

RICARDO DUTRA DA SILVA

**Segmentação de Imagens Baseada em Descritores de Textura  
Extraídos por Transformadas Wavelets**

Dissertação apresentada como requisito parcial à  
obtenção do grau de Mestre. Programa de Pós-  
Graduação em Informática, Setor de Ciências Exatas,  
Universidade Federal do Paraná.  
Orientador: Prof. Dr. Hélio Pedrini

CURITIBA

2008

## SUMÁRIO

<b>LISTA DE FIGURAS</b>	<b>iv</b>
<b>LISTA DE TABELAS</b>	<b>v</b>
<b>LISTA DE ALGORITMOS</b>	<b>vi</b>
<b>RESUMO</b>	<b>vii</b>
<b>ABSTRACT</b>	<b>viii</b>
<b>1 INTRODUÇÃO</b>	<b>1</b>
1.1 Objetivos . . . . .	2
1.2 Justificativa . . . . .	3
1.3 Organização do Trabalho . . . . .	4
<b>2 REVISÃO BIBLIOGRÁFICA</b>	<b>5</b>
2.1 Segmentação de Imagens . . . . .	5
2.2 Wavelets . . . . .	6
2.2.1 Famílias de funções . . . . .	10
2.2.2 Análise de multirresolução . . . . .	11
2.2.3 Bancos de filtros . . . . .	14
2.2.4 Lifting . . . . .	18
2.2.5 Transformadas bidimensionais . . . . .	19
2.3 Análise de Texturas . . . . .	26
2.4 Cores . . . . .	31
2.5 Normalização . . . . .	32
2.6 Ordenação de Variáveis . . . . .	33

<b>3</b>	<b>METODOLOGIA PROPOSTA</b>	<b>36</b>
3.1	Seleção de Bases Wavelets . . . . .	37
3.2	Ordenação de Características . . . . .	39
3.3	Invariância à Rotação . . . . .	40
3.4	Modelos de Cores e Tamanhos de Blocos . . . . .	42
3.5	Algoritmo de Segmentação . . . . .	43
3.5.1	Segmentação Inicial . . . . .	44
3.5.2	Segmentação Final . . . . .	49
<b>4</b>	<b>RESULTADOS EXPERIMENTAIS</b>	<b>51</b>
4.1	Seleção de Bases . . . . .	53
4.2	Ordenação de Variáveis . . . . .	55
4.3	Invariância à Rotação . . . . .	59
4.4	Níveis de Decomposição . . . . .	65
4.5	Modelos de Cores e Tamanhos de Blocos . . . . .	68
4.6	Método de Segmentação . . . . .	71
<b>5</b>	<b>CONCLUSÕES</b>	<b>80</b>
	<b>REFERÊNCIAS BIBLIOGRÁFICAS</b>	<b>85</b>
	<b>APÊNDICE A FILTROS PARA AS TRANSFORMADAS WAVELETS</b>	<b>93</b>
A.1	Daubechies . . . . .	93
A.2	Symlets . . . . .	95
A.3	Coiflets . . . . .	97
A.4	Biortogonal . . . . .	99
A.5	Neville . . . . .	102
A.6	MaxMax e MaxMin . . . . .	105
	<b>ÍNDICE REMISSIVO</b>	<b>106</b>

## LISTA DE FIGURAS

2.1	Uma wavelet diferencia-se de uma onda comum por possuir a sua energia localizada no tempo. . . . .	7
2.2	Função chapéu mexicano em diferentes escalas. . . . .	8
2.3	Translações da função chapéu mexicano. . . . .	9
2.4	Esquema de decomposição usando bancos de filtros. . . . .	17
2.5	Dois níveis de decomposição usando bancos de filtros. . . . .	17
2.6	Esquema de reconstrução usando bancos de filtros. . . . .	17
2.7	Decomposição wavelet pelo esquema de <i>lifting</i> . . . . .	18
2.8	Reconstrução wavelet pelo esquema de <i>lifting</i> . . . . .	19
2.9	Decomposição de um nível da imagem Monalisa utilizando o filtro de Haar. As sub-bandas de detalhes foram equalizadas para uma melhor visualização. . . .	21
2.10	Transformadas bidimensionais. (a) decomposição em um nível (b) decomposição em dois níveis. . . . .	21
2.11	Transformada bidimensional padrão. . . . .	22
2.12	Transformada bidimensional não-padrão. . . . .	22
2.13	Grade retangular composta por duas grades <i>quincunx</i> . . . . .	24
2.14	Grade diagonal composta por duas grades <i>quincunx</i> . . . . .	25
2.15	Filtros Neville de ordem 4. . . . .	26
2.16	Exemplos de texturas. . . . .	27
3.1	Ordenação de variáveis. . . . .	39
3.2	Textura em diferentes rotações. As figuras 3.2(a)-(h) mostram a textura variando de $0^\circ$ até $105^\circ$ com passo de rotação igual a $15^\circ$ . . . . .	41
3.3	Diagrama do método proposto. . . . .	43
3.4	Efeito de blocagem produzido pela segmentação inicial baseada em blocos: (a) imagem original e (b) imagem segmentada. . . . .	46

3.5	Em (a) são mostrados os agrupamentos e os centróides resultantes do algoritmo <i>k-means</i> . Em (b), pontos que representam blocos heterogêneos são marcados com o símbolo +. . . . .	48
3.6	Blocos heterogêneos marcados em branco. . . . .	49
3.7	Segmentação final. . . . .	50
4.1	Exemplos de mosaicos. Os mosaicos (a), (b) e (c) são compostos respectivamente por 2, 3 e 4 regiões de textura. . . . .	52
4.2	Resultado da ordenação pelo método não-supervisionado. . . . .	57
4.3	Resultado da ordenação pelo método supervisionado. . . . .	58
4.4	Variação de características para imagens com rotações. A base bior3.1 foi utilizada. . . . .	60
4.5	Variação de características para imagens com rotações. Wavelets separáveis. . . . .	61
4.6	Variação de características para imagens com rotações. Wavelets não-separáveis. . . . .	62
4.7	Variação de wavelets para imagens com rotações. . . . .	63
4.8	Variação de wavelets para imagens com rotações. . . . .	64
4.9	Correção da segmentação variando níveis de decomposição com blocos de tamanho $64 \times 64$ pixels. . . . .	66
4.10	Correção da segmentação variando níveis de decomposição com blocos de tamanho $32 \times 32$ pixels. . . . .	67
4.11	Segmentação de mosaico com diferentes modelos de cores e blocos de $64 \times 64$ pixels. . . . .	69
4.12	Comparação de modelos de cores para segmentação de mosaicos. . . . .	70
4.13	Segmentação de mosaico com diferentes modelos de cores e blocos de $128 \times 128$ pixels. . . . .	71
4.14	Imagens de satélite segmentadas pelo método proposto. . . . .	73
4.15	Imagens naturais segmentadas pelo método proposto. . . . .	75
4.16	Mosaicos segmentados pelo método proposto. . . . .	77

## LISTA DE TABELAS

4.1	Grupos de mosaicos para testes. . . . .	53
4.2	Conjunto de wavelets utilizado nos testes de segmentação. . . . .	53
4.3	Número de ocorrências em que cada wavelet obteve a $n$ -ésima melhor segmentação. Foram realizados 550 testes para cada wavelet. Os resultados referem-se às 25 primeiras posições. . . . .	54
4.4	Níveis máximos de decomposição das bases wavelets dependendo do tamanho do bloco. . . . .	66
4.5	Bases wavelets usadas pelo método de segmentação. . . . .	71
4.6	Informações sobre os parâmetros do método de segmentação e sobre as imagens de satélite. . . . .	72
4.7	Informações sobre os parâmetros do método de segmentação e sobre as imagens naturais. . . . .	74
4.8	Informações sobre os parâmetros do método de segmentação e sobre os mosaicos. . . . .	76
4.9	Matrizes de confusão da segmentação dos mosaicos. Os valores referem-se a porcentagens. . . . .	78

## LISTA DE ALGORITMOS

3.1	Técnica de agrupamento <i>k-means</i> . . . . .	38
3.2	Cálculo do coeficiente de rotação relativo a texturas com rotação. . . . .	42
3.3	Segmentação inicial. . . . .	45
3.4	Identificação de blocos heterogêneos. . . . .	49
3.5	Segmentação final. . . . .	50

## RESUMO

A segmentação de imagens é um processo fundamental em diversas aplicações de imagens e vídeos, cujo principal propósito é permitir uma discriminação significativa entre objetos de interesse. Técnicas tradicionais para extração de texturas baseadas em pixels geralmente demandam alto custo computacional. Este trabalho apresenta um método de segmentação baseado em transformadas wavelets para extrair características das informações de luminância e crominância em imagens digitais. Para reduzir o custo, o método é composto por dois estágios. Inicialmente, a imagem é decomposta em blocos de pixels e uma transformada wavelet é aplicada sobre cada bloco para identificar regiões homogêneas da imagem, atribuindo o bloco inteiro para sua classe respectiva. Um estágio de refinamento é aplicado nos pixels restantes, descobertos no primeiro estágio como pertencentes a regiões heterogêneas. Como o método opera sem qualquer conjunto de treinamento, além de reduzir o custo das computações, o primeiro estágio também é responsável por extrair descritores para identificar cada região da imagem. Estudos preliminares são desenvolvidos para avaliar as transformadas wavelets, descritores e modelos de cores. Transformadas wavelets separáveis e não-separáveis são um tópico importante considerado nos estudos iniciais, estando diretamente relacionado com a invariância à rotação. O método desenvolvido, testado em várias imagens de texturas, mosaicos e imagens reais, apresenta alta adaptabilidade às regiões das imagens e custo computacional relativamente baixo.



## ABSTRACT

Image segmentation is a fundamental process in several image and video applications, whose main purpose is to allow a meaningful discrimination among constituent objects of interest. Traditional feature extraction techniques based on individual pixels usually demand high computational cost. This work presents an image segmentation method based on wavelet transforms for extracting a number of texture features from image luminance and chrominance information. To reduce the cost, the approach is composed of two main stages. Initially, the image is decomposed into blocks of pixels and a wavelet transform is applied to each block to identify homogeneous regions of the image, assigning the entire block to a respective class. A refinement stage is applied to the remaining pixels, found in the first stage as belonging to heterogeneous regions. Since the method operates without any training set, besides reducing the computational cost, the first stage is also responsible for extracting descriptors to identify each region in the images. Preliminary studies are developed to evaluate wavelet transformations, features and color models. Separable and non-separable wavelet transformations are an important topic considered in the early studies, being directly related to rotation invariance. The developed method, tested on several textures, mosaics and real images, presents high adaptability to image regions and relatively low computational cost.

## CAPÍTULO 1

### INTRODUÇÃO

O crescente avanço nas técnicas de aquisição de imagens torna complexa a análise automática ou semi-automática das imagens, que consiste na criação de algoritmos capazes de reproduzir total ou parcialmente a capacidade humana de entendimento e interpretação de imagens. O reconhecimento de objetos e regiões que compõem uma imagem é essencial para aplicações de visão robótica, monitoração industrial, sensoriamento remoto, diagnóstico médico assistido e recuperação de imagens por conteúdo.

Em análise de imagens, a segmentação é uma atividade de extrema importância. A segmentação procura particionar o conjunto de objetos ou regiões que compõem uma imagem, produzindo um conjunto de estruturas de alto nível que descrevem cada um destes objetos ou regiões.

Para que a segmentação possa ser efetuada de maneira satisfatória, deve-se definir um conjunto de características que identifiquem unicamente cada uma das regiões. Características de texturas têm sido intensamente exploradas para realizar essa identificação [15, 73, 74]. A importância provém do fato de que texturas apresentam excelentes informações estruturais e estão entre as principais características percebidas pelo sistema visual humano [44, 54].

As abordagens clássicas para análise de imagens costumam realizar a extração de características de texturas sobre níveis de cinza. No entanto, é possível explorar também informações presentes em cores e melhorar o desempenho dos sistemas.

Atualmente, a extração de características de texturas utilizando transformadas wavelets tornou-se um dos principais pontos de estudo em análise de imagens. As transformadas wavelets apresentam propriedades que proporcionam uma representação com informações espaciais e de frequência, além do aspecto de multirresolução, que permite a extração de descritores em diversas escalas [53]. Estas propriedades tornam as transformadas wavelets apropriadas, superando outros tipos de transformadas, como, por exemplo, a transformada de Fourier.

## 1.1 Objetivos

Este trabalho tem como objetivo explorar a técnica de segmentação de imagens baseada em transformadas wavelets. O estudo de características de textura e das transformadas wavelets deve servir como embasamento para o desenvolvimento de um algoritmo de segmentação de imagens. Em especial, descritores das classes de texturas, que formam os parâmetros para a tarefa de segmentação, devem ser adquiridos diretamente da imagem, não existindo qualquer tipo de conhecimento prévio.

A pesquisa tem alguns principais pontos de aprofundamento que, além de levarem ao desenvolvimento do algoritmo de segmentação, devem servir de auxílio para outras pesquisas em análise de imagens. Um dos pontos prevê a formação de um estudo comparativo de descritores que agregam informações de texturas em intensidade de nível de cinza e cores. Estendendo a capacidade dos descritores, é importante o estudos daquelas medidas mais apropriadas para a descrição de texturas em diferentes rotações.

Variando os modelos de cores, pretende-se avaliar diferentes formas de caracterização das propriedades presentes em imagens, visando a uma complementação robusta das propriedades de textura em níveis de cinza. A partir da avaliação, deve ser evidenciada a capacidade e importância da descrição conjunta entre textura e cor.

Outro ponto é o levantamento e descrição de bases e propriedades que sejam interessantes para a descrição de regiões texturizadas, definindo transformadas ideais para a tarefa de segmentação de imagens. Para que as texturas possam ser descritas observando invariâncias, é necessário que as bases wavelets apresentem propriedades que permitam este tipo de análise. Portanto, a pesquisa deve aprofundar-se em bases wavelets e aspectos de multirresolução.

O resultado do estudo de técnicas de segmentação baseadas em wavelets é empregado em um método de segmentação que trabalha apenas com as informações obtidas diretamente da imagem, sem imagens auxiliares para treinamento. Dividido em duas etapas, o método busca acelerar a segmentação e adquirir bons descritores na primeira etapa, possibilitando a aplicação de um melhoramento da segmentação em apenas um conjunto reduzido da imagem. Como os estudos preliminares são independentes do método de segmentação proposto, tanto o estudo das características quanto a investigação das transformadas wavelets devem formar

referências para a utilização em análise de imagens digitais.

## 1.2 Justificativa

Este trabalho vem suprir algumas deficiências existentes em grande parte das pesquisas relacionadas à segmentação de imagens utilizando wavelets, passando desde a escolha de bases wavelets, características de texturas e modelos de cores, até a sua aplicação e as tarefas relacionadas com a segmentação de imagens digitais.

Há uma carência sobre a definição de bases wavelets preferenciais para uso em sistemas de segmentação. Na literatura não são comumente encontradas comparações que justifiquem a escolha de determinadas bases wavelets, Todas as bases possuem certas propriedades que as identificam como wavelets, no entanto, há também propriedades particulares que diferenciam as bases e as tornam apropriadas para aplicações específicas. As propriedades individuais permitem a adaptação ou mesmo construção de diversas bases. O conjunto de bases relatadas já é bastante extenso, o que justifica a avaliação para promover a melhoria de descritores empregados em análise de imagem.

Também pretende-se verificar a eficiência dos descritores de texturas em níveis de cinza e em cores tomados conjuntamente. Poucos trabalhos exploram essa junção para a extração de características sobre texturas. Entre os que fazem uso de cores, a abordagem é geralmente empregada através de conceitos não relacionados com texturas. Na pesquisa desenvolvida, as informações de cores são utilizadas como complemento às informações de intensidade, buscando uma descrição mais completa dos padrões de texturas.

Entre as medidas usadas para extrair descritores baseados em coeficientes da transformada wavelet, a energia figura como a mais popular. Em muitos trabalhos é também a única medida considerada para obtenção dos descritores de texturas. Embora a importância da energia seja evidente por todos os trabalhos que a abordam, existem outras medidas cuja avaliação não pode ser ignorada e que servem tanto como alternativas à energia quanto colaboradoras na tarefa de descrição.

Além dos fatos anteriores, o estudo de segmentação sem imagens para treinamento produz contribuições em uma área ainda incipiente e que precisa ser melhor pesquisada.

### **1.3 Organização do Trabalho**

O restante do trabalho é dividido como segue. O capítulo 2 apresenta uma revisão bibliográfica sobre os conceitos necessários para o desenvolvimento da segmentação através de wavelets. Entre os assuntos mais importantes encontram-se a teoria de wavelets e aspectos relacionados com análise de texturas. A metodologia de pesquisa é apresentada no capítulo 3, onde são descritas as etapas para a construção do método de segmentação. Os testes e resultados obtidos com a aplicação da metodologia são demonstrados no capítulo 4. Por fim, o capítulo 5 apresenta as conclusões e considerações finais, juntamente com a citação dos trabalhos futuros. O apêndice A apresenta uma referência aos filtros wavelets considerados no desenvolvimento do método de segmentação.

## CAPÍTULO 2

### REVISÃO BIBLIOGRÁFICA

#### 2.1 Segmentação de Imagens

Devido ao crescente avanço nas técnicas de aquisição de imagens em áreas como sensoriamento remoto e medicina, possibilitando um aumento significativo na resolução das imagens e no volume dos dados gerados, a análise ou interpretação de imagens torna-se bastante complexa. É crucial a aplicação de um processo intermediário de segmentação, o qual efetua a partição dos objetos ou regiões que compõem uma imagem [35], produzindo estruturas de mais alto nível que auxiliarão na análise [6, 16, 27, 64]. Baseando-se em descontinuidades ou similaridades das características de uma imagem, algumas abordagens clássicas para a segmentação são encontradas na literatura, como a *limiarização*, *crescimento por regiões* e *divisão e fusão* [35].

A identificação correta da forma, topologia e localização dos objetos na imagem é um requisito fundamental para que o processo de segmentação possa fornecer informações confiáveis. Entretanto, a etapa de segmentação apresenta-se ainda como um grande desafio, pois processar uma imagem de modo a segmentar um número de objetos, possivelmente em posições distintas e com diferentes tamanhos e formas, é uma tarefa que apresenta alta complexidade.

Métodos de segmentação são comumente classificados em supervisionados e não-supervisionados. Quando os parâmetros são estimados usando informações conhecidas previamente, como um conjunto de treinamento que descreve cada uma das regiões da imagem, o método é supervisionado. As técnicas de segmentação não-supervisionadas, nas quais os parâmetros não são conhecidos inicialmente e precisam ser extraídos da própria imagem, ainda apresentam muitos desafios. Na literatura, a segmentação não-supervisionada não foi tão explorada quanto a segmentação supervisionada, apresentando poucas metodologias e resultados [31].

A definição de um conjunto de características, capaz de descrever de maneira precisa as regiões presentes nas imagens, é de extrema importância. Nesse aspecto, é comum o uso de

características de texturas, formas de objetos, intensidade dos níveis de cinza e cores. Métodos de análise de texturas têm obtido resultados significativos, dentre eles, destacam-se os métodos baseados nas transformadas wavelets [1, 32, 51, 66, 71].

## 2.2 Wavelets

A deficiência da transformada de Fourier em fornecer informações de um sinal nos domínios da frequência e do tempo levou à busca de ferramentas com a capacidade de obter análises localizadas, descrevendo melhor as partes de um sinal [34, 37]. Desta busca surgiram as transformadas wavelets [3, 24, 37, 53].

No apêndice da tese de Albert Haar [41], em 1910, houve a primeira menção ao sistema ortogonal de funções mais simples encontrado na literatura. Atualmente, essas funções são conhecidas como as wavelets de Haar. Entre as décadas de 1960 e 1980, importantes trabalhos foram desenvolvidos por Coifman, Grossmann e Morlet [20, 38]. O termo wavelet, que em inglês significa pequena onda, foi introduzido por Morlet [60]. Após 1980, Yves Meyer e Stephane Mallat aplicaram a idéia em uma teoria chamada multirresolução [52, 53]. Mallat descobriu importantes relações entre bancos de filtros (QMF, *quadrature mirror filters*), algoritmos de pirâmide e bases wavelets ortonormais. Baseando-se nos trabalhos de Mallat, Ingrid Daubechies desenvolveu uma família de funções wavelets de base ortonormal que se tornou bastante importante [25].

As wavelets possuem sua energia concentrada no tempo, ou espaço para imagens, sendo ondas localizadas, de curta duração, o que possibilita a análise de sinais transitórios, não-estacionários ou variáveis no tempo. Pode-se dizer que wavelets são sinais oscilantes e não-nulos em um curto intervalo de tempo.

Wavelets diferenciam-se de ondas mais comuns por terem sua energia localizada no tempo. Como exemplo, a figura 2.1 mostra a diferença entre elas. Na figura 2.1(a), a onda oscila em todo o espaço de tempo, já a wavelet (figura 2.1(b)) oscila em apenas um intervalo do espaço de tempo.

Com o propósito de desenvolver a teoria das transformadas wavelets, será comum falar em sinais, já que a teoria foi inicialmente formulada para processamento de sinais digitais. Desta

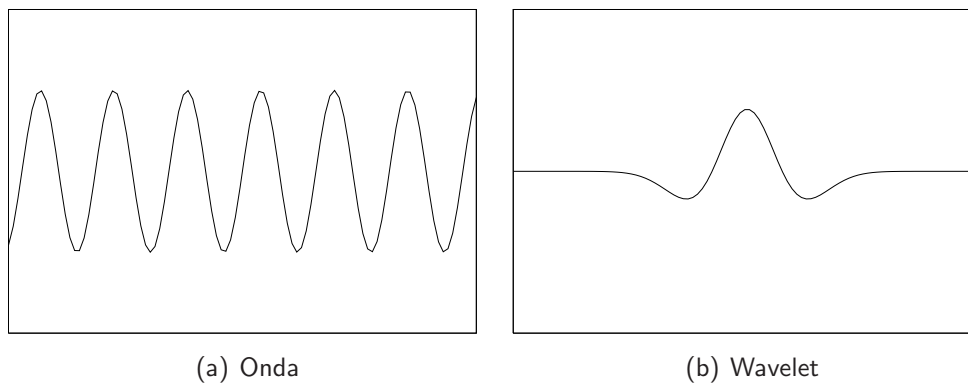


Figura 2.1: Uma wavelet diferencia-se de uma onda comum por possuir a sua energia localizada no tempo.

forma, é possível pensar em imagens como sinais unidimensionais. Ao longo deste capítulo serão mostradas maneiras de expandir as transformações para espaços bidimensionais.

Um sinal pode ser representado por uma função  $f(t)$  (equação 2.1) formada pela combinação linear de um conjunto de funções reais  $\psi_k(t)$  com coeficientes reais  $c_k$ , sendo  $k$  um índice inteiro.

$$f(t) = \sum_k c_k \psi_k(t) \quad (2.1)$$

Se a expansão é única, ou seja, se todas as funções  $f(t)$  em um espaço vetorial possuem um único conjunto de coeficientes  $\{c_k\}$  para a expansão da equação 2.1, então o conjunto  $\{\psi_k(t)\}$  é uma *base* para as funções do espaço.

A *expansão wavelet* é um sistema constituído de dois índices inteiros  $j, k$  e funções de expansão  $\psi_{j,k}(t)$ .

$$f(t) = \sum_j \sum_k c_{j,k} \psi_{j,k}(t) \quad (2.2)$$

A equação 2.2 é a *transformada wavelet inversa*, pois o sinal  $f(t)$  é obtido pela combinação linear das funções de expansão. Os coeficientes  $\{c_{j,k}\}$ , desta expansão, formam a chamada *transformada wavelet discreta* (DWT, *discrete wavelet transform*), consistindo na representação do sinal  $f(t)$  no espaço definido pelas funções  $\{\psi_{j,k}(t)\}$ . Para cada espaço de funções com um conjunto base  $\{\psi_{j,k}(t)\}$ , existe uma *base dual*  $\{\tilde{\psi}_{j,k}(t)\}$  [76]. Por esta



última é possível calcular os coeficientes da equação 2.2 tomando o produto interno

$$c_{j,k} = \langle \tilde{\psi}_{j,k}(t), f(t) \rangle = \int \tilde{\psi}_{j,k}(t) f(t) dt. \quad (2.3)$$

As funções  $\psi_{j,k}(t)$  são criadas a partir de translações e escalamentos de uma função de base  $\psi$ , chamada *wavelet mãe* [12]. A criação do conjunto de funções wavelets é realizada conforme a equação 2.4a, com  $j$  guiando a escala e  $k$  a translação.

$$\psi_{j,k}(t) = 2^{j/2} \psi(2^j t - k) \quad (2.4a)$$

Tomando  $j = 0$  e  $k = 0$ , então a função  $\psi_{0,0}(t)$  é a própria wavelet mãe  $\psi(t)$

$$\psi_{0,0}(t) = 2^0 \psi(2^0 t - 0) = \psi(t). \quad (2.4b)$$

O fator de escala representa uma contração ou dilatação do sinal. Para valores de  $j < 0$ , o sinal é dilatado, enquanto que, para valores de  $j > 0$ , o sinal sofre contração. Se  $j$  pertence aos inteiros  $\mathbb{Z}$  e  $k = 0$ , então

$$\psi_{j,0}(t) = 2^j \psi(2^j t - 0) = \psi(2^j t). \quad (2.4c)$$

O efeito pode ser observado na figura 2.4. Na figura 2.2(a), com  $j = -1$ , o sinal sofre dilatação. Quando  $j = 0$  (figura 2.2(b)) não há nem dilatação nem contração. Uma contração ocorre na figura 2.2(c), com  $j = 1$ . Existe uma relação inversa entre escala e frequência. Em escalas baixas tem-se alta frequência e em escalas altas a frequência é baixa.

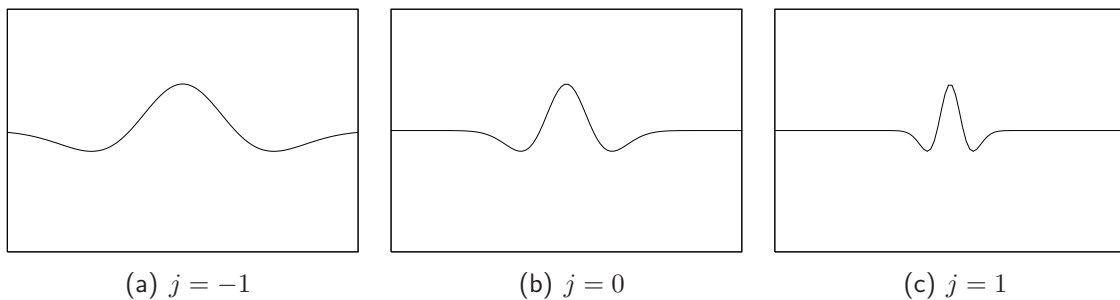


Figura 2.2: Função chapéu mexicano em diferentes escalas.

A translação, ou deslocamento, das funções  $\psi_{j,k}(t)$  é dada pelo fator  $k$ . Fazendo  $j = 0$  e  $k$  em  $\mathbb{Z}$

$$\psi_{0,k}(t) = 2^0 \psi(2^0 t - k) = \psi(t - k). \quad (2.4d)$$

A função  $\psi(t)$  deslocada é denotada por  $\psi(t - k)$ . Exemplos de translação podem ser vistos nas figura 2.3(a), 2.3(b) e 2.3(c), sendo  $k$  igual a  $-5$ ,  $0$  e  $5$ , respectivamente.

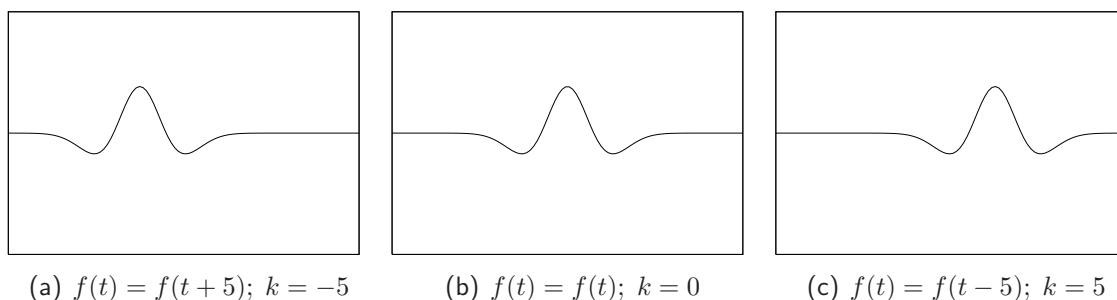


Figura 2.3: Translações da função chapéu mexicano.

Funções que respeitam as características de ondas citadas anteriormente são wavelets. Existem diversas funções wavelets conhecidas. Algumas propriedades que diferenciam as funções wavelets são discutidas a seguir.

## Suporte

O suporte de uma função  $f(t)$  de um conjunto  $X$ , no espaço dos números reais, é um subconjunto  $Y$  de  $X$  de forma que  $f(t)$  é zero para todo  $x$  em  $X$  que não está em  $Y$ . O suporte é fechado se o subconjunto de  $X$  é fechado. Assim, o suporte é a intersecção de todos os subconjuntos fechados.

As wavelets possuem suporte compacto [19], o que significa que seu suporte além de fechado é limitado, com o intervalo variando conforme a wavelet. Devido ao suporte compacto, os coeficientes de uma transformada contêm informações locais de uma função analisada. Se a função analisada fosse alterada em um certo intervalo, somente os coeficientes relacionados com o intervalo modificado sofreriam alteração. A maioria dos coeficientes permaneceria igual.

## Momentos de desvanecimento

Os momentos de desvanecimento de uma função estão relacionados com sua capacidade de reproduzir polinômios. Se uma função tem  $n$  momentos de desvanecimento, então, para  $l = 0, 1, \dots, n - 1$ ,

$$\int t^l f(t) dt = 0. \quad (2.5)$$

Esses fatores são importantes para o grau de convergência das aproximações wavelet de funções suaves e para a detecção de singularidades usando wavelets [34].

## Regularidade

A regularidade da função é útil para obter algumas características como a suavidade na reconstrução. A suavidade também corresponde a uma melhor localização das frequências dos filtros.

## Simetria

A simetria da função é outro fator de diferenciação. A simetria é responsável por características de defasagem do sinal. Esta propriedade pode ser explorada para a quantização de imagens em aplicações de compressão e, também, para processar bordas usando extensão simétrica.

### 2.2.1 Famílias de funções

A possibilidade de construir funções wavelets é muito grande, fato percebido pela quantidade de funções já existentes. As funções costumam ser separadas em famílias que compartilham propriedades semelhantes. A opção por uma função ou conjunto de funções em geral está condicionada à aplicação.

As famílias wavelets são divididas conforme as características ou propriedades que compartilham [55, 63]. Uma das formas de classificar tais famílias refere-se a elas serem ortogonais ou biortogonais.

## Wavelets ortogonais

Se para o conjunto de funções  $\{\psi_{j,k}\}$ , o produto interno para qualquer par  $\langle \psi_{j,k}, \psi_{j,l} \rangle = 0$ ,  $k \neq l$ , então a transformada wavelet é chamada ortogonal. O conjunto de base neste caso é igual ao seu conjunto dual. Entre as famílias de wavelets ortogonais estão as famílias Daubechies, Symlets, Coiflets, entre outras. Wavelets ortogonais não podem ser simétricas, exceto pela wavelet de Haar. No entanto, é possível projetar wavelets com maior ou menor assimetria.

As características das wavelets de Daubechies são analisadas de acordo com a ordem  $N$  das suas funções [63]. O suporte das funções  $\phi$  e  $\psi$  da Daubechies  $dbN$  é dado por  $[0, 2N - 1]$  e o número de momentos de desvanecimento é igual a  $N$ . A maioria destas funções é não-simétrica exceto pela  $db1$  que é a mesma wavelet de Haar. A regularidade aumenta conforme a ordem da função.

Apesar do nome Symlets, as wavelets  $symN$  são apenas quase-simétricas. Ingrid Daubechies propôs mudanças em suas wavelets de modo que fosse diminuída a assimetria. Fora a simetria, as outras propriedades das Symlets são semelhantes às das Daubechies.

As Coiflets de ordem  $N$ ,  $coifN$ , possuem funções  $\psi$  com  $2N$  momentos iguais a zero e funções  $\phi$  com  $2N - 1$  momentos iguais a zero. O suporte destas funções é igual a  $[0, 6N - 1]$ . As funções  $\phi$  e  $\psi$  das Coiflets possuem maior simetria em relação às mesmas funções Daubechies.

## Wavelets biortogonais

Para as wavelets biortogonais, o conjunto de base e o conjunto dual são diferentes. Essas wavelets apresentam duas ordens,  $Nr$  e  $Nd$ , que estão relacionadas às bases e suas duais [55]. As vantagens em relação às wavelets ortogonais consiste em uma maior flexibilidade das biortogonais. Essas wavelets podem possuir simetria e perfeita reconstrução, algo incompatível para o caso ortogonal.

São usados pares de funções de escala,  $\phi$  e  $\tilde{\phi}$ , e wavelets,  $\psi$  e  $\tilde{\psi}$ . As ordens  $Nr$  e  $Nd$  estão relacionadas à reconstrução e à decomposição. Os suportes são  $[0, 2Nr + 1]$  e  $[0, 2Nd + 1]$ ,

enquanto os momentos de desvanecimento para a decomposição são  $Nd$ .

## 2.2.2 Análise de multirresolução

A análise de multirresolução em wavelets foi formulada em 1986, em trabalhos de Mallat e Meyer [53]. O método consiste em representar funções como um conjunto de coeficientes que fornecem informação sobre a posição e a frequência da função em resoluções diferentes.

A multirresolução é bastante útil para análise de imagens. Ela permite que objetos difíceis de serem identificados em uma determinada resolução possam ser identificados a partir de uma resolução mais apropriada, seja esta mais alta ou mais baixa. Em uma imagem, esses objetos são entendidos como regiões que possuem texturas ou pixels com intensidades semelhantes. Objetos maiores possivelmente não necessitam de uma resolução muito alta para serem identificados, já para objetos pequenos uma resolução alta pode ser necessária. A idéia é poder transitar entre diferentes resoluções em busca de melhores análises.

A análise de multirresolução consiste em um conjunto de espaços de aproximação sucessivos  $V_j$ ,  $j \in \mathbb{Z}$ , que satisfazem algumas condições. A primeira, mostrada pela equação 2.6, expressa que, ao passar de um espaço de resolução  $V_j$  para um espaço  $V_{j+1}$ , novos detalhes são acrescentados à aproximação de um sinal. Por outro lado, à medida que uma função  $f(t)$  é aproximada por níveis de resolução menores, como, por exemplo, passando de  $V_j$  para  $V_{j-1}$ , informação é perdida. A diferença de informação entre duas resoluções de um sinal é chamada de *sinal de detalhe* [59].

$$\dots \subset V_{-2} \subset V_{-1} \subset V_0 \subset V_1 \subset V_2 \subset \dots \quad (2.6)$$

Através da equação 2.6, tem-se que os espaços são relacionados por

$$f(t) \in V_j \iff f(2t) \in V_{j+1} \quad (2.7)$$

o que significa que elementos em um espaço são versões escaladas no próximo espaço.

Cada subespaço  $V_j$  deve ser gerado através de todas as translações inteiras de uma única

função  $f(t)$ , verificando-se que

$$f(t) \in V_j \iff f(t - k) \in V_j, \quad \text{com } k \in \mathbb{Z}. \quad (2.8)$$

Como consequência das equações 2.7 e 2.8, surge a necessidade de se definir em  $V_j$  uma função que desempenha um papel fundamental na análise de multirresolução e na geração de sistemas wavelets. Esta função é denominada *função de escala*, sendo comumente representada na literatura por  $\phi(t)$ .

Para a análise de multirresolução, o conceito de escala é bastante importante. Diferentes escalas permitem a visualização de diferentes detalhes de um objeto observado. Em um mapa, usando uma escala pequena, são percebidas características macroscópicas da região detalhada, enquanto que em escalas maiores podem ser representados detalhes menores [34]. Qualquer função definida em  $V_j$  pode ser expressa como combinação linear das funções que constituem a base do subespaço  $V_{j+1}$ . Em particular, pode-se escrever a função de escala como na equação 2.9, na qual  $2^{j/2}$  é um fator de normalização.

$$\phi_{j,k}(t) = 2^{j/2} \phi(2^j t - k) \quad (2.9)$$

Aqui,  $k$  determina a posição da função e  $j$  a abrangência da função ao longo do eixo considerado. O fator  $2^{j/2}$  controla a amplitude.

Para estender as diferenças entre os subespaços gerados pela função de escala, é preciso definir um segundo conjunto de funções, representadas na literatura por  $\psi(t)$  e determinadas pela equação 2.10. Essas são as funções wavelets.

$$\psi_{j,k}(t) = 2^{j/2} \psi(2^j t - k) \quad (2.10)$$

O espaço vetorial  $W_j$  formado por todas as funções de  $V_{j+1}$  que são ortogonais a todas as funções de  $V_j$ , segundo algum produto interno adotado, é chamado *complemento ortogonal* de  $V_j$ . A partir das funções  $\phi_{j,k}(t)$  de  $V_j$  e  $\psi_{j,k}(t)$  de  $W_j$  é possível construir uma base para

o espaço  $V_{j+1}$ . Deste modo, pode-se definir

$$V_1 = V_0 \oplus W_0 \quad (2.11a)$$

que pode ser estendido para

$$V_2 = V_0 \oplus W_0 \oplus W_1 \quad (2.11b)$$

ou seja

$$V_l = V_j \oplus W_j \oplus W_{j+1} \oplus \dots \oplus W_{l-1}, \quad \text{para } l > j \quad (2.11c)$$

sendo arbitrária a escala  $j$  do espaço inicial.

Qualquer função  $f(t)$  pode ser escrita em termos de uma série através das funções de escala e funções wavelets. A expansão em série é mostrada pela equação 2.12. O primeiro somatório é a função  $f(t)$  em uma resolução menor. O somatório duplo adiciona detalhes de mais alta resolução através dos coeficientes de detalhes, ou coeficientes wavelets,  $d_{j,k}$ .

$$f(t) = \sum_{k=-\infty}^{\infty} c_k \phi_k(t) + \sum_{j=0}^{\infty} \sum_{k=-\infty}^{\infty} d_{j,k} \psi_{j,k}(t) \quad (2.12)$$

### 2.2.3 Bancos de filtros

Pela análise de multirresolução, as transformadas wavelets são semelhantes a um banco de filtros [53] formado por dois filtros:  $\mathbf{l}$  (passa-baixa) e  $\mathbf{h}$  (passa-alta), que formam as bases para as transformadas. *Filtro* é um termo empregado para designar sistemas que alteram apenas algumas freqüências de um sinal de entrada [34]. Em processamento de imagens, o sinal é a própria imagem. A relação entre o cálculo de coeficientes em uma transformada wavelet e bancos de filtros foi mostrada por diversos pesquisadores, entre eles, Mallat e Daubechies.

Duas operações importantes para o estudo de filtros são as operações de *decimação* (*down-sampling*) e *interpolação* (*upsampling*).

A operação de decimação consiste em descartar os termos de uma seqüência  $\mathbf{u}$  cujos índices  $i$  não são múltiplos de um inteiro  $q$ . Para as transformadas wavelets, costuma-se utilizar  $q = 2$ , descartando, deste modo, elementos alternados em uma seqüência. Se a seqüência original,

composta por  $n$  valores, é dada por

$$\mathbf{u} = (u_i)_{i=1}^n = (u_1, u_2, \dots, u_n) \quad (2.13)$$

após a decimação, a nova seqüência  $\mathbf{u}'$  será dada por

$$\mathbf{u}' = (u_{2i})_{i=1}^{\lfloor \frac{n}{2} \rfloor} \quad (2.14)$$

A matriz para este operador é dada por

$$\begin{bmatrix} 1 & & & & \\ 0 & 0 & 1 & & \\ & & 0 & 0 & 1 \\ & & & 0 & \dots \\ & & & \vdots & \ddots \end{bmatrix} \quad (2.15)$$

O operador de interpolação adiciona zeros em uma seqüência de termos [34]. Para  $q = 2$ , o operador de interpolação intercala zeros em uma seqüência. Seja uma seqüência  $\mathbf{u} = (u_1, u_2, \dots, u_n)$ , após a interpolação, a nova seqüência será  $\mathbf{u}' = (u_1, 0, u_2, 0, \dots, 0, u_n)$ .

A matriz deste operador é dada por

$$\begin{bmatrix} 1 & 0 & & & \\ & 0 & 0 & & \\ & & 1 & 0 & 0 \\ & & & 0 & 0 \\ & & & & 1 & 0 & 0 & \dots \\ & & & & & \vdots & \ddots \end{bmatrix} \quad (2.16)$$

A convolução dos filtros permite obter coeficientes em um nível seguinte de resolução. A definição da função de escala  $\phi$  que relaciona a transformada ao aspecto de multirresolução é



dada pela equação 2.17.

$$\phi(t) = 2^{1/2} \sum_k \mathbf{l}(k) \phi(2t - k) \quad (2.17)$$

Para expandir as diferenças entre resoluções e completar a representação wavelet, a função wavelet  $\psi$  é definida na equação 2.18.

$$\psi(t) = 2^{1/2} \sum_k \mathbf{h}(k) \phi(2t - k) \quad (2.18)$$

Os coeficientes de  $\mathbf{l}$  e  $\mathbf{h}$  são dados por

$$\mathbf{l}(k) = 2^{1/2} \int_{-\infty}^{\infty} \phi(t) \phi(2t - k) dt \quad (2.19)$$

$$\mathbf{h}(k) = 2^{1/2} \int_{-\infty}^{\infty} \psi(t) \phi(2t - k) dt \quad (2.20)$$

e relacionados pela equação

$$\mathbf{h}(k) = (-1)^k \mathbf{l}(n - k - 1) \quad k \in 0, 1, \dots, n - 1 \quad (2.21)$$

sendo  $n$  o tamanho do filtro.

Alguns exemplos de filtros são dados abaixo, o filtro de Haar na equação 2.22 e o filtro de Daubechies 2 na equação 2.23. Outros filtros são listados no apêndice A e uma boa referência introdutória é encontrada no trabalho [33].

$$\mathbf{l} = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix} \quad \mathbf{h} = \begin{bmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix} \quad (2.22)$$

$$\mathbf{l} = \begin{bmatrix} \frac{1+\sqrt{3}}{4\sqrt{2}} & \frac{3+\sqrt{3}}{4\sqrt{2}} & \frac{3-\sqrt{3}}{4\sqrt{2}} & \frac{1-\sqrt{3}}{4\sqrt{2}} \end{bmatrix} \quad \mathbf{h} = \begin{bmatrix} \frac{1-\sqrt{3}}{4\sqrt{2}} & -\frac{3-\sqrt{3}}{4\sqrt{2}} & \frac{3+\sqrt{3}}{4\sqrt{2}} & -\frac{1+\sqrt{3}}{4\sqrt{2}} \end{bmatrix} \quad (2.23)$$

Em [12], a partir das equações 2.17 e 2.18 são derivadas as equações

$$c_{j,k} = \sum_m \mathbf{l}(m - 2k) c_{j+1,m} \quad (2.24)$$

$$d_{j,k} = \sum_m \mathbf{h}(m - 2k) c_{j+1,m}. \quad (2.25)$$

que permitem que se trabalhe apenas com os coeficientes da transformada wavelet. O índice  $m$  é responsável por indexar os coeficientes dos filtros e os coeficientes da transformada em um nível  $j + 1$ .

Essas equações mostram que os coeficientes de escala e wavelet em diferentes níveis de escala podem ser obtidos pela convolução dos coeficientes na escala  $j + 1$ . Para isso são usados os coeficientes invertidos no tempo  $l(-t)$  e  $h(-t)$  e então aplica-se a decimação para obter os novos coeficientes na próxima escala,  $j$ . A implementação é mostrada na figura 2.4.

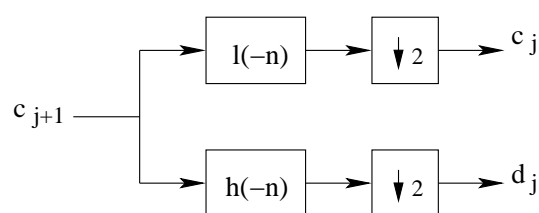


Figura 2.4: Esquema de decomposição usando bancos de filtros.

O processo de filtragem e decimação pode ser repetido nos coeficientes de aproximação, obtendo-se a decomposição em dois níveis mostrada na figura 2.5. Portanto, pela iteração do banco de filtros são obtidos novos níveis de decomposição.

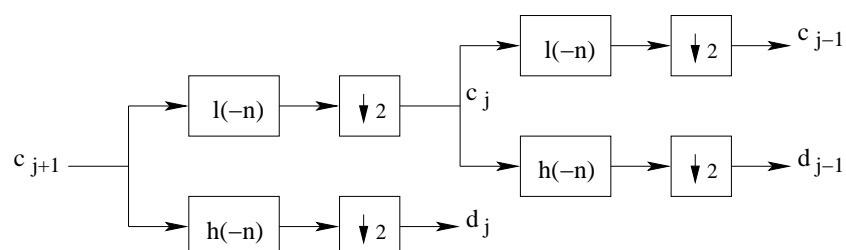


Figura 2.5: Dois níveis de decomposição usando bancos de filtros.

Para a reconstrução obtém-se

$$c_{j+1,k} = \sum_m c_{j,m} l(k - 2m) + \sum_m d_{j,m} h(k - 2m) \quad (2.26)$$

possibilitando a reconstrução da resolução  $j+1$  a partir de coeficientes em uma menor resolução  $j$  [12], com os coeficientes da transformada wavelet indexados pelo índice  $m$ . O processo de reconstrução é realizado pela interpolação seguida da convolução com os filtros  $l(n)$  e  $h(n)$ , como ilustrado na figura 2.6.

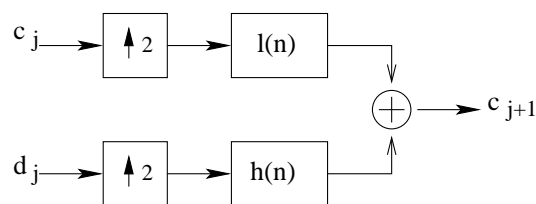


Figura 2.6: Esquema de reconstrução usando bancos de filtros.

## 2.2.4 Lifting

*Lifting* é um esquema para calcular a transformada wavelet discreta. A idéia foi explicada por Sweldens [69]. O esquema foi primeiramente aplicado para sistemas biortogonais e posteriormente estendido para sistemas ortogonais. Existem algumas vantagens que tornam o *lifting* vantajoso, por exemplo, não é preciso memória extra para os cálculos pois a transformação é local e, além disso, a eficiência é aumentada por uma redução de operações [70].

A transformação wavelet por *lifting* (figura 2.7) é tipicamente constituída de três passos: divisão (*split*), predição (*predict*) e atualização (*update*). Esses passos são usados para decompor um sinal  $s_j$ , composto por  $2^j$  amostras, em um sinal de menor resolução  $s_{j-1}$  e um sinal de detalhes  $d_{j-1}$ .

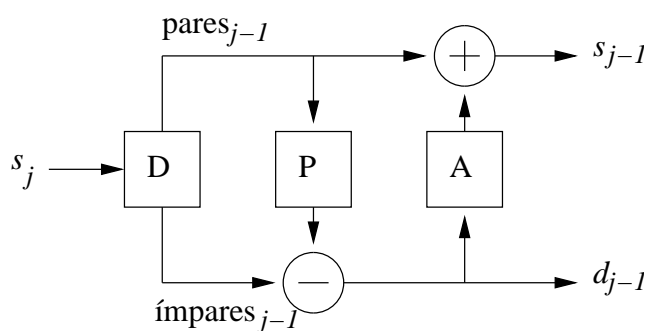


Figura 2.7: Decomposição wavelet pelo esquema de *lifting*.

A divisão consiste em separar os valores de amostra do sinal em dois subconjuntos disjuntos, formados pelas amostras de índices pares,  $s_{2l}$ , e ímpares  $s_{2l+1}$ . Cada um dos subconjuntos contém metade das amostras do sinal inicial. Assim, define-se o operador de divisão como

$$D(s_j) = (\text{pares}_{j-1}, \text{ímpares}_{j-1}). \quad (2.27)$$

O operador de predição permite que valores de um conjunto possam ser obtidos dos valores do outro conjunto, com acurácia, explorando a correlação entre os valores dos dois conjuntos. Usando o conjunto dos pares para prever o valores ímpares

$$d_{j-1} = \text{ímpares}_{j-1} - P(\text{pares}_{j-1}). \quad (2.28)$$

Sinais em uma resolução menor têm o mesmo valor médio do sinal original [70], o que leva à definição do operador de atualização

$$s_{j-1} = \text{pares}_{j-1} - A(\text{ímpares}_{j-1}). \quad (2.29)$$

A reconstrução do sinal é dada pelo esquema inverso mostrado na figura 2.8, novamente composto por três estágios. Na inversão da atualização, os valores pares são recuperados subtraindo a informação da atualização, tendo os sinais  $s_{j-1}$  e  $d_{j-1}$

$$\text{pares}_{j-1} = s_{j-1} - A(d_{j-1}). \quad (2.30)$$

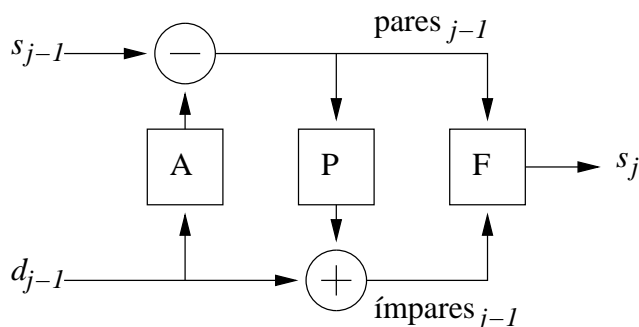


Figura 2.8: Reconstrução wavelet pelo esquema de *lifting*.

A partir dos valores  $\text{pares}_{j-1}$  e  $d_{j-1}$ , os valores ímpares são obtidos adicionando a informação da predição

$$\text{ímpares}_{j-1} = d_{j-1} + P(\text{pares}_{j-1}). \quad (2.31)$$

Por fim, a seqüência original do sinal é recuperada pela fusão dos coeficientes nos dois

subconjuntos, ímpares e pares,

$$F(\text{pares}_{j-1}, \text{ímpares}_{j-1}) = s_j. \quad (2.32)$$

## 2.2.5 Transformadas bidimensionais

As transformações bidimensionais em imagens são em sua maioria obtidas a partir de transformações unidimensionais aplicadas separadamente nas direções vertical e horizontal. Assim, as wavelets são facilmente estendidas para imagens pelo produto de funções de escala e wavelets. Os produtos geram uma função de escala

$$\phi(x, y) = \phi(x)\phi(y) \quad (2.33)$$

e três funções wavelets

$$\psi(x, y)^H = \phi(x)\psi(y) \quad (2.34)$$

$$\psi(x, y)^V = \psi(x)\phi(y) \quad (2.35)$$

$$\psi(x, y)^D = \psi(x)\psi(y). \quad (2.36)$$

As transformadas wavelets bidimensionais obtidas pelo produto das funções unidimensionais são chamadas *separáveis*. Essas funções wavelets medem variações em diferentes direções:  $\psi(x, y)^H$  mede variações horizontais,  $\psi(x, y)^V$  mede variações verticais e  $\psi(x, y)^D$  mede variações nas diagonais.

As funções de base agora são denotadas por um parâmetro de escala  $j$  e parâmetros  $k$  e  $l$  para as translações horizontal e vertical, respectivamente,

$$\phi_{j,k,l}(x, y) = 2^{j/2}\phi(2^j x - k, 2^j y - l) \quad (2.37)$$

$$\psi_{j,k,l}^{or}(x, y) = 2^{j/2}\psi(2^j x - k, 2^j y - l) \quad or = H, V, D. \quad (2.38)$$

A aplicação dos filtros sobre uma imagem nas direções vertical e horizontal gera um nível

de decomposição e produz quatro sub-bandas, LL, LH, HL e HH, como pode ser visto nas figuras 2.9 e 2.10. A decomposição pode ser realizada recursivamente na sub-banda LL (figura 2.10(b)), obtendo níveis adicionais de decomposição.

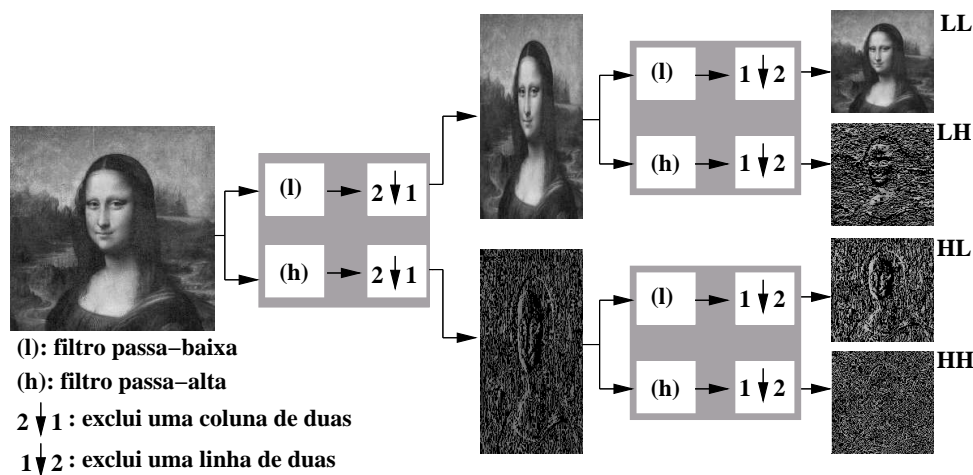


Figura 2.9: Decomposição de um nível da imagem Mona Lisa utilizando o filtro de Haar. As sub-bandas de detalhes foram equalizadas para uma melhor visualização.

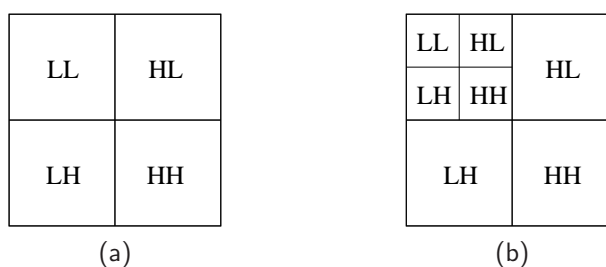


Figura 2.10: Transformadas bidimensionais. (a) decomposição em um nível (b) decomposição em dois níveis.

As sub-bandas LL e HH denotam, respectivamente, as freqüências baixas e altas da imagem, enquanto LH e HL descrevem as freqüências intermediárias presentes na imagem. As sub-bandas LH, HL e HH correspondem às imagens de detalhe; enquanto que a sub-banda de baixa freqüência, LL, é a aproximação da imagem em uma resolução menor, estando relacionada à informação espacial.

Há dois métodos para a aplicação dos filtros QMF em imagens, chamados *decomposição padrão* e *decomposição não-padrão*. Na decomposição padrão, aplica-se recursivamente a transformada unidimensional para cada linha da imagem até que reste apenas uma coluna

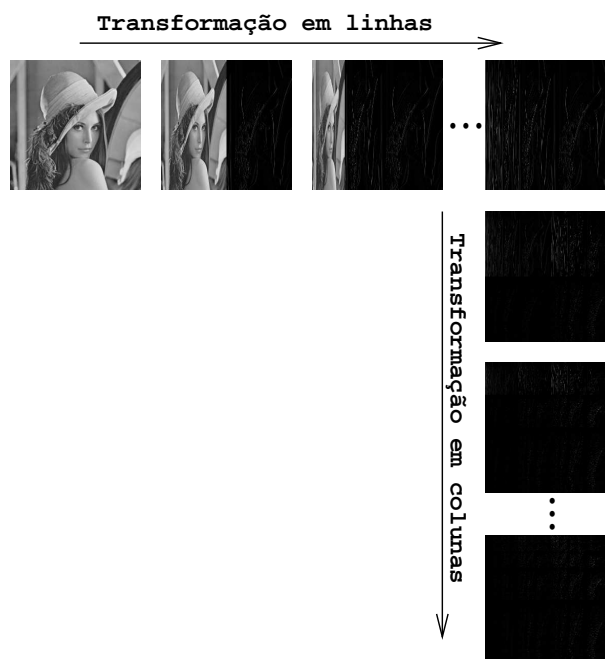


Figura 2.11: Transformada bidimensional padrão.

com coeficientes de escala e o restante com coeficientes wavelets. O mesmo processo é aplicado para cada coluna. O resultado é um único coeficiente global de escala com o restante sendo coeficientes de wavelets. Um exemplo deste método é mostrado na figura 2.11.

A decomposição não-padrão é realizada aplicando-se as operações em linhas e colunas, alternadamente, até que reste apenas um único coeficiente global de escala com o restante dos coeficientes sendo de wavelets. Um exemplo é mostrado na figura 2.12.

Muitos dos esforços de pesquisa em wavelets bidimensionais baseiam-se em filtros separáveis, construídos por produtos tensoriais dos filtros unidimensionais. Mais recentemente, a procura por filtros *não-separáveis*, verdadeiramente bidimensionais ou mesmo multidimensionais, tornou-se o objetivo de algumas pesquisas [21, 42, 47, 49, 50]. No entanto, a construção de tais esquemas é bem mais complexa do que no caso unidimensional, além de exigirem maior custo para execução computacionalmente.

Esquemas não-separáveis permitem a obtenção de sistemas mais adaptados ao sistema visual humano [48], proporcionando uma solução para a falta de isotropia inerente aos esquemas separáveis. A decimação no esquema separável é realizada nas direções vertical e horizontal, removendo metade dos valores em cada dimensão. O problema é que o sistema visual humano



Figura 2.12: Transformada bidimensional não-padrão.

é mais sensível a mudanças na vertical e horizontal do que na diagonal. A decimação dos esquemas não-separáveis difere-se dessa remoção na vertical e horizontal, mantendo importantes informações visuais.

A principal diferença em relação aos esquemas unidimensionais consiste na utilização de *reticulados* (*lattices*), que são subespaços discretos formados por todos os vetores gerados por uma matriz  $D$ . Essa matriz é a responsável pela amostragem no novo esquema, ou seja, pela decimação e interpolação. A matriz  $D$  não é única. Um esquema de amostragem bastante utilizado é o chamado *quincunx*, para o qual  $|\det D| = 2$ , o que significa que este é um caso bidimensional não-separável de dois canais [47]. Alguns exemplos de matrizes para o esquema *quincunx* são

$$D_1 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \quad D_2 = \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix}. \quad (2.39)$$

O reticulado da matriz  $D_1$  é formado pelos vetores  $[1 \ 1]^t$  e  $[1 \ -1]^t$ , enquanto que o reticulado da matriz  $D_2$  é formado pelos vetores  $[1 \ 1]^t$  e  $[-1 \ 1]^t$ . A amostragem no esquema *quincunx* difere do esquema separável em relação à quantidade de redução. No caso separável



são geradas quatro sub-bandas, cada uma com um quarto do número de amostras anterior. No esquema *quincunx* cada sub-banda tem metade do número de amostras. O fator é de  $1/\sqrt{2}$  em cada direção [2].

Existe um requisito de que as matrizes produzam dilatações em todas as dimensões. Para isso, é preciso que os autovalores das matrizes sejam estritamente maiores do que 1. As funções de escala e wavelet para as transformadas com *quincunx* são semelhantes às das transformadas unidimensionais, porém o valor de escala 2 é substituído pela matriz  $D$ . Portanto

$$\phi(t) = \sum_n \mathbf{l}(n) 2^{1/2} \phi(Dt - n), \quad n \in \mathbb{Z} \quad (2.40)$$

$$\psi(t) = \sum_n \mathbf{h}(n) 2^{1/2} \phi(Dt - n), \quad n \in \mathbb{Z}. \quad (2.41)$$

A decomposição wavelet não-separável é realizada também por filtros, agora bidimensionais, e decimação. No caso separável, a convolução em linhas e colunas é considerada como se a imagem fosse um conjunto de sinais não conectados. No caso não-separável, a imagem é entendida como um conjunto de áreas.

## A transformada vermelho-preta

Na tese [75], desenvolveu-se um esquema de *lifting* para wavelets bidimensionais baseado em reticulados *quincunx*. A divisão (*split*) entre pares e ímpares do esquema unidimensional agora é substituída por uma divisão de tabuleiro de xadrez (*checkerboard*), com posições vermelhas e pretas, formada por duas grades *quincunx*, como mostra a figura 2.13.

Há, portanto, um conjunto com valores vermelhos  $\lambda_j$  e um conjunto com valores pretos  $\gamma_j$ . Os valores pretos são preditos pelos valores vermelhos vizinhos e, com base nos novos valores pretos, os valores vermelhos são atualizados para preservar o valor médio dos dados.

Os dados em menor resolução são agora formados pelo conjunto  $\lambda_j$ . No próximo nível de decomposição não é possível realizar as mesmas operações anteriores. Como pode ser visto pela figura 2.13, os valores do conjunto  $\lambda_j$  estão organizados nas diagonais da grade retangular. A operações devem seguir esse direcionamento. Para isso, novamente são definidos dois conjuntos de valores: o azul  $\lambda_{j-1}$  e o amarelo  $\gamma_{j-1}$ , mostrados na figura 2.14. Agora os

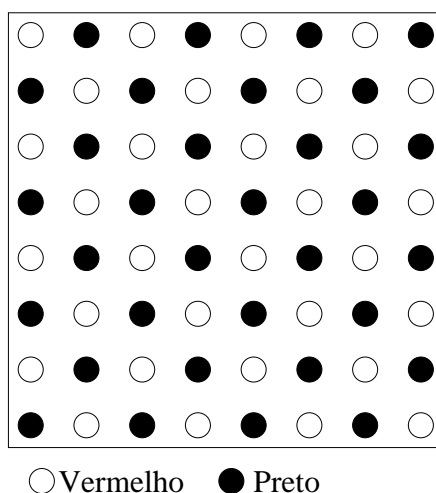


Figura 2.13: Grade retangular composta por duas grades *quincunx*.

passos de decomposição podem ser computados para os valores azuis e amarelos.

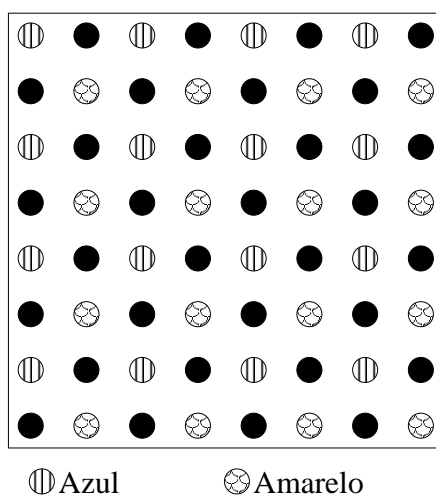


Figura 2.14: Grade diagonal composta por duas grades *quincunx*.

Após a decomposição, os valores azuis contêm a aproximação e os amarelos os detalhes. Um próximo nível de decomposição deve ser aplicado sobre o conjunto azul, que assume um posicionamento nas direções horizontal e vertical novamente. Portanto, a próxima decomposição segue o padrão vermelho-preto.

Conclui-se que o algoritmo para a transformação wavelet alterna, em cada nível, operações computadas nas direções horizontal e vertical com operações computadas na diagonal. Com base nessas idéias é possível formular o esquema de *lifting*.

A etapa de divisão consiste em separar a imagem  $\lambda_j$  em conjuntos  $\lambda_{j-1}$  e  $\gamma_{j-1}$  que,

conforme o nível de decomposição, seguirá divisão vermelho-preta da grade na figura 2.13 ou a divisão azul-amarela da grade na figura 2.14.

Na etapa de predição, os valores vermelhos (azuis) são calculados com base nos valores pretos (amarelos)

$$P(x, y) = \sum_{n, m \in \mathbf{h}} \mathbf{h}(n, m) I(x + m, y + n), \quad x \bmod 2 \neq y \bmod 2 \quad (2.42)$$

na qual  $\mathbf{h}(n, m)$  corresponde ao coeficiente indexado pelos inteiros  $n$  e  $m$  no filtro bidimensional  $\mathbf{h}$ . O operador  $\bmod$  retorna o resto da divisão inteira entre os dois argumentos.

Da mesma forma, a atualização é calculada com bases nos coeficientes de um filtro  $\mathbf{l}$

$$A(x, y) = \sum_{n, m \in \mathbf{l}} \mathbf{l}(n, m) I(x + m, y + n), \quad x \bmod 2 = y \bmod 2. \quad (2.43)$$

Exemplos de filtros são mostrados na figura 2.15. A figura 2.15(a) ilustra um filtro usado para um nível de decomposição vermelho-preta e a figura 2.15(b) ilustra um filtro usado para um nível de decomposição azul-amarela. Outros filtros são listados no apêndice A. Os trabalhos [30, 75] apresentam um bom material sobre a transformação e sobre a construção de filtros.

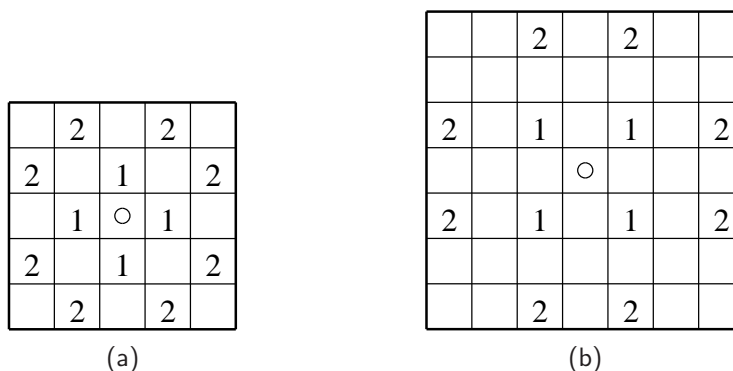


Figura 2.15: Filtros Neville de ordem 4.

## 2.3 Análise de Texturas

Apesar de seu grande uso e de apresentar uma noção bastante intuitiva, não há uma definição precisa para texturas. Informalmente, uma textura está relacionada com as variações, regulares ou aleatórias, nos níveis de cinza ou de cores em uma imagem [36]. Exemplos de texturas são mostrados na figura 2.16.

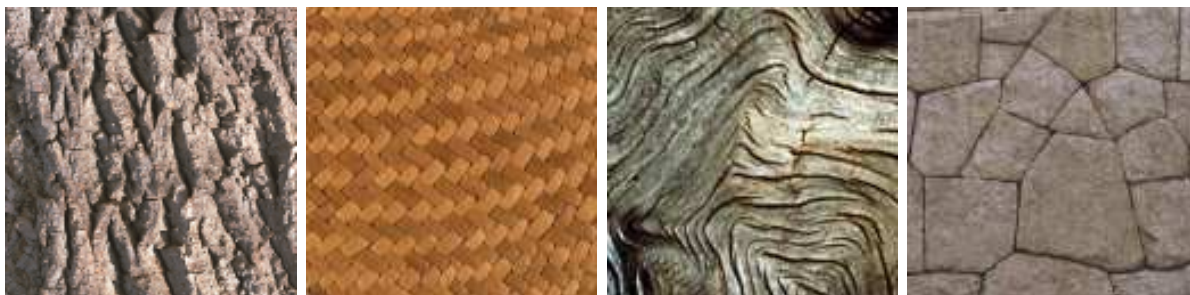


Figura 2.16: Exemplos de texturas.

A análise de texturas está relacionada com tarefas de reconhecimento de padrões, tendo como objetivo tentar descrever e identificar objetos presentes em uma imagem. Para que essas tarefas possam ser realizadas, deve-se representar os objetos através de um conjunto discreto de medidas ou propriedades. Cada uma dessas medidas ou propriedades é chamada *característica* (*feature*):  $x_1, x_2, \dots, x_n$ . O conjunto de características forma um *vetor de características* (*feature vector*),  $\mathbf{x} = (x_1, x_2, \dots, x_n)$ , que representa um padrão em um espaço  $n$ -dimensional, chamado *espaço de características* (*feature space*). Cada vetor de características deve identificar unicamente um objeto ou região de textura. Através de vetores de características, o processo de análise de imagens representa mais sucintamente cada um dos objetos presentes em imagens.

Os métodos de descrição de texturas podem ser divididos em *estatísticos*, *estruturais* e *espectrais*. Os métodos estatísticos dividem as texturas em, por exemplo, suaves e grossas, entre outras possíveis divisões. Nos métodos estruturais, a classificação baseia-se em como são formadas as texturas a partir de elementos mais primitivos, como linhas. O uso de transformações é base para métodos espectrais.

Vários métodos de extração de características têm sido propostos [5]. Inicialmente, a análise de texturas era baseada em medidas estatísticas de primeira e segunda ordem. Então,

foram propostos métodos baseados em Campos Gaussianos Aleatórios de Markov e de Gibbs. Mais tarde, características baseadas em transformações lineares e espectro de texturas passaram a ser descritas.

Uma deficiência da maioria desses esquemas de análise de texturas reside no fato de que a imagem é analisada em uma única resolução [28]. Surgiram, baseados em estudos do sistema visual humano [7, 18], métodos que exploram representações em múltiplas resoluções, chamados multirresolução. Em especial, decomposições realizadas por filtros de Gabor e transformadas wavelets [1, 14, 32, 78]. A teoria de wavelets destacou-se entre os métodos por prover simultaneamente informações sobre os domínios espacial e de frequência [13].

Uma das vantagens das transformadas wavelets sobre a transformada de Gabor reside no fato de que os seus filtros são mutuamente ortogonais. Isto evita as redundâncias entre características de texturas, diferentemente do que ocorre com a transformada de Gabor [4].

Cada uma das texturas presentes em uma imagem pode ser representada a partir de medidas extraídas sobre as sub-bandas de coeficientes da transformada. Os coeficientes da transformada evidenciam os padrões texturais presentes nas imagens. Regiões com texturas suaves possuem principalmente coeficientes baixos. Nas regiões com transições, os coeficientes tendem a ser maiores, enfatizando as mudanças.

Uma característica bastante utilizada para a caracterização de texturas é a *energia*. A energia calculada para uma sub-banda  $Sb$ , adquirida através da transformada wavelet e composta por  $N$  coeficientes, é descrita pela equação 2.44, na qual  $Sb(x, y)$  representa o coeficiente da sub-banda  $Sb$  na coordenada  $(x, y)$ .

$$\text{Energia}_{Sb} = \sqrt{\frac{1}{N} \sum_{x,y} (Sb(x, y))^2}. \quad (2.44)$$

O *desvio médio* (equação 2.45) é uma alternativa à energia, sendo formado pela média dos valores absolutos dos coeficientes em  $Sb$ . O desvio médio é uma medida semelhante à energia mas que evita o cálculo de potências.

$$\text{Desvio Médio}_{Sb} = \frac{1}{N} \sum_{x,y} |Sb(x, y)|. \quad (2.45)$$

Em alguns trabalhos a *entropia* (equação 2.46) medida sobre os coeficientes da transformada é usada.

$$\text{Entropia}_{\text{Sb}} = -\frac{1}{N} \sum_{x,y} |Sb(x,y)| \log |Sb(x,y)|. \quad (2.46)$$

Semelhantes à energia e ao desvio médio são as normas de ordem 1 e 2 dos coeficientes, a diferença consiste em não haver divisão pelo número de elementos em cada sub-banda. A norma de ordem  $r$  é dada pela fórmula

$$\text{Norma}_{\text{Sb}}^r = \sqrt[r]{\sum_{x,y} |Sb(x,y)|^r}. \quad (2.47)$$

Em [26] são apresentadas outras medidas estatísticas de primeira ordem, juntamente com medidas de co-ocorrência como complemento das primeiras. Entre as medidas estatísticas de primeira ordem, podem ser usadas, por exemplo, média, mediana, variância e desvio padrão. As medidas de co-ocorrência auxiliam quando não existe preocupação com invariância à rotação.

Além da média, da mediana, da variação e do desvio padrão, os momentos centrais podem ser utilizados. O momento central de ordem  $r$  é dado por

$$\text{Momento}_{\text{Sb}}^r = \sum_{x,y} (Sb(x,y) - \mu)^r p(Sb(x,y)) \quad (2.48)$$

sendo  $\mu$  a média dos coeficientes na sub-banda  $Sb$  e  $p(Sb(x,y))$  a probabilidade do coeficiente  $Sb(x,y)$ . O primeiro momento é igual a zero, o segundo momento é igual à variância calculada usando um divisor  $n$ , o terceiro momento é a obliquidade e o quarto momento é a curtose.

Pela variância pode-se construir mais uma medida, chamada de suavidade. Essa medida tem valor igual a zero para regiões com coeficientes constantes e cresce tendendo a 1 à medida que a variância cresce. A suavidade é definida como

$$\text{Suavidade}_{\text{Sb}} = 1 - \frac{1}{1 + \sigma^2} \quad (2.49)$$

sendo  $\sigma^2$  a variância dos coeficientes da sub-banda  $Sb$ .

Mais uma medida que pode ser encontrada para caracterização de texturas é o resíduo

médio [61]

$$\text{Resíduo Médio}_{\text{Sb}} = \sum_{x,y} (\text{Sb}(x,y) - \mu)^2 \quad (2.50)$$

sendo  $\mu$  a média dos coeficientes da sub-banda  $\text{Sb}$ .

O *momento* de ordem  $(p + q)$  para uma função  $f(x, y)$  é expresso por

$$m_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^p y^q f(x, y) dx dy \quad p, q = 0, 1, 2, \dots \quad (2.51)$$

A equação 2.52, com centróides  $\bar{x} = m_{10}/m_{00}$  e  $\bar{y} = m_{01}/m_{00}$ , é usada para encontrar os *momentos centrais* para uma sub-banda  $\text{Sb}$ .

$$\mu_{pq} = \sum_x \sum_y (x - \bar{x})^p (y - \bar{y})^q \text{Sb}(x, y) \quad (2.52)$$

Os *momentos centrais normalizados* [35], denotados por  $\eta_{pq}$ , são definidos como

$$\eta_{pq} = \left( \frac{1}{\mu_{pq}} \right)^{\gamma-1} \quad (2.53)$$

em que

$$\gamma = \frac{p+q}{2} + 1 \quad (2.54)$$

para  $p + q = 2, 3, \dots$

A partir dos segundos e terceiros momentos, Hu [45] deriva sete momentos invariantes à translação, rotação e mudanças de escala

$$\omega_1 = \eta_{20} + \eta_{02} \quad (2.55)$$

$$\omega_2 = (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2 \quad (2.56)$$

$$\omega_3 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2 \quad (2.57)$$

$$\omega_4 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2 \quad (2.58)$$

$$\omega_5 = (\eta_{30} + \eta_{12}) + (\eta_{21} + \eta_{03})^2 [(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] + \quad (2.59)$$

$$(3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03}) [3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2]$$

$$\omega_6 = (\eta_{20} + \eta_{02}) [(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03}) \quad (2.60)$$

$$\omega_7 = (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12}) [(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] + \quad (2.61)$$

$$(3\eta_{12} - \eta_{30})(\eta_{21} + \eta_{03}) [(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2].$$

Em muitos problemas de análise de texturas é importante que os resultados sejam independentes quanto à orientação da textura. A transformação bidimensional com filtros separáveis produz transformadas orientadas nas direções horizontal e vertical. Algumas técnicas propostas buscam a solução produzindo filtros com rotações e interpolando suas respostas ou, então, utilizando filtros não-separáveis. Em [26] são apresentados métodos que usam *frames* wavelets isotrópicos e não-isotrópicos obtidos através da discretização de transformadas wavelets contínuas bidimensionais. As características são extraídas dos momentos.

## 2.4 Cores

Texturas e cores são duas das propriedades mais importantes para análise de imagens. O acréscimo de características obtidas a partir de propriedades extraídas das bandas de cores melhora o desempenho dos esquemas de segmentação, em geral, baseados apenas nas informações de níveis de cinza. A combinação dessas informações é ainda pouco explorada. No entanto, características de cores fornecem informações sobre a distribuição espectral complementares às dos níveis de cinza, que representam o total de luz visível na posição de cada pixel [26, 29].

A primeira etapa para a inclusão de características de cores é a escolha de um modelo de cores, o que pode afetar o desempenho da segmentação. Existem diversos modelos descritos na literatura, entre eles, os mais importantes são RGB, CMY, YIQ, HSI, HSV e YCbCr. Alguns modelos são avaliados em [26] para utilização em segmentação de imagens. Embora os modelos RGB e CMY sejam amplamente aplicados com o objetivo de visualização e impressão, respectivamente, nestes modelos as componentes de intensidade e cor não estão separadas. Diferentemente, modelos como YCbCr e HSV, nos quais a intensidade (luminância) e a formação de cores (crominância) estão melhor relacionadas com a percepção visual humana,



permitem explorar mais convenientemente as propriedades de cores. Mais informações sobre modelos de cores e como transformar pixels de um modelo para outro podem ser encontradas em [43].

Algumas abordagens já foram propostas para a junção de características de textura e cor. Muitas das técnicas existentes usam histogramas de cores. Em [17] são usadas características wavelets para texturas e, para cores, um método que agrupa cores dominantes presentes na imagem. Em [26] são extraídas características de cores a partir de coeficientes wavelets. Em [56] são comparados descritores de cores baseados apenas nos valores originais dos pixels ou em coeficientes de transformações wavelets.

## 2.5 Normalização

Características diferentes normalmente assumem valores em intervalos diferentes. Valores maiores tendem a exercer maior influência em funções como, por exemplo, a distância Euclidiana. No entanto, essa influência pode não refletir diretamente a importância da característica para a descrição de uma textura. A normalização das características visa à uniformização das importâncias, fazendo com que as características assumam valores em intervalos similares.

Uma forma direta de realizar a normalização utiliza o cálculo da média e do desvio padrão das várias amostras de uma característica. Para a  $i$ -ésima característica dentre  $l$ , com  $m$  amostras disponíveis, tem-se

$$\begin{aligned}\bar{x}_i &= \frac{1}{m} \sum_{k=1}^m x_{k,i}, & i = 1, 2, \dots, l \\ \sigma_i &= \sqrt{\frac{1}{m-1} \sum_{k=1}^m (x_{k,i} - \bar{x}_i)^2} \\ \hat{x}_{k,i} &= \frac{x_{k,i} - \bar{x}_i}{\sigma_i}.\end{aligned}\tag{2.62}$$

Todas as novas características  $\hat{x}_{k,i}$  possuirão média zero e variância unitária [72]. Esse esquema é linear. Existem esquemas não lineares, baseados em funções como a logarítmica. Um exemplo de esquema não linear, que limita os valores no intervalo  $[0, 1]$ , pode ser obtido

por dois passos, com  $r$  definido pelo usuário

$$\begin{aligned} y &= \frac{x_{k,i} - \bar{x}_i}{r\sigma_i} \\ \hat{x}_{k,i} &= \frac{1}{1 + e^{-y}}. \end{aligned} \quad (2.63)$$

## 2.6 Ordenação de Variáveis

A ordenação de variáveis pode ser considerada como uma primeira etapa no processo de seleção de características, ou mesmo como uma maneira mais simples, menos rígida, de realizar a seleção, cujos principais objetivos são a melhora da classificação das amostras e a diminuição do custo computacional.

A ordenação de variáveis consiste em atribuir valores de importância às variáveis que compõem um vetor de características conforme sua capacidade de discriminar texturas diferentes. A importância das variáveis é atribuída utilizando algum critério avaliativo. Portanto, considerando-se um conjunto de amostras, com cada amostra sendo um vetor de características,  $\mathbf{x}_k$ ,  $k = 1, \dots, m$ , composto por  $l$  variáveis  $x_{k,i}$ ,  $i = 1, \dots, l$  e um conjunto de identificadores de classes  $\mathbf{y}_k$ , uma função de pontuação  $P(i)$  é usada como critério de avaliação para ordenar as variáveis conforme sua importância para a predição das texturas. Há uma diversidade de algoritmos para realizar a tarefa de ordenação de variáveis, usando medidas como a entropia e a correlação [39, 46, 68]. Além disso alguns métodos permitem identificar redundâncias presentes nas variáveis.

Os algoritmos de ordenação abrangem duas categorias principais: os operadores de empacotamento (*wrappers*) e os operadores de filtragem (*filters*). Os primeiros consideram algoritmos de aprendizado para avaliar a qualidade de um subconjunto de variáveis selecionadas, métodos baseados em SVM (*Support Vector Machine*) estão inseridos dentro dos operadores de empacotamento [11, 40]. Os operadores de filtragem não baseiam-se em algoritmos de aprendizagem, são atribuídas pontuações (medida de correlação, medidas de informação) para as variáveis conforme sua capacidade de predição. As variáveis são ordenadas conforme a pontuação. Os operadores de filtragem são algoritmos mais simples e que exigem menos

esforço computacional [39, 65].

No trabalho [23], a ordenação é realizada usando a entropia dos dados. O método é não-supervisionado, ou seja, somente os vetores de características  $\mathbf{x}_k$  são considerados. O algoritmo analisa as variáveis conforme sua importância na separação de amostras em classes diferentes, o que é feito retirando e repondo as variáveis uma a uma. A retirada de uma variável faz com que os dados sejam projetados de um espaço de dimensão  $d$  para um espaço de dimensão  $d - 1$  ( $d \leq l$ ), observa-se, então, se no novo espaço os agrupamentos mantêm-se distintos ou se a retirada provocou alguma confusão na separação. O critério de avaliação é baseado na entropia e dado por

$$\text{ERV} = - \sum_{k=1}^m \sum_{j=1, j \neq k}^m (S(k, j) \log S(k, j) + (1 - S(k, j)) \log (1 - S(k, j))) \quad (2.64)$$

sendo  $S(k, j)$  a similaridade entre  $\mathbf{x}_k$  e  $\mathbf{x}_j$  normalizada entre  $[0, 1]$  e dada por

$$S(k, j) = e^{-\alpha D(k, j)} \quad (2.65)$$

com

$$\alpha = \frac{-\log 0.5}{\bar{D}} \quad (2.66)$$

sendo  $\bar{D}$  a distância média entre os vetores. A distância entre os vetores  $\mathbf{x}_k$  e  $\mathbf{x}_j$  excluindo uma variável do espaço  $d$  é

$$D(k, j) = \left[ \sum_{i=1}^{d-1} \left( \frac{x_{k,i} - x_{j,i}}{\max_i - \min_i} \right)^2 \right]^{1/2} \quad (2.67)$$

sendo  $\max_i$  e  $\min_i$  os valores máximo e mínimo. A variável menos importante é retirada a cada iteração. No final, as variáveis estarão ordenadas conforme a importância para predição das classes.

A abordagem do algoritmo *SPE-ranker* (*simplified polynomial expansion*), apresentado em [65], constrói a ordenação supervisionada usando um critério baseado em correlação. Uma medida estimada da correlação é mostrada na equação 2.68 [39]. O algoritmo é dividido em duas etapas e suas entradas são formadas pelos vetores de características  $\mathbf{x}_k$  e pelo vetor de identificação das classes  $\mathbf{y}_k$ . Na primeira etapa as variáveis recebem pontuações e na segunda

são identificadas redundâncias através da ortogonalização das variáveis.

$$R(i) = \frac{\sum_{k=1}^m (x_{k,i} - \bar{x}_i)(y_k - \bar{y})}{\sqrt{\sum_{k=1}^m (x_{k,i} - \bar{x}_i)^2 \sum_{k=1}^m (y_k - \bar{y})^2}}. \quad (2.68)$$

O *SPE-ranker* permite tanto a avaliação da importância de cada variável para tarefas de classificação quanto a identificação de redundâncias entre as variáveis.

## CAPÍTULO 3

### METODOLOGIA PROPOSTA

Neste capítulo é descrita a metodologia para o desenvolvimento de uma ferramenta de segmentação de imagens. A construção envolve diversas etapas de avaliação, com o objetivo de que cada uma das partes que constituem o método de segmentação explore, da melhor forma possível, os recursos disponíveis na teoria de transformadas wavelets para descrição e segmentação de imagens.

O estudo das transformadas wavelets em processamento digital de imagens envolve aplicações diversas, onde compressão, segmentação, recuperação de imagens por conteúdo e remoção de ruído estão entre os principais tópicos desenvolvidos. A variedade e a quantidade de bases wavelets encontradas em trabalhos relacionados com processamento de imagens formam um conjunto importante para avaliação. O desenvolvimento das bases permite que cada problema explore as wavelets com melhores propriedades para alcançar a solução desejada. Portanto, uma das etapas de avaliação a serem realizadas neste trabalho envolve o estudo e seleção de bases wavelets para segmentar imagens, abrangendo tanto as bases separáveis tradicionais quanto as bases não-separáveis desenvolvidas mais recentemente.

As características para descrição de imagens baseiam-se em informações de textura extraídas dos coeficientes obtidos através de transformadas wavelets. Outra tarefa importante e complexa consiste em definir um conjunto de características capaz de descrever e separar texturas diferentes de forma efetiva. Para a construção de um segmentador robusto deve-se avaliar as capacidades de cada característica. Essa questão, atrelada à questão das bases, permite o estudo de conjuntos apropriados para a descrição de texturas, em especial, a descrição quando estas adquirem rotações dentro da imagem, tornando a tarefa ainda mais desafiadora.

Na busca de um conjunto de características que descreva as regiões em uma imagem de forma robusta, pretende-se testar descritores que utilizem tanto informações de textura em níveis de cinza quanto em cores. A maioria dos métodos de segmentação encontrados na

literatura utiliza apenas informações de níveis de cinza, no entanto, esquemas que agregam características de cores comprovam melhorias na representação das informações de texturas [29]. Sabendo que uma textura não é formada apenas pelos padrões de intensidade luminosa das imagens mas também pelas variações de cores, a definição do espaço de cores influencia no resultado final da segmentação. A escolha do modelo de cores é um parâmetro importante para melhorar a descrição das medidas usadas como características.

A decomposição wavelet pode ser realizada em diversos níveis de resolução, no entanto, muito provavelmente a eficiência da descrição não melhora após determinado nível. Além de não melhorar a descrição, como as características são extraídas para cada uma das sub-bandas nos diversos níveis de decomposição, a dimensão do vetor de características aumenta, fazendo com que o tempo de processamento fique maior. Assim, serão realizadas experiências comparativas com diferentes níveis de decomposição para adequar o método ao problema de segmentação.

Por fim, os resultados das avaliações serão aplicados em um algoritmo de segmentação. Como nem sempre informações prévias sobre as imagens encontram-se disponíveis, o método é modelado de modo a extrair as informações necessárias diretamente da imagem a ser segmentada. Dessa forma, reduz-se de maneira significativa a necessidade de intervenção humana durante o processamento.

Nas próximas seções são descritas as etapas preliminares e, posteriormente, aquelas diretamente envolvidas com a construção do método de segmentação proposto.

### 3.1 Seleção de Bases Wavelets

Esta seção envolve a avaliação da segmentação de regiões de textura em imagens de mosaicos através de diversas bases wavelets. O objetivo principal é a classificação das bases pela capacidade de melhor descrever regiões de textura. A ordenação das wavelets pelas suas capacidades de descrição deve servir como critério para escolha de um subconjunto de bases para uso em segmentação de imagens.

Cada mosaico é composto por  $R$  regiões, de  $N_I \times N_I$  pixels, que pertencem a um conjunto de  $T$  texturas diferentes. O método divide a imagem em um conjunto de blocos de tamanho

$N_B \times N_B$ , com  $N_B \leq N_I$ . Os blocos devem ser discriminados e rotulados como pertencentes a uma das  $T$  texturas que formam cada mosaico.

O desvio médio dos coeficientes em cada sub-banda wavelet, definido pela equação 2.45, é utilizado para formar os descritores dos blocos que subdividem um mosaico. A medida foi escolhida por ser uma das características com maior destaque em segmentação de imagens por wavelets. Para a extração desses descritores, os blocos de textura são decompostos por uma base wavelet, gerando quatro sub-bandas (LL, LH, HL e HH), das quais são extraídos os quatro desvios médios que formam um vetor de características 4-dimensional para cada bloco.

Após possuírem os seus vetores de características  $\mathbf{x}$ , é preciso que todos os  $m$  blocos sejam agrupados pela semelhança de seus vetores. A etapa consiste em agrupar os blocos em  $T$  conjuntos de texturas usando o algoritmo *k-means* (algoritmo 3.1). O método *k-means* agrupa os vetores semelhantes em  $k$  grupos, que no caso são as  $T$  texturas, portanto  $k = T$ . A primeira etapa consiste em inicializar aleatoriamente os centros  $\mathbf{c}_i, i = 1, \dots, k$  de cada grupo  $C_i, i = 1, \dots, k$ . Em seguida, para cada agrupamento são atribuídos os vetores  $\mathbf{x}$  mais próximos de seus centros. Os centros são atualizados e o processo repete-se até que nenhum vetor mude de agrupamento. O resultado é então confrontado com as  $R$  regiões do mosaico original para calcular a porcentagem de blocos corretamente rotulados.

```

escolhe aleatoriamente  $k$  centros iniciais  $C = \mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_k$ ;
repita
  para cada  $i$  de 1 até  $k$  faça
    atribui ao agrupamento  $C_i$  todos os vetores  $\mathbf{x}$  mais próximos de  $\mathbf{c}_i$  do que de
    qualquer  $\mathbf{c}_j$  ( $\mathbf{c}_i \neq \mathbf{c}_j$ );
  fim
  para cada  $i$  de 1 até  $k$  faça
    recalcula o centro  $\mathbf{c}_i$  como  $\mathbf{c}_i = \frac{1}{|C_i|} \sum_{\mathbf{x} \in C_i} \mathbf{x}$ ;
  fim
até  $C$  não mudar ;

```

Algoritmo 3.1: Técnica de agrupamento *k-means*.

As etapas de divisão da imagem em blocos, extração de seus descritores, agrupamento e verificação da porcentagem de acerto devem ser realizadas utilizando diferentes famílias e bases wavelets. O último passo consiste em ordenar as bases por ordem de acerto para selecionar um subconjunto com as bases mais eficazes.

## 3.2 Ordenação de Características

O conjunto de características que podem ser escolhidas para a formação do descritor das regiões em uma imagem é bastante grande. De todas as características mostradas na seção 2.3, algumas já têm o seu uso bastante divulgado, em especial, a energia.

A maioria dos trabalhos que envolvem wavelets para segmentação não apresenta justificativa ou comparação das capacidades descritivas de cada medida. Visando obter uma boa descrição para as regiões presentes em uma imagem, as medidas apresentadas neste trabalho devem ser comparadas usando algoritmos de ordenação de variáveis. Serão realizados os dois tipos de ordenação descritos na seção 2.6, supervisionada e não-supervisionada. O método é esquematizado na figura 3.1.

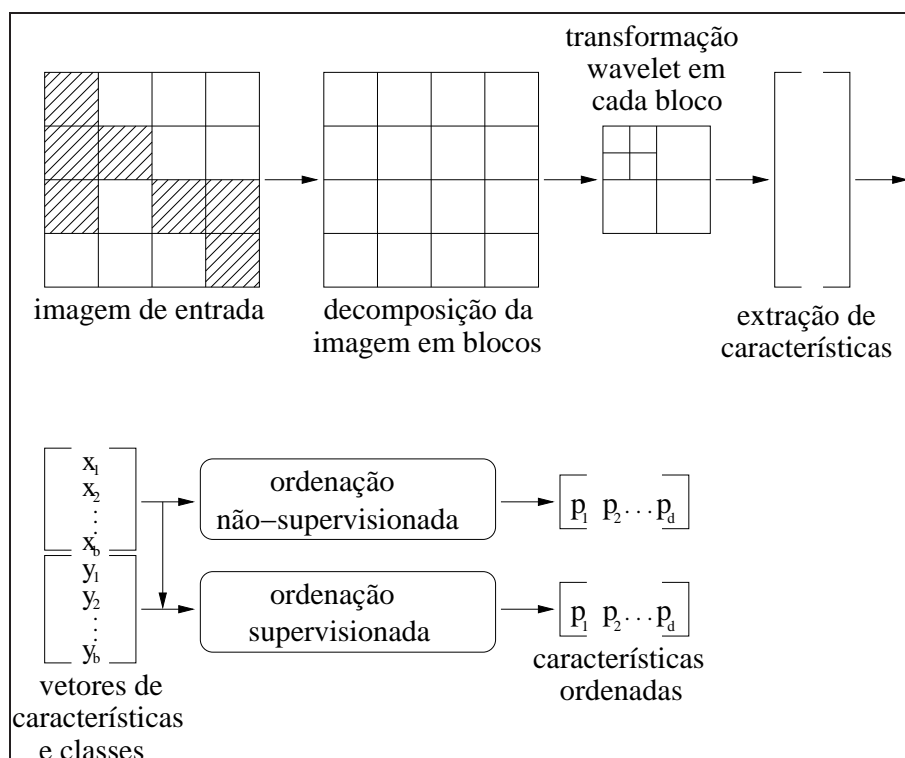


Figura 3.1: Ordenação de variáveis.

Dados mosaicos compostos por  $m$  regiões de textura de tamanho  $N_I \times N_I$ , serão aplicadas transformações wavelets em blocos de tamanho igual ao das regiões de textura ( $N_I \times N_I$ ), com o objetivo de extrair vetores de características para cada uma das regiões da imagem através de transformadas wavelets. Esse conjunto de vetores compostos por  $l$  características,  $\mathbf{x}_{k,i}, k = 1, \dots, m; i = 1, \dots, l$ , será usado como entrada para os algoritmos de ordenação.



O algoritmo supervisionado exige ainda um segundo conjunto de entrada  $y_k, k = 1, \dots, m$ , que relaciona o bloco à sua classe correta.

Primeiramente, a imagem é dividida em blocos de tamanho  $N_I \times N_I$  que são decompostos por uma transformada wavelet. Todas as  $l$  medidas de características são extraídas em cada uma das sub-bandas dos blocos transformados, formando os vetores de características  $x_{k,i}$  dos  $m$  blocos contidos na imagem.

O algoritmo não-supervisionado já pode ser aplicado usando os vetores de características para identificar a ordem de importância das  $l$  variáveis que os compõem. Através de medidas de entropia, cada característica é avaliada conforme sua capacidade de separar texturas diferentes. No fim, as variáveis são retornadas em ordem de importância para diferenciar suas capacidades de discriminação.

Para o algoritmo supervisionado, além dos vetores de características, é preciso tornar disponível o vetor que identifica a classe para cada vetor de característica, o vetor  $y_k$ . Disponíveis as entradas, o *SPE-ranker*, pela correlação das variáveis, identifica as importâncias e retorna as variáveis ordenadas.

A seleção das características que melhor separam texturas diferentes se faz pela avaliação direta dos resultados retornados pelos algoritmos de ordenação.

### 3.3 Invariância à Rotação

Esta seção apresenta o método de avaliação de wavelets e características de texturas para serem usadas como descritores invariantes à rotação. As características são extraídas dos coeficientes de transformadas wavelets. O cálculo das características deve ser feito tomando imagens de uma textura com diferentes rotações, a fim de avaliar a dispersão dos valores à medida em que varia a rotação. Um exemplo de imagem com diferentes rotações é mostrado na figura 3.2.

Existem dois problemas a resolver usando as imagens de texturas com diferentes rotações: o primeiro é a avaliação de quanto uma característica varia, o segundo envolve a avaliação do comportamento de wavelets diferentes. Em especial, a comparação de wavelets separáveis e não-separáveis deve ser realizada. Em teoria, as wavelets não-separáveis proporcionam de-

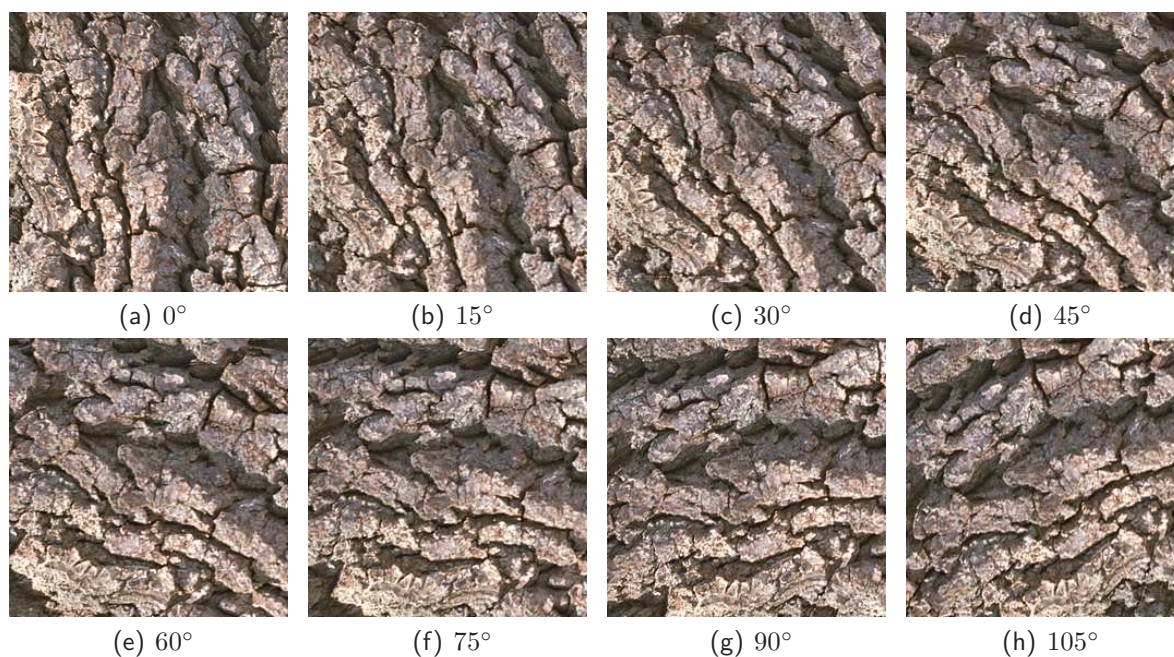


Figura 3.2: Textura em diferentes rotações. As figuras 3.2(a)-(h) mostram a textura variando de  $0^\circ$  até  $105^\circ$  com passo de rotação igual a  $15^\circ$ .

composições mais invariantes para texturas que apresentam rotações, o que se deve ao fato de serem isotrópicas e contornarem os direcionamentos intrínsecos à decomposição bidimensional obtida por wavelets separáveis.

Para o primeiro problema, dada uma imagem de textura de  $N \times N$  pixels, uma transformada wavelet específica de  $l$  níveis é aplicada sobre a imagem. Tomando-se uma das características apresentadas na seção 2.3, o cálculo das características é efetuado sobre as sub-bandas da decomposição. O conjunto de características de cada sub-banda forma o vetor de características que descreve a imagem de textura.

Então, é necessário que a textura sofra uma rotação de  $\alpha$  graus, com  $\alpha$  entre  $0^\circ$  e  $360^\circ$ . Para diferentes valores de  $\alpha$ , o processo de decomposição e extração das características é realizado. Ao final, haverá um vetor de características para cada rotação da imagem. A medida tomada como característica é avaliada quanto a sua sensibilidade à rotação através de uma medida estatística de dispersão. O coeficiente de variação (equação 3.1), razão entre o desvio padrão  $\sigma$  e a média  $\mu$  para um conjunto de valores, é uma medida apropriada, pois permite a comparação direta entre diferentes conjuntos de dados, o que não é possível através

da variação ou do desvio padrão.

$$c_v = \frac{\sigma}{\mu} \quad (3.1)$$

O procedimento deve ser realizado para diferentes medidas de características de maneira que elas possam ser confrontadas para a wavelet escolhida. De maneira análoga, o segundo problema se resolve fixando uma característica e variando as bases wavelets. Resumidamente, fixando um dos parâmetros (medida de característica ou base wavelet) e variando o outro, o processo pode ser descrito pelo algoritmo 3.2.

**para cada**  $\alpha$  de  $0^\circ$  a  $360^\circ$  com incremento  $\beta$  **faça**  
 aplica rotação de  $\alpha$  graus na imagem de textura original;  
 aplica a transformada wavelet sobre a imagem de textura após rotação;  
 calcula as características para as sub-bandas;  
 armazena o vetor de características;  
**fim**  
 calcula o coeficiente de variação dos vetores de características;

Algoritmo 3.2: Cálculo do coeficiente de rotação relativo a texturas com rotação.

Por fim, comparam-se os coeficientes de variação do parâmetro variável para selecionar as wavelets e características que melhor descrevem as texturas com rotações.

### 3.4 Modelos de Cores e Tamanhos de Blocos

A seleção do modelo de cores é importante para a extração de características. As informações de luminância (intensidade de níveis de cinza) costumam ser empregadas isoladamente para caracterizar texturas, no entanto, explorar as informações retidas em bandas de cores complementam e auxiliam na separação de padrões de texturas diferentes.

Os testes devem esclarecer a questão sobre qual modelo de cores é mais apropriado para adquirir o complemento de cores de forma a auxiliar as informações de luminância e proporcionar melhor separação entre texturas.

Como nos testes anteriores, a segmentação baseia-se em blocos, aplicando-se determinada wavelet sobre os blocos e extraíndo-se as características sobre os coeficientes. O agrupamento de blocos semelhantes é obtido para avaliar a separação de texturas proporcionada pelos descritores.

O processo de agrupamento e avaliação da correção é realizado variando o tamanho dos blocos para cada modelo de cores. Assim será possível identificar os modelos mais robustos e também a influência do tamanho dos blocos para uma boa segmentação.

Para que regiões de uma imagem sejam corretamente segmentadas é preciso que o tamanho dos blocos seja grande o suficiente para a obtenção dos padrões texturais.

### 3.5 Algoritmo de Segmentação

A segmentação baseia-se na identificação de regiões em uma imagem que possuem características uniformes. O método de segmentação desenvolvido (proposto em [22]) é composto de duas etapas, como pode ser visto na figura 3.3. O propósito desta divisão é diminuir o custo computacional, realizando uma segmentação inicial em blocos e posteriormente refinando a segmentação ponto-a-ponto. Além disso, a primeira etapa serve para extrair descritores para cada uma das regiões ou objetos presentes na imagem.

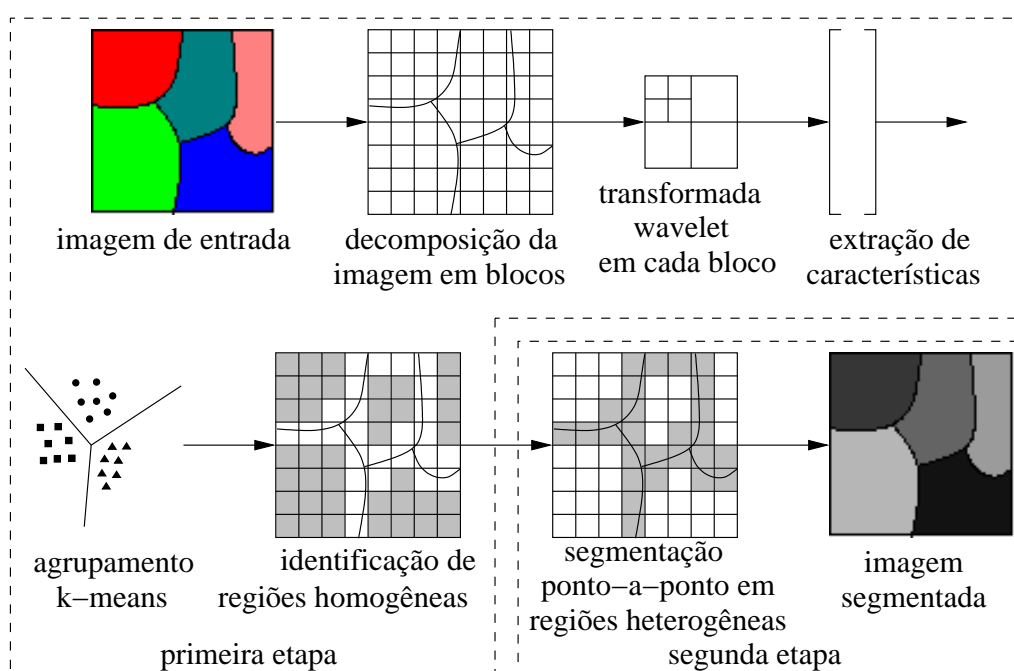


Figura 3.3: Diagrama do método proposto.

A segmentação inicial agrupa, em classes, blocos que possuem texturas homogêneas. O método de agrupamento aplicado é o *k-means*, como alternativa pode ser usado o melhoramento do *k-means* proposto no trabalho [67]. Nesta etapa também são identificados blocos

com características heterogêneas ou que estejam presentes em regiões de transição entre classes. Posteriormente, na segunda etapa, efetua-se uma segmentação ponto-a-ponto sobre estes últimos blocos, obtendo uma segmentação mais fina e eliminando efeitos de blocagem.

Como regiões com texturas homogêneas costumam ser formadas por uma grande quantidade de pixels, a primeira etapa do algoritmo reduz consideravelmente o tempo de processamento. Apesar da primeira abordagem em blocos, o resultado da segmentação não é comprometido. A qualidade é garantida pela identificação de regiões de heterogeneidade. A segmentação final é realizada para prover uma boa delimitação das regiões de heterogeneidade. Embora a etapa final da segmentação seja realizada ponto-a-ponto, o tempo de processamento ainda é baixo, já que esta etapa é realizada sobre um conjunto menor de pixels.

O método de segmentação baseado em duas etapas já foi implementado usando medidas de energia e de estatísticas de primeira ordem. Os artigos [22] e [58] demonstram bons resultados obtidos pela aplicação do método em imagens médicas e de sensoriamento remoto.

### 3.5.1 Segmentação Inicial

A segmentação inicial consiste em particionar a imagem de entrada, com dimensões  $N_I \times M_I$  pixels, em blocos de tamanho  $N_B \times M_B$  pixels, onde  $N_B$  e  $M_B$  são múltiplos de  $N_I$  e  $M_I$ , respectivamente. Para cada um dos blocos que compõem a imagem, tratados como regiões de textura, aplica-se uma transformada wavelet com  $l$  níveis de decomposição. Pelos coeficientes das sub-bandas de detalhe LH, HL e HH, em cada um dos subníveis de decomposição, juntamente com a sub-banda de aproximação LL, extraem-se as características que representam a textura.

Imagens coloridas usam o modelo de cores selecionado a partir da comparação entre RGB, YCbCr, YIQ e HSV proposta na seção 3.4 deste capítulo. Imagens em níveis de cinza usam a banda de intensidade para decomposição e extração das características. Em imagens coloridas, cada uma das bandas relacionadas com o modelo de cor é decomposta e usada para a extração de características. Dessa forma, além das informações de intensidade, o método de segmentação considera a influência das cores para a formação das texturas.

Como não existem informações prévias sobre o conteúdo das imagens, cada classe presente

na imagem deve ser caracterizada por meio de informações extraídas durante o processo de segmentação. Essa é uma das importantes funções da primeira etapa do método, que obtém descritores para os blocos e possibilita que estes sejam agrupados em  $k$  conjuntos com características semelhantes, diferenciando as  $k$  regiões distintas de uma imagem da melhor maneira possível.

O algoritmo *k-means* (algoritmo 3.1) pode ter como resultado apenas um ótimo local, não existindo garantia de obtenção do ótimo global. O resultado retornado pelo algoritmo é dependente dos centros escolhidos aleatoriamente em seu início. Uma maneira comum de tentar contornar o problema é repetir a execução do algoritmo de agrupamento seguidas vezes. No final, toma-se o melhor resultado retornado. O pseudocódigo para a segmentação inicial é mostrado no algoritmo 3.3.

```

divide imagem em blocos de tamanho  $N_B \times M_B$  pixels;
para cada bloco  $b$  da imagem faça
  aplica transformada wavelet sobre  $b$ ;
  extrai vetor de características  $\mathbf{x}_b$  de  $b$ ;
fim
agrupa vetores  $\{\mathbf{x}_b\}$  usando k-means;
identifica blocos de heterogeneidade;
inicia centros das classes com os valores dos blocos não heterogêneos pertencentes às classes;

```

Algoritmo 3.3: Segmentação inicial.

A segmentação por blocos torna a identificação de fronteiras entre classes pouco precisa, produzindo um efeito de blocagem, como mostrado na figura 3.4. A figura 3.4(a) mostra a imagem original e a figura 3.4(b) mostra as regiões segmentadas pela primeira fase. Além da blocagem em fronteiras, é possível que pequenas regiões internas sejam segmentadas erroneamente. Os blocos localizados nessas regiões são detectados para serem segmentados com maior precisão na etapa final.

## Identificação de regiões heterogêneas

Como ilustrado na figura 3.4, a abordagem baseada em blocos da primeira etapa produz efeitos de blocagem na imagem. Por esse motivo, as fronteiras entre regiões diferentes não são

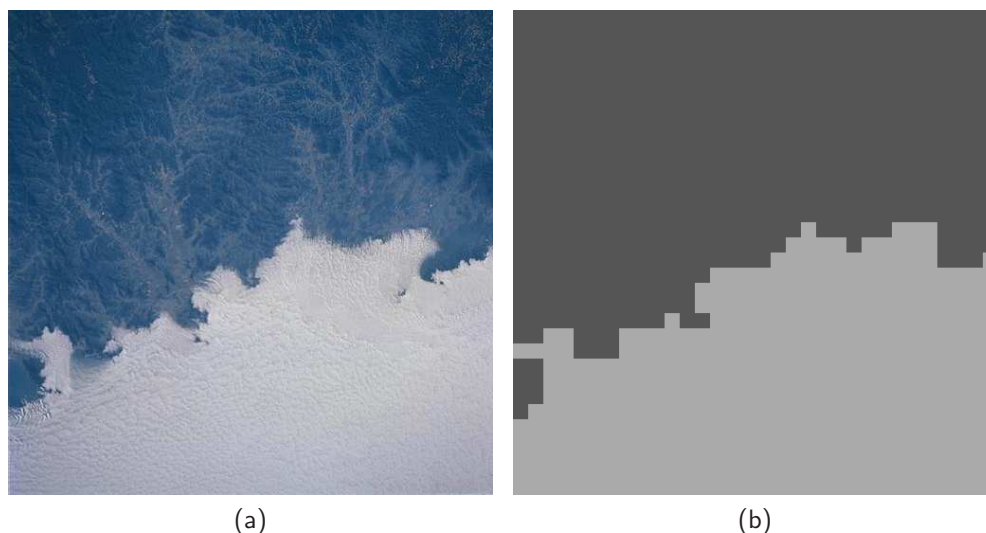


Figura 3.4: Efeito de blocagem produzido pela segmentação inicial baseada em blocos: (a) imagem original e (b) imagem segmentada.

bem delineadas. Blocos situados em fronteiras podem conter pixels pertencentes a diferentes texturas, provocando segmentação errônea, já que todos os pixels dentro do bloco acabam sendo atribuídos para a classe de textura que apresenta maior influência sobre o método de agrupamento.

A influência majoritária de uma textura pode também ocultar regiões ou objetos de pequenas áreas presentes em regiões ou objetos maiores. Dependendo da aplicação, a segmentação correta dessas pequenas regiões é importante. Portanto, a identificação de regiões heterogêneas não deve se restringir apenas às fronteiras de texturas.

A determinação dos blocos pertencentes a regiões de heterogeneidade apresenta mais um fator de importância: como a primeira etapa do método forma descritores para as regiões, a influência de blocos heterogêneos é prejudicial, pois acaba deslocando o vetor de características das classes para longe do que seria o vetor ideal. Blocos heterogêneos devem ser anulados para a computação dos centros de cada classe, visando à obtenção de descritores mais puros e capazes de separar melhor as texturas.

Inicialmente, utilizou-se uma medida de heterogeneidade  $w_i$  proposta em [58]. A medida, que localiza regiões com diferentes texturas é dada pela equação 3.2, onde  $Sb(x, y)$  representa, para o bloco considerado, o coeficiente da transformada na coordenada  $(x, y)$  da sub-banda  $Sb$ , e  $nrl$  é usado para normalizar a medida de heterogeneidade entre zero e um.

$$w_i = \frac{\sum_{LH} |Sb(x, y)| + \sum_{HL} |Sb(x, y)| + \sum_{HH} |Sb(x, y)|}{nr_l} \quad (3.2)$$

Os coeficientes wavelets nas sub-bandas de detalhe (LH, HL e HH) medem as flutuações locais dos valores dos pixels em uma dada escala. É possível usar os coeficientes wavelets de detalhe para caracterizar uma função localmente, bem como localizar contornos e regiões de fronteira entre texturas.

Em regiões homogêneas ou de suavidade, onde os pixels são iguais ou bastante próximos, têm-se que os coeficientes wavelets das sub-bandas LH, HL e HH são nulos ou desprezíveis. Isto se deve à correlação local no espaço, fator presente em uma imagem real. O inverso acontece em regiões de fronteira entre texturas e contornos, onde os pixels da imagem possuem valores bastante variados. Nestas regiões, os coeficientes wavelets são consideráveis e evidenciam as transições.

Caso  $w_i$  seja maior que um limiar determinado experimentalmente, o bloco em questão é considerado durante a segmentação final. Um dos grandes problemas para identificar regiões heterogêneas é a necessidade de estipular limiares para separar o que é heterogêneo e o que é homogêneo. A medida  $w_i$  era bastante dependente da imagem a ser segmentada, havia uma dificuldade real para estipular o limiar.

Uma alternativa apresentada neste trabalho faz uso das informações de distância entre os vetores de características dos blocos em relação aos blocos de sua classe e em relação aos blocos das outras classes. Exploram-se as informações de como o agrupamento foi obtido, no caso, pelo *k-means*. Ainda não é possível eliminar a necessidade de um limiar, no entanto, este se torna um pouco mais estável do que anteriormente.

A nova medida para identificação de heterogeneidade é uma medida de similaridade, ou seja, considera quão similar é um bloco comparado com os blocos em sua classe. A medida assume valores entre  $-1$  e  $1$ . Quanto mais baixo é o valor que a medida assume para um bloco menor a semelhança entre ele e os outros blocos da sua classe. A figura 3.5(a) ilustra a divisão de um conjunto de pontos, representando os vetores de características, em dois agrupamentos retornados pelo *k-means*, um dos agrupamentos é representado pelo símbolo  $\circ$  e o outro pelo símbolo  $\times$ . Os centróides de cada agrupamento são representados pelo símbolo  $*$ .



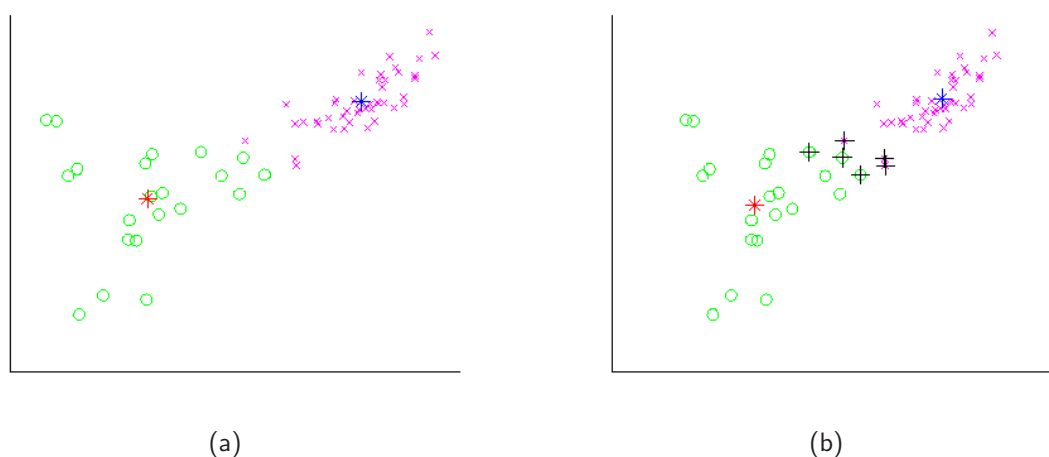


Figura 3.5: Em (a) são mostrados os agrupamentos e os centróides resultantes do algoritmo *k-means*. Em (b), pontos que representam blocos heterogêneos são marcados com o símbolo +.

A medida de similaridade é definida pela equação 3.3, na qual  $S(b)$  é a medida para o bloco  $b$ ,  $D(b, c)$  é a distância média entre o bloco  $b$  e os blocos da classe  $c$ , e  $d(b)$  é a distância média do bloco em relação aos blocos de sua própria classe.

$$\text{Similaridade}(b) = \frac{\min(D(b, c), 2) - d(b)}{\max(d(b), \min(D(b, c), 2))} \quad (3.3)$$

O resultado de  $|S(b) - \mu_c| > L\sigma_c$  é calculado para cada bloco  $b$ , onde  $L$  é um limiar,  $\mu_c$  e  $\sigma_c$  são a média e o desvio padrão das similaridades para o agrupamento ao qual o bloco  $b$  pertence. Se a relação for válida, o bloco é marcado como heterogêneo. A figura 3.6 mostra a identificação dos blocos heterogêneos para a imagem da figura 3.4, marcados em branco.

Blocos heterogêneos são excluídos de suas classes para que seus pixels sejam segmentados na etapa final do método. Com a exclusão, os descritores das classes são atualizados, eliminando a influência de blocos heterogêneos e adquirindo uma medida mais pura para a segmentação final. Na figura 3.5(b), os pontos dos blocos heterogêneos, localizados na fronteira entre os dois agrupamentos, são marcados com o símbolo +. Os pontos centrais dos agrupamentos sofrem um pequeno ajuste em virtude da exclusão dos pontos heterogêneos. O algoritmo 3.4 mostra o pseudocódigo para encontrar os blocos heterogêneos.

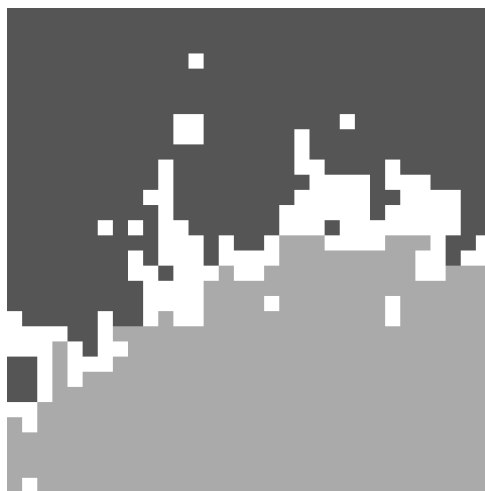


Figura 3.6: Blocos heterogêneos marcados em branco.

```

para cada bloco  $b$  da imagem faça
    calcula  $S(b)$ ;
fim
para cada classe  $c$  faça
    calcula média  $\mu_c$  das similaridades dos blocos de  $c$ ;
    calcula desvio padrão  $\sigma_c$  das similaridades dos blocos de  $c$ ;
fim
para cada bloco  $b$  da imagem faça
    se  $|S(b) - \mu_c| > L\sigma_c$  então
        bloco  $b$  é heterogêneo;
    fim
fim

```

Algoritmo 3.4: Identificação de blocos heterogêneos.

### 3.5.2 Segmentação Final

A segmentação final é uma etapa de refinamento ponto-a-ponto, que determina à qual classe deve pertencer cada pixel contido nos blocos heterogêneos identificados durante a segmentação inicial. O pseudocódigo para a segmentação final é mostrado no algoritmo 3.5.

Nesta etapa são estimados descritores para cada pixel e, a partir destes, o pixel é rotulado para uma das classes criadas na primeira etapa do método. A estimação é feita a partir de uma janela centrada sobre o pixel na imagem original. Aplica-se a transformada wavelet sobre a janela e extrai-se as características de seus coeficientes, formando o vetor de características do pixel.

O vetor de características do pixel é então comparado com os vetores de características

**para cada pixel  $p$  em blocos de heterogeneidade faça**  
amostra uma janela de tamanho  $N_B \times M_B$  sobre a imagem original com  $p$  centralizado;  
aplica transformada wavelet sobre a janela;  
extrai vetor de características  $\mathbf{x}_p$  de  $p$ ;  
atribui  $p$  para a classe  $c$  cuja distância entre  $\mathbf{x}_p$  e  $\mathbf{C}_c$  seja mínima;  
**fim**

Algoritmo 3.5: Segmentação final.

das classes (centros das classes) de texturas descobertas durante a segmentação inicial. O pixel é atribuído à classe cuja distância euclidiana entre os vetores do pixel e da classe seja mínima. A segmentação final da imagem na figura 3.4(a) é mostrada na figura 3.7.



Figura 3.7: Segmentação final.

## CAPÍTULO 4

### RESULTADOS EXPERIMENTAIS

Nesta seção são discutidos os resultados obtidos com a metodologia proposta. O método de segmentação será testado sobre uma base variada de imagens. Os diferentes conjuntos de imagens são importantes por oferecerem a oportunidade de analisar e validar diversos aspectos da metodologia. Os algoritmos foram implementados com o pacote MATLAB [57] versão 7.0.1.24704, no sistema operacional Linux.

Mosaicos de texturas podem auxiliar a avaliação dos descritores usados. Por outro lado, imagens de sensoriamento remoto, entre outras imagens naturais, oferecem importantes aplicações reais. Além disso, esse tipo de imagens contém estruturas e características irregulares que podem ser usadas para avaliar a adaptabilidade do algoritmo.

Os primeiros resultados referem-se ao estudo dos parâmetros para o método de segmentação. O passo inicial constitui uma avaliação de bases wavelets separáveis dentro de um conjunto numeroso, com o objetivo de selecionar um subconjunto de bases capazes de fornecer informações importantes sobre imagens de texturas.

Após a seleção das bases, será realizada a avaliação de características de textura extraídas dos coeficientes das transformadas. O intuito dos testes sobre as características é avaliá-las individualmente para verificar a capacidade de separar texturas diferentes.

Testes de características e de bases mais propensas à descrição de texturas com rotações são realizados na seqüência. O uso de bases não-separáveis é um ponto importante neste estágio. Bases não-separáveis foram construídas tentando-se obter transformações wavelets isotrópicas, superando os direcionamentos verticais e horizontais produzidos pelas transformadas separáveis.

A escolha de um modelo de cores é a última etapa para a calibração do segmentador. A partir de então, os resultados referem-se à segmentação de imagens através do método de duas etapas descrito no capítulo 3.

Para as etapas de avaliação de parâmetros foram construídas imagens sintéticas de mosaicos com o objetivo de facilitar as comparações e a obtenção dos resultados. As texturas que compõem as imagens pertencem à coleção Vision Texture Database [77]. Foram usadas texturas de cortiças, tecidos, alimentos, metais, rochas e madeira.

Cada mosaico construído tem dimensões de  $512 \times 512$  pixels e regiões de textura de  $64 \times 64$  pixels, formando, dessa forma, mosaicos compostos por 64 regiões de textura. O subconjunto de texturas que constituem um determinado mosaico foi tomado aleatoriamente dentro do conjunto disponível, podendo conter texturas pertencentes a grupos de naturezas diferentes. Ou seja, é possível que um mosaico seja formado por uma textura inserida em alimentos e outra inserida em tecidos. O objetivo é avaliar as bases wavelets na caracterização de texturas independente da natureza destas.

Formado o subconjunto de texturas para um mosaico específico, cada uma das regiões de  $64 \times 64$  pixels foi preenchida com uma região de igual tamanho de uma imagem escolhida aleatoriamente dentro do subconjunto formador do mosaico. A região de preenchimento também foi obtida aleatoriamente dentro de cada imagem de textura, evitando a escolha de um região única, o que tornaria o processo de segmentação trivial.

O modo de construção dos mosaicos permitiu que regiões obtidas de uma mesma imagem contivessem variações significativas das posições dos padrões de textura e, também, permitiu que houvesse variações de coloração e luminosidade. Exemplos de mosaicos são mostrados na figura 4.1.

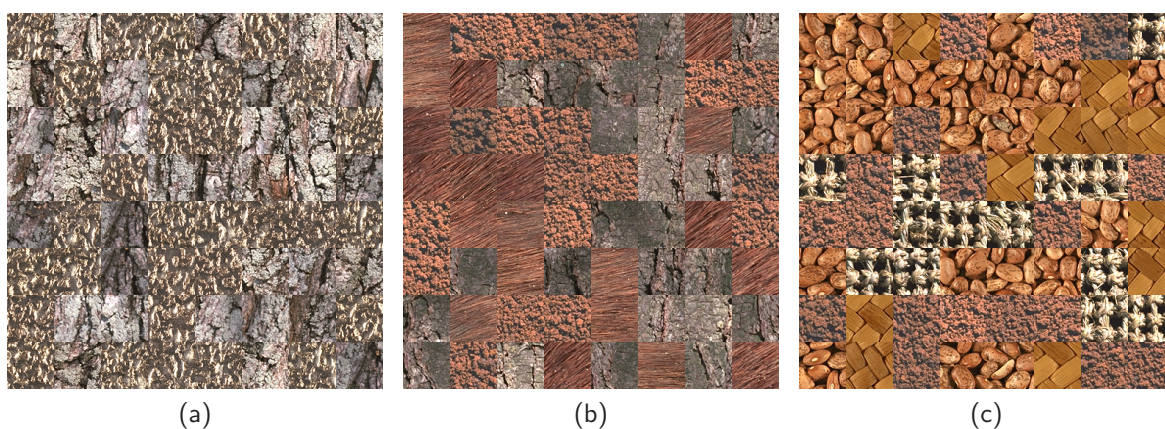


Figura 4.1: Exemplos de mosaicos. Os mosaicos (a), (b) e (c) são compostos respectivamente por 2, 3 e 4 regiões de textura.

É possível observar na figura 4.1(a) que existe variação significativa nos padrões de formação das texturas para cada um dos blocos. Nas figuras 4.1(b) e 4.1(c), além da variação dos padrões de textura, é possível perceber variações de orientação e de luminosidade.

O conjunto construído tem o total de 275 mosaicos compostos por 2, 3 ou 4 texturas diferentes. A divisão exata contabiliza 91, 92 e 92 mosaicos compostos por 2, 3 e 4 texturas, respectivamente. Para facilitar a referência nas seções de teste, os mosaicos formados serão identificados pelas informações mostradas na tabela 4.1.

Tabela 4.1: Grupos de mosaicos para testes.

Grupo	Texturas	Mosaicos
I	2	91
II	3	92
III	4	92

## 4.1 Seleção de Bases

Os testes de discriminação de texturas utilizando diferentes bases wavelets foram aplicados sobre os conjuntos de mosaicos I, II e III identificados na tabela 4.1. O algoritmo foi executado com blocos de tamanho  $64 \times 64$  pixels e também  $32 \times 32$  pixels.

Os 275 mosaicos foram segmentados usando 36 wavelets diferentes, pertencentes a 5 famílias: Haar, Daubechies, Symlets, Coiflets e Biortogonais. Os nomes específicos das wavelets são mostrados na tabela 4.2.

Tabela 4.2: Conjunto de wavelets utilizado nos testes de segmentação.

Família	Wavelets
Daubechies	1 (Haar), 2-10
Symlets	2-8
Coiflets	1-5
Biortogonais	1.1, 1.3, 1.5, 2.2, 2.4, 2.6, 2.8, 3.1, 3.3, 3.5, 3.7, 3.9, 4.4, 5.5, 6.8

Para cada uma das bases foi usado um nível de decomposição wavelet. Os vetores de características foram formados pela medida de energia, dada pela equação 2.44. Como as

wavelets foram executadas com blocos de  $64 \times 64$  pixels e  $32 \times 32$  pixels, o total de testes, para cada wavelet, foi de 550.

A análise dos resultados da segmentação baseou-se nas imagens de referência (*ground truth*) de cada mosaico, ou seja, as imagens verdade que possuem a divisão exata das regiões formadoras das imagens de texturas. As regiões segmentadas foram comparadas com as regiões das imagens de referência. Para todos os mosaicos, as wavelets foram ordenadas da que teve a melhor segmentação para a que teve a pior segmentação, podendo haver empate.

Os resultados finais são resumidos na tabela 4.3. Esta tabela mostra que algumas wavelets biortogonais apresentam os melhores resultados para a descrição de texturas. Em especial, as primeiras cinco colocadas são wavelets biortogonais: bior3.1, bior3.3, bior3.5, bior3.7 e bior1.1.

Tabela 4.3: Número de ocorrências em que cada wavelet obteve a  $n$ -ésima melhor segmentação. Foram realizados 550 testes para cada wavelet. Os resultados referem-se às 25 primeiras posições.

Wavelet	Ordenação																								
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
bior3.1	295	73	40	22	18	20	14	9	3	7	10	2	3	8	1	4	1	1	5	3	5	1	1	0	1
bior3.3	287	95	31	21	13	23	16	12	5	8	5	3	2	1	3	0	5	7	0	4	2	2	1	1	0
bior3.5	262	74	47	30	21	21	11	9	11	8	8	8	5	7	2	2	3	4	4	3	1	1	1	3	1
bior3.7	258	73	34	38	17	19	14	12	13	6	13	8	4	5	7	6	5	0	4	6	1	2	0	2	1
bior1.1	241	79	50	37	31	17	12	16	6	7	7	7	9	2	7	4	5	1	0	0	1	2	2	2	2
db1	237	79	48	37	34	17	13	17	6	7	8	8	8	2	7	3	5	1	0	1	1	2	2	2	1
bior1.5	219	60	40	39	31	23	17	22	14	10	16	8	9	4	7	8	4	4	4	1	3	3	1	0	0
bior2.2	216	76	59	33	41	23	9	19	15	9	7	2	5	6	6	1	2	3	3	4	2	2	1	3	1
sym2	209	63	71	56	20	21	19	15	13	9	7	5	2	6	7	2	5	3	2	3	0	4	2	1	2
db2	206	67	64	58	20	23	20	15	14	9	6	6	4	6	5	3	5	4	2	2	1	3	2	1	1
bior2.6	204	53	30	43	28	23	26	22	15	13	8	9	12	14	9	6	4	8	8	3	4	1	3	1	1
bior2.4	203	63	41	35	33	28	17	18	17	10	16	15	8	10	6	5	4	8	4	2	2	3	0	0	0
sym5	201	54	44	43	33	21	23	17	18	10	18	5	11	5	8	5	4	7	5	5	4	5	1	0	1
coif5	197	79	53	45	35	27	13	20	11	16	6	5	5	10	4	4	3	4	1	2	1	2	2	2	1
sym3	195	60	46	36	32	16	22	23	17	22	16	8	12	6	12	7	5	2	5	0	3	1	2	0	0
bior2.8	195	52	40	44	35	22	16	19	26	10	9	12	12	7	9	7	6	10	2	5	6	1	0	1	0
db3	194	61	46	37	31	17	24	24	17	22	15	5	10	6	13	8	5	1	5	0	2	1	2	1	1
coif1	193	55	51	50	37	28	21	15	25	10	7	14	4	8	7	4	0	5	2	4	3	2	1	1	1
bior1.3	191	55	49	41	23	29	29	16	15	11	14	8	8	12	7	10	6	4	6	6	4	2	2	0	0
sym4	187	51	54	41	25	27	25	17	15	18	12	10	12	8	11	5	7	4	5	2	3	3	3	2	1
db4	186	64	37	33	38	22	16	19	28	13	16	14	12	13	6	7	2	7	3	2	6	1	3	0	0
bior4.4	184	52	42	42	35	26	20	18	15	21	10	14	14	13	5	7	10	5	7	1	3	3	1	0	0
db10	179	68	42	32	36	24	30	25	11	18	16	13	13	4	6	4	5	6	3	5	2	1	1	2	1
sym6	174	50	47	42	25	30	22	22	12	15	12	12	14	9	5	9	10	8	7	9	2	4	4	0	2
coif4	171	48	43	40	42	35	34	29	21	17	12	6	5	11	9	4	4	2	4	5	1	1	2	0	0
db5	169	60	44	34	36	20	12	30	15	12	19	15	16	11	10	10	8	4	9	2	4	4	3	1	0
sym7	167	46	45	42	38	21	23	27	15	13	11	17	15	8	13	10	5	6	7	4	3	4	3	3	0
coif2	166	53	46	45	31	27	26	18	13	16	15	15	12	7	10	9	7	6	7	4	3	4	4	1	2
bior6.8	166	48	39	42	32	30	26	17	21	18	15	11	14	11	7	8	13	9	4	3	3	3	2	1	4
sym8	164	38	51	40	33	24	25	21	21	14	7	21	9	9	9	12	6	9	10	8	3	2	4	3	2
db9	164	66	37	33	25	26	23	22	23	26	9	11	24	5	13	8	8	6	7	3	3	1	1	1	0
db7	161	57	45	32	35	29	19	26	18	13	11	18	14	9	14	7	17	10	1	4	1	1	3	1	1
coif3	159	48	53	33	34	24	28	31	28	13	12	7	11	9	12	14	8	7	4	5	1	3	0	1	2
db8	158	66	45	21	33	26	25	19	20	24	16	12	17	10	11	13	11	6	6	3	1	1	1	0	0
db6	158	61	45	36	31	30	19	16	22	13	12	24	15	12	8	6	12	11	4	3	4	1	2	2	1
bior5.5	157	50	45	41	25	28	23	17	25	12	16	17	12	13	15	12	5	7	6	5	10	3	2	1	0

Nas outras famílias, entre as bases com melhor posicionamento, destacam-se:

- Daubechies: db1 e db2, linhas 6 e 10;
- Symlets: sym2 e sym5, nas linhas 9 e 13;
- Coiflets: coif5, na linha 14.

Na família Daubechies, as bases com melhores resultados foram db1 e db2, as mais simples da família. A base db1 é a base de Haar, a mais simples de todas as bases wavelets e a única com simetria entre as bases ortogonais.

Pelos resultados é possível perceber que não existe uma base que possa ser considerada indiscutivelmente a melhor. Uma base wavelet não mantém constante seu posicionamento em relação às outras bases. De uma imagem para outra, uma mesma base pode obter resultados bastante diferentes, podendo ser a melhor wavelet para uma imagem e figurar entre as piores em outra.

No entanto, resultados extremos não são comuns. As bases tendem a seguir um comportamento relativamente estável. Existe uma flutuação entre as bases, mas aquelas que obtêm melhores resultados tendem a se manter em um grupo líder, possibilitando a escolha final entre as bases com melhores resultados.

Para manter a variação das propriedades das bases wavelets, o subconjunto de bases será escolhido entre todas as famílias. As bases melhor colocadas de cada família e que serão especialmente utilizadas nos próximos testes são:

- Biortogonal: bior3.1, bior3.3;
- Daubechies: db1 (Haar), db2;
- Coiflets: coif5;
- Symlets: sym2, sym5.

## 4.2 Ordenação de Variáveis

Para a aplicação e avaliação das características de textura usando transformadas wavelets foram utilizados mosaicos sintéticos do conjunto explicado no início deste capítulo. Foi considerado o grupo I da tabela 4.1, com 91 imagens formadas por 2 texturas diferentes, já



que os algoritmos de ordenação são lineares. A execução foi realizada com blocos de tamanho  $64 \times 64$  pixels, dessa forma os blocos coincidem com as regiões de textura, não existindo texturas diferentes nas regiões cobertas pelos blocos. Isso permite que os algoritmos de ordenação funcionem sem perturbações, exceto pelas variações encontradas em regiões diferentes de uma mesma textura, como explicado anteriormente quando apresentado o processo de criação dos mosaicos.

O primeiro conjunto de testes foi realizado com o algoritmo de ordenação não-supervisionado. Os testes usaram as seguintes características: desvio médio (dm), energia (en), entropia (ent), normas de ordem 1 e 2 ( $n_1$  e  $n_2$ ), momentos centrais de ordem 2, 3 e 4 ( $m_2$ ,  $m_3$  e  $m_4$ ), resíduo médio (rm), suavidade (sm) e os sete momentos invariantes de Hu ( $im_1$ - $im_7$ ). As medidas foram normalizadas através da equação 2.62, de maneira que todas têm média zero e desvio padrão unitário.

Os resultados para o algoritmo não supervisionado são mostrados na figura 4.2. Conforme a colocação retornada pelo algoritmo para uma variável, foi atribuída a esta um peso de importância, distribuído linearmente conforme o número de características usado, que nos casos de teste somam 21. A variável melhor colocada recebeu o maior peso, 21; a segunda recebeu 20; a última 1. Os pesos foram acumulados para os 91 mosaicos e, por fim, transformados para assumir valores entre  $[0, 1]$ .

Pela figura 4.2, a característica com maior peso para a discriminação dentro do vetor de características foi a variância, com peso aproximadamente igual a 1. Com peso entre 0.8 e 0.9 encontram-se o momento central de ordem 2 e as normas de ordem 1 e 2. Como citado na seção 2.3, o momento central de ordem 2 é a variância dividida pelo número de elementos da sub-banda ao invés do número de elementos menos um. Algumas características bastante empregadas para descrição usando coeficientes wavelets aparecem com peso entre 0.6 e 0.7, como a energia, o desvio médio e a entropia. Com peso semelhante estão o resíduo médio, os momentos centrais de ordem 3 e 4, a norma de ordem 2, a média, o desvio padrão, a suavidade e o primeiro momento invariante. Pouco acima de 0.4 estão somente a mediana e o segundo momento invariante. As outras características tiveram importância menor que 0.4.

De todas as características iniciais, o algoritmo não-supervisionado destacou a variância,

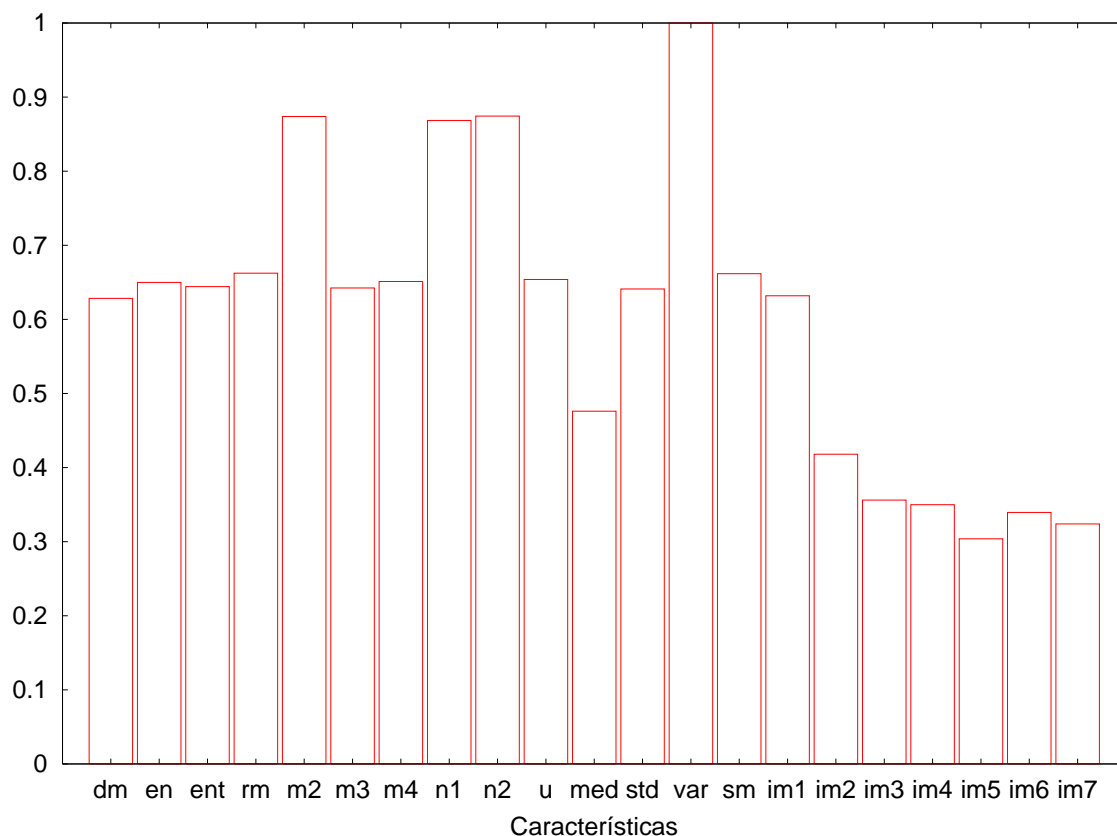


Figura 4.2: Resultado da ordenação pelo método não-supervisionado.

o momento central de ordem 2 e as normas de ordem 1 e 2 como principais características. Em seguida, aparecem as características de energia, desvio médio, entropia, resíduo médio, momentos centrais de ordem 3 e 4, média, desvio padrão, suavidade e primeiro momento invariante.

O segundo teste considera o algoritmo supervisionado. A partir das imagens de referência de cada mosaico foi definido o parâmetro de entrada  $y_k$  que informa as classes às quais cada vetor de característica  $x_k$  pertence. Desta forma, para o algoritmo SPE, cada valor de  $y_k$  foi estabelecido como  $-1$  ou  $1$ , diferenciando as duas classes existentes em cada mosaico.

Da mesma forma como no teste com o algoritmo não-supervisionado, cada característica recebeu um peso de importância (figura 4.3). O maior peso, de aproximadamente 1, foi atribuído à característica de suavidade. Entre 0.9 e 1 destacam-se o desvio médio, energia, entropia, normas de ordem 1 e 2 e desvio padrão. Com aproximadamente 0.9 estão resíduo médio, momento central de ordem 2 e variância. A proximidade entre momento central de ordem 2 e variância fica mais evidente pela grande semelhança de peso. Mesmo sem análise de

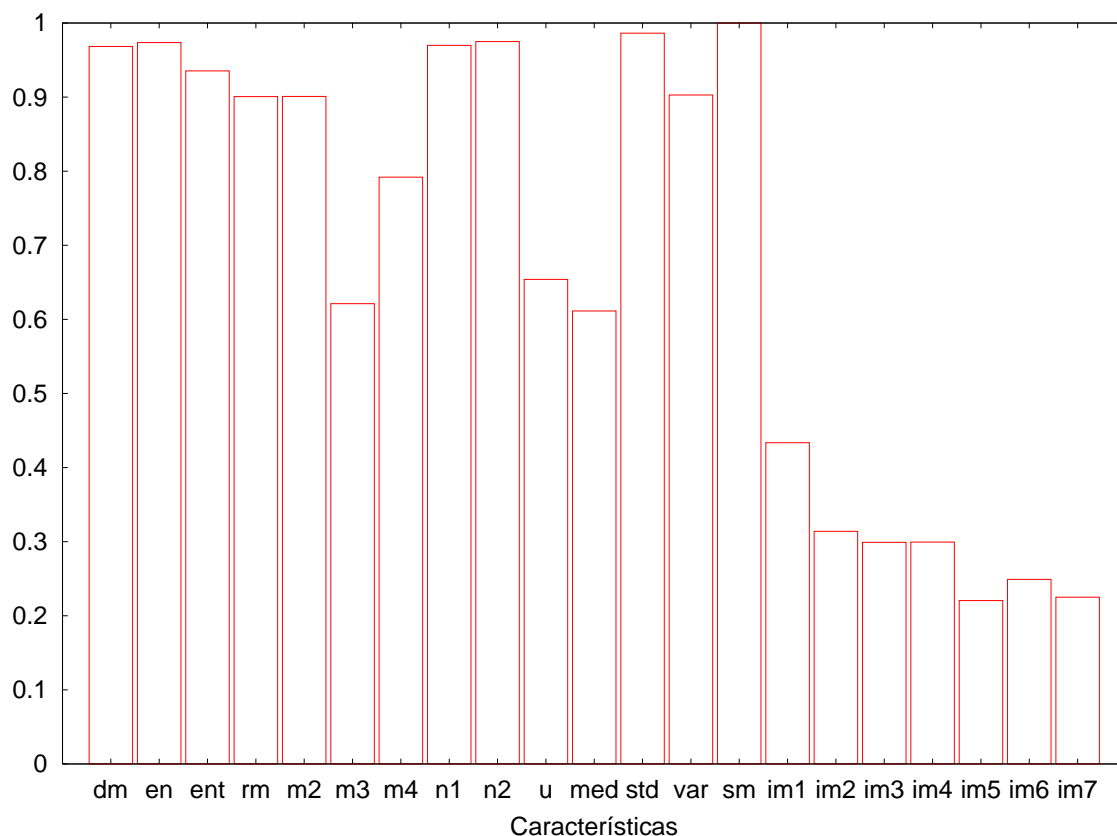


Figura 4.3: Resultado da ordenação pelo método supervisionado.

redundância é possível afirmar que o uso das duas medidas em conjunto não acrescenta muito à capacidade de descrição além da que cada uma faria isoladamente. Com peso pouco inferior a 0.8 aparece o momento central de ordem 4. O momento central de ordem 3, juntamente com a média e mediana aproximam-se com peso 0.6. O restante das características possuem peso inferior a 0.4, exceto o primeiro momento invariante que possui peso pouco superior a 0.4.

Das características com maior destaque para o algoritmo não-supervisionado, a média manteve o valor de importância, a variância e o primeiro momento invariante sofreram queda, enquanto as outras características melhoraram o posicionamento. Ainda assim, a variância obteve resultado satisfatório. Entre energia e desvio médio, duas características semelhantes, a energia manteve-se um pouco melhor. Se uma escolha tivesse de ser feita entre as duas, provavelmente considerações de custo computacional pudessem ser importantes. Não havendo necessidade de diminuir o custo computacional, a energia deve ser usada. Por outro lado, se o custo é importante, o desvio médio, evitando cálculo de potências e raízes, reduziria o tempo

com resultado quase equivalente.

Ambos os métodos de ordenação evidenciam um conjunto de características que produzem melhor separação de texturas, fato importante para confirmar sua importância. As características de variância, desvio padrão e momento central de ordem 2 medem as mesmas propriedades de uma textura; não é necessário que as três sejam utilizadas conjuntamente, basta que apenas uma seja escolhida. O mesmo ocorre com o desvio médio, a energia e as normas.

Os próximos testes ainda consideram as características como parâmetro de escolha. Dessa forma, serão mantidas as redundâncias para a próxima etapa. As características com maior destaque para diferenciação de texturas, sem ordem, foram o resíduo médio, o desvio médio, a energia, a entropia, momento de segunda ordem, normas de primeira e segunda ordem, suavidade, desvio padrão e variância.

### 4.3 Invariância à Rotação

Esta seção considera a adaptação de características e bases wavelets em texturas com diferentes orientações. Os testes foram aplicados em um conjunto de texturas pertencente à coleção Vision Texture Database [77], usando as seguintes imagens de textura: Bark.0008, Bark.0009, Bark.0011, Brick.0001, Brick.0004, Fabric.0000, Fabric.0006, Fabric.0007, Fabric.0009, Fabric.0011, Fabric.0014, Fabric.0015, Fabric.0017, Fabric.0018, Food.0000, Food.0001, Food.0002, Food.0003, Food.0004, Food.0005, Food.0006, Grass.0001, Metal.0000, Metal.0001, Metal.0002, Metal.0004, Misc.0002, Stone.0002, Stone.0004, Tile.0007, Tile.0009, Wood.0001, Wood.0002. A composição das imagens de teste foi feita aplicando rotações sobre a imagem original de 15 em 15 graus e extraíndo a parte central, composta por  $256 \times 256$  pixels em tons de cinza.

As transformadas foram realizadas com wavelets separáveis das famílias Biortogonal (bior), Coiflets (coif), Daubechies (db) e Symlets (sym). Especificamente bior3.1, bior3.3, coif5, db1 (Haar), db2, sym2 e sym5. Os filtros de wavelets não-separáveis utilizados foram Neville 2 (nev2), Neville 4 (nev4), Neville 6 (nev6), Neville 8 (nev8), MaxMax (MM), MaxMin (Mm). Cada uma delas foi decomposta com dois níveis.

As características consideradas foram o resíduo médio, o desvio médio, a energia, a entropia, momento de segunda ordem, normas de primeira e segunda ordem, suavidade, desvio padrão e variância.

Os primeiros resultados comparam a variação de características. Uma wavelet é fixada e as características extraídas. Os gráficos mostram o coeficiente de variação de cada característica para a base wavelet escolhida.

O gráfico da figura 4.4 demonstra os resultados obtidos usando a base wavelet biortogonal bior3.1. Dentre todas, a medida de suavidade tem grande destaque por sofrer pouca variação, bastante abaixo das outras. Enquanto a suavidade possui coeficiente de variação igual a 0.00233, a energia e a norma de ordem 2, que aparecem com a segunda melhor colocação, possuem coeficiente de variação igual a 0.10184, uma aumento bem significativo se comparado com a suavidade.

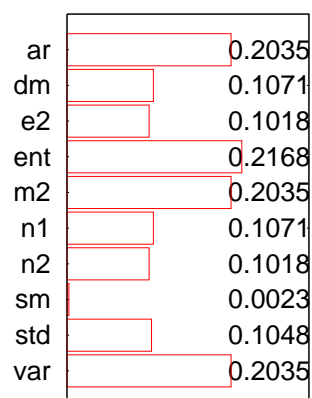


Figura 4.4: Variação de características para imagens com rotações. A base bior3.1 foi utilizada.

Seguindo uma linha de variação bastante similar, encontram-se 5 características: desvio médio, energia, norma 1, norma 2 e desvio padrão. A diferença entre o desvio médio e a energia, bem como entre as duas medidas de norma, é pequena. Como essas medidas são semelhantes, a escolha de uma, caso tenha-se a intenção de evitar redundâncias ou mesmo diminuir a dimensionalidade do vetor de características, pode ser embasada sobre o cálculo computacional. A energia e a norma 2 envolvem cálculos de maior custo mas possuem leve superioridade em relação às outras. Assim, dependendo da aplicação, uma ou outra pode ser

usada.

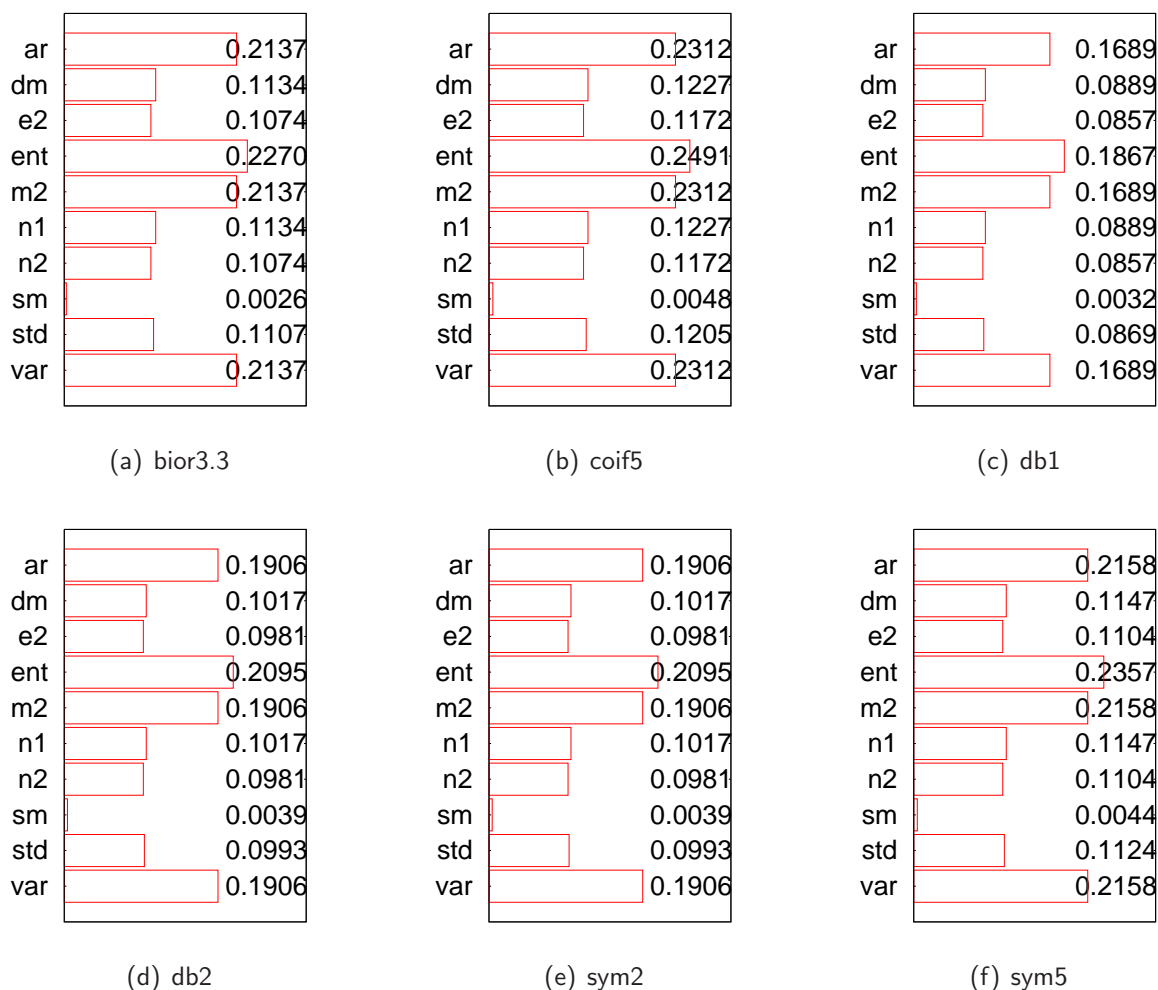


Figura 4.5: Variação de características para imagens com rotações. Wavelets separáveis.

É possível notar que o coeficiente de variação entre o par desvio médio e norma 1, também entre o par energia e norma 2 são idênticos, outro indício da semelhança entre as medidas. O mesmo ocorre entre momento central de ordem 2 e variância. A opção pelo desvio padrão em detrimento à variância é justificável pois o desvio padrão possui dispersão bem menor do que a variância.

Variando ainda mais do que o resíduo médio, o momento de ordem 2 e a variância, encontra-se a entropia. Como o objetivo desta seção é a avaliação e escolha de medidas que melhor se adaptem a imagens nas quais as texturas apresentam rotações, a variação dos coeficientes dessas últimas quatro medidas é proibitiva para seu uso. Para as outras transformadas separáveis o resultado segue o mesmo comportamento descrito acima. Os

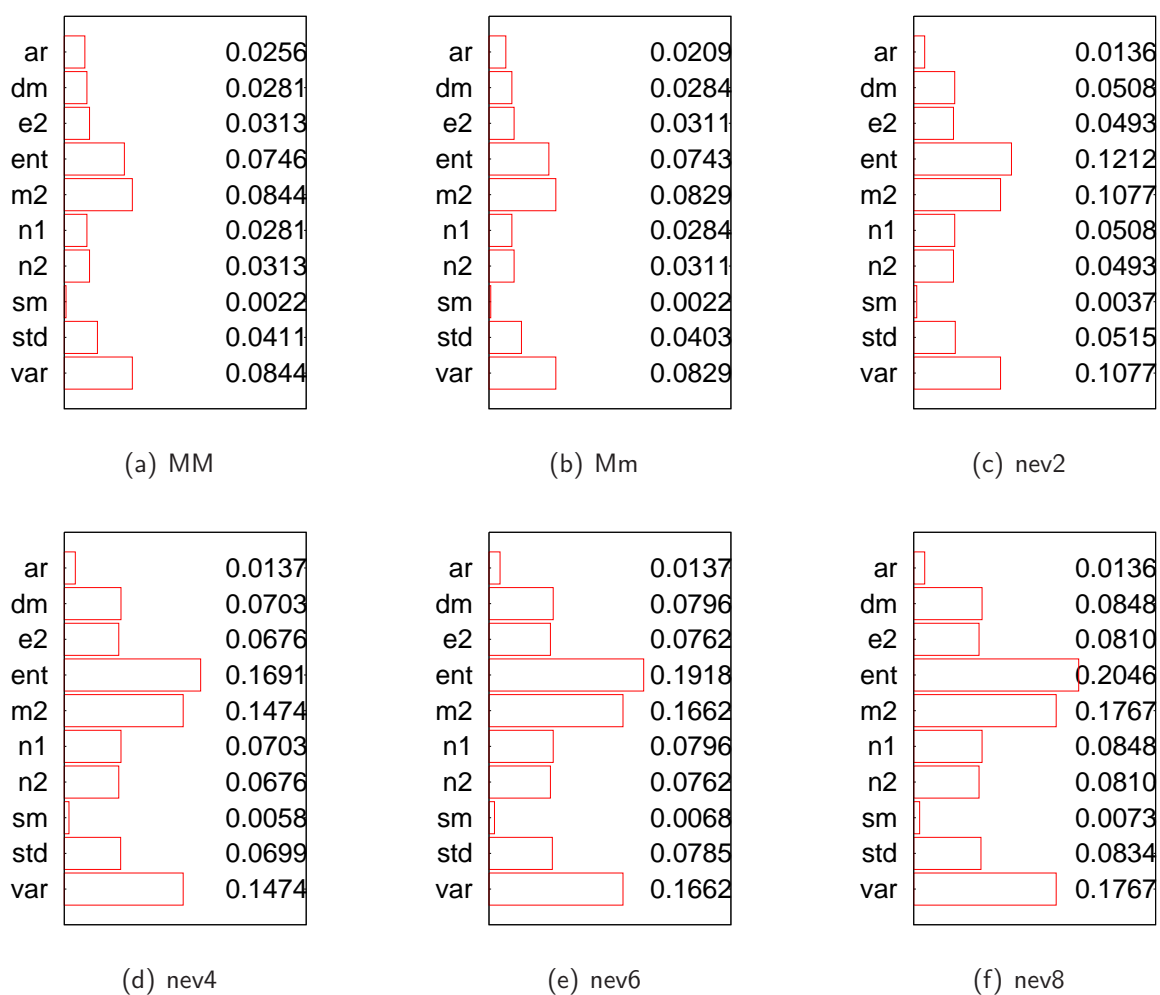


Figura 4.6: Variação de características para imagens com rotações. Wavelets não-separáveis.

gráficos são mostrados na figura 4.5.

O comportamento das wavelets não-separáveis modifica-se um pouco se comparado com as wavelets separáveis. A suavidade continua sendo a medida com menor variação para todas as wavelets. As medidas de desvio médio, energia e normas mantêm a semelhança entre si, mas agora a energia e a norma 2 possuem variação ligeiramente maior. A maior diferença percebida entre os dois tipos de base ocorre para a medida de resíduo médio, que em bases separáveis apresentava resultados ruins e em bases não-separáveis um dos melhores resultados.

A entropia, o momento central de ordem 2 e a variância mantiveram os maiores desvios. No entanto, para as bases MaxMax e MaxMin, a variação é bem inferior à observada usando wavelets separáveis.

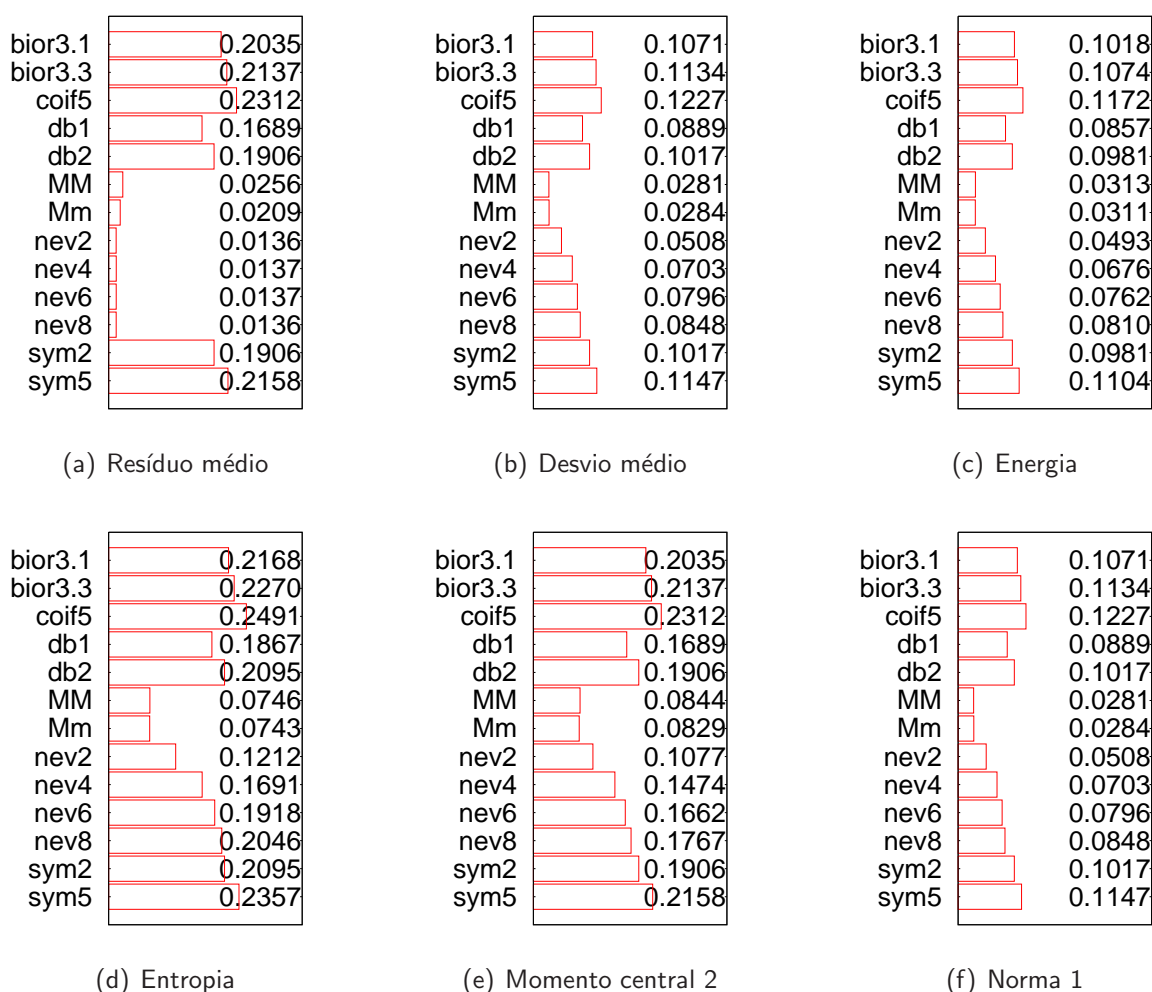


Figura 4.7: Variação de wavelets para imagens com rotações.

O próximo conjunto de resultados complementa os resultados anteriores, agora avaliando as wavelets mais apropriadas para conseguir descritores quando as texturas apresentam rotações.

O gráfico da figura 4.7(a) mostra os resultados para o resíduo médio. As wavelets não-separáveis foram as únicas a obterem resultados baixos, entre 0.01357 e 0.02556. Os resultados para as wavelets separáveis ficaram entre 0.16886 e 0.23124. Somente para o resíduo médio os filtros de Neville apresentaram os melhores resultados dentre os filtros não-separáveis.

Pode ser observado a partir dos gráficos das figuras 4.7 e 4.8 que as wavelets com melhores resultados são as wavelets MaxMax e MaxMin. Já para os filtros de Neville, à medida que aumenta a ordem do filtro, aumentam também os coeficientes de variação. As transformadas nev2 até nev4 possuem valores relativamente próximos aos de MM e Mm. Os filtros nev6 e nev8 se aproximam mais dos filtros separáveis.



Os resultados dos gráficos na figura 4.8 mostram dois fatos importantes. Para o desvio padrão, todas as wavelets têm coeficientes de variação menores que 0.10 e relativamente semelhantes. A outra característica importante é a suavidade, os coeficientes de variação são bastante pequenos para qualquer transformada, todos iguais a zero até a segunda casa decimal.

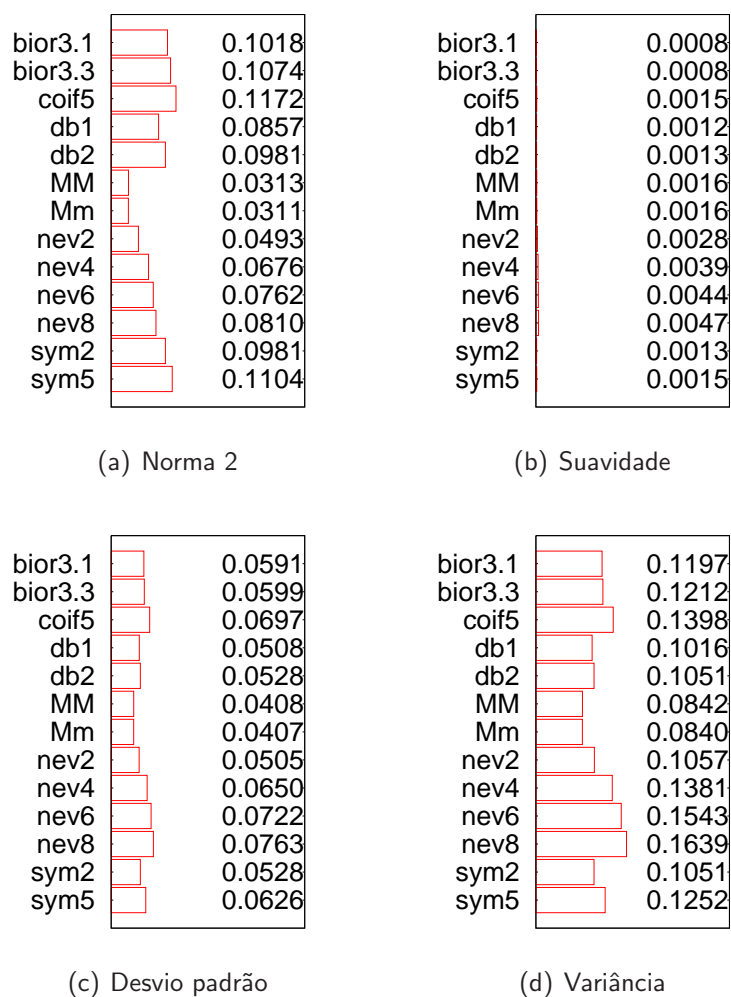


Figura 4.8: Variação de wavelets para imagens com rotações.

A partir dos gráficos, a característica menos sujeita à variação é a suavidade. Da mesma forma como são semelhantes para a separação de texturas, o desvio médio, a energia e as normas comportam-se de maneira equivalente em relação à rotação. O desvio padrão, no entanto, mostrou-se uma característica menos suscetível às mudanças de rotação do que a variância e o momento central de ordem 2. Quando a transformada wavelet é obtida por filtros não-separáveis, a variação do resíduo médio torna-se bem pequena, tornando a medida

adequada para segmentação.

#### 4.4 Níveis de Decomposição

Os testes a seguir avaliam o comportamento da segmentação variando o nível de decomposições wavelets. O nível de decomposição pode influenciar na qualidade da segmentação, já que maiores níveis de decomposição tendem a identificar arranjos que não são identificados em níveis iniciais de decomposição.

Aumentando o nível de decomposição, aumenta-se linearmente a dimensionalidade do vetor de características. Para uma decomposição com  $l$  níveis, as wavelets separáveis possuem  $3l + 1$  bandas e as não-separáveis  $2l - 1$  bandas para níveis maiores do que dois e 2 bandas para o primeiro nível. A dimensão dos vetores de características para uma imagem em níveis de cinza é igual número de bandas multiplicado por uma constante relacionada ao número de medidas consideradas. Caso a imagem seja colorida a dimensão é triplicada.

Os testes foram aplicados sobre o conjunto de mosaicos do grupo I identificados na tabela 4.1. A execução considerou blocos de tamanho  $64 \times 64$  pixels e  $32 \times 32$  pixels. Os 91 mosaicos foram segmentados usando 7 wavelets diferentes: Daubechies 1 (db1, Haar), Daubechies 2 (db2), Biortogonal 3.1 (bior3.1), MaxMax (MM), MaxMin (Mm), Neville 2 (nev2) e Neville 4 (nev4). Os níveis de decomposição testados variam de 1 até o nível máximo de decomposição permitido para cada uma das bases com o tamanho do bloco considerado. A tabela 4.4 mostra as informações sobre níveis máximos de decomposição por tamanho de bloco.

As bases que utilizam o esquema bidimensional *quincunx*, ou seja, as não-separáveis (MaxMax, MaxMin, Neville 2 e Neville 4), necessitam de um nível mínimo de 2 decomposições em razão do próprio algoritmo, que somente opera a cada dois níveis de decomposição. Como mostrado na figura 4.9, os resultados para estas bases ocorrem de dois em dois níveis. As características usadas para decomposição foram a energia, o desvio padrão e a suavidade, normalizadas pela equação 2.62.

Os resultados para blocos de tamanho  $64 \times 64$  pixels são mostrados na figura 4.9. As bases separáveis não obtiveram ganhos notáveis com o aumento dos níveis de decomposição.

Tabela 4.4: Níveis máximos de decomposição das bases wavelets dependendo do tamanho do bloco.

Wavelet	Nível máximo por tamanho do bloco	
	$32 \times 32$ pixels	$64 \times 64$ pixels
bior3.1	1	3
db1	5	6
db2	3	4
MM	6	8
Mm	6	8
nev2	6	8
nev4	4	6

A base biortogonal, com nível máximo de decomposição igual a três, manteve uma taxa de segmentação correta próxima a 97% para os dois primeiros níveis e queda para o terceiro nível, 94%. As outras bases separáveis apresentaram comportamento semelhante, houve equilíbrio nos primeiros níveis e queda posterior.

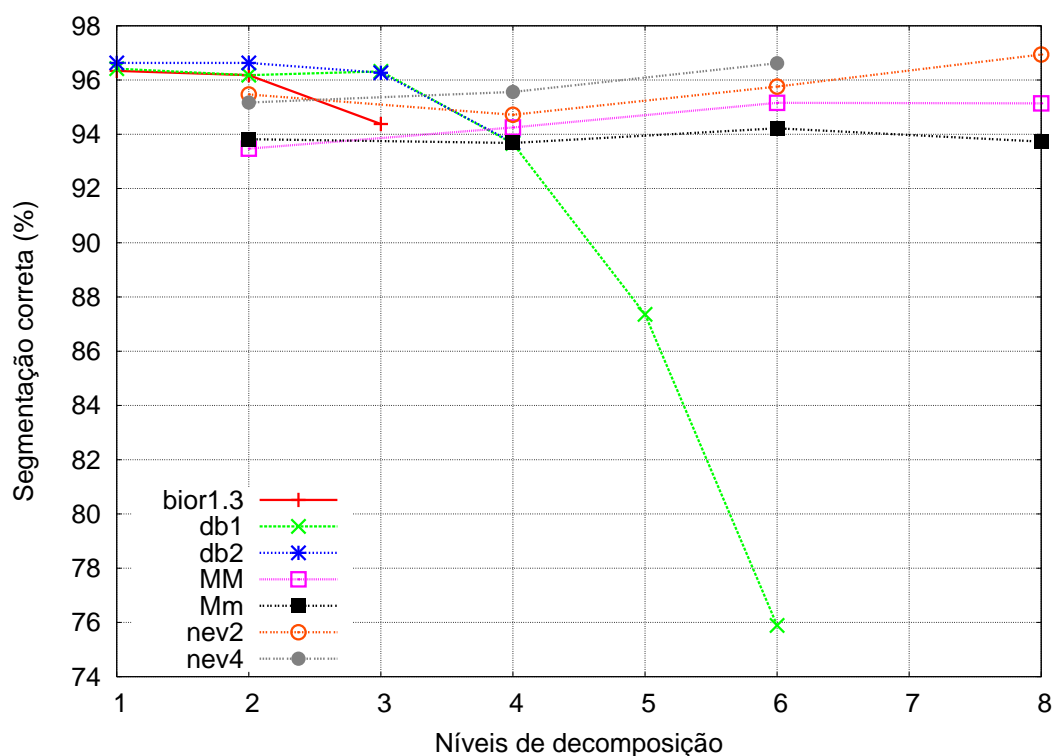


Figura 4.9: Correção da segmentação variando níveis de decomposição com blocos de tamanho  $64 \times 64$  pixels.

As wavelets db1 e db2 mantiveram resultados semelhantes até o terceiro nível de decom-

posição, próximos a 96%. Na seqüência, a queda foi significativa, chegando a 76% para o sexto nível da base db1. Como a base db2 tem nível máximo de decomposição igual a 4, a taxa de segmentação correta diminuiu 2%. No entanto, é possível perceber que o comportamento de db1 e db2 é comparável.

As bases não-separáveis, ao contrário, obtiveram leve melhora com o aumento dos níveis de decomposição, exceto pela base Mm. As bases MM, nev2 e nev4 aumentaram a segmentação correta em aproximadamente 2% entre o nível inicial e o nível máximo de decomposição. A base Mm manteve a taxa de segmentação correta em aproximadamente 94% para todos os níveis.

A figura 4.10 mostra os resultados para blocos de  $32 \times 32$  pixels. A queda para as wavelets separáveis iniciou-se logo no segundo nível de decomposição. Os melhores resultados são, portanto, aqueles em que apenas um nível de decomposição wavelet é utilizado. Embora não comparável às bases separáveis, também houve queda para as bases não-separáveis. Somente a base MM conseguiu pequena melhora com o aumento do nível de decomposição.

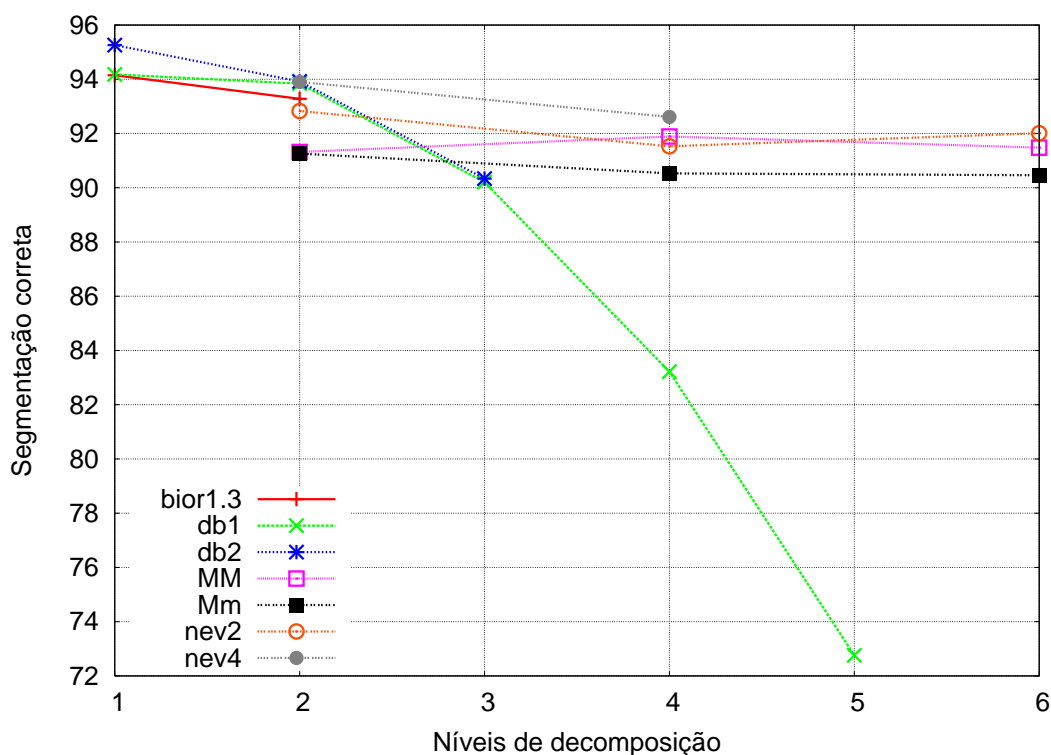


Figura 4.10: Correção da segmentação variando níveis de decomposição com blocos de tamanho  $32 \times 32$  pixels.

Muito embora os resultados demonstrados acima sejam elucidativos para o tipo de texturas testadas, a questão do nível de decomposição não pode ser generalizada. Se o tamanho do bloco ou o tamanho da imagem utilizada for pequeno, o uso de muitos níveis de decomposição não ajuda muito na identificação de maiores detalhes por serem as sub-bandas pequenas ou porque os níveis de decomposição adicionais trabalham em regiões muito pequenas. Por exemplo, o desvio padrão de uma sub-banda com  $32 \times 32$  pixels contém muito mais informação do que o desvio de uma sub-banda com  $2 \times 2$  pixels. Imagens grandes o suficiente, ou para as quais a transformação wavelet seja aplicada sobre blocos grandes, justificam o uso de vários níveis de decomposição, pois nestes casos é possível adquirir novas informações.

Como as imagens segmentadas neste trabalho exigem blocos de no máximo  $64 \times 64$  pixels, um nível de decomposição wavelet produz resultados satisfatórios. Para algumas bases, como visto, apenas um nível é essencial. Dessa forma, o método de segmentação utilizará blocos de até  $64 \times 64$  pixels e um nível de decomposição, em geral. Em alguns casos, como quando forem usadas bases não-separáveis, serão aplicados maiores níveis de decomposição.

A questão do tamanho de bloco para segmentar determinada imagem é ainda mais crítica. Blocos maiores, abrangendo a totalidade de uma textura, certamente exploram melhor as variações presentes na imagem. Para as imagens testadas,  $64 \times 64$  pixels é o tamanho ideal para o bloco, pois as regiões de textura apresentam o mesmo tamanho. No entanto, não se pode dizer que o melhor é usar sempre blocos desse tamanho.

Na verdade, a escolha do tamanho de bloco é uma tarefa específica para o tipo de imagens aplicadas. Esse fato será melhor analisado na próxima seção, juntamente com o estudo de modelos de cores.

## 4.5 Modelos de Cores e Tamanhos de Blocos

Os experimentos sobre cores utilizam os modelos HSV, YCbCr e YIQ, escolhidos por manterem informações de luminância e crominância separadas. Também foram consideradas imagens em tons de cinza para avaliar a melhora obtida com a adição de informações de cores.

Os experimentos foram realizados em mosaicos formados por texturas da coleção Vis-Tex [77]. Cada mosaico foi constituído por quatro texturas de dimensões  $512 \times 512$  pixels e

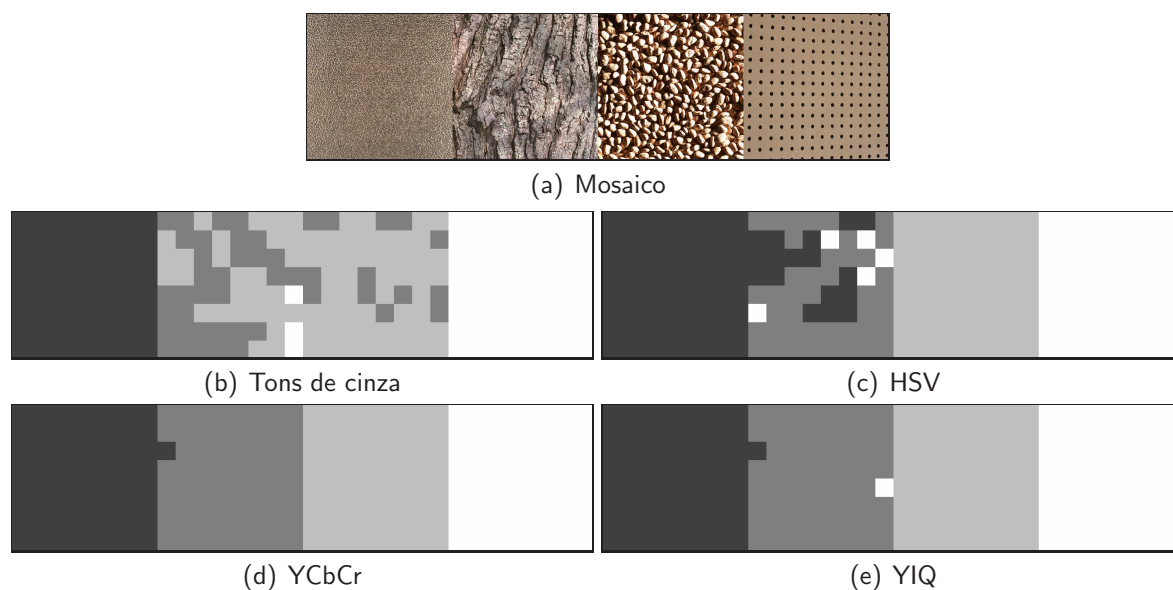


Figura 4.11: Segmentação de mosaico com diferentes modelos de cores e blocos de  $64 \times 64$  pixels.

diferentes justapostas lado a lado, como a da figura 4.11(a). As texturas foram então segmentadas com os modelos de cor HSV, YCbCr, YIQ e também somente com a intensidade de níveis de cinza; usando blocos de diferentes tamanhos. A transformada wavelet foi realizada usando a base não-separável MaxMax com dois níveis de decomposição. Como características foram usadas as medidas normalizadas de energia, desvio padrão e suavidade. A figuras 4.11(b)-(e) exemplificam a segmentação de um mosaico variando os parâmetros de cores e usando blocos de tamanho  $64 \times 64$  pixels.

Um conjunto de 20 mosaicos foi testado. Os resultados são mostrados no gráfico da figura 4.12. Os tamanhos de blocos testados para a segmentação foram  $16 \times 16$ ,  $32 \times 32$ ,  $64 \times 64$ ,  $128 \times 128$  e  $256 \times 256$  pixels. O acréscimo de informações de cores auxiliou a descrição das texturas, como pode ser visto pelo gráfico. O modelo de cores que proporcionou maior porcentagem de segmentação correta foi o YCbCr, seguidos pelos modelos YIQ e HSV.

Um fator de importância mostrado no gráfico é a influência do tamanho dos blocos para a segmentação. Com blocos de tamanho  $16 \times 16$  pixels a porcentagem de segmentação correta não é grande, o melhor resultado foi 78% para os modelos YCbCr e HSV. O tamanho do bloco neste caso exerce grande influência, como cada uma das 4 texturas é composta por  $512 \times 512$  pixels, o uso de apenas  $16 \times 16$  pixels para o bloco de segmentação é insuficiente,

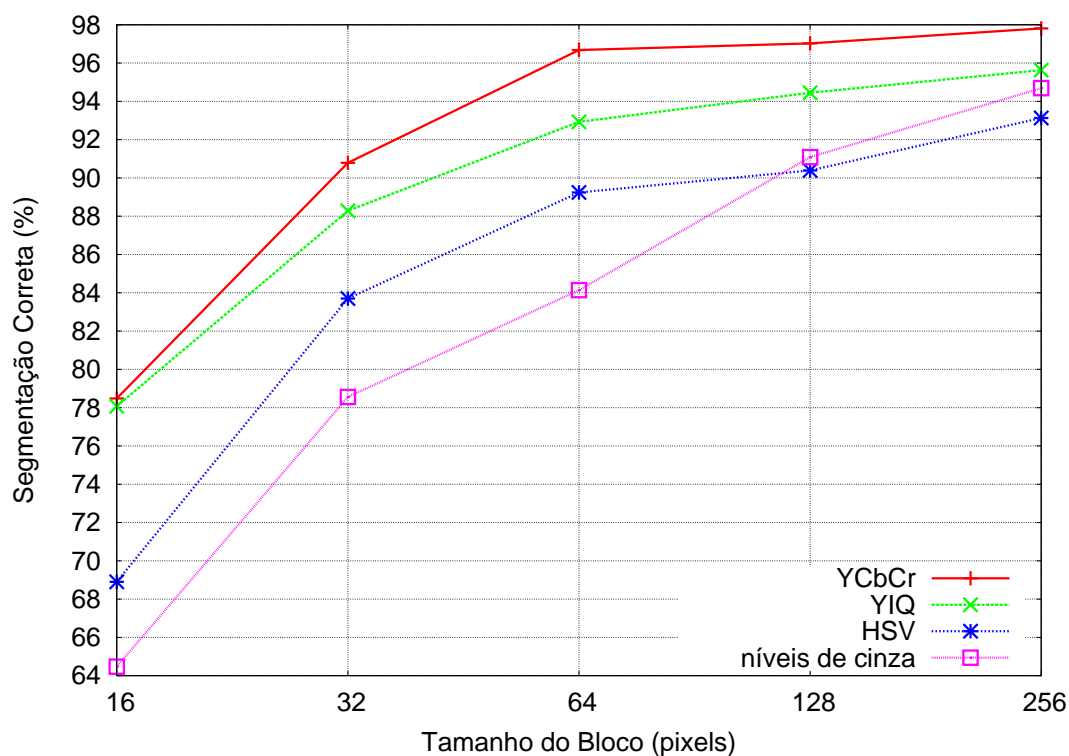


Figura 4.12: Comparação de modelos de cores para segmentação de mosaicos.

não é possível identificar os padrões formadores das texturas.

À medida que aumenta o tamanho do bloco, aumenta a taxa de segmentação correta. Com bloco de  $64 \times 64$  pixels, o modelo YCbCr chega a aproximadamente 97% de acerto. Em níveis de cinza, a segmentação correta está em 84%, o uso das informações de cores proporcionou um ganho de 13%. Esse resultado é importante pois é possível perceber pelas imagens da figura 4.11 que o bloco não é suficientemente grande para descrever alguns dos padrões das texturas. Em especial, a segunda textura da figura 4.11(a) possui ranhuras difíceis de serem identificadas e descritas usando blocos pequenos. Com blocos maiores,  $128 \times 128$  e  $256 \times 256$ , mesmo sem informação de cores a segmentação correta passa de 90% (figura 4.13). Com YCbCr, fica acima de 96%.

A escolha do tamanho do bloco é crucial e dependente do tipo de imagem que se deseja segmentar. Para algumas imagens, como aquelas produzidas por satélites ou tomógrafos, blocos grandes podem abranger regiões maiores do que o tamanho dos próprios objetos componentes da imagem. Isso geraria erro de segmentação. Blocos menores seriam de melhor utilidade nesses casos. Mosaicos de texturas ou imagens com dimensões grandes podem suportar ou

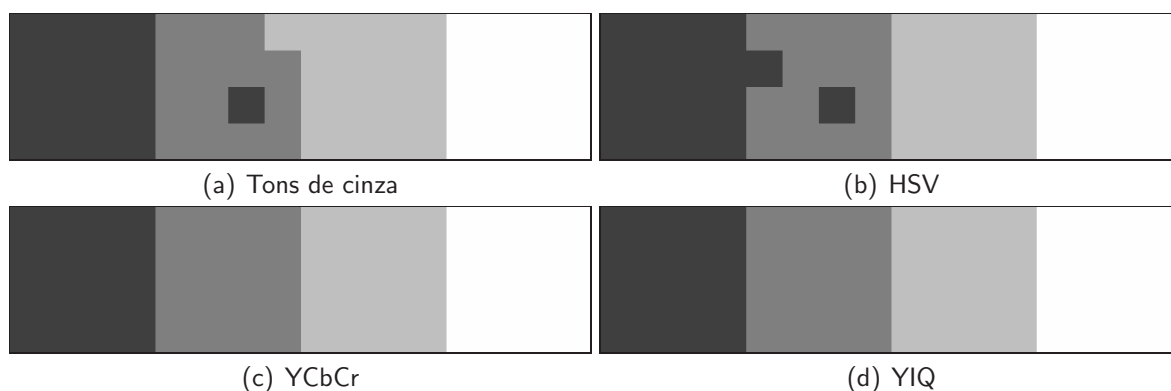


Figura 4.13: Segmentação de mosaico com diferentes modelos de cores e blocos de  $128 \times 128$  pixels.

mesmo necessitar de blocos maiores para que suas regiões componentes sejam bem descritas.

## 4.6 Método de Segmentação

Nesta seção são realizados testes com o método de segmentação descrito na seção 3.5. Para demonstrar a eficácia do método, a segmentação é aplicada em imagens de diversas naturezas: mosaicos de texturas, imagens de satélites e outras imagens naturais. Além das imagens segmentadas, o ganho computacional relacionado com o número de transformadas wavelets aplicadas pelo método é demonstrado.

Os parâmetros para o método foram escolhidos com base nos testes descritos anteriormente. As transformadas wavelets utilizam as bases mostradas na tabela 4.5. Os vetores de características são formados por medidas de energia ( $e_2$ ), desvio padrão ( $std$ ) e suavidade ( $sm$ ).

Tabela 4.5: Bases wavelets usadas pelo método de segmentação.

Família	Bases Wavelets
Daubechies	db1, db2
Coiflets	coif5
Symlets	sym2, sym5
Biorotogonal	bior3.1, bior3.3
Não-separável	MM, Mm, nev2

O modelo de cores usado é o YCbCr. A medida de heterogeneidade, que identifica blocos com vetores de característica divergindo da maioria dos vetores em sua classe, necessita que



o limiar  $L$  seja especificado. Os resultados a seguir foram obtidos com  $L = 0.8$ , determinado experimentalmente usando como critério a identificação de blocos em fronteiras e de detalhes isolados. O agrupamento realizado pelo algoritmo *k-means* requer que o número de classes seja informado.

O primeiro conjunto de imagens usado para testar o método de segmentação é formado por imagens de satélites, obtidas da base de dados provida pelo *Image Science and Analysis Laboratory, NASA-Johnson Space Center* [62]. As informações sobre as imagens e parâmetros para a segmentação são mostradas na tabela 4.6.

Tabela 4.6: Informações sobre os parâmetros do método de segmentação e sobre as imagens de satélite.

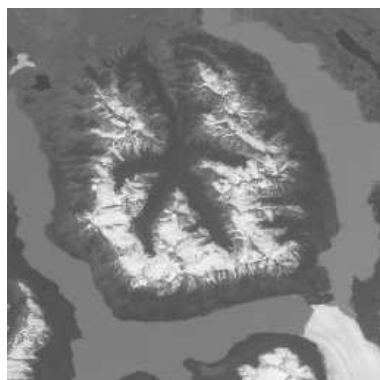
Imagem	Dimensão da Imagem (pixels)	Dimensão do Bloco (pixels)	Base Wavelet	Taxa de Ganho
Geleira Moreno	$340 \times 340$	$4 \times 4$	db1	0.52
Baía Shark	$512 \times 512$	$8 \times 8$	db1	0.74
Palm Island	$512 \times 512$	$8 \times 8$	nev2	0.72
Lagos Viedma	$512 \times 512$	$8 \times 8$	db2	0.72

Como a etapa final de refinamento é aplicada apenas para uma região reduzida da imagem, o custo computacional reduz-se significativamente. A quinta coluna da tabela 4.6 mostra a taxa de ganho proporcionada pela divisão da segmentação em duas etapas. A taxa de ganho é uma medida calculada pela razão entre o número de transformações wavelets realizadas pelo método proposto e o número de transformações wavelets necessárias para realizar a segmentação ponto-a-ponto em toda a imagem. Se o número máximo de transformações wavelets  $T_{\text{imagem}}$ , que ocorre quando cada pixel é segmentado separadamente, é igual ao produto das dimensões da imagem  $NM$ , e o número de transformações calculadas pelo método é  $T_{\text{método}}$ , a taxa de ganho é definida como

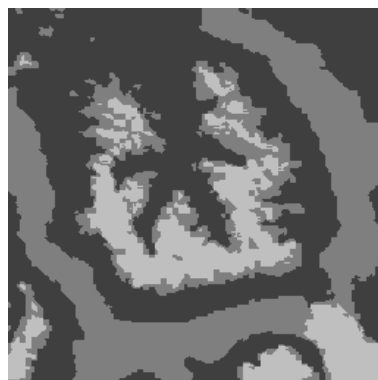
$$\text{TG} = 1 - \frac{T_{\text{método}}}{T_{\text{imagem}}}. \quad (4.1)$$

A primeira coluna da figura 4.14 mostra as imagens originais de terrenos. A segunda coluna mostra os resultados do método de segmentação proposto. As imagens das figuras 4.14(a) e

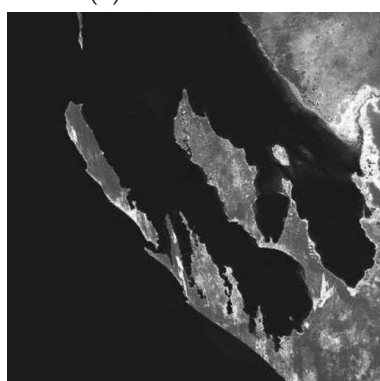
(c) são imagens em níveis de cinza, as outras duas, das figuras 4.14(e) e (g), são coloridas.



(a) Geleira Moreno



(b) Imagem segmentada



(c) Baía Shark



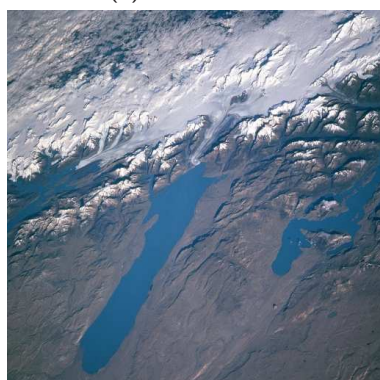
(d) Imagem segmentada



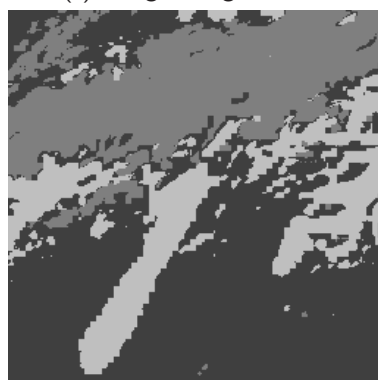
(e) Palm Island



(f) Imagem segmentada



(g) Lagos Viedma



(h) Imagem segmentada

Figura 4.14: Imagens de satélite segmentadas pelo método proposto.

Por apresentarem muitos detalhes, a opção por blocos grandes não é apropriada. Além de mascarar detalhes, a segunda etapa pode ser bastante custosa, pois muitos dos blocos abrangendo diferentes regiões da imagem serão marcados como heterogêneos, gerando um número elevado de pixels para refinamento na etapa final do método. Outro problema é que, nesse tipo de imagens, blocos grandes não oferecem boas informações para a formação dos centros das classes, provocando erros de segmentação. A segmentação foi aplicada usando blocos de  $4 \times 4$  ou  $8 \times 8$  pixels, de modo a identificar melhor os detalhes, o que não é possível de se obter com blocos de maior tamanho.

A taxa de ganho para a imagem da figura 4.14(a) foi de 0.52, o que significa que muitos blocos passaram para a segunda fase, marcados como heterogêneos e, mesmo assim, mais da metade das transformações foi evitada. Para as imagens das figuras 4.14(c), (e) e (g), o ganho foi superior, com taxas de 0.74, 0.72 e 0.72, respectivamente.

A segmentação da Geleira Moreno teve as três classes presentes na imagem bem separadas, mantendo os detalhes entre as fronteiras de água, terra e neve. Também foram mantidos os detalhes de reentrâncias da Baía Shark, delineando a costa mesmo com blocos de tamanho  $8 \times 8$  pixels. A identificação deve-se especialmente à segunda etapa do método que suavizou os encontros de classes, eliminando a maior parte dos efeitos provocados pelos blocos. A terceira imagem diferencia-se da primeira por apresentar cores. O destaque desta imagem é a segmentação da ilha artificial em forma de palmeira. A quarta imagem, também colorida, é de mais difícil segmentação. Três classes formam a imagem, com fronteiras recortadas e misturadas. A segmentação mostrou-se bastante satisfatória.

Tabela 4.7: Informações sobre os parâmetros do método de segmentação e sobre as imagens naturais.

<b>Imagem</b>	<b>Dimensão da Imagem (pixels)</b>	<b>Dimensão do Bloco (pixels)</b>	<b>Base Wavelet</b>	<b>Taxa de Ganho</b>
Campo	$256 \times 352$	$16 \times 16$	sym2	0.94
Lobo	$256 \times 352$	$8 \times 8$	coif5	0.93
Pirâmides	$256 \times 352$	$8 \times 8$	sym5	0.86

O segundo conjunto de resultados, mostrado na figura 4.15, é formado por imagens naturais da base de dados disponível pelo *The Berkeley Segmentation Dataset and Benchmark* [8]. As

informações sobre este conjunto são mostradas na tabela 4.7. As imagens do segundo conjunto não apresentam tantas reentrâncias e pequenos detalhes quanto as do primeiro conjunto, por esse motivo, o uso de blocos maiores possibilitam melhor identificação dos padrões que formam as regiões.

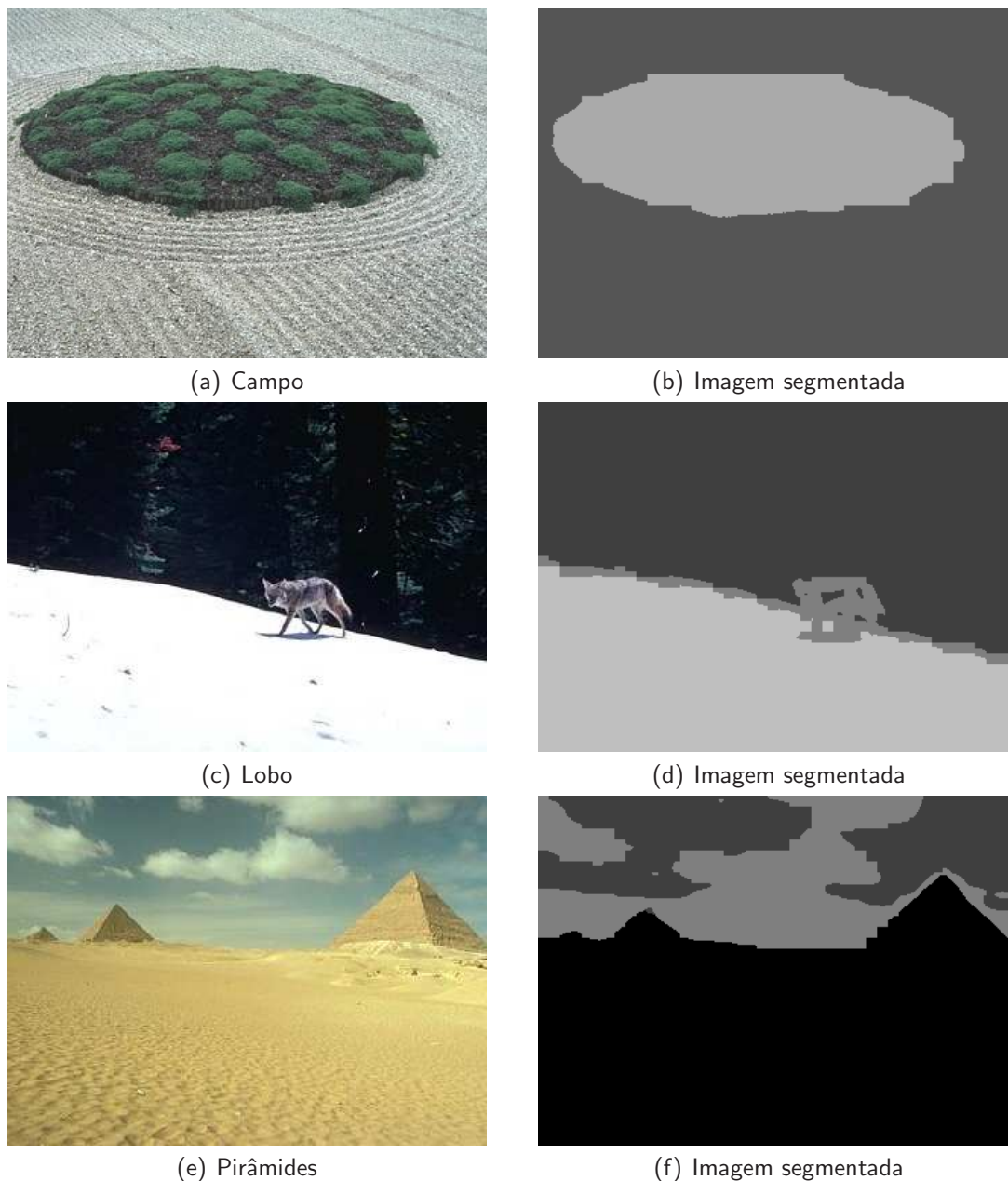


Figura 4.15: Imagens naturais segmentadas pelo método proposto.

A primeira imagem, de um campo, possui texturas bem definidas, uma mais fina e outra mais grossa. Os blocos de tamanho  $16 \times 16$  pixels permitem identificar os padrões formadores das duas classes, especialmente da classe central, com estruturas maiores. Com três classes

distintas, a imagem do lobo, destaca-se pela identificação do animal. Os traços ficaram mais grossos, o que poderia ser contornado usando blocos menores. Na imagem das pirâmides houve a divisão de céu, nuvens e areia. O destaque desta imagem é a segmentação das inclinações nas pirâmides, suavizadas pela segunda etapa do método. Houve um pequeno efeito de bloqueio na pirâmide maior, o que se deve ao fato de não terem sido marcados blocos heterogêneos naquela área. Uma pequena mudança do limiar é suficiente para resolver o problema sem diminuir muito o ganho computacional, já que este foi alto, 0.86 (tabela 4.7).

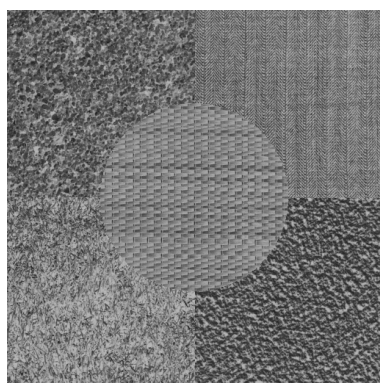
As taxas de ganho para as imagens da figura 4.15 foram bastante grandes para o segundo conjunto de imagens, como pode ser observado na quinta coluna da tabela 4.7. Os ganhos foram de 0.94, 0.93 e 0.86 para as imagens Campo, Lobo e Pirâmides, respectivamente. Isso significa que o número de pixels na segunda etapa é reduzido e, portanto, a mudança do limiar para identificação de blocos heterogêneos pode ser realizada para melhorar os contornos em imagens naturais.

O terceiro conjunto de imagens é formado por mosaicos sintéticos coloridos, construídos usando texturas disponíveis por Paul Bourke [9], exceto o Mosaico 1, o único em intensidade de níveis de cinza e formado por texturas disponíveis pela base *Brodatz* [10]. A tabela 4.8 contém informações para o conjunto. A identificação dos padrões nos mosaicos requer, em geral, blocos maiores para a extração dos descritores das classes. Nesse conjunto foram utilizados blocos de  $16 \times 16$ ,  $32 \times 32$  e  $64 \times 64$  pixels.

Tabela 4.8: Informações sobre os parâmetros do método de segmentação e sobre os mosaicos.

<b>Imagem</b>	<b>Dimensão da Imagem (pixels)</b>	<b>Dimensão do Bloco (pixels)</b>	<b>Base Wavelet</b>	<b>Taxa de Ganho</b>
Mosaico 1	$512 \times 512$	$64 \times 64$	MaxMin	0.81
Mosaico 2	$512 \times 512$	$16 \times 32$	MaxMax	0.73
Mosaico 3	$512 \times 512$	$16 \times 16$	MaxMin	0.88
Mosaico 4	$512 \times 512$	$32 \times 32$	MaxMax	0.87

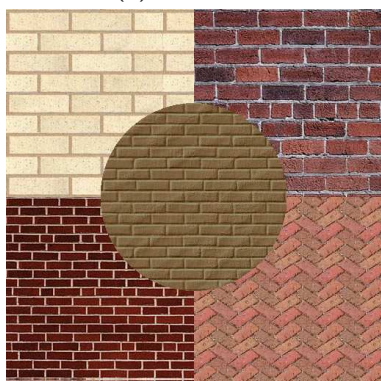
O método de segmentação identificou as cinco classes que formam cada um dos mosaicos. O interior de cada textura apresentou poucos pontos segmentados erroneamente. Em geral, a homogeneidade das regiões foi identificada de modo eficaz. Aliada ao reconhecimento



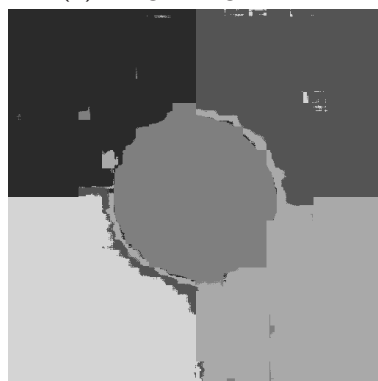
(a) Mosaico 1



(b) Imagem segmentada



(c) Mosaico 2



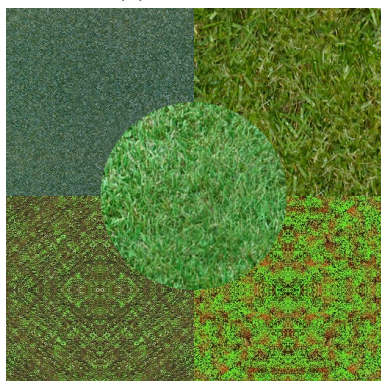
(d) Imagem segmentada



(e) Mosaico 3



(f) Imagem segmentada



(g) Mosaico 4



(h) Imagem segmentada

Figura 4.16: Mosaicos segmentados pelo método proposto.

de blocos heterogêneos, a primeira etapa do método permite a extração de descritores que separam bem as regiões de textura e permite a formação de centros pouco influenciados por regiões com encontro de texturas.

Os erros de segmentação entre texturas ocorreram principalmente na segunda etapa do método, quase que exclusivamente nos locais em que as janelas amostradas para a segmentação ponto-a-ponto abrangiam mais de uma classe de textura. Os maiores erros ocorreram nas fronteiras do círculo central dos dois primeiros mosaicos. O erro ocorrido tem como uma causa a fragilidade do classificador da segunda etapa, que consiste basicamente na distância Euclidiana. Por ser uma medida relativamente simples, a distância Euclidiana está sujeita às mudanças dos vetores de características nas regiões em que as janelas contêm mais de uma textura.

Os mosaicos 3 e 4 não tiveram segmentações com erros como os dos primeiros mosaicos. O aparecimento de uma terceira classe nas fronteiras foi bem mais raro. A fragilidade da distância Euclidiana abre a oportunidade de estudar classificadores com maior robustez, o que surge entre um dos objetivos futuros, como citado posteriormente no capítulo 5.

Tabela 4.9: Matrizes de confusão da segmentação dos mosaicos. Os valores referem-se a porcentagens.

(a) Mosaico 1						(b) Mosaico 2					
	t1	t2	t3	t4	t5		t1	t2	t3	t4	t5
t1	96.74	0.00	1.56	0.43	1.26	t1	98.07	0.71	0.51	0.13	0.58
t2	0.00	98.85	0.73	0.42	0.00	t2	0.00	99.42	0.00	0.58	0.00
t3	4.29	0.34	89.28	5.31	0.78	t3	3.54	6.32	78.18	0.56	10.40
t4	0.06	0.00	1.81	97.48	0.65	t4	0.00	1.12	0.00	98.88	0.00
t5	0.00	0.00	0.61	0.00	99.39	t5	0.00	0.94	0.73	0.32	98.01

(c) Mosaico 3						(d) Mosaico 4					
	t1	t2	t3	t4	t5		t1	t2	t3	t4	t5
t1	99.99	0.00	0.01	0.00	0.00	t1	96.26	0.16	3.58	0.00	0.00
t2	0.21	98.48	0.61	0.70	0.00	t2	0.00	98.00	2.00	0.00	0.00
t3	2.06	2.72	95.11	0.11	0.00	t3	0.00	0.07	95.04	1.07	3.82
t4	0.30	0.50	0.23	98.97	0.00	t4	0.00	0.00	0.32	99.54	0.13
t5	0.00	0.12	2.84	0.00	97.04	t5	0.00	0.00	0.09	0.00	99.91

O tamanho dos blocos variou para as imagens. As imagens coloridas obtiveram bons resultados com blocos de tamanho maior ou igual a  $16 \times 16$  pixels. O mosaico 1 necessitou

de blocos de  $64 \times 64$  pixels para uma segmentação com menores erros. A diferença evidencia a importância das bandas de cores para melhorar a eficiência da segmentação. O uso de informações de cores contribui significativamente na extração dos descritores, permitindo o uso de blocos menores para uma boa segmentação.

As porcentagens de acertos e erros da segmentação podem ser analisadas pelas matrizes de confusão mostradas na tabela 4.9. As cinco texturas que formam os mosaicos são identificadas pelos rótulos t1, t2, t3, t4 e t5. Para cada linha, a coluna com rótulo correspondente indica a segmentação correta, portanto, a segmentação correta é mostrada na diagonal principal. As colunas restantes de cada linha mostram a segmentação errada atribuída aos rótulos indicados nas colunas. Conforme os valores apresentados nas diagonais principais das matrizes de confusão, a segmentação obteve alto grau de acerto.



## CAPÍTULO 5

### CONCLUSÕES

Este trabalho descreveu uma abordagem para segmentação de imagens baseada em wavelets, oferecendo um método novo para realizar uma das etapas mais importantes e complexas na análise de imagens. A abordagem de segmentação em duas etapas proporciona rapidez sem diminuir a qualidade dos resultados, pois o subsequente refinamento em regiões de heterogeneidade garante a qualidade do processo.

A pesquisa proposta envolveu pontos importantes para o desenvolvimento da segmentação de imagens digitais por wavelets, principalmente a segmentação sem imagens para treinamento, o que torna a segmentação ainda mais difícil.

A maioria dos algoritmos de segmentação utiliza características extraídas dos valores de pixels em intensidade de níveis de cinza. O método proposto, além desses valores, referentes à luminância, constata a importância do uso das bandas de crominância, complementando a descrição de regiões de uma imagem com informações de cores.

Um conjunto de 36 bases wavelets separáveis foi analisado inicialmente para a tarefa de segmentação. O objetivo foi selecionar, entre as bases, aquelas que melhor responderam à tarefa de caracterização de texturas e que, conseqüentemente, obtivessem melhor segmentação.

Algumas wavelets biortogonais obtiveram resultados superiores aos de wavelets de outras famílias, principalmente as wavelets Biortogonal 3.1 e Biortogonal 3.3. Como pode ser visto nos resultados da seção 4.1, ainda outras três wavelets biortogonais aparecem melhor colocadas antes da wavelet Daubechies 1.

A escolha de bases wavelets não é uma tarefa simples dentro de um conjunto muito grande e, como pôde ser visto pelos primeiros resultados, a diferença entre bases não é muito significativa. Muitas bases possuem características e capacidades equiparáveis. Este trabalho, no entanto, ajudou na escolha de um subconjunto de bases dentro do conjunto inicial. Pela escolha pretendia-se manter uma certa diversidade de propriedades, o que foi feito selecionando-se

bases das 4 famílias consideradas: Daubechies, Coiflets, Symlets e Biortogonal. Especificamente, as bases escolhidas foram: Daubechies 1, Daubechies 2, Coiflets 5, Symlets 2, Symlets 5, Biortogonal 3.1 e Biortogonal 3.3.

As características para descrição de regiões em uma imagem são obtidas a partir dos coeficientes da transformada wavelet, que apresentam informações sobre as variações locais de uma imagem com o diferencial, em relação a outras transformadas, de serem capazes de reter informações espaciais. A literatura apresenta um grande conjunto de medidas que podem ser usadas para descrição com wavelets. Um dos melhores e mais comuns descritores é a energia. No entanto, existem outros descritores capazes de representar as informações providas pelas transformadas.

Um número significativo de medidas foi apresentado com o objetivo de examinar suas capacidades individuais de descrição e separação de texturas diferentes. O uso de métodos para ordenação de variáveis apresentou-se como uma maneira apropriada, dada a importância destes no processo de reconhecimento de padrões e por apresentarem na literatura um estudo importante.

Em especial, 10 características obtiveram resultados bastante satisfatórios, podendo ser consideradas boas medidas para descrever regiões texturais. Sem ordem e ignorando redundâncias, as características destaque são: energia, desvio médio, entropia, resíduo médio, momento central de ordem 2, normas de ordem 1 e 2, desvio padrão, variância e suavidade.

Para avaliar a invariância à rotação, foram analisadas 10 características de texturas (escolhidas pelo teste de características) e 13 bases wavelets diferentes, das quais 7 eram as wavelets separáveis selecionadas a partir dos primeiros testes com as bases e 6 eram wavelets não-separáveis (MaxMax, MaxMin, Neville 2, Neville 4, Neville 6 e Neville 8).

As wavelets não-separáveis começaram a ser desenvolvidas recentemente, um dos reflexos desse fato é que poucos filtros são encontrados na literatura. No entanto, mesmo os filtros relativamente simples utilizados neste trabalho, demonstraram maior capacidade para descrever texturas considerando a invariância à rotação do que os filtros separáveis. Dos filtros não-separáveis considerados, os filtros MaxMax e MaxMin possibilitaram que texturas com diferentes orientações fossem descritas por vetores de características com menor variação entre

eles.

A avaliação das características demonstrou que a medida mais robusta para descrição invariante à rotação é a suavidade. Além de ser a mais robusta, os resultados obtidos com a medida são bastante superiores aos resultados das outras medidas. Um segundo grupo pode então ser definido com possíveis medidas a serem usadas. Considerando o uso tanto de wavelets separáveis quando de wavelets não-separáveis, tem-se as seguintes medidas: desvio médio, energia, norma 1, norma 2 e desvio padrão. Caso a avaliação seja restringida às wavelets não-separáveis, a medida de resíduo médio apresenta bons resultados.

O método de segmentação proposto não usa dados para treinamento e apresenta boa confiabilidade. Na primeira etapa do método, a segmentação baseada em blocos acelera o processo e produz descritores para a o agrupamento dos blocos e também para a formação dos vetores de características que identificam as classes diferentes pertencentes a uma imagem.

O efeito de blocagem produzido pela primeira etapa pode ser suavizado ou eliminado a partir da identificação de blocos heterogêneos pela medida de heterogeneidade, que mostrou-se eficaz ao apresentar certo grau de independência mantendo-se o mesmo limiar para todas as imagens. A escolha do limiar para a medida é o fator mais delicado, sendo decisivo para o nível de refinamento na etapa final mas que envolve considerações de custo computacional. Nos testes realizados, um bom equilíbrio entre qualidade final da segmentação e custo computacional foi obtido com o limiar igual a 0.8.

A etapa final do algoritmo proveu uma segmentação mais fina, melhorando o resultado de regiões com detalhes. O resultado foi satisfatório e os casos de segmentação errônea podem ser evitados usando classificadores mais robustos.

## **Trabalhos Futuros**

Alguns pontos de estudo futuro são importantes para melhorar o poder de classificação do algoritmo e ainda diminuir a intervenção humana necessária.

Transformadas bidimensionais não-separáveis demonstraram-se mais eficazes no fornecimento de informações sobre texturas e adaptaram-se melhor às variações de rotação. A avaliação de transformadas não-separáveis, estendida para esquemas de decomposição diferentes,

além dos baseados no esquema *quincunx*, pode resultar em melhores explorações dos padrões de textura.

A algoritmo de pirâmide é o mais comum para realizar as transformações wavelets em diversos níveis de decomposição. A multirresolução é obtida aplicando decomposições sobre a banda de aproximação (LL) em cada um dos níveis, como descrito na subseção 2.2.2. Outra forma de algoritmo para multirresolução é o chamado *wavelet packet*, neste a transformação assemelha-se a uma árvore binária. Os diversos níveis de decomposição são aplicados sobre todas as sub-bandas e não somente na sub-banda de aproximação. Em alguns casos, as sub-bandas que devem ser subseqüentemente decompostas são escolhidas pela entropia, que indica as sub-bandas com maiores informações como propícias para novas decomposições. Algumas pesquisas afirmam que *wavelet packets* são mais apropriados para extrair descritores. Este é um bom ponto de pesquisa futura.

As bases wavelets neste trabalho foram escolhidas empiricamente mas mantendo-se a variação de famílias devido às suas propriedades particulares. Uma estudo matemático sobre as propriedades das famílias e seus efeitos para a extração de descritores é uma importante consideração.

A informação do número de classes requer dependência humana. Há trabalhos que exploram especificamente o problema de definição automática do número de agrupamentos em conjuntos de dados [79], baseados em critérios que avaliam propriedades dos agrupamentos, otimização de funções de probabilidade ou heurísticas baseadas em abordagens variadas. Esse estudo deve contribuir para tornar o método mais automático.

O limiar para identificação de blocos heterogêneos é mais um fator complexo de padronização e aplicação independente da imagem. A análise do limiar é um ponto importante de aprofundamento.

Dados os erros observados nos resultados, algoritmos de classificação para a fase final da segmentação devem ser estudados para contornar as fragilidades da medida Euclidiana pura. A consideração de um melhoramento do *k-means* também pode favorecer a melhora na formação de centros para as classes.

A comparação do acerto obtido com o método de segmentação desenvolvido e do acerto

obtido com outros métodos disponíveis também deve ser realizada. Em especial, comparando outros métodos que não usem imagens de treinamento, baseados ou não em transformadas wavelets.

## REFERÊNCIAS BIBLIOGRÁFICAS

- [1] ACHARYYA, M., DE, R. K. E KUNDU, M. K. Segmentation of remotely sensed images using wavelet features and their evaluation in soft computing framework. *IEEE Transactions on Geoscience and Remote Sensing* 41, 12 (Dezembro 2003), 2900–2905.
- [2] ANDREWS, R. E NGUYEN, D. Separable and quincunx wavelet image coding. *6 th IEEE International Workshop on Signal Processing and Communication Systems* (Novembro 1998), 1–5.
- [3] ANTONINI, M., BARLAUD, M., MATHIEU, P. E DAUBECHIES, I. Image coding using wavelet transform. *IEEE Transactions on Image Processing* 1, 2 (Abril 1992), 205–220.
- [4] ARIVAZHAGAN, S. E GANESAN, L. Texture classification using wavelet transform. *Pattern Recognition Letters* 24, 9 (Junho 2003), 1513–1521.
- [5] ARIVAZHAGAN, S. E GANESAN, L. Texture segmentation using wavelet transform. *Pattern Recognition Letters* 24, 16 (Dezembro 2003), 3197–3203.
- [6] AUJOL, J.-F., AUBERT, G. E BLANC-FERAUD, L. Wavelet-based level set evolution for classification of textured images. *IEEE Transactions on Image Processing* 12, 12 (Dezembro 2003), 1634–1641.
- [7] BECK, J., SUTTER, A. E IVRY, R. Spatial frequency channels and perceptual grouping in texture segregation. *Computer Vision, Graphics and Image Processing* 37, 2 (1987), 299–325.
- [8] BERKELEY. The Berkeley Segmentation Dataset and Benchmark. <http://www.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/segbench/>, acesso em: 05 de março de 2008.
- [9] BOURKE, P. Texture and Color. [http://local.wasp.uwa.edu.au/~pbourke/texture\\_colour/](http://local.wasp.uwa.edu.au/~pbourke/texture_colour/), acesso em: 05 de março de 2008.

- [10] BRODATZ. Brodatz. <http://www.ux.uis.no/~tranden/brodatz.html>, acesso em: 05 de março de 2008.
- [11] BURGESS, C. J. C. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery* 2, 2 (1998), 121–167.
- [12] BURRUS, C. S., GOPINATH, R. A. E GUO, H. *Introduction to Wavelets and Wavelet Transforms*. Prentice Hall, Nova Jersey, Estados Unidos, 1998. págs. 268.
- [13] CASTAÑON, C. A. B. Recuperação de imagens por conteúdo através de análise multiresolução por wavelets. Dissertação de Mestrado, Instituto de Ciências Matemáticas e de Computação, São Paulo, Brasil, 2003. págs. 112.
- [14] CHANG, T. E KUO, C.-C. J. A wavelet transform approach to texture analysis. *IEEE International Conference on Acoustics, Speech, and Signal Processing* 4 (Março 1992), 661–664.
- [15] CHANG, T. E KUO, C.-C. J. Texture analysis and classification with tree-structured wavelet transform. *IEEE Transactions on Image Processing* 2, 4 (Outubro 1993), 429–441.
- [16] CHARALAMPIDIS, D. E KASPARIS, T. Wavelet-based rotational invariant roughness features for texture classification and segmentation. *IEEE Transactions on Image Processing* 11, 8 (Agosto 2002), 825–837.
- [17] CHEN, J., PAPPAS, T., MOJSILOVIC, A. E ROGOWITZ, B. Adaptive image segmentation based on color and texture. In *IEEE International Conference on Image Processing* (Rochester, NY, Estados Unidos, Setembro 2002), pp. 1–4.
- [18] CHOI, H.-I. E WILLIAMS, W. J. Improved time-frequency representation of multi-component signals using exponential kernels. *IEEE Transactions on Acoustics, Speech and Signal Processing* 37, 6 (Junho 1989), 862–871.
- [19] CHRISTENSEN, O. E CHRISTENSEN, K. L. *Approximation theory: from Taylor polynomials to wavelets*. Springer-Verlag, Maio 2004. págs. 165.

- [20] COIFMAN, R. R. Wavelet analysis and signal processing. Springer-Verlag, pp. 59–68.
- [21] CUI, L., LI, H. E ZHOU, Y. Theory of 2-D non-separable optimal M-band wavelets and its applications. In *GMP '04: Proceedings of the Geometric Modeling and Processing 2004* (Washington, DC, Estados Unidos, 2004), IEEE Computer Society, pp. 331–334.
- [22] DA SILVA, R. D., MINETTO, R. E PEDRINI, H. Segmentação de imagens médicas utilizando transformadas wavelets. In *Workshop de Iniciação Científica em Computação Gráfica e Processamento de Imagens* (Outubro 2005), pp. 1–8.
- [23] DASH, M., LIU, H. E YAO, J. Dimensionality reduction of unsupervised data. In *Tools with Artificial Intelligence, 1997. Proceedings., Ninth IEEE International Conference on* (November 1997), no. 3-8, pp. 532–539.
- [24] DAUBECHIES, I. The wavelet transform, time-frequency localization and signal analysis. *IEEE Transactions on Information Theory* 36 (Setembro 1990), 961–1005.
- [25] DAUBECHIES, I. *Ten Lectures on Wavelets*. CBMS-NSF Regional Conference Series in Applied Mathematics. SIAM, 1992. págs. 377.
- [26] DE WOUWER, G. V. *Wavelets for Multiscale Texture Analysis*. PhD thesis, Universiteit Antwerpen, Antwerpen, Bélgica, Maio 1998. págs. 169.
- [27] DE WOUWER, G. V., SCHEUNDERS, P. E D. VAN DYCK, D. Statistical texture characterization from discrete wavelet representations. *IEEE Transactions on Image Processing* 8, 4 (Abril 1999), 592–598.
- [28] DE WOUWER, G. V., SCHEUNDERS, P. E DYCK, D. V. Statistical texture characterization from discrete wavelet representations. *IEEE Transactions on Image Processing* 8, 4 (Abril 1999), 592–598.
- [29] DE WOUWER, G. V., SCHEUNDERS, P., LIVENS, S. E DYCK, D. V. Wavelet correlation signatures for color texture characterization. *Pattern Recognition* 32, 3 (Março 1999), 443–451.



- [30] DE ZEEUW, P. M. A toolbox for the lifting scheme on quincunx grids. Tech. rep., Centrum Voor Wiskunde en Informatica, Dezembro 2002. págs. 25.
- [31] DENG, Y. E MANJUNATH, B. Unsupervised segmentation of color-texture regions in images and video. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 23, 8 (Agosto 2001), 800–810.
- [32] FUKUDA, S. E HIROSAWA, H. A wavelet-based texture feature set applied to classification of multifrequency polarimetric SAR images. *IEEE Transactions on Geoscience and Remote Sensing* 37, 5 (Setembro 1999), 2282–2286.
- [33] GETREUER, P. Filter coefficients to popular wavelets, Maio 2006. <http://www.mathworks.com/matlabcentral/fileexchange/>.
- [34] GOMES, J. E VELHO, L. *From fourier analysis to wavelets*. SIGGRAPH'98 Course Notes, SIGGRAPH-ACM publication, Orlando, Estados Unidos, Julho 1998. págs. 210.
- [35] GONZALEZ, R. C. E WOODS, R. E. *Processamento de Imagens Digitais*. Edgard Blücher, São Paulo, Brasil, 2000. págs. 793.
- [36] GOSE, E., JOHNSONBAUGH, R. E JOST, S. *Pattern Recognition and Image Analysis*. Prentice Hall PTR, Upper Saddle River, Estados Unidos, 1996. págs. 483.
- [37] GRAPS, A. An introduction to wavelets. *IEEE Computational Science & Engineering* 2, 2 (1995), 50–61.
- [38] GROSSMANN, A. E MORLET, J. Decomposition of hardy functions into square integrable wavelets of constant shape. *SIAM Journal on Mathematical Analysis* 15, 4 (1984), 723–736.
- [39] GUYON, I. E ELISSEEFF, A. An introduction to variable and feature selection. *Journal of Machine Learning Research* 3 (2003), 1157–1182.
- [40] GUYON, I., WESTON, J., BARNHILL, S. E VAPNIK, V. Gene selection for cancer classification using support vector machines. *Machine Learning* 46, 1-3 (2002), 389–422.

- [41] HAAR, A. Zur theorie der orthogonalen funktionensysteme. *Mathematische Annalen* (1910), 331–371.
- [42] HE, W. E LAI, M.-J. Examples of bivariate nonseparable compactly supported orthonormalcontinuous wavelets. *IEEE Transactions on Image Processing* 9, 5 (2000), 949–953.
- [43] HEARN, D. E BAKER, M. P. *Computer graphics: C version*, 2 ed. Prentice-Hall, Inc., Upper Saddle River, NJ, Estados Unidos, 1997. págs. 652.
- [44] HEEGER, D. J. E BERGEN, J. R. Pyramid-based texture analysis/synthesis. In *SIGGRAPH '95: Proceedings of the 22nd annual conference on Computer graphics and interactive techniques* (Nova Iorque, NY, Estados Unidos, 1995), ACM, pp. 229–238.
- [45] HU, M.-K. Visual pattern recognition by moment invariants. *IEEE Transactions on Information Theory* 8, 2 (Fevereiro 1962), 179–187.
- [46] JONG, K., MARY, J., CORNUÉJOLS, A., MARCHIORI, E. E SEBAG, M. Ensemble feature ranking. In *PKDD '04: Proceedings of the 8th European Conference on Principles and Practice of Knowledge Discovery in Databases* (Nova Iorque, NY, Estados Unidos, 2004), Springer-Verlag New York, Inc., pp. 267–278.
- [47] KOVACEVIC, J. E VETTERLI, M. Nonseparable multidimensional perfect reconstruction filter banks and wavelet bases for  $\mathbb{R}^2$ . *IEEE Transactions on Information Theory* 38, 2 (1992), 533–555.
- [48] KOVACEVIC, J. E VETTERLI, M. Nonseparable two- and three-dimensional wavelets. *IEEE Transactions on Signal Processing* 43, 5 (1995), 1269–1273.
- [49] LENDERS, P. E SJOSTROM, A. On the implementation of nonseparable two-dimensional Haar wavelet transforms. *IEEE Transactions on Signal Processing* 47, 11 (Novembro 1999), 3137–3139.
- [50] LINDEMANN, M. *Approximation properties of non-separable wavelet bases with isotropic scaling matrices – and their relation to Besov spaces*. PhD thesis, Universidade de Bremen, Alemanha, 2005. págs. 138.

- [51] LIU, Y., WU, S. E ZHOU, X. Texture segmentation based on features in wavelet domain for image retrieval. In *Visual Communications and Image Processing* (2003), pp. 2026–2034.
- [52] MALLAT, S. Multifrequency channel decompositons of images and wavelet models. *IEEE Transactions on Acoustics, Speech and Signal Processing* 37, 12 (Dezembro 1989), 2091–2110.
- [53] MALLAT, S. A theory for multiresolution signal decomposition: the wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 11, 7 (Julho 1989), 674–693.
- [54] MALLAT, S. Wavelets for a vision. *Proceedings of the IEEE* 84, 4 (1996), 604–614.
- [55] MALLAT, S. *A Wavelet Tour of Signal Processing*. AP Professional, London, 1997. págs. 620.
- [56] MANJUNATH, B. S., OHM, J.-R., VASUDEVAN, V. V. E YAMADA, A. Color and texture descriptors. *IEEE Transactions Circuits and Systems for Video Technology* 11, 6 (Junho 2001), 703–715.
- [57] MATHWORKS, T. MATLAB®- the language of technical computing. <http://www.mathworks.com/products/matlab/>, acesso em: 11 de março de 2008.
- [58] MINETTO, R., DA SILVA, R. D., SCHWARTZ, W. E PEDRINI, H. Segmentação de imagens utilizando abordagem espectral por transformadas Wavelet e de Fourier. *IV Colóquio Brasileiro de Ciências Geodésicas* (Maio 2005), 1–6.
- [59] MORETTIN, P. A. *Ondas e ondaletas - da análise de fourier à análise de ondaletas*. Edusp - Editora da Universidade de São Paulo, São Paulo, Brasil, 1999. págs. 276.
- [60] MORLET, J., ARENS, G., FORGEAU, I. E GIARD, D. Wave propagation and sampling theory. *SIAM Journal on Mathematical Analysis* 15, 4 (1984), 723–736.

- [61] MUNESWARAN, K., GANESAN, L., ARUMUGAM, S. E SOUNДАР, K. R. Texture classification with combined rotation and scale invariant wavelet features. *Pattern Recognition* 38, 10 (2005), 1495–1506.
- [62] NASA. Earth from Space. <http://earth.jsc.nasa.gov/sseop/efs/>, acesso em: 05 de março de 2008.
- [63] PARAMESWARIAH, C. E COX, M. The ways of wavelet filters. *IEEE Potentials* 25, 1 (Janeiro 2006), 11–15.
- [64] PENG, L., ZHAO, Z. M. E MA, J. L. Texture image segmentation based on wavelet-domain hidden markov models. In *IEEE International Geoscience and Remote Sensing Symposium* (Setembro 2004), vol. 6, pp. 3829–3832.
- [65] QUEVEDO, J. R., BAHAMONDE, A. E LUACES, O. A simple and efficient method for variable ranking according to their usefulness for learning. *Computational Statistics & Data Analysis* 42, 1 (Setembro 2007), 578–595.
- [66] RANDEN, T. E HUSØY, J. H. Texture segmentation using filters with optimized energy separation. *IEEE Transactions on Image Processing* 8, 4 (Abril 1999), 571–582.
- [67] SCHWARTZ, W. R., DA SILVA, R. D., MINETTO, R. E PEDRINI, H. An improved k-means clustering algorithm for image segmentation. In *II Workshop de Visão Computacional* (Outubro 2006), pp. 1–6.
- [68] STOPPIGLIA, H., DREYFUS, G., DUBOIS, R. E OUSSAR, Y. Ranking a random feature for variable and feature selection. *Journal of Machine Learning Research* 3 (2003), 1399–1414.
- [69] SWELDENS, W. The lifting scheme: a custom-design construction of biorthogonal wavelets. *Applied and Computational Harmonic Analysis* 3, 2 (Abril 1996), 186–200.
- [70] SWELDENS, W. E SCHRÖDER, P. Building your own wavelets at home. In *Wavelets in Computer Graphics*. ACM SIGGRAPH Course notes, 1996, pp. 15–87.

- [71] TANG, Y. Y., LIU, J., YANG, L. H. E MA, H. *Wavelet theory and its Application to pattern recognition*. World Scientific Publishing Company, 2000. págs. 300.
- [72] THEODORIDIS, S. E KOUTROUMBAS, K. *Pattern Recognition*. Elsevier Academic Press, San Diego, CA, Estados Unidos, 2003. págs. 640.
- [73] UNSER, M. Texture classification and segmentation using wavelet frames. *IEEE Transactions on Image Processing* 4, 11 (Novembro 1995), 1549 – 1560.
- [74] UNSER, M. E EDEN, M. Multiresolution feature extraction and selection for texture segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 11, 7 (Julho 1989), 717–728.
- [75] UYTTERHOEVEN, G. *Wavelets: software and applications*. PhD thesis, Katholieke Universiteit Leuven, Leuven, Holanda, 1999. págs. 195.
- [76] VETTERLI, M. E KOVACEVIC, J. *Wavelets and Subband Coding*. Prentice-Hall, Englewood Cliffs, NJ, Estados Unidos, 1995. págs. 488.
- [77] VISTEX. Vision Texture Database. <http://vismod.media.mit.edu/vismod/imagery/VisionTexture/distribution.html>, acesso em: 05 de março de 2008.
- [78] WU, G., ZHANG, Y. E LIN, X. Wavelet transform-based texture classification with feature weighting. In *EEE International Conference on Image Processing* (1999), pp. 435–439.
- [79] XU, R. E II, D. W. Survey of clustering algorithms. *IEEE Transactions on Neural Networks* 16, 3 (Maio 2005), 645–678.

## APÊNDICE A

### FILTROS PARA AS TRANSFORMADAS WAVELETS

A seguir são mostrados os filtros para decomposição wavelet para transformadas separáveis e não-separáveis. Os filtros LD e HD referem-se respectivamente aos filtros passa-baixa e passa-alta para a decomposição. Os filtros LR e HR referem-se respectivamente aos filtros passa-baixa e passa-alta para a reconstrução.

#### A.1 Daubechies

##### Daubechies 1 (Haar)

$$LD = \begin{bmatrix} 0.707107 \\ 0.707107 \end{bmatrix} \quad HD = \begin{bmatrix} -0.707107 \\ 0.707107 \end{bmatrix} \quad LR = \begin{bmatrix} 0.707107 \\ 0.707107 \end{bmatrix} \quad HR = \begin{bmatrix} 0.707107 \\ -0.707107 \end{bmatrix}$$

##### Daubechies 2

$$LD = \begin{bmatrix} -0.129410 \\ 0.224144 \\ 0.836516 \\ 0.482963 \end{bmatrix} \quad HD = \begin{bmatrix} -0.482963 \\ 0.836516 \\ -0.224144 \\ -0.129410 \end{bmatrix} \quad LR = \begin{bmatrix} 0.482963 \\ 0.836516 \\ 0.224144 \\ -0.129410 \end{bmatrix} \quad HR = \begin{bmatrix} -0.129410 \\ -0.224144 \\ 0.836516 \\ -0.482963 \end{bmatrix}$$

##### Daubechies 3

$$LD = \begin{bmatrix} 0.035226 \\ -0.085441 \\ -0.135011 \\ 0.459878 \\ 0.806892 \\ 0.332671 \end{bmatrix} \quad HD = \begin{bmatrix} -0.332671 \\ 0.806892 \\ -0.459878 \\ -0.135011 \\ 0.085441 \\ 0.035226 \end{bmatrix} \quad LR = \begin{bmatrix} 0.332671 \\ 0.806892 \\ 0.459878 \\ -0.135011 \\ -0.085441 \\ 0.035226 \end{bmatrix} \quad HR = \begin{bmatrix} 0.035226 \\ 0.085441 \\ -0.135011 \\ -0.459878 \\ 0.806892 \\ -0.332671 \end{bmatrix}$$

##### Daubechies 4

$$LD = \begin{bmatrix} -0.010597 \\ 0.032883 \\ 0.030841 \\ -0.187035 \\ -0.027984 \\ 0.630881 \\ 0.714847 \\ 0.230378 \end{bmatrix} \quad HD = \begin{bmatrix} -0.230378 \\ 0.714847 \\ -0.630881 \\ -0.027984 \\ 0.187035 \\ 0.030841 \\ -0.032883 \\ -0.010597 \end{bmatrix} \quad LR = \begin{bmatrix} 0.230378 \\ 0.714847 \\ 0.630881 \\ -0.027984 \\ -0.187035 \\ 0.030841 \\ 0.032883 \\ -0.010597 \end{bmatrix} \quad HR = \begin{bmatrix} -0.010597 \\ -0.032883 \\ 0.030841 \\ 0.187035 \\ -0.027984 \\ -0.630881 \\ 0.714847 \\ -0.230378 \end{bmatrix}$$

### Daubechies 5

$$\begin{array}{l}
 \text{LD} = \begin{bmatrix} 0.003336 \\ -0.012581 \\ -0.006241 \\ 0.077571 \\ -0.032245 \\ -0.242295 \\ 0.138428 \\ 0.724309 \\ 0.603829 \\ 0.160102 \end{bmatrix} \\
 \text{HD} = \begin{bmatrix} -0.160102 \\ 0.603829 \\ -0.724309 \\ 0.138428 \\ 0.242295 \\ -0.032245 \\ -0.077571 \\ -0.006241 \\ 0.012581 \\ 0.003336 \end{bmatrix} \\
 \text{LR} = \begin{bmatrix} 0.160102 \\ 0.603829 \\ 0.724309 \\ 0.138428 \\ -0.242295 \\ -0.032245 \\ 0.077571 \\ -0.006241 \\ -0.012581 \\ 0.003336 \end{bmatrix} \\
 \text{HR} = \begin{bmatrix} 0.003336 \\ 0.012581 \\ -0.006241 \\ -0.077571 \\ -0.032245 \\ 0.242295 \\ 0.138428 \\ -0.724309 \\ 0.603829 \\ -0.160102 \end{bmatrix}
 \end{array}$$

### Daubechies 6

$$\begin{array}{l}
 \text{LD} = \begin{bmatrix} -0.001077 \\ 0.004777 \\ 0.000554 \\ -0.031582 \\ 0.027523 \\ 0.097502 \\ -0.129767 \\ -0.226265 \\ 0.315250 \\ 0.751134 \\ 0.494624 \\ 0.111541 \end{bmatrix} \\
 \text{HD} = \begin{bmatrix} -0.111541 \\ 0.494624 \\ -0.751134 \\ 0.315250 \\ 0.226265 \\ -0.129767 \\ -0.097502 \\ 0.027523 \\ 0.031582 \\ 0.000554 \\ -0.004777 \\ -0.001077 \end{bmatrix} \\
 \text{LR} = \begin{bmatrix} 0.111541 \\ 0.494624 \\ 0.751134 \\ 0.315250 \\ -0.226265 \\ -0.129767 \\ 0.097502 \\ 0.027523 \\ -0.031582 \\ 0.000554 \\ 0.004777 \\ -0.001077 \end{bmatrix} \\
 \text{HR} = \begin{bmatrix} -0.001077 \\ -0.004777 \\ 0.000554 \\ 0.031582 \\ 0.027523 \\ -0.097502 \\ -0.129767 \\ 0.226265 \\ 0.315250 \\ -0.751134 \\ 0.494624 \\ -0.111541 \end{bmatrix}
 \end{array}$$

### Daubechies 7

$$\begin{array}{l}
 \text{LD} = \begin{bmatrix} 0.000354 \\ -0.001802 \\ 0.000430 \\ 0.012551 \\ -0.016575 \\ -0.038030 \\ 0.080613 \\ 0.071309 \\ -0.224036 \\ -0.143906 \\ 0.469782 \\ 0.729132 \\ 0.396539 \\ 0.077852 \end{bmatrix} \\
 \text{HD} = \begin{bmatrix} -0.077852 \\ 0.396539 \\ -0.729132 \\ 0.469782 \\ 0.143906 \\ -0.224036 \\ -0.071309 \\ 0.080613 \\ 0.038030 \\ -0.016575 \\ -0.012551 \\ 0.000430 \\ 0.001802 \\ 0.000354 \end{bmatrix} \\
 \text{LR} = \begin{bmatrix} 0.077852 \\ 0.396539 \\ 0.729132 \\ 0.469782 \\ -0.143906 \\ -0.224036 \\ 0.071309 \\ 0.080613 \\ -0.038030 \\ -0.016575 \\ 0.012551 \\ 0.000430 \\ -0.001802 \\ 0.000354 \end{bmatrix} \\
 \text{HR} = \begin{bmatrix} 0.000354 \\ 0.001802 \\ 0.000430 \\ -0.012551 \\ -0.016575 \\ 0.038030 \\ 0.080613 \\ -0.071309 \\ -0.224036 \\ 0.143906 \\ 0.469782 \\ -0.729132 \\ 0.396539 \\ -0.077852 \end{bmatrix}
 \end{array}$$

### Daubechies 8

$$\begin{array}{l}
 \text{LD} = \begin{bmatrix} -0.000117 \\ 0.000675 \\ -0.000392 \\ -0.004870 \\ 0.008746 \\ 0.013981 \\ -0.044088 \\ -0.017369 \\ 0.128747 \\ 0.000472 \\ -0.284016 \\ -0.015829 \\ 0.585355 \\ 0.675631 \\ 0.312872 \\ 0.054416 \end{bmatrix} \\
 \text{HD} = \begin{bmatrix} -0.054416 \\ 0.312872 \\ -0.675631 \\ 0.585355 \\ 0.015829 \\ -0.284016 \\ -0.000472 \\ 0.128747 \\ 0.017369 \\ -0.044088 \\ -0.013981 \\ 0.008746 \\ 0.004870 \\ -0.000392 \\ -0.000675 \\ -0.000117 \end{bmatrix} \\
 \text{LR} = \begin{bmatrix} 0.054416 \\ 0.312872 \\ 0.675631 \\ 0.585355 \\ -0.015829 \\ -0.284016 \\ 0.000472 \\ 0.128747 \\ -0.017369 \\ -0.044088 \\ 0.013981 \\ 0.008746 \\ -0.004870 \\ -0.000392 \\ 0.000675 \\ -0.000117 \end{bmatrix} \\
 \text{HR} = \begin{bmatrix} -0.000117 \\ -0.000675 \\ -0.000392 \\ 0.004870 \\ 0.008746 \\ -0.013981 \\ -0.044088 \\ 0.017369 \\ 0.128747 \\ -0.000472 \\ -0.284016 \\ 0.015829 \\ 0.585355 \\ -0.675631 \\ 0.312872 \\ -0.054416 \end{bmatrix}
 \end{array}$$

## Daubechies 9

$$\begin{array}{l}
 \text{LD} = \begin{bmatrix} 0.000039 \\ -0.000252 \\ 0.000230 \\ 0.001848 \\ -0.004282 \\ -0.004723 \\ 0.022362 \\ 0.000251 \\ -0.067633 \\ 0.030726 \\ 0.148541 \\ -0.096841 \\ -0.293274 \\ 0.133197 \\ 0.657288 \\ 0.604823 \\ 0.243835 \\ 0.038078 \end{bmatrix} \\
 \text{HD} = \begin{bmatrix} -0.038078 \\ 0.243835 \\ -0.604823 \\ 0.657288 \\ -0.133197 \\ -0.293274 \\ 0.096841 \\ 0.148541 \\ -0.030726 \\ -0.067633 \\ -0.000251 \\ 0.022362 \\ 0.004723 \\ -0.004282 \\ -0.001848 \\ 0.000230 \\ 0.000252 \\ 0.000039 \end{bmatrix} \\
 \text{LR} = \begin{bmatrix} 0.038078 \\ 0.243835 \\ 0.604823 \\ 0.657288 \\ 0.133197 \\ -0.293274 \\ -0.096841 \\ 0.148541 \\ 0.030726 \\ -0.067633 \\ 0.000251 \\ 0.022362 \\ -0.004723 \\ -0.004282 \\ 0.001848 \\ 0.000230 \\ -0.000252 \\ 0.000039 \end{bmatrix} \\
 \text{HR} = \begin{bmatrix} 0.000039 \\ 0.000252 \\ 0.000230 \\ -0.001848 \\ -0.004282 \\ 0.004723 \\ 0.022362 \\ -0.000251 \\ -0.067633 \\ -0.030726 \\ 0.148541 \\ 0.096841 \\ -0.293274 \\ -0.133197 \\ 0.657288 \\ -0.604823 \\ 0.243835 \\ -0.038078 \end{bmatrix}
 \end{array}$$

## Daubechies 10

$$\begin{array}{l}
 \text{LD} = \begin{bmatrix} -0.000013 \\ 0.000094 \\ -0.000116 \\ -0.000686 \\ 0.001992 \\ 0.001395 \\ -0.010733 \\ 0.003607 \\ 0.033213 \\ -0.029458 \\ -0.071394 \\ 0.093057 \\ 0.127369 \\ -0.195946 \\ -0.249846 \\ 0.281172 \\ 0.688459 \\ 0.527201 \\ 0.188177 \\ 0.026670 \end{bmatrix} \\
 \text{HD} = \begin{bmatrix} -0.026670 \\ 0.188177 \\ -0.527201 \\ 0.688459 \\ -0.281172 \\ -0.249846 \\ 0.195946 \\ 0.127369 \\ -0.093057 \\ -0.071394 \\ 0.029458 \\ 0.033213 \\ -0.003607 \\ -0.010733 \\ -0.001395 \\ 0.001992 \\ 0.000686 \\ -0.000116 \\ -0.000094 \\ -0.000013 \end{bmatrix} \\
 \text{LR} = \begin{bmatrix} 0.026670 \\ 0.188177 \\ 0.527201 \\ 0.688459 \\ 0.281172 \\ -0.249846 \\ -0.195946 \\ 0.127369 \\ 0.093057 \\ -0.071394 \\ -0.029458 \\ 0.033213 \\ 0.003607 \\ -0.010733 \\ 0.001395 \\ 0.001992 \\ -0.000686 \\ -0.000116 \\ 0.000094 \\ -0.000013 \end{bmatrix} \\
 \text{HR} = \begin{bmatrix} -0.000013 \\ -0.000094 \\ -0.000116 \\ 0.000686 \\ 0.001992 \\ -0.001395 \\ -0.010733 \\ -0.003607 \\ 0.033213 \\ 0.029458 \\ -0.071394 \\ -0.093057 \\ 0.127369 \\ 0.195946 \\ -0.249846 \\ -0.281172 \\ 0.688459 \\ -0.527201 \\ 0.188177 \\ -0.026670 \end{bmatrix}
 \end{array}$$

## A.2 Symlets

### Symlet 2

$$\begin{array}{l}
 \text{LD} = \begin{bmatrix} -0.129410 \\ 0.224144 \\ 0.836516 \\ 0.482963 \end{bmatrix} \\
 \text{HD} = \begin{bmatrix} -0.482963 \\ 0.836516 \\ -0.224144 \\ -0.129410 \end{bmatrix} \\
 \text{LR} = \begin{bmatrix} 0.482963 \\ 0.836516 \\ 0.224144 \\ -0.129410 \end{bmatrix} \\
 \text{HR} = \begin{bmatrix} -0.129410 \\ -0.224144 \\ 0.836516 \\ -0.482963 \end{bmatrix}
 \end{array}$$

### Symlet 3

$$\begin{array}{l}
 \text{LD} = \begin{bmatrix} 0.035226 \\ -0.085441 \\ -0.135011 \\ 0.459878 \\ 0.806892 \\ 0.332671 \end{bmatrix} \\
 \text{HD} = \begin{bmatrix} -0.332671 \\ 0.806892 \\ -0.459878 \\ -0.135011 \\ 0.085441 \\ 0.035226 \end{bmatrix} \\
 \text{LR} = \begin{bmatrix} 0.332671 \\ 0.806892 \\ 0.459878 \\ -0.135011 \\ -0.085441 \\ 0.035226 \end{bmatrix} \\
 \text{HR} = \begin{bmatrix} 0.035226 \\ 0.085441 \\ -0.135011 \\ -0.459878 \\ 0.806892 \\ -0.332671 \end{bmatrix}
 \end{array}$$



## Symlet 4

$$\begin{array}{cccc}
 \text{LD} = \begin{bmatrix} -0.075766 \\ -0.029636 \\ 0.497619 \\ 0.803739 \\ 0.297858 \\ -0.099220 \\ -0.012604 \\ 0.032223 \end{bmatrix} &
 \text{HD} = \begin{bmatrix} -0.032223 \\ -0.012604 \\ 0.099220 \\ 0.297858 \\ -0.803739 \\ 0.497619 \\ 0.029636 \\ -0.075766 \end{bmatrix} &
 \text{LR} = \begin{bmatrix} 0.032223 \\ -0.012604 \\ -0.099220 \\ 0.297858 \\ 0.803739 \\ 0.497619 \\ -0.029636 \\ -0.075766 \end{bmatrix} &
 \text{HR} = \begin{bmatrix} -0.075766 \\ 0.029636 \\ 0.497619 \\ -0.803739 \\ 0.297858 \\ 0.099220 \\ -0.012604 \\ -0.032223 \end{bmatrix}
 \end{array}$$

## Symlet 5

$$\begin{array}{cccc}
 \text{LD} = \begin{bmatrix} 0.027333 \\ 0.029519 \\ -0.039134 \\ 0.199398 \\ 0.723408 \\ 0.633979 \\ 0.016602 \\ -0.175328 \\ -0.021102 \\ 0.019539 \end{bmatrix} &
 \text{HD} = \begin{bmatrix} -0.019539 \\ -0.021102 \\ 0.175328 \\ 0.016602 \\ -0.633979 \\ 0.723408 \\ -0.199398 \\ -0.039134 \\ -0.029519 \\ 0.027333 \end{bmatrix} &
 \text{LR} = \begin{bmatrix} 0.019539 \\ -0.021102 \\ -0.175328 \\ 0.016602 \\ 0.633979 \\ 0.723408 \\ 0.199398 \\ -0.039134 \\ 0.029519 \\ 0.027333 \end{bmatrix} &
 \text{HR} = \begin{bmatrix} 0.027333 \\ -0.029519 \\ -0.039134 \\ -0.199398 \\ 0.723408 \\ -0.633979 \\ 0.016602 \\ 0.175328 \\ -0.021102 \\ -0.019539 \end{bmatrix}
 \end{array}$$

## Symlet 6

$$\begin{array}{cccc}
 \text{LD} = \begin{bmatrix} 0.015404 \\ 0.003491 \\ -0.117990 \\ -0.048312 \\ 0.491056 \\ 0.787641 \\ 0.337929 \\ -0.072638 \\ -0.021060 \\ 0.044725 \\ 0.001768 \\ -0.007801 \end{bmatrix} &
 \text{HD} = \begin{bmatrix} 0.007801 \\ 0.001768 \\ -0.044725 \\ -0.021060 \\ 0.072638 \\ 0.337929 \\ -0.787641 \\ 0.491056 \\ 0.048312 \\ -0.117990 \\ -0.003491 \\ 0.015404 \end{bmatrix} &
 \text{LR} = \begin{bmatrix} -0.007801 \\ 0.001768 \\ 0.044725 \\ -0.021060 \\ -0.072638 \\ 0.337929 \\ 0.787641 \\ 0.491056 \\ -0.048312 \\ -0.117990 \\ 0.003491 \\ 0.015404 \end{bmatrix} &
 \text{HR} = \begin{bmatrix} 0.015404 \\ -0.003491 \\ -0.117990 \\ 0.048312 \\ 0.491056 \\ -0.787641 \\ 0.337929 \\ 0.072638 \\ -0.021060 \\ -0.044725 \\ 0.001768 \\ 0.007801 \end{bmatrix}
 \end{array}$$

## Symlet 7

$$\begin{array}{cccc}
 \text{LD} = \begin{bmatrix} 0.002682 \\ -0.001047 \\ -0.012636 \\ 0.030516 \\ 0.067893 \\ -0.049553 \\ 0.017441 \\ 0.536102 \\ 0.767764 \\ 0.288630 \\ -0.140047 \\ -0.107808 \\ 0.004010 \\ 0.010268 \end{bmatrix} &
 \text{HD} = \begin{bmatrix} -0.010268 \\ 0.004010 \\ 0.107808 \\ -0.140047 \\ -0.288630 \\ 0.767764 \\ -0.536102 \\ 0.017441 \\ 0.049553 \\ 0.067893 \\ -0.030516 \\ -0.012636 \\ 0.001047 \\ 0.002682 \end{bmatrix} &
 \text{LR} = \begin{bmatrix} 0.010268 \\ 0.004010 \\ -0.107808 \\ -0.140047 \\ 0.288630 \\ 0.767764 \\ 0.536102 \\ 0.017441 \\ -0.049553 \\ 0.067893 \\ 0.030516 \\ -0.012636 \\ -0.001047 \\ 0.002682 \end{bmatrix} &
 \text{HR} = \begin{bmatrix} 0.002682 \\ 0.001047 \\ -0.012636 \\ -0.030516 \\ 0.067893 \\ 0.049553 \\ 0.017441 \\ -0.536102 \\ 0.767764 \\ -0.288630 \\ -0.140047 \\ 0.107808 \\ 0.004010 \\ -0.010268 \end{bmatrix}
 \end{array}$$

## Symlet 8

$$\begin{array}{l}
 \text{LD} = \begin{bmatrix} -0.003382 \\ -0.000542 \\ 0.031695 \\ 0.007607 \\ -0.143294 \\ -0.061273 \\ 0.481360 \\ 0.777186 \\ 0.364442 \\ -0.051946 \\ -0.027219 \\ 0.049137 \\ 0.003809 \\ -0.014952 \\ -0.000303 \\ 0.001890 \end{bmatrix} \\
 \text{HD} = \begin{bmatrix} -0.001890 \\ -0.000303 \\ 0.014952 \\ 0.003809 \\ -0.049137 \\ -0.027219 \\ 0.051946 \\ 0.364442 \\ -0.777186 \\ 0.481360 \\ 0.061273 \\ -0.143294 \\ -0.007607 \\ 0.031695 \\ 0.000542 \\ -0.003382 \end{bmatrix} \\
 \text{LR} = \begin{bmatrix} 0.001890 \\ -0.000303 \\ -0.014952 \\ 0.003809 \\ 0.049137 \\ -0.027219 \\ -0.051946 \\ 0.364442 \\ 0.777186 \\ 0.481360 \\ -0.061273 \\ -0.143294 \\ 0.007607 \\ 0.031695 \\ -0.000542 \\ -0.003382 \end{bmatrix} \\
 \text{HR} = \begin{bmatrix} -0.003382 \\ 0.000542 \\ 0.031695 \\ -0.007607 \\ -0.143294 \\ 0.061273 \\ 0.481360 \\ -0.777186 \\ 0.364442 \\ 0.051946 \\ -0.027219 \\ -0.049137 \\ 0.003809 \\ 0.014952 \\ -0.000303 \\ -0.001890 \end{bmatrix}
 \end{array}$$

## A.3 Coiflets

### Coiflets 1

$$\begin{array}{l}
 \text{LD} = \begin{bmatrix} -0.015656 \\ -0.072733 \\ 0.384865 \\ 0.852572 \\ 0.337898 \\ -0.072733 \end{bmatrix} \\
 \text{HD} = \begin{bmatrix} 0.072733 \\ 0.337898 \\ -0.852572 \\ 0.384865 \\ 0.072733 \\ -0.015656 \end{bmatrix} \\
 \text{LR} = \begin{bmatrix} -0.072733 \\ 0.337898 \\ 0.852572 \\ 0.384865 \\ -0.072733 \\ -0.015656 \end{bmatrix} \\
 \text{HR} = \begin{bmatrix} -0.015656 \\ 0.072733 \\ 0.384865 \\ -0.852572 \\ 0.337898 \\ 0.072733 \end{bmatrix}
 \end{array}$$

### Coiflets 2

$$\begin{array}{l}
 \text{LD} = \begin{bmatrix} -0.000721 \\ -0.001823 \\ 0.005611 \\ 0.023680 \\ -0.059434 \\ -0.076489 \\ 0.417005 \\ 0.812724 \\ 0.386110 \\ -0.067373 \\ -0.041465 \\ 0.016387 \end{bmatrix} \\
 \text{HD} = \begin{bmatrix} -0.016387 \\ -0.041465 \\ 0.067373 \\ 0.386110 \\ -0.812724 \\ 0.417005 \\ 0.076489 \\ -0.059434 \\ -0.023680 \\ 0.005611 \\ 0.001823 \\ -0.000721 \end{bmatrix} \\
 \text{LR} = \begin{bmatrix} 0.016387 \\ -0.041465 \\ -0.067373 \\ 0.386110 \\ 0.812724 \\ 0.417005 \\ -0.076489 \\ -0.059434 \\ 0.023680 \\ 0.005611 \\ -0.001823 \\ -0.000721 \end{bmatrix} \\
 \text{HR} = \begin{bmatrix} -0.000721 \\ 0.001823 \\ 0.005611 \\ -0.023680 \\ -0.059434 \\ 0.076489 \\ 0.417005 \\ -0.812724 \\ 0.386110 \\ 0.067373 \\ -0.041465 \\ -0.016387 \end{bmatrix}
 \end{array}$$

### Coiflets 3

$$\begin{array}{l}
 \text{LD} = \begin{bmatrix} -0.000035 \\ -0.000071 \\ 0.000466 \\ 0.001118 \\ -0.002575 \\ -0.009008 \\ 0.015881 \\ 0.034555 \\ -0.082302 \\ -0.071800 \\ 0.428483 \\ 0.793777 \\ 0.405177 \\ -0.061123 \\ -0.065772 \\ 0.023453 \\ 0.007783 \\ -0.003794 \end{bmatrix} \\
 \text{HD} = \begin{bmatrix} 0.003794 \\ 0.007783 \\ -0.023453 \\ -0.065772 \\ 0.061123 \\ 0.405177 \\ -0.793777 \\ 0.428483 \\ 0.071800 \\ -0.082302 \\ -0.034555 \\ 0.015881 \\ 0.009008 \\ -0.002575 \\ -0.001118 \\ 0.000466 \\ 0.000071 \\ -0.000035 \end{bmatrix} \\
 \text{LR} = \begin{bmatrix} -0.003794 \\ 0.007783 \\ 0.023453 \\ -0.065772 \\ -0.061123 \\ 0.405177 \\ 0.793777 \\ 0.428483 \\ -0.071800 \\ -0.082302 \\ 0.034555 \\ 0.015881 \\ -0.009008 \\ -0.002575 \\ 0.001118 \\ 0.000466 \\ -0.000071 \\ -0.000035 \end{bmatrix} \\
 \text{HR} = \begin{bmatrix} -0.000035 \\ 0.000071 \\ 0.000466 \\ -0.001118 \\ -0.002575 \\ 0.009008 \\ 0.015881 \\ -0.034555 \\ -0.082302 \\ 0.071800 \\ 0.428483 \\ -0.793777 \\ 0.405177 \\ 0.061123 \\ -0.065772 \\ -0.023453 \\ 0.007783 \\ 0.003794 \end{bmatrix}
 \end{array}$$

## Coiflets 4

$$\begin{array}{l}
 \text{LD} = \begin{bmatrix} -0.000002 \\ -0.000003 \\ 0.000031 \\ 0.000062 \\ -0.000260 \\ -0.000589 \\ 0.001267 \\ 0.003751 \\ -0.005658 \\ -0.015212 \\ 0.025082 \\ 0.039334 \\ -0.096220 \\ -0.066627 \\ 0.434386 \\ 0.782239 \\ 0.415308 \\ -0.056077 \\ -0.081267 \\ 0.026682 \\ 0.016069 \\ -0.007346 \\ -0.001629 \\ 0.000892 \end{bmatrix} \\
 \text{HD} = \begin{bmatrix} -0.000892 \\ -0.001629 \\ 0.007346 \\ 0.016069 \\ -0.026682 \\ -0.081267 \\ 0.056077 \\ 0.415308 \\ -0.782239 \\ 0.434386 \\ 0.066627 \\ -0.096220 \\ -0.039334 \\ 0.025082 \\ 0.015212 \\ -0.005658 \\ -0.003751 \\ 0.001267 \\ 0.000589 \\ -0.000260 \\ -0.000062 \\ 0.000031 \\ 0.000003 \\ -0.000002 \end{bmatrix} \\
 \text{LR} = \begin{bmatrix} 0.000892 \\ -0.001629 \\ -0.007346 \\ 0.016069 \\ 0.026682 \\ -0.081267 \\ -0.056077 \\ 0.415308 \\ 0.782239 \\ 0.434386 \\ -0.066627 \\ -0.096220 \\ 0.039334 \\ 0.025082 \\ -0.015212 \\ -0.005658 \\ 0.003751 \\ 0.001267 \\ -0.000589 \\ -0.000260 \\ 0.000062 \\ 0.000031 \\ -0.000003 \\ -0.000002 \end{bmatrix} \\
 \text{HR} = \begin{bmatrix} -0.000002 \\ 0.000003 \\ 0.000031 \\ -0.000062 \\ -0.000260 \\ 0.000589 \\ 0.001267 \\ -0.003751 \\ -0.005658 \\ 0.015212 \\ 0.025082 \\ -0.039334 \\ -0.096220 \\ 0.066627 \\ 0.434386 \\ -0.782239 \\ 0.415308 \\ 0.056077 \\ -0.081267 \\ -0.026682 \\ 0.016069 \\ 0.007346 \\ -0.001629 \\ -0.000892 \end{bmatrix}
 \end{array}$$

## Coiflets 5

$$\begin{array}{l}
 \text{LD} = \begin{bmatrix} -0.000000 \\ -0.000000 \\ 0.000002 \\ 0.000004 \\ -0.000021 \\ -0.000041 \\ 0.000141 \\ 0.000302 \\ -0.000638 \\ -0.001663 \\ 0.002433 \\ 0.006764 \\ -0.009164 \\ -0.019762 \\ 0.032684 \\ 0.041289 \\ -0.105574 \\ -0.062036 \\ 0.437992 \\ 0.774290 \\ 0.421566 \\ -0.052043 \\ -0.091920 \\ 0.028168 \\ 0.023408 \\ -0.010131 \\ -0.004159 \\ 0.002178 \\ 0.000359 \\ -0.000212 \end{bmatrix} \\
 \text{HD} = \begin{bmatrix} 0.000212 \\ 0.000359 \\ -0.002178 \\ -0.004159 \\ 0.010131 \\ 0.023408 \\ -0.028168 \\ -0.091920 \\ 0.052043 \\ 0.421566 \\ -0.774290 \\ 0.437992 \\ 0.062036 \\ -0.105574 \\ -0.041289 \\ 0.032684 \\ 0.019762 \\ -0.009164 \\ -0.006764 \\ 0.002433 \\ 0.001663 \\ -0.000638 \\ -0.000302 \\ 0.000141 \\ 0.000041 \\ -0.000021 \\ -0.000004 \\ 0.000002 \\ 0.000000 \\ -0.000000 \end{bmatrix} \\
 \text{LR} = \begin{bmatrix} -0.000212 \\ 0.000359 \\ 0.002178 \\ -0.004159 \\ -0.010131 \\ 0.023408 \\ 0.028168 \\ -0.091920 \\ -0.052043 \\ 0.421566 \\ 0.774290 \\ 0.437992 \\ -0.062036 \\ -0.105574 \\ 0.041289 \\ 0.032684 \\ -0.019762 \\ -0.009164 \\ 0.006764 \\ 0.002433 \\ -0.001663 \\ -0.000638 \\ 0.000302 \\ 0.000141 \\ -0.000041 \\ -0.000021 \\ 0.000004 \\ 0.000002 \\ -0.000000 \\ -0.000000 \end{bmatrix} \\
 \text{HR} = \begin{bmatrix} -0.000000 \\ 0.000000 \\ 0.000002 \\ -0.000004 \\ -0.000021 \\ 0.000041 \\ 0.000141 \\ -0.000302 \\ -0.000638 \\ 0.001663 \\ 0.002433 \\ -0.006764 \\ -0.009164 \\ 0.019762 \\ 0.032684 \\ -0.041289 \\ -0.105574 \\ 0.062036 \\ 0.437992 \\ -0.774290 \\ 0.421566 \\ 0.052043 \\ -0.091920 \\ -0.028168 \\ 0.023408 \\ 0.010131 \\ -0.004159 \\ -0.002178 \\ 0.000359 \\ 0.000212 \end{bmatrix}
 \end{array}$$

## A.4 Biortogonal

### Biortogonal 1.3

$$\begin{aligned}
 \text{LD} &= \begin{bmatrix} -0.088388 \\ 0.088388 \\ 0.707107 \\ 0.707107 \\ 0.088388 \\ -0.088388 \end{bmatrix} &
 \text{HD} &= \begin{bmatrix} -0.000000 \\ 0.000000 \\ -0.707107 \\ 0.707107 \\ -0.000000 \\ 0.000000 \end{bmatrix} &
 \text{LR} &= \begin{bmatrix} 0.000000 \\ 0.000000 \\ 0.707107 \\ 0.707107 \\ 0.000000 \\ 0.000000 \end{bmatrix} &
 \text{HR} &= \begin{bmatrix} -0.088388 \\ -0.088388 \\ 0.707107 \\ -0.707107 \\ 0.088388 \\ 0.088388 \end{bmatrix}
 \end{aligned}$$

### Biortogonal 1.5

$$\begin{aligned}
 \text{LD} &= \begin{bmatrix} 0.016573 \\ -0.016573 \\ -0.121534 \\ 0.121534 \\ 0.707107 \\ 0.707107 \\ 0.121534 \\ -0.121534 \\ -0.016573 \\ 0.016573 \end{bmatrix} &
 \text{HD} &= \begin{bmatrix} -0.000000 \\ 0.000000 \\ -0.000000 \\ 0.000000 \\ -0.707107 \\ 0.707107 \\ -0.000000 \\ 0.000000 \\ -0.000000 \\ 0.000000 \end{bmatrix} &
 \text{LR} &= \begin{bmatrix} 0.000000 \\ 0.000000 \\ 0.000000 \\ 0.000000 \\ 0.707107 \\ 0.707107 \\ 0.000000 \\ 0.000000 \\ 0.000000 \\ 0.000000 \end{bmatrix} &
 \text{HR} &= \begin{bmatrix} 0.016573 \\ 0.016573 \\ -0.121534 \\ -0.121534 \\ 0.707107 \\ -0.707107 \\ 0.121534 \\ 0.121534 \\ -0.016573 \\ -0.016573 \end{bmatrix}
 \end{aligned}$$

### Biortogonal 2.2

$$\begin{aligned}
 \text{LD} &= \begin{bmatrix} 0.000000 \\ -0.176777 \\ 0.353553 \\ 1.060660 \\ 0.353553 \\ -0.176777 \end{bmatrix} &
 \text{HD} &= \begin{bmatrix} -0.000000 \\ 0.353553 \\ -0.707107 \\ 0.353553 \\ -0.000000 \\ 0.000000 \end{bmatrix} &
 \text{LR} &= \begin{bmatrix} 0.000000 \\ 0.353553 \\ 0.707107 \\ 0.353553 \\ 0.000000 \\ 0.000000 \end{bmatrix} &
 \text{HR} &= \begin{bmatrix} 0.000000 \\ 0.176777 \\ 0.353553 \\ -1.060660 \\ 0.353553 \\ 0.176777 \end{bmatrix}
 \end{aligned}$$

### Biortogonal 2.4

$$\begin{aligned}
 \text{LD} &= \begin{bmatrix} 0.000000 \\ 0.033146 \\ -0.066291 \\ -0.176777 \\ 0.419845 \\ 0.994369 \\ 0.419845 \\ -0.176777 \\ -0.066291 \\ 0.033146 \end{bmatrix} &
 \text{HD} &= \begin{bmatrix} -0.000000 \\ 0.000000 \\ -0.000000 \\ 0.353553 \\ -0.707107 \\ 0.353553 \\ -0.000000 \\ 0.000000 \\ -0.000000 \\ 0.000000 \end{bmatrix} &
 \text{LR} &= \begin{bmatrix} 0.000000 \\ 0.000000 \\ 0.000000 \\ 0.353553 \\ 0.707107 \\ 0.353553 \\ 0.000000 \\ 0.000000 \\ 0.000000 \\ 0.000000 \end{bmatrix} &
 \text{HR} &= \begin{bmatrix} 0.000000 \\ -0.033146 \\ -0.066291 \\ 0.176777 \\ 0.419845 \\ -0.994369 \\ 0.419845 \\ 0.176777 \\ -0.066291 \\ -0.033146 \end{bmatrix}
 \end{aligned}$$

### Biortogonal 2.6

$$\begin{aligned}
 \text{LD} &= \begin{bmatrix} 0.000000 \\ -0.006905 \\ 0.013811 \\ 0.046956 \\ -0.107723 \\ -0.169871 \\ 0.447466 \\ 0.966748 \\ 0.447466 \\ -0.169871 \\ -0.107723 \\ 0.046956 \\ 0.013811 \\ -0.006905 \end{bmatrix} &
 \text{HD} &= \begin{bmatrix} -0.000000 \\ 0.000000 \\ -0.000000 \\ 0.000000 \\ -0.000000 \\ 0.353553 \\ -0.707107 \\ 0.353553 \\ -0.000000 \\ 0.000000 \\ -0.000000 \\ 0.000000 \\ -0.000000 \\ 0.000000 \end{bmatrix} &
 \text{LR} &= \begin{bmatrix} 0.000000 \\ 0.000000 \\ 0.000000 \\ 0.000000 \\ 0.000000 \\ 0.353553 \\ 0.707107 \\ 0.353553 \\ 0.000000 \\ 0.000000 \\ 0.000000 \\ 0.000000 \\ 0.000000 \\ 0.000000 \end{bmatrix} &
 \text{HR} &= \begin{bmatrix} 0.000000 \\ 0.006905 \\ 0.013811 \\ -0.046956 \\ -0.107723 \\ 0.169871 \\ 0.447466 \\ -0.966748 \\ 0.447466 \\ 0.169871 \\ -0.107723 \\ -0.046956 \\ 0.013811 \\ 0.006905 \end{bmatrix}
 \end{aligned}$$

## Biortogonal 2.8

$$\begin{array}{l}
 \text{LD} = \begin{bmatrix} 0.000000 \\ 0.001511 \\ -0.003021 \\ -0.012948 \\ 0.028916 \\ 0.052998 \\ -0.134913 \\ -0.163829 \\ 0.462571 \\ 0.951642 \\ 0.462571 \\ -0.163829 \\ -0.134913 \\ 0.052998 \\ 0.028916 \\ -0.012948 \\ -0.003021 \\ 0.001511 \end{bmatrix} \\
 \text{HD} = \begin{bmatrix} -0.000000 \\ 0.000000 \\ -0.000000 \\ 0.000000 \\ -0.000000 \\ 0.000000 \\ 0.000000 \\ 0.353553 \\ -0.707107 \\ 0.353553 \\ -0.000000 \\ 0.000000 \\ -0.000000 \\ 0.000000 \\ -0.000000 \\ 0.000000 \\ -0.000000 \\ 0.000000 \end{bmatrix} \\
 \text{LR} = \begin{bmatrix} 0.000000 \\ 0.000000 \\ 0.000000 \\ 0.000000 \\ 0.000000 \\ 0.000000 \\ 0.000000 \\ 0.353553 \\ 0.707107 \\ 0.353553 \\ 0.000000 \\ 0.000000 \\ 0.000000 \\ 0.000000 \\ 0.000000 \\ 0.000000 \\ 0.000000 \\ 0.000000 \end{bmatrix} \\
 \text{HR} = \begin{bmatrix} 0.000000 \\ -0.001511 \\ -0.003021 \\ 0.012948 \\ 0.028916 \\ -0.052998 \\ -0.134913 \\ 0.163829 \\ 0.462571 \\ -0.951642 \\ 0.462571 \\ 0.163829 \\ -0.134913 \\ -0.052998 \\ 0.028916 \\ 0.012948 \\ -0.003021 \\ -0.001511 \end{bmatrix}
 \end{array}$$

## Biortogonal 3.1

$$\begin{array}{l}
 \text{LD} = \begin{bmatrix} -0.353553 \\ 1.060660 \\ 1.060660 \\ -0.353553 \end{bmatrix} \\
 \text{HD} = \begin{bmatrix} -0.176777 \\ 0.530330 \\ -0.530330 \\ 0.176777 \end{bmatrix} \\
 \text{LR} = \begin{bmatrix} 0.176777 \\ 0.530330 \\ 0.530330 \\ 0.176777 \end{bmatrix} \\
 \text{HR} = \begin{bmatrix} -0.353553 \\ -1.060660 \\ 1.060660 \\ 0.353553 \end{bmatrix}
 \end{array}$$

## Biortogonal 3.3

$$\begin{array}{l}
 \text{LD} = \begin{bmatrix} 0.066291 \\ -0.198874 \\ -0.154680 \\ 0.994369 \\ 0.994369 \\ -0.154680 \\ -0.198874 \\ 0.066291 \end{bmatrix} \\
 \text{HD} = \begin{bmatrix} -0.000000 \\ 0.000000 \\ -0.176777 \\ 0.530330 \\ -0.530330 \\ 0.176777 \\ -0.000000 \\ 0.000000 \end{bmatrix} \\
 \text{LR} = \begin{bmatrix} 0.000000 \\ 0.000000 \\ 0.176777 \\ 0.530330 \\ 0.530330 \\ 0.176777 \\ 0.000000 \\ 0.000000 \end{bmatrix} \\
 \text{HR} = \begin{bmatrix} 0.066291 \\ 0.198874 \\ -0.154680 \\ -0.994369 \\ 0.994369 \\ 0.154680 \\ -0.198874 \\ -0.066291 \end{bmatrix}
 \end{array}$$

## Biortogonal 3.5

$$\begin{array}{l}
 \text{LD} = \begin{bmatrix} -0.013811 \\ 0.041432 \\ 0.052481 \\ -0.267927 \\ -0.071816 \\ 0.966748 \\ 0.966748 \\ -0.071816 \\ -0.267927 \\ 0.052481 \\ 0.041432 \\ -0.013811 \end{bmatrix} \\
 \text{HD} = \begin{bmatrix} -0.000000 \\ 0.000000 \\ -0.000000 \\ 0.000000 \\ -0.176777 \\ 0.530330 \\ -0.530330 \\ 0.176777 \\ -0.000000 \\ 0.000000 \\ -0.000000 \\ 0.000000 \end{bmatrix} \\
 \text{LR} = \begin{bmatrix} 0.000000 \\ 0.000000 \\ 0.000000 \\ 0.000000 \\ 0.176777 \\ 0.530330 \\ 0.530330 \\ 0.176777 \\ 0.000000 \\ 0.000000 \\ 0.000000 \\ 0.000000 \end{bmatrix} \\
 \text{HR} = \begin{bmatrix} -0.013811 \\ -0.041432 \\ 0.052481 \\ 0.267927 \\ -0.071816 \\ -0.966748 \\ 0.966748 \\ 0.071816 \\ -0.267927 \\ -0.052481 \\ 0.041432 \\ 0.013811 \end{bmatrix}
 \end{array}$$

### Biortogonal 3.7

$$\begin{array}{cccc}
 \text{LD} = & \begin{bmatrix} 0.003021 \\ -0.009063 \\ -0.016832 \\ 0.074664 \\ 0.031333 \\ -0.301159 \\ -0.026499 \\ 0.951642 \\ 0.951642 \\ -0.026499 \\ -0.301159 \\ 0.031333 \\ 0.074664 \\ -0.016832 \\ -0.009063 \\ 0.003021 \end{bmatrix} & \text{HD} = & \begin{bmatrix} -0.000000 \\ 0.000000 \\ -0.000000 \\ 0.000000 \\ -0.000000 \\ 0.000000 \\ -0.176777 \\ 0.530330 \\ -0.530330 \\ 0.176777 \\ -0.000000 \\ 0.000000 \\ -0.000000 \\ 0.000000 \\ -0.000000 \\ 0.000000 \end{bmatrix} & \text{LR} = & \begin{bmatrix} 0.000000 \\ 0.000000 \\ 0.000000 \\ 0.000000 \\ 0.000000 \\ 0.000000 \\ 0.176777 \\ 0.530330 \\ 0.530330 \\ 0.176777 \\ 0.000000 \\ 0.000000 \\ 0.000000 \\ 0.000000 \\ 0.000000 \\ 0.000000 \end{bmatrix} & \text{HR} = & \begin{bmatrix} 0.003021 \\ 0.009063 \\ -0.016832 \\ -0.074664 \\ 0.031333 \\ 0.301159 \\ -0.026499 \\ -0.951642 \\ 0.951642 \\ 0.026499 \\ -0.301159 \\ -0.031333 \\ 0.074664 \\ 0.016832 \\ -0.009063 \\ -0.003021 \end{bmatrix}
 \end{array}$$

### Biortogonal 3.9

$$\begin{array}{cccc}
 \text{LD} = & \begin{bmatrix} -0.000680 \\ 0.002039 \\ 0.005060 \\ -0.020619 \\ -0.014113 \\ 0.099135 \\ 0.012300 \\ -0.320192 \\ 0.002050 \\ 0.942126 \\ 0.942126 \\ 0.002050 \\ -0.320192 \\ 0.012300 \\ 0.099135 \\ -0.014113 \\ -0.020619 \\ 0.005060 \\ 0.002039 \\ -0.000680 \end{bmatrix} & \text{HD} = & \begin{bmatrix} -0.000000 \\ 0.000000 \\ -0.000000 \\ 0.000000 \\ -0.000000 \\ 0.000000 \\ -0.000000 \\ 0.000000 \\ -0.176777 \\ 0.530330 \\ -0.530330 \\ 0.176777 \\ -0.000000 \\ 0.000000 \\ -0.000000 \\ 0.000000 \\ -0.000000 \\ 0.000000 \\ -0.000000 \\ 0.000000 \end{bmatrix} & \text{LR} = & \begin{bmatrix} 0.000000 \\ 0.000000 \\ 0.000000 \\ 0.000000 \\ 0.000000 \\ 0.000000 \\ 0.000000 \\ 0.000000 \\ 0.176777 \\ 0.530330 \\ 0.530330 \\ 0.176777 \\ 0.000000 \\ 0.000000 \\ 0.000000 \\ 0.000000 \\ 0.000000 \\ 0.000000 \\ 0.000000 \\ 0.000000 \end{bmatrix} & \text{HR} = & \begin{bmatrix} -0.000680 \\ -0.002039 \\ 0.005060 \\ 0.020619 \\ -0.014113 \\ -0.099135 \\ 0.012300 \\ 0.320192 \\ 0.002050 \\ -0.942126 \\ 0.942126 \\ -0.002050 \\ -0.320192 \\ -0.012300 \\ 0.099135 \\ 0.014113 \\ -0.020619 \\ -0.005060 \\ 0.002039 \\ 0.000680 \end{bmatrix}
 \end{array}$$

### Biortogonal 4.4

$$\begin{array}{cccc}
 \text{LD} = & \begin{bmatrix} 0.000000 \\ 0.037828 \\ -0.023849 \\ -0.110624 \\ 0.377403 \\ 0.852699 \\ 0.377403 \\ -0.110624 \\ -0.023849 \\ 0.037828 \end{bmatrix} & \text{HD} = & \begin{bmatrix} -0.000000 \\ -0.064539 \\ 0.040689 \\ 0.418092 \\ -0.788486 \\ 0.418092 \\ 0.040689 \\ -0.064539 \\ -0.000000 \\ 0.000000 \end{bmatrix} & \text{LR} = & \begin{bmatrix} 0.000000 \\ -0.064539 \\ -0.040689 \\ 0.418092 \\ 0.788486 \\ 0.418092 \\ -0.040689 \\ -0.064539 \\ 0.000000 \\ 0.000000 \end{bmatrix} & \text{HR} = & \begin{bmatrix} 0.000000 \\ -0.037828 \\ -0.023849 \\ 0.110624 \\ 0.377403 \\ -0.852699 \\ 0.377403 \\ 0.110624 \\ -0.023849 \\ -0.037828 \end{bmatrix}
 \end{array}$$

## Biortogonal 5.5

$$\begin{array}{l}
 \text{LD} = \begin{bmatrix} 0.000000 \\ 0.000000 \\ 0.039687 \\ 0.007948 \\ -0.054464 \\ 0.345605 \\ 0.736660 \\ 0.345605 \\ -0.054464 \\ 0.007948 \\ 0.039687 \\ 0.000000 \end{bmatrix}
 \end{array}
 \begin{array}{l}
 \text{HD} = \begin{bmatrix} -0.013457 \\ -0.002695 \\ 0.136707 \\ -0.093505 \\ -0.476803 \\ 0.899506 \\ -0.476803 \\ -0.093505 \\ 0.136707 \\ -0.002695 \\ -0.013457 \\ 0.000000 \end{bmatrix}
 \end{array}
 \begin{array}{l}
 \text{LR} = \begin{bmatrix} 0.013457 \\ -0.002695 \\ -0.136707 \\ -0.093505 \\ 0.476803 \\ 0.899506 \\ 0.476803 \\ -0.093505 \\ -0.136707 \\ -0.002695 \\ 0.013457 \\ 0.000000 \end{bmatrix}
 \end{array}
 \begin{array}{l}
 \text{HR} = \begin{bmatrix} 0.000000 \\ -0.000000 \\ 0.039687 \\ -0.007948 \\ -0.054464 \\ -0.345605 \\ 0.736660 \\ -0.345605 \\ -0.054464 \\ -0.007948 \\ 0.039687 \\ -0.000000 \end{bmatrix}
 \end{array}$$

## Biortogonal 6.8

$$\begin{array}{l}
 \text{LD} = \begin{bmatrix} 0.000000 \\ 0.001909 \\ -0.001914 \\ -0.016991 \\ 0.011935 \\ 0.049733 \\ -0.077263 \\ -0.094059 \\ 0.420796 \\ 0.825923 \\ 0.420796 \\ -0.094059 \\ -0.077263 \\ 0.049733 \\ 0.011935 \\ -0.016991 \\ -0.001914 \\ 0.001909 \end{bmatrix}
 \end{array}
 \begin{array}{l}
 \text{HD} = \begin{bmatrix} 0.000000 \\ -0.000000 \\ 0.000000 \\ 0.014426 \\ -0.014468 \\ -0.078722 \\ 0.040368 \\ 0.417849 \\ -0.758908 \\ 0.417849 \\ 0.040368 \\ -0.078722 \\ -0.014468 \\ 0.014426 \\ 0.000000 \\ -0.000000 \\ 0.000000 \\ -0.000000 \end{bmatrix}
 \end{array}
 \begin{array}{l}
 \text{LR} = \begin{bmatrix} -0.000000 \\ -0.000000 \\ -0.000000 \\ 0.014426 \\ 0.014468 \\ -0.078722 \\ -0.040368 \\ 0.417849 \\ 0.758908 \\ 0.417849 \\ -0.040368 \\ -0.078722 \\ 0.014468 \\ 0.014426 \\ -0.000000 \\ -0.000000 \\ -0.000000 \\ -0.000000 \end{bmatrix}
 \end{array}
 \begin{array}{l}
 \text{HR} = \begin{bmatrix} 0.000000 \\ -0.001909 \\ -0.001914 \\ 0.016991 \\ 0.011935 \\ -0.049733 \\ -0.077263 \\ 0.094059 \\ 0.420796 \\ -0.825923 \\ 0.420796 \\ 0.094059 \\ -0.077263 \\ -0.049733 \\ 0.011935 \\ 0.016991 \\ -0.001914 \\ -0.001909 \end{bmatrix}
 \end{array}$$

Os próximos filtros referem-se às wavelets não-separáveis. São mostrados apenas os filtros para decomposição. Os filtros para reconstrução são os mesmos, o que muda são os sinais conforme o esquema de *lifting*. Os filtros são mostrados aos pares. O primeiro par refere-se às decomposições vermelho-pretas e o segundo par refere-se às decomposições azul-amarelas, conforme o algoritmo descrito no capítulo 2.

## A.5 Neville

### Neville 2

$$\text{LD}_{vp} = \begin{bmatrix} 0 & 0.2500 & 0 \\ 0.2500 & 0 & 0.2500 \\ 0 & 0.2500 & 0 \end{bmatrix}
 \quad
 \text{HD}_{vp} = \begin{bmatrix} 0 & 0.1250 & 0 \\ 0.1250 & 0 & 0.1250 \\ 0 & 0.1250 & 0 \end{bmatrix}$$

$$\text{LD}_{aa} = \begin{bmatrix} 0.2500 & 0 & 0.2500 \\ 0 & 0 & 0 \\ 0.2500 & 0 & 0.2500 \end{bmatrix}
 \quad
 \text{HD}_{aa} = \begin{bmatrix} 0.1250 & 0 & 0.1250 \\ 0 & 0 & 0 \\ 0.1250 & 0 & 0.1250 \end{bmatrix}$$







## A.6 MaxMax e MaxMin

O esquema de aplicação dos filtros MaxMax e MaxMin segue a transformação vermelha-preta realizada para os filtros de Neville, no entanto, os filtros são diferentes.

Quando o esquema aplica a decomposição em um nível ímpar usando MaxMin, o filtro de predição é o valor máximo dentro da grade do filtro. Quando o esquema aplica a decomposição em um nível par, o filtro de predição é o valor mínimo dentro da grade. Da mesma forma são definidos os filtros de atualização.

A aplicação para MaxMax é semelhante, porém, os filtros são definidos sempre com o valor máximo.

## ÍNDICE REMISSIVO

- análise de multirresolução, 11
- análise de texturas, 26
- banco de filtros, 14
- base, 7, 37
- característica, 1, 27
  - de textura, 1, 28
  - desvio médio, 28
  - energia, 28
  - entropia, 28
  - momentos centrais, 29
  - momentos de Hu, 30
  - normas, 28
  - resíduo médio, 29
  - suavidade, 29
- cores, 31
- decimação, 14
- espaço de características, 27
- expansão wavelet, 7
- função de escala, 12
- função wavelet, 13
- interpolação, 14, 15
- k-means, 38
- lifting, 18, 24
  - atualização, 18, 19
  - divisão, 18
  - predição, 18
- medida de heterogeneidade, 46, 47
- modelo de cores, 31
- normalização, 32
- ordenação de variáveis, 33, 39
  - não-supervisionada, 34
  - supervisionada, 34
- quincunx, 23, 24
- reticulados, 23
- segmentação, 1, 5, 37
  - não-supervisionada, 5
  - supervisionada, 5
- sinal, 7
- sub-banda, 20, 23
- textura, 26
- transformada wavelet, 1, 6
  - bidimensional, 19–21
  - discreta, 7
  - inversa, 7
  - não-separável, 22
  - separável, 20
- vetor de características, 27
- wavelet, 6, 37
  - biortogonal, 11
  - Coiflets, 11
  - Daubechies, 11
  - famílias, 10
  - mãe, 8
  - momentos de desvanecimento, 9
  - ortogonal, 10
  - regularidade, 10
  - simetria, 10
  - Symlets, 11

## PUBLICAÇÕES CIENTÍFICAS DO AUTOR

### Capítulo de Livro

- Rodrigo Minetto, Ricardo Dutra da Silva, William Robson Schwartz, Hélio Pedrini. *Abordagem Espectral para Segmentação de Imagens Baseada em Transformadas Wavelet e de Fourier*. Série em Ciências Geodésicas, Volume V, pp. 204-218, 2005. (ISBN: 85-88783-88-8)

### Artigos em Revistas

- Ricardo Dutra da Silva, Hélio Pedrini. *Estudo Comparativo de Transformadas Wavelets para Segmentação de Imagens*. Revista Eletrônica de Iniciação Científica (REIC), Volume 6, Número 2, pp. 18-29, Junho de 2006.
- (Submetido) Ricardo Dutra da Silva, Rodrigo Minetto, Hélio Pedrini. *Segmentação de Imagens Médicas Utilizando Transformadas Wavelets*. Revista TEMA (Tendências em Matemática Aplicada e Computacional).

### Artigos Completos em Eventos

- (Submetido) Ricardo Dutra da Silva, Hélio Pedrini. *Segmentation of Color Textured Images Using Wavelet Transforms*. IEEE International Conference on Image Processing (ICIP'2008). São Diego - CA, Estados Unidos, pp. 1-4, 12 e 15 de Outubro de 2008.
- Ricardo Dutra da Silva, Hélio Pedrini. *Compressão de Imagens Médicas Sem Perdas por Transformadas Wavelets*. Anais do VII Workshop de Informática Médica (WIM) do SBQS, Porto de Galinhas - PE, Brasil, pp. 176-185, 25 e 26 de Junho de 2007.
- William Robson Schwartz, Ricardo Dutra da Silva, Rodrigo Minetto, Hélio Pedrini. *An Improved K-means Clustering Algorithm for Image Segmentation*. Anais do II Workshop

de Visão Computacional (WVC), São Carlos - SP, Brasil, pp. 1-6, 16 a 18 de Outubro de 2006.

- Ricardo Dutra da Silva, Rodrigo Minetto, Hélio Pedrini. *Segmentação de Imagens Médicas Utilizando Transformadas Wavelets*. Anais do III Workshop de Iniciação Científica em Computação Gráfica e Processamento de Imagens (WICCGPI) do SIBGRAPI, Natal - RN, Brasil, pp. 1-8, 9 a 12 de Outubro de 2005.
- Rodrigo Minetto, Ricardo Dutra da Silva, William Robson Schwartz, Hélio Pedrini. *Segmentação de Imagens Utilizando Abordagem Espectral por Transformadas Wavelet e de Fourier*. Anais do IV Colóquio Brasileiro de Ciências Geodésicas (CBCG'2005), Curitiba - PR, Brasil, pp. 1-8, 16 a 20 de Maio de 2005.

## Resumos em Eventos

- Ricardo Dutra da Silva, Rodrigo Minetto, Hélio Pedrini. *Segmentação de Imagens Digitais Utilizando Transformadas Wavelets*. Anais do XIII Evento de Iniciação Científica (EVINCI'2005), Curitiba - PR, Brasil, pp. 232, 5 a 7 de Outubro de 2005.
- Ricardo Dutra da Silva, Rodrigo Minetto, Hélio Pedrini. *Atenuação de Ruídos em Imagens Digitais Utilizando Transformadas Wavelets*. Anais do XIII Evento de Iniciação Científica (EVINCI'2005), Curitiba - PR, Brasil, pp. 232, 5 a 7 de Outubro de 2005.
- Vinícius Máximo da Silva, A.F.B. Godói, Dilmari Seidel, Ricardo Dutra da Silva, Rodrigo Minetto, Cristina Duarte Murta. *Utilizando Teoria de Filas para Modelar e Planejar a Capacidade do Restaurante Universitário da UFPR*. Anais do XIII Evento de Iniciação Científica (EVINCI'2005), Curitiba - PR, Brasil, pp. 234, 5 a 7 de Outubro de 2005.
- Ricardo Dutra da Silva, Rodrigo Minetto, Hélio Pedrini. *Compressão Adaptativa de Imagens Utilizando Transformadas Wavelets*. Anais do XII Evento de Iniciação Científica (EVINCI'2004), Curitiba - PR, Brasil, pp. 298, 13 a 15 de Outubro de 2004.
- D.N. Freitas, V.C. Brand, J.G.G. Borges, Ricardo Dutra da Silva, Rodrigo Minetto, R.M. Maeda, Hélio Pedrini. *Simulação de Futebol de Robôs Móveis Autônomos*. Anais do

XII Evento de Iniciação Científica (EVINCI'2004), Curitiba - PR, Brasil, pp. 299, 13 a 15 de Outubro de 2004.

- V.C. Brand, Rodrigo Minetto, Ricardo Dutra da Silva, D. Martineschen, V. Palodeto, Hélio Pedrini. *Simulador para Futebol de Robôs*. Anais do XI Evento de Iniciação Científica (EVINCI'2003), Curitiba - PR, Brasil, pp. 303, 10 a 12 de Novembro de 2003.