apresentar uma metodologia não supervisionada para segmentação de imagens com texturas foi alcançado utilizando como principais ferramentas a transformada *wavelet packet* e o ARIA.

7.2 Trabalhos Futuros

O fato da metodologia aqui apresentada ter sido construída em blocos facilita o teste de novos algoritmos, substituindo apenas o bloco desejado, sem necessidade de alteração do restante do código.

A primeira possibilidade de melhoramento é avaliar os resultados obtidos utilizando outras famílias de *wavelets*. Por exemplo: *wavelets daubechies* e *wavelets* bilineares. Uma outra possibilidade de mudança é na função de avaliação de custo citada na Seção 3.6.2. Além da função de entropia adotada neste trabalho, a função logaritmo da energia também tem sido largamente utilizada na literatura.

No caso das imagens naturais, pode ser relevante efetuar um pré-processamento na imagem, para que problemas do tipo iluminação irregular tenham menor interferência no resultado final. Além disso, outros métodos não supervisionados para clusterização podem

ser estudados e implementados.

E, por último, pode-se realizar um estudo sobre o custo computacional do algoritmo proposto, fazendo melhorias onde necessário.

Referências Bibliográficas

- [1] R. Jenssen e T. Eltoft. Independent component analysis for texture segmentation. *Pattern Recognition*, 36(10):2301–2315, 2003.
- [2] E. O. Nunes e A. Conci. Segmentação por textura e localização do contorno de regiões em imagens multibandas. *IEEE Latin America Transactions*, 5(3):185–192, 2007.
- [3] T. Tasdizen, R. T. Whitaker e S. P. Awate. Unsupervised texture segmentation with nonparametric neighborhood statistics. *Lecture Notes in Computer Science*, (3952):494–507, 2006.
- [4] A. Pagamisse. *Discriminação de Texturas Pela Transformada Wavelet*. Tese de Doutorado, Instituto de Matemática, Estatística e Computação Científica, UNICAMP, 2003.
- [5] L. Wang e J. Liu. Texture segmentation based on mrmrf modeling. *Pattern Recognition Letters*, 21(2):189 – 200, 2000.
- [6] T. Yamawaki, H. Tamura e S. Mori. Textural features corresponding to visual perception. *IEEE Transactions on Systems, Man and Cybernetics*, 8(6):460–473, 1978.
- [7] Jack Sklansky. Image segmentation and feature extraction. *IEEE Transactions on Systems, Man and Cybernetics*, 8(4):237–247, 1978.
- [8] R. M. Haralick. Statistical and structural approaches to texture. *Proceedings of the IEEE*, 67(5):786–804, 1979.
- [9] S. W. Zucker e K. Kant. Multiple-level representations for texture discrimination. *Proc. IEEE Conf. On Pattern Recognition and Image Processing*, pp. 609–614, 1981.

- [10] W. Richards e A. Polit. Texture matching. *Biological Cybernetics*, 16(3):155–162, 2004.
- [11] M. D. Levine. *Vision in Man and Machine*. Mcgraw Hill, 1985.
- [12] T. Randen. Brodatz textures. <http://www.ux.uis.no/~tranden/brodatz.html>.
- [13] A. Materka e M. Strzelecki. Texture analysis methods - a review. *COST B11 report*, 1998.
- [14] J. Sobus, B. Pourdeyhimi, B. Xu, e Y. Ulcay. Evaluating loss of texture definition in carpets using mathematical morphology: Covariance. *Textile Research Journal*, 62(1):26–39, 1992.
- [15] X. Liu e D. Wang. Texture classification using spectral histograms. *IEEE Transactions on Image Processing*, 12(6):661–670, 2003.
- [16] T. Jaaskelainen, R. Lenz, M. Hauta-Kasari e J. Parkkinen. Generalized co-occurrence matrix for multispectral texture analysis. *Proceedings of the 13th International Conference on Pattern Recognition*, 2:785–789, 1996.
- [17] A. L. M. Cruz. *Procedimentos para método híbrido de compressão de imagens digitais utilizando transformadas Wavelet e codificação fractal*. Tese de Doutorado, Universidade Estadual de Campinas. Faculdade de Engenharia Elétrica e de Computação, 2005.
- [18] M. Strzelecki e A. Materka. Markov random fields as model of textured biomedical images. *Proc. 20th National Conf. Circuit Theory and Electric Networks KTOiUE'97*, pp. 493–498, 1997.
- [19] B.J. Hosticka, A. Teuner e O. Pichler. Unsupervised texture segmentation of images using tuned matched gabor filters. *IEEE Transactions on Image Processing*, 4(6):863–870, 1995.
- [20] M. Unser. Texture classification and segmentation using wavelet frames. *IEEE Transactions on Image Processing*, 4(11):1549–1560, 1995.

- [21] L. R. Schad, D. Boyce, S. Blüml, I. Zuna, R. A. Lerski e K. Straughan. Mr image texture analysis : an approach to tissue characterization : Tissue characterization by magnetic resonance spectroscopy and imaging. *Magnetic Resonance Imaging*, 11(6):873–887, 1993.
- [22] W.S. Geisler, A.C. Bovik e M. Clark. Multichannel texture analysis using localized spatial filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(1):55–73, 1990.
- [23] Ma Li e R.C. Staunton. Optimum gabor filter design and local binary patterns for texture segmentation. *Pattern Recognition Letters*, 29(5):664 – 672, 2008.
- [24] M. Idrissa e M. Acheroy. Texture classification using gabor filters. *Pattern Recognition Letters*, 23(9):1095–1102, 2002.
- [25] G.M. Haley e B.S. Manjunath. Rotation-invariant texture classification using modified gabor filters. *Image Processing, International Conference on*, pp. 1:262, 1995.
- [26] A. K. Jain e F. Farrokhnia. Unsupervised texture segmentation using gabor filters. *Pattern Recognition*, 24(12):1167–1186, 1991.
- [27] D. Casanova. Identificação de espécies vegetais por meio da análise de textura foliar. Dissertação de Mestrado, Universidade de São Paulo, 2008.
- [28] C. P. Santos, J. Monteiro e M. J. Ferreira. Texture cue based tracking system using wavelet transform and a fuzzy grammar. *5th IEEE International Conference on Industrial Informatics*, volume 1, pp. 393–398, 2007.
- [29] X.-J. HE, J.-H. Han e Y. Zhang. Texture feature-based image classification using wavelet package transform. *Lecture Notes in Computer Science*, 3644:165–173, 2005.
- [30] A. Laine e J. Fan. Texture classification by wavelet packet signatures. *IEEE Transactions On Pattern Analysis And Machine Intelligence*, 15(11):1186–1191, 1993.
- [31] T. Chang e C.-C.J. Kuo. Texture analysis and classification with tree-structured wavelet transform. *IEEE Transactions on Image Processing*, 2(4):429–441, 1993.

- [32] M. A. Helvie, M. M. Goodsitt, D. D. Adler, A. Petrosian e H.-P. Chan. Computer-aided diagnosis in mammography: classification of mass and normal tissue by texture analysis. *Physics in Medicine and Biology*, 39(12):2273–2288, 1994.
- [33] J. Jan. Retinal image analysis aimed at blood vessel segmentation and neural layer detection. *Proceedings of the 11th Biennial Baltic Electronics Conference*, pp. 31–38, 2008.
- [34] S. Nawaz e A. H. Dar. Hepatic lesions classification by ensemble of svms using statistical features based on co-occurrence matrix. *Proceedings - 4th IEEE International Conference on Emerging Technologies 2008*, pp. 21–26, 2008.
- [35] Y. Li, Y. Liu e L. Ma. Sar image texture analysis of oil spill. *37th COSPAR Scientific Assembly*, 1861, 2008.
- [36] F. Farrokhnia. *Multi-Channel Filtering Techniques for Texture Segmentation and Surface Quality Inspection*. Tese de Doutorado, Michigan State University, 1990.
- [37] D.-M. Tsai e T.-Y. Huang. Automated surface inspection for statistical textures. *Image and Vision Computing*, 21(4):307 – 323, 2003.
- [38] P. Geurts, L. Wehenkel e R. Maree. Random subwindows and extremely randomized trees for image classification in cell biology. *BMC Cell Biology*, 8(Suppl 1):S2, 2007.
- [39] M. Penas-Centeno, D.-H. Xu, L. Shapiro, R. Ladner, E. Riskin, W. Hol e D. Meldrum S. Pan, G. Shavit. Automated classification of protein crystallization images using support vector machines with scale-invariant texture and Gabor features. *Acta Crystallographica Section D*, 62(3):271–279, 2006.
- [40] L. Velho, S. Goldenstein e J. Gomes. *Wavelets: Teoria, Software e Aplicacoes*. IMPA publication (21º Colóquio Brasileiro de Matemática), Rio de Janeiro, 1997.
- [41] A. Grossmann e J. Morlet. Decomposition of hardy functions into square integrable wavelets of constant shape. *SIAM Journal on Mathematical Analysis*, 15(4):723–736, 1984.
- [42] Y. Meyer. Principe d’incertitude, bases hilbertiennes et algebres d’operateurs. (662), 1985-1986.

- [43] G. Battle. A block spin construction of ondelettes. part i: Lemarié functions. 110(4):601–615, 1987.
- [44] P. G. Lemarié. Ondelettes à localisation exponentielles. *J. Math. Pures et Appl.*, 67:227–236, 1988.
- [45] I. Daubechies. Orthonormal bases of compactly supported wavelets. *Communications on Pure and Applied Mathematics*, 41(7):909–996, 1988.
- [46] A. Pagarra. Aplicação da transformada wavelet packet na análise e classificação de sinais de vozes patológicas. Dissertação de Mestrado, Universidade Federal do Rio Grande do Sul, 2002.
- [47] I. Daubechies. The wavelet transform, time-frequency localization and signal analysis. *IEEE Transactions on Information Theory*, 36, 1990.
- [48] S. Mallat. A theory for multiresolution signal decomposition: the wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11:674–693, 1989.
- [49] S. Mallat. Multifrequency channel decompositions of images and wavelet models. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 37(12), 1989.
- [50] M. C. Q. Farias. Aplicação da transformada wavelet na compressão de imagens. Dissertação de Mestrado, FEEC/UNICAMP, 1998.
- [51] S. Mallat. Multiresolution approximations and wavelet orthonormal bases of $L_2(\mathbb{R})$. *Transactions of the American Mathematical Society*, 315(1), 1989.
- [52] M. V. Wickerhauser. *Adapted Wavelet Packet Analysis from Theory to software*. AK Peters, 1994.
- [53] A. C. Paris. Análise da eficiência de recuperação por conteúdo de imagens médicas, utilizando extratores de textura baseados em wavelet e wavelet packet. Dissertação de Mestrado, Universidade de São Paulo, 2008.
- [54] X. Ding e C. Liu X. Wang. Gabor filters-based feature extraction for character recognition. *Pattern Recognition*, 38(3):369 – 379, 2005.

- [55] V. B. Sablon, R. F. L. Chavez e Y. Iano. Processo de reconhecimento de Íris humana: Localização rápida de Íris. *Telecomunicações - INATEL*, 9:1–7, 2006.
- [56] B. Chen e N. Morgan Q. Zhu, A. Stolcke. Using mlp features in sri's conversational speech recognition system. *Interspeech*, pp. 2141–2144, 2005.
- [57] Y.-K. Huang e S.-N. Yu. Recognition of microcalcifications in digital mammograms using high order markov random field model. *World Congress on Medical Physics and Biomedical Engineering 2006*, 14:2276–2279, 2007.
- [58] S. Haykin. *Neural Networks*. Prentice Hall, 2 edition, 1999.
- [59] R. C. Gonzalez e R. E. Woods. *Processamento de Imagens Digitais*. Edgard Blucher, 2000.
- [60] T. B. S. de Oliveira. Clusterização de dados utilizando técnicas de redes complexas e computação bioinspirada. Dissertação de Mestrado, Universidade de São Paulo, 2008.
- [61] S. Theodoridis e K. Koutroumbas. *Pattern Recognition*. AcademicPress, 1999.
- [62] J. Sander, M. Ester, H.-P. Kriegel e X. Xu. Density-based clustering in spatial databases: The algorithm gbscan and its applications. *Data Mining and Knowledge Discovery*, pp. 169–194, 2004.
- [63] L. N. de Castro, F. J. Von Zuben, G. B. Bezerra e T. V. Barra. Adaptive radius immune algorithm for data clustering. *Lecture Notes in Computer Science*, 3627:290–303, 2005.
- [64] C. T. Zahn. Graph-theoretical methods for detecting and describing gestalt clusters. *IEEE Transactions on Computers*, C-20(1):68–86, 1971.
- [65] Randen e J. H. Husoy. Filtering for supervised texture segmentation: A comparative study. *IEEE Transactions Pattern Analysis and Machine Intelligence*, 21(4):291–310, 1999.
- [66] J.M. Coggins e A.K. Jain. A spatial filtering approach to texture analysis. *Pattern Recognition Letters*, 3(3):195–203, 1985.

- [67] A.K. Jain e F. Farrokhnia. Unsupervised texture segmentation using gabor filters. *Pattern Recognition*, 24(12):1167–1186, 1991.
- [68] I. Ng, T. Tan e J. Kittler. On local linear transform and gabor filter representation of texture. *Proc. Int'l Conf. Pattern Recognition*, pp. 627–631, 1992.
- [69] R. Haralick, K. Shanmugam e I. Dinstein. Textural features for image classification. *IEEE Trans. Systems, Man, Cybernetics*, 3:610–621, 1973.
- [70] M. Tuceryan e A.K. Jain. *Handbook Pattern Recognition and Computer Vision*, chapter Texture Analysis, pp. 235–276. Singapore: World Scientific, 1993.
- [71] T. Randen e J.H. Husøy. Multichannel filtering for image texture segmentation. *Optical Eng.*, 33:2617–2625, 1994.
- [72] F. Ade. Characterization of texture by 'eigenfilter'. *Signal Processing*, 5(5):451–457, 1983.
- [73] T. Randen. *Filter and Filter Bank Design for Image Texture Recognition*. Tese de Doutorado, Norwegian Univ. of Science and Technology, 1997.
- [74] D.F. Dunn e W.E. Higgins. Optimal gabor filters for texture segmentation. *IEEE Trans. Image Processing*, 4:947–964, 1995.
- [75] T.P. Weldon e W.E. Higgins. Design of multiple gabor filters for texture segmentation. *Proc. Int'l Conf. Acoustic Speech, Signal Proc.*, pp. 2243–2246, 1996.
- [76] INPE. *SPRING: Tutorial de Geoprocessamento*. [<http://www.dpi.inpe.br/spring/portugues/tutorial/index.html>]. Acesso em: 14/04/2010.