Universidade Federal de Pernambuco
Centro de Informática

Pós-graduação em Ciência da Computação

Visão Computacional e Segmentação de Imagens por Discriminação de Textura

Daniel de Filgueiras Gomes

Dissertação de Mestrado

Recife
8 de Setembro de 2009

Universidade Federal de Pernambuco
Centro de Informática

Daniel de Filgueiras Gomes

Visão Computacional e Segmentação de Imagens por Discriminação de Textura

Trabalho apresentado ao Programa de Pós-graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco como requisito parcial para obtenção do grau de Mestre em Inteligência Computacional.

Orientador: *Prof. Dr. Aluizio Fausto Ribeiro Araújo*

Recife
8 de Setembro de 2009

Gomes, Daniel de Filgueiras

Visão computacional e segmentação de imagens por discriminação de textura / Daniel de Filgueiras Gomes. - Recife: O Autor, 2009.

xv, 120 folhas : il., fig., tab.

Dissertação (mestrado) – Universidade Federal de Pernambuco. CIn. Ciência da Computação, 2009.

Inclui bibliografia e apêndices.

1. Inteligência computacional. 2. Inteligência artificial. 3. Processamento de imagem. 4. Segmentação de imagens. I. Título.

006.8 CDD (22. ed.) MEI2010 – 070

Dissertação de Mestrado apresentada por **Daniel de Filgueiras Gomes** Pós-Graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco, sob o título **"Visão Computacional e Segmentação de Imagens por Discriminação de Textura"**, orientado pelo **Prof. Aluízio Fausto Ribeiro Araújo** e aprovada pela Banca Examinadora formada pelos professores:

Prof. George Darmiton da Cunha Cavalcanti
Centro de Informática / UFPE

Prof. Herman Martins Gomes
Departamento de Sistemas Computacionais /UFCG

Prof. Aluízio Fausto Ribeiro Araújo
Centro de Informática / UFPE

Visto e permitida a impressão.
Recife, 08 de setembro de 2009.

Prof. Nelson Souto Rosa
Vice-Coordenador da Pós-Graduação em Ciência da Computação do

Para Mary, Regina, Marcelo e Raquel.

Agradecimentos

Um especial agradecimento a Linus Torvald e Richard Stallman's, sem os quais nada disso seria possível...

Today's scientists have substituted mathematics for experiments, and they wander off through equation after equation, and eventually build a structure which has no relation to reality.

—NIKOLA TESLA ("Radio Power Will Revolutionize the World". Modern Mechanics and Inventions, July, 1934.)

Resumo

A segmentação é uma parte fundamental do processo de particionamento do espaço de dados em regiões salientes e é um pré-requisito para processos subsequentes como reconhecimento de objetos e interpretação de cena. A etapa de segmentação de imagens e a eficiência com que é realizada afeta diretamente a performance de um sistema automático de análise de imagem. A grande maioria dos algoritmos de segmentação existentes trabalha com características como cor e luminosidade sem levar em consideração os micro-padrões de textura formados pela combinação destas características. Inicialmente motivada como ferramenta de segmentação de paisagens em fotos aéreas e de satélite a análise de textura tem sido estudada por um longo período de tempo usando muitas abordagens distintas. Vários métodos realizam a análise de textura sobre estatísticas de segunda ordem de pixels ou componentes espectrais presentes na imagem. Estudos sobre padrões de textura presentes em imagens tem revelado que a informação sobre a textura de um determinado objeto pode ser tão específica a ponto de poder ser utilizada, não só para a discriminação de regiões, mas também para a identificação de objetos em uma cena. O presente trabalho discute as diversas questões e problemas envolvendo o processamento e a discriminação de texturas em imagens digitais e propõe métodos de segmentação utilizando uma abordagem não-supervisionada com redes neurais artificiais.

Palavras-chave: Visão Computacional, Inteligência Artificial, Processamento de Imagens, Segmentação de Textura, Discriminação de Textura.

Abstract

Segmentation is a fundamental part of the process of partitioning the image space into salient regions and a prerequisite for subsequent processes such as object recognition and scene interpretation. The stage of segmentation of images and the efficiency with which it is held directly affects the performance of an automatic system for image analysis. Most existing segmentation algorithms work with features like color and brightness without considering the patterns of micro-texture formed by the combination of these characteristics. Initially motivated as tool for landscapes in aerial and satellite photos, the analysis of texture has been studied for a long period of time using many different approaches. Several methods perform texture analysis based on the second-order statistics of pixels or spectral components in the image. Studies on texture patterns present in images has shown that information about the texture of a particular object can be so specific to the point it can be used not only for discrimination of regions, but also for the identification of objects in a scene. This work discusses the various issues and problems involving the processing and discrimination of textures in digital images, and proposes methods for segmentation using an approach with non-supervised artificial neural networks.

Keywords: Computer Vision, Artifitial Intelligence, Image Processing, Texture Segmentation, Texture Discrimination.

Sumário

1 Introdução **1**
1.1 Objetivos 2
1.2 Modelo de Imagem 2
Modelo de Imagem em Domínio Contínuo 3
Modelo de Imagem de Domínio Discreto 3
Definição de Textura 4
1.3 Etapas de um Sistema de Processamento de Imagens 4
1.3.1 Aquisição de Imagens 4
1.3.2 Pré-Processamento 5
1.3.3 Segmentação 6
1.3.4 Representação e Descrição 6
1.3.5 Reconhecimento e Interpretação 7

2 Fundamentação Teórica **8**
2.1 Filtros 8
2.1.1 *Wavelets* 11
2.2 Matriz de Co-ocorrência de Níveis de Cinza 15
2.3 Teoria dos Textons 19
2.4 Redes Neurais 20
2.4.1 Abordagem Não Supervisionada e Mapas Auto-Organizáveis 21
2.4.1.1 Mapas de Kohonen 22
2.4.1.2 Growing Neural Gas 24
2.4.1.3 Growing When Required 27
2.4.1.4 LARFSOM 28
2.5 Discussão Final 30

3 Descrição do Problema **31**
3.1 Métodos Clássicos de Segmentação de Imagens 32
3.1.1 Segmentação por Limiarização de Histograma 32
3.1.2 Watershed 34
3.1.3 Crescimento de Regiões 34
3.2 Informações Relevantes para Segmentação 35
3.3 Formalização do Problema 38
3.4 Métodos de Avaliação 39
3.4.1 Erro de Classificação 40

3.4.2 Não Uniformidade de Região 40
41
3.4.3 Detecções Corretas 42

4 Abordagem Proposta 43
4.1 Requisitos e Restrições da Solução e do Problema Proposto 47
4.1.1 Tipos de Janelas Utilizadas no Modelo e seu Funcionamento 50
4.1.2 Descritores Implementados 53
4.1.3 Mapas Auto-Organizáveis 56
4.2 Revisitando o Algoritmo do LARFSOM 57
4.3 Apresentação Formal da Solução 63
4.4 Resultados Esperados 65

5 Experimentos Realizados 66
5.1 Estudo Comparativo de Modelos de Mapas Auto-organizáveis 67
5.2 Objetivos dos Experimentos de Segmentação 70
5.3 Procedimentos Experimentais 72
5.4 Parâmetros dos Mapas 75
5.5 Parâmetros do Descritor de Haralick 76
5.6 Primeira Fase de Testes 77
5.6.1 Análise Visual dos Experimentos 78
5.6.2 Análise Quantitativa dos Experimentos 82
5.7 Segunda Fase de Testes 90
5.7.1 Análise Quantitativa dos Experimentos 91
5.8 Discussão Final 99

6 Conclusões e Trabalhos Futuros 100
6.1 Trabalhos Futuros 102

A Testes de Segmentação de Textura em Imagens Naturais 104

B Padrões do Brodatz Utilizados nos Mosaicos de Teste 112

Referências Bibliográficas 116

Lista de Figuras

1.1 Comparativo ilustrativo de filtragem. 6

2.1 Ilustração da operação de convolução 11
2.2 Comparativo ilustrativo de compressão de imagens com *wavelets* e transformada de cosseno discreto. 15
2.3 Ilustração da utilização da matriz de co-ocorrência 17
2.4 Ilustração dos padrões utilizados por Julesz 20
2.5 Exemplo de uma rede SOM 23

3.1 Exemplo de uma limiarização simples 33

4.1 Diagrama em bloco da Abordagem 1 proposta para segmentação por textura. 47
4.2 Diagrama em bloco do Abordagem 2 proposta para reconhecimento de textura. 48
4.3 Movimentação da janela sobre a imagem teste 52
4.4 Tipos de janelas de segmentação utilizadas 52
4.5 Janela para detecção de bordas 53

5.1 LARFSOM "cobrindo" um nuvem de pontos. 69
5.2 Gráficos comparativos de desempenho entre SOM's 69
5.3 Equivalência entre segmentação por borda e por área. 74
5.4 Ilustrações das configurações utilizadas no descritor de Haralick 77
5.5 Imagem do mosaicos de teste. 78
5.6 Comparativo de segmentação 80
5.7 Plot do espaço do descritor MV 81
5.8 Teste de segmentação. 84
5.9 Configuração dos mosaicos utilizados na segunda fase de testes. 90
5.10 Exemplos de mosaicos utilizados na segunda fase de testes. 91
5.11 Comparativo de desempenho entre os mapas. 94
5.12 Comparativo de tempo de treinamento. 95
5.13 Comparativo entre o tempo de treinamento e o tempo total de processamento. 96
5.14 Comparativo de desempenho entre os mapas. 97
5.15 Comparativo das segmentações corretas em função dos descritores utilizados. 98

A.1 Teste de segmentação de borda com o descritor MVSK e o Mapa GNG com $g = 5$ 105
A.2 Teste de segmentação de borda com o descritor MVSK e o Mapa GNG com $g = 10$ 106
A.3 Teste de segmentação de área utilizando o descritor MV e MVSK e o mapa GNG 107

A.4 Teste de segmentação de borda com o descritor MVSK e o mapa GNG 108
A.5 Teste de segmentação de textura por área utilizando os descritores MV e MVSK e o mapa GNG 109
A.6 Teste de segmentação de borda com o descritor histograma 110
A.7 Teste de segmentação de área utilizando os descritores MV e MVSK e o mapa GNG 111

Lista de Tabelas

5.1 Parâmetros do GNG utilizado no modelo de segmentação. 75
5.2 Parâmetros do LARFSOM utilizado no modelo de segmentação. 75
5.3 Parâmetros do LARF2 utilizado no modelo de segmentação. 76
5.4 Parâmetros do LARF22 utilizado no modelo de segmentação. 76
5.5 Combinações de descritores e mapas 82
5.6 Desempenho de segmentação com GNG e com descritor MV 85
5.7 Desempenho de segmentação com LARFSOM e com o descritor MV 86
5.8 Desempenho de segmentação com GNG e com o descritor MVSK 86
5.9 Desempenho de segmentação com LARFSOM 86
5.10 Desempenho de segmentação com GNG 87
5.11 Desempenho de segmentação com LARFSOM 87
5.12 Desempenho de segmentação com GNG 87
5.13 Desempenho de segmentação com LARFSOM 88
5.14 Desempenho de segmentação com GNG 88
5.15 Desempenho de segmentação com LARFSOM 88
5.16 Combinações de descritores e mapas 92

B.1 Mosaicos com dois padrões 113
B.2 Mosaicos com quatro padrões 114
B.3 Mosaicos com cinco padrões 115

Lista de Algoritmos

1	Kohonen SOM	24
2	GNG-SOM	26
3	GWR-SOM	28
4	LARFSOM	29
5	LARFSOM Generalizado	58
6	LARFSOM Generalizado Variação LARF2	61
7	LARFSOM Generalizado Variação LARF22	62
8	Algoritmo de Segmentação de Imagens por Área	64
9	Algoritmo de Segmentação de Imagens por Borda	65

Lista de Abreviaturas e Siglas

SOM	–	*Self-organizing Map*
GNG	–	*Growing Neural Gas*
GWR	–	*Grow When Required*
IA	–	Inteligência Artificial
JPEG	–	*Joint Photographic Experts Group*
LARFSOM	–	*Local Adaptive Receptive Field Self-organizing Map*
MLP	–	*Multilayer Perceptron*
MSE	–	*Mean Squared Error*
RBF	–	*Radial Basis Function*
RNA	–	Redes Neurais Artificiais
CCD	–	*Charge-Coupled Device*
DSP	–	*Digital Signal Processing*
LIT	–	Linear Invariante no Tempo
BMU	–	*Best-Matching Unit*
GIMP	–	*GNU Image Manipulation Program*
ME	–	*Misclassification Error*
NU	–	*Region Nonuniformity*
CS	–	*Correct Detection*
MV	–	Descritor de Média e Variância
MVSK	–	Descritor de Média, Variância, Skewness e Kurtosis
Ha	–	Descritor de Haralick
GLCM	–	*Grey Level Co-occurrence Matrix*

Notação Utilizada

Notações	
x,y,z	Variáveis escalares utilizadas para representar o posicionamento espacial
$\mathbf{s}$	Vetor posição
t	Tempo
i	Intensidade escalar
f	Usado preferencialmente para representar imagens em geral
$\hat{f}$	Transformada de Fourier da função f
f^*	Conjugado complexo da função f
g	Usado preferencialmente para representar imagens afetadas com ruídos ou interferências
h	Utilizado normalmente para representar funções transferências ou máscaras de filtros
a	Ruído multiplicativo
b	Ruído aditivo
u	Sinal de interferência
$\mathbf{i}$	Vetor intensidade de cor, onde $dim(\mathbf{i}) > 1$
i_r,i_g,i_b	Intensidades Red, Green e Blue do sinal de cor de uma determinada imagem
$i = f(x)$	Imagem monocromática unidimensionais.
$i = f(x,y)$	Imagens monocromáticas bidimensionais
$\mathbf{i} = f(x,y)$	Imagem colorida bidimensional

CAPÍTULO 1

Introdução

A segmentação e a análise de texturas são duas áreas de extrema importância em processamento de imagens e visão computacional. A segmentação subdivide uma imagem em regiões menores ou em objetos que a constituem. A quantidade de subdivisões realizada sobre uma imagem depende da aplicação e deve ser suficiente para separar os objetos de interesse presentes na imagem (Gonzalez & Woods, 2000). A segmentação de uma imagem em várias partes pode ocorrer obedecendo a diversos critérios diferentes. Textura é uma importante característica que pode ser utilizada na segmentação para identificar ou discriminar objetos e regiões de interesse em imagens (Reed & Dubuf, 1993). Através da análise da informação de textura é possível agrupar regiões que apresentam combinações complexas de cor e brilho em áreas comuns. Objetos que não apresentam uma forma bem definida, como árvores, pedras e tipos de terreno podem ser identificados pela análise de sua informação de textura. Segmentação e análise de texturas são pré-requisitos para processos subsequentes como reconhecimento de objetos e interpretação de cena. Em geral, a segmentação autônoma é uma das tarefas mais difíceis em processamento de imagens (Gonzalez & Woods, 2000). A etapa de segmentação de imagens e a eficiência com que é realizada afeta diretamente a performance de um sistema automático de análise de imagem. Inicialmente motivada como ferramenta de segmentação de paisagens em fotos aéreas e de satélite a análise de textura tem sido estudada por um longo período de tempo usando muitas abordagens distintas (Haralick et al., 1973; Haralick, 1979; Zheng et al., 2006). Vários métodos realizam a análise de textura sobre estatísticas de segunda ordem de pixels (Haralick et al., 1973; Palm, 2004) ou componentes espectrais presentes na imagem (Weszka et al., 1976; Kim & Kang, 2007). Estudos sobre padrões de textura presentes em imagens tem

revelado que a informação sobre a textura de um determinado objeto pode ser tão específica a ponto de poder ser utilizada, não só para a discriminação de regiões, mas também para a identificação de objetos em cena (Luo & Crandall, 2006).

1.1 Objetivos

Esta dissertação realiza uma revisão sobre as áreas de segmentação de imagem e análise de texturas, propõe um novo sistema de segmentação de imagem por discriminação de texturas baseado em computação inteligente e mapas auto-organizáveis. O sistema proposto tem inspiração no sistema visual de pré-atenção humano, que é o responsável pela "detecção" de padrões primitivos de textura na visão humana (Julesz, 1981). Os resultados da utilização do sistema proposto são apresentados e avaliados por diferentes critérios.

1.2 Modelo de Imagem

No processamento digital de imagens, define-se, classicamente, imagem como sendo uma representação bidimensional, na forma de matriz, dos níveis de intensidade de um determinado sinal captado de um cenário através de um determinado sensor, como por exemplo, uma câmera fotográfica. O sensor não necessariamente precisa captar "luz visível". Podendo ser um radar, um sonar, etc. Nesta dissertação, são adotados dois modelos matemáticos para representar imagens, um de Domínio contínuo e outro de Domínio discreto. A opção de um ou de outro será apenas de acordo com a conveniência matemática.

Modelo de Imagem em Domínio Contínuo

Para facilitar a descrição matemática e análise de alguns métodos que envolvem operações de cálculo, como integrais ou derivadas, uma imagem será considerada como uma função de duas ou três dimensões de Domínio contínuo. Sendo representada por f(x,y) ou g(x,y), no caso bidimensional, e por f(x,y,z) ou g(x,y,z) para imagens tridimensionais. Como exemplo, a "imagem" de uma circunferência de raio 2 utilizando esta notação fica definida pela Equação 1.1 (Considerando-se os 1's, pontos pretos, e os 0's, pontos brancos). Nota-se que é uma representação matemática e, por isso, só será utilizada em alguns casos, onde uma abordagem direta de domínio discreto se tornaria pouco didática. Apesar de todas as implementações dos algoritmos em computadores serem sempre de natureza discreta.

$$f(x,y) = \begin{cases} 1 \rightarrow x^2 + y^2 = 2 \\ 0 \rightarrow x^2 + y^2 \neq 2 \end{cases} \tag{1.1}$$

Modelo de Imagem de Domínio Discreto

Essa é a abordagem mais utilizada e também a mais próxima da implementação real. Nesta, a imagem é uma matriz bidimensional ou tridimensional, em que as coordenadas x e y são as linhas e colunas da matriz. No caso tridimensional, x e y continuam sendo as linhas e colunas e a coordenada z informa sobre qual é a camada ou plano da matriz a qual os valores de x e y estão associados. A representação fica sendo da forma $f[x,y]$ ou $g[x,y]$, no caso bidimensional, e $f[x,y,z]$ ou $g[x,y,z]$ no caso de imagens tridimensionais. A substituição dos parênteses ( ) pelos colchetes [ ] é para enfatizar o Domínio discreto. Como exemplo, a imagem de um quadrado de lado 3, utilizando esta notação, pode ser visto através da matriz $f[x,y]$, Equação 1.2.

$$f[x,y] = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{1.2}$$

Definição de Textura

Textura é um elemento chave para classificação e segmentação de imagens em muitas áreas de processamento de imagens e de visão computacional. Apesar desta importância, curiosamente, não existe uma definição matemática precisa para textura (Zhu et al., 2005; Gonzalez & Woods, 2000; Julesz, 1981). Textura é um conceito extenso e subjetivo. Normalmente está associado a padrões de luminosidade e de cores perceptíveis ao ser humano. Este conceito será tratado com mais detalhes na Seção 2.3.

1.3 Etapas de um Sistema de Processamento de Imagens

1.3.1 Aquisição de Imagens

A aquisição de imagens é todo o processo de obtenção dos dados da imagem que se deseja processar. Partindo desde o dispositivo físico, propriamente, até a digitalização das informações. Sendo assim, o processo de aquisição possui dois elementos básicos: O dispositivo físico de leitura sensível à informação que deseja-se registrar e o sistema de digitalização de informação. Os dispositivos físicos de leitura mais usuais são as câmeras que trabalham no espectro eletromagnético visível ou no infravermelho, onde a leitura do valor do pixel é proporcional a um nível de energia percebida. Entretanto, nem sempre imagens digitais são formadas a partir

de câmeras, equipamentos modernos como sonar e radar podem também serem utilizados para produzir imagens digitais. Especificamente nestes dois casos, a leitura do valor de pixel contém, normalmente, uma informação métrica de distância. Ou seja, o valor do pixel corresponde a uma distância e não a uma intensidade de energia. O segundo dispositivo necessário à aquisição de imagens, o digitalizador, pode parecer um pouco "anacrônico", pois a grande maioria dos equipamentos disponíveis atualmente já possuem saídas digitais. Entretanto, a grande maioria dos equipamentos digitais modernos tem, internamente, um sensor analógico, como um CCD, acoplado a um digitalizador, normalmente embutido em um DSP. E a maneira como a digitalização é realizada pode influenciar na qualidade final de uma imagem digital.

1.3.2 Pré-Processamento

O pré-processamento, de maneira geral, tem como função melhorar e corrigir algumas características chaves da imagem bruta para aumentar a eficiência dos processos subsequentes. Uma das características básicas do pré-processamento e de suas rotinas é a atuação na imagem por completo sem fazer nenhum tipo de distinção morfológica ou por área. As operações básicas utilizadas em pré-processamento de imagens(Gonzalez & Woods, 2000) são as seguintes:

- Correção de histograma;
- Correções geométricas;
- Filtragem.

(a) Original (b) Imagem com ruído (c) Imagem após a utilização de um filtro de Mediana 3x3

Figura 1.1 Exemplo ilustrativo da remoção de ruído em uma imagem digital utilizando um filtro de mediana com máscara de 3x3.

1.3.3 Segmentação

A segmentação é a etapa do processamento de imagens em que uma imagem de entrada é dividida em regiões salientes de acordo com uma determinada métrica. É uma parte fundamental do processo de particionamento do espaço de dados e é um pré-requisito para processos subsequentes mais complexos como reconhecimento de objetos e interpretação de cena. A etapa de segmentação de imagens e a eficiência com que é realizada afeta diretamente a performance de um sistema automático de análise de imagem. Neste processo, uma imagem pode ser dividida em regiões homogêneas em relação a uma dada característica de textura ou cor para que, posteriormente, as diversas regiões possam ser analisadas de forma individual ou combinadas em macrorregiões.

1.3.4 Representação e Descrição

A escolha da representação dos dados fornecidos pela etapa de segmentação é um passo importante para os processos subsequentes e depende da finalidade do sistema. Em sistemas que utilizam, por exemplo, limiarização, para segmentação, é comum que a imagem segmentada tenha a representação e o armazenamento alterada para uma forma binária independentemente da imagem adquirida na etapa de aquisição ser colorida ou em tons de cinza. Já em sistemas

de reconhecimento de objetos pelo seu perfil é usual converter a imagem segmentada em uma lista de pontos.

1.3.5 Reconhecimento e Interpretação

Reconhecimento é o processo que atribui um rótulo a um objeto, baseado na informação fornecida pelo descritor. A interpretação envolve a atribuição de significado a um conjunto de objetos reconhecidos. Estas duas últimas etapas, frequentemente, exigem um grau elevado de processamento, em comparação com as etapas anteriores. Abordagens baseadas em inteligência artificial e reconhecimento de padrões são muito utilizados para desempenhar estes papéis.

CAPÍTULO 2

Fundamentação Teórica

Este capítulo é composto por uma breve introdução e revisão de ferramentas matemáticas, conceitos e algoritmos importantes para a compreensão e para o desenvolvimento subsequente da discussão sobre segmentação de imagens por discriminação de textura.

Na Seção 2.1 é realizada uma revisão sobre filtros lineares, convolução e transformada de Fourier. Na Subseção 2.1.1, complementando a parte de filtros, é realizada uma breve introdução sobre *wavelets*. Este trabalho utiliza apenas alguns conceitos relacionados com *wavelets* e não as utiliza diretamente. A importância maior das *wavelets* está nas perspectivas futuras deste trabalho. Na Seção 2.2 é apresentado o conceito de matriz de co-ocorrência e como este é utilizado para detectar padrões em imagens digitais. A Seção 2.3 mostra o conceito de *textons* e o seu estudo. A Seção 2.4 faz uma introdução sobre redes neurais e apresenta alguns modelos de redes neurais relevantes para o trabalho.

2.1 Filtros

Filtros, de modo geral, tem por objetivo "remover" características indesejáveis ou "limpar" ruídos em uma imagem, facilitando o trabalho de etapas posteriores do processamento da imagem (Gonzalez & Woods, 2000). De um modo mais formal, em processamento de imagens, filtros são transformações aplicadas sobre imagens que alteram os valores dos níveis de intensidade dos pixels, sem alterar as suas respectivas posições (Gonzalez & Woods, 2000). Os filtros podem atuar no domínio da frequência ou no espaço. O que determina um domínio ou outro é a conveniência matemática, ou seja, o domínio no qual a operação do filtro possa ser mais

facilmente resolvida. A grande maioria dos filtros usados no pré-processamento de imagens são transformações lineares invariantes no tempo (LIT) e tem a forma matemática, no domínio espacial, expressa através de uma integral de convolução, Equação 2.1.

$$g(x,y) = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f(\alpha,\beta)h(x-\alpha,y-\beta)d\alpha d\beta \tag{2.1}$$

onde $f(x,y)$ é a imagem original, $h(x,y)$ é o núcleo do filtro ou "janela" e $g(x,y)$ é a imagem filtrada. A função bidimensional $h(x,y)$ varia de acordo com o tipo de operação de filtragem. Uma notação mais compacta da integral de convolução pode ser expressa como na Equação 2.2.

$$g(x,y) = h(x,y) * f(x,y) \tag{2.2}$$

No domínio da frequência a forma básica de um filtro linear é expressa pela multiplicação da transformada de Fourier da imagem original ($F(u,v)$) por uma função transferência $H(u,v)$, ilustrado pela Equação 2.3.

$$g(x,y) = \mathscr{F}^{-1}\{H(u,v) \cdot F(u,v)\} \tag{2.3}$$

onde, $\mathscr{F}\{\cdot\}$ e $\mathscr{F}^{-1}\{\cdot\}$ são as transformadas direta e inversa de Fourier, respectivamente. A função de transferência $H(u,v)$ é exatamente a função dual, no espaço de Fourier, do núcleo de um filtro convolutivo, $h(x,y)$. A prova é simples e basta utilizar a propriedade da dualidade da convolução no tempo e na frequência de teoria de sistemas lineares. Deste modo, sabendo que a multiplicação de duas transformadas de funções no domínio da frequência é equivalente a convolver estas duas funções no domínio do espaço e vice-versa (Lathi, 2007). Assim, pode-se reescrever a Equação 2.2 da seguinte forma:

$$
\begin{aligned}
g(x,y) &= h(x,y) * f(x,y) && (2.4)\\
&= \mathscr{F}^{-1}\{\mathscr{F}\{h(u,v)\} \cdot \mathscr{F}\{f(u,v)\}\} && (2.5)\\
&= \mathscr{F}^{-1}\{\mathscr{F}\{h(u,v)\} \cdot F(u,v)\} && (2.6)
\end{aligned}
$$

Comparando o lado direito da Equação 2.2 com a Equação 2.6 nota-se que a função transferência de um filtro no domínio da frequência é realmente a transformada de Fourier do núcleo do filtro da Equação 2.2, ou seja:

$$h(x,y) = \mathscr{F}^{-1}\{H(u,v)\} \tag{2.7}$$

Deste modo, realmente, utilizar filtros no domínio espacial ou na frequência é fundamentalmente uma questão de conveniência. As Equações de 2.1 à 2.7 dão uma ideia do que é, matematicamente, um filtro. A implementação pratica é um pouco diferente. Como computadores não realizam cálculos no domínio contínuo, nem, no intervalo de mais infinito a menos infinito, todas as equações são discretizadas e as integrais dão lugar aos somatórios. A Equação 2.1 em sua forma discreta se torna:

$$g[x,y] = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f[x-m, y-n] \cdot h[m,n] \tag{2.8}$$

A substituição dos parêntesis pelos colchetes nas funções $g[]$,$f[]$ e $h[]$ é simplesmente para denotar que as variáveis x e y, agora, são discretas (1, 2, 3, etc), embora o valor das funções ($g[]$,$f[]$ e $h[]$) possa ser analógico ou digital. As variáveis M e N representam os valores máximos do núcleo do filtro $h[m,n]$. A Equação 2.8 é bem simples e consiste em realizar o somatório de $h[m,n]$, nos intervalos de 0 a $M-1$ e 0 a $N-1$, com $f[x,y]$. Para, exemplificar, em um caso mais simples, unidimensional, a convolução discreta assume a forma da Equação 2.9.

$$g[x] = \sum_{m=0}^{M-1} f[x-m] \cdot h[m] \tag{2.9}$$

A convolução da matriz $\mathbf{v_1} = [111]$ com a matriz $\mathbf{v_2} = [000111000]$ segue o processo ilustrado na Figura 2.1.

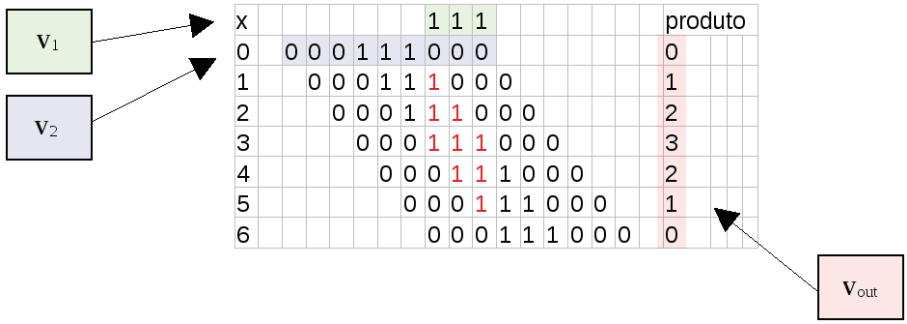


Figura 2.1 Convolução, unidimensional, entre a matriz $\mathbf{v_1} = [111]$ e a matriz $\mathbf{v_2} = [000111000]$. A matriz resultante é $\mathbf{v_{out}} = [0123210]$

O caso bidimensional é exatamente análogo ao unidimensional, só que as matrizes são de MxN e não $1xN$. Assim, normalmente uma das matrizes é a própria imagem que se deseja filtrar ($f[x,y]$) e a outra matriz ($h[x,y]$), normalmente menor, é denominada "máscara" ou "janela" do filtro. A "janela" é na realidade o núcleo da transformada do filtro. Para máscaras pequenas, com dimensão de até 25x25, a implementação direta do filtro no espaço é, normalmente, mais rápida. Para máscaras de dimensão maior é mais eficiente realizar todas as operações no domínio da frequência.

2.1.1 *Wavelets*

Apesar do conceito de filtros no domínio espacial e no domínio da frequência serem duais e, como visto anteriormente, a partir da implementação de uma máscara de um filtro espacial ser possível obter uma função transferência que no domínio da frequência realizará exatamente a mesma operação, não foi trabalhado, simultaneamente, em nenhum momento com informações

do domínio espacial e do domínio da frequência. Ou seja, na análise clássica ou se trabalha observando as características de uma imagem no espaço ou se trabalho no domínio da frequência. Entretanto, as vezes, é mais eficiente implementar filtros que trabalhem, simultaneamente, parte no espaço e parte na frequência. Mesmo sabendo que a informação contida nestes dois domínios é a mesma. Para realizar a análises em espaço-frequência uma nova teoria foi desenvolvida baseada em *wavelets*(“ondinhas”)

Wavelets (Mallat, 1999) são funções($\psi(.)$) que apresentam propriedades especiais que possibilitam a construção de representações espectrais unívocas de funções de energia finita (ex: imagens digitais). Escolhendo-se uma determinada família de *wavelets*($\psi(.)_{u,s}$) a transformada wavelet permite decompor, de maneira univoca, uma função $f(.)$ de energia finita em uma combinação de *wavelets*. O processo é muito semelhante ao da transformada de Fourier. Entretanto, ao contrário de Fourier, a representação espectral de uma função com Wavelets preserva parte das informações sobre a evolução temporal de um determinado sinal. Do ponto de vista geométrico, decompor uma função $f(.)$ em uma representação espectral com *wavelets* é representar a função $f(.)$ como um ponto no espaço de Hilbert formado por uma base de autofunções *wavelets* $\psi(.)_{u,s}$ calculadas através da transformada *wavelet*.

Wavelets estão interconectadas com uma gama enorme de áreas e aplicações matemáticas, físicas e de engenharia. As principais aplicações de Wavelets são:

- Compressão de dados;
- Análise espectral;
- Interpolação de dados;
- Classificação de dados;
- Construção de filtros.

Em processamento de imagens há muitos estudos fisiológicos e de visão computacional que motivam o uso de transformada wavelet (Mallat, 1999). Especificamente nas áreas relacionadas com identificação e discriminação de textura, *wavelets* podem ser utilizadas para implementar filtros sensíveis a texturas (Kim & Kang, 2007), para implementação de descritores de imagem para redes neurais (Bashar et al., 2005) e para implementar redes neurais (J. Zhang et al., 1995).

As *wavelets* precisam obedecer a certas propriedades matemáticas, duas destas propriedades que todas as funções ψ *wavelets* obedecem são as seguintes:

$$\int_{-\infty}^{+\infty} \psi(t)dt = 0 \tag{2.10}$$

$$\int_{-\infty}^{+\infty} \mid \psi(t) \mid^2 dt < +\infty \tag{2.11}$$

Qualquer função que obedeça a estas propriedades e tenha $\|\psi\| = 1$ pode ser usada para formar uma base vetorial de autofunções no espaço de Hilbert. Para isto, uma função ψ wavelet precisa ser modificada. A maneira usual de se "construir" uma base com *wavelets* é através de ajustes de escala e de deslocamentos no tempo, como ilustrado pela Equação 2.12.

$$\psi_{u,s}(t) = \frac{1}{\sqrt{s}}\psi(\frac{t-u}{s}) \tag{2.12}$$

A transformação da Equação 2.12 não altera a norma da função, ou seja, $\|\psi_{u,s}\| = 1$. Utilizando esta transformação pode-se definir a transformada Wavelet de $f \in \mathbf{L}^2(R)$ no tempo t e na escala s do seguinte modo:

$$Wf(u,s) = \langle f, \psi_{u,s} \rangle = \int_{-\infty}^{+\infty} f(t)\frac{1}{\sqrt{s}}\psi^*(\frac{t-u}{s})dt \tag{2.13}$$

onde ψ^* denota o conjugado complexo de ψ. A transformada inversa é dada por:

$$f(t) = \frac{1}{C_\psi} \int_0^{+\infty} \int_{-\infty}^{+\infty} Wf(u,s) \frac{1}{\sqrt{s}} \psi(\frac{t-u}{s}) du \frac{ds}{s^2} \tag{2.14}$$

Em que C_ψ é expressa por:

$$C_\psi = \int_0^{+\infty} \frac{\mid \hat{\psi}(w) \mid^2}{w} dw < +\infty \tag{2.15}$$

As Equações 2.13 à 2.15 são utilizadas no caso contínuo e podem utilizar *wavelets* para formar bases ortogonais ou não ortogonais. Entretanto, a utilização de uma base ortogonal discreta trás muitas vantagens matemáticas e computacionais. Do ponto de vista computacional, a implementação de uma base discreta é muito mais fácil e rápida de ser utilizada em computadores digitais. Assim, a grande maioria dos algoritmos implementados para calcular a transformada *wavelet* se restringe apenas a funções *wavelets* discretas e ortogonais. Uma base ortogonal discreta pode ser obtida através de dilatações e translações de uma *wavelet*, conforme a Equação 2.16 :

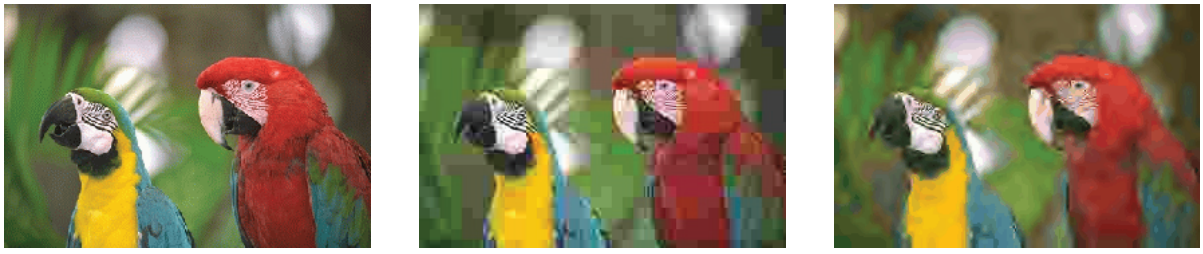
$$\psi_{j,n}(t) = \frac{1}{\sqrt{2^j}} \psi(\frac{t-2^j n}{2^j}) \qquad (j,n) \in \mathbb{Z}^2 \tag{2.16}$$

Note que, neste caso, o deslocamento no tempo (termo $2^j n$) se torna dependente da escala (j). A representação de uma função f através de uma base ortogonal discreta assume a seguinte forma:

$$f = \Sigma_{j=-\infty}^{+\infty} \Sigma_{n=-\infty}^{+\infty} \langle f, \psi_{j,n} \rangle \psi_{j,n} \tag{2.17}$$

Wavalets são frequentemente utilizadas, por exemplo, em análise de textura ou para discriminação de textura porque é possível escolher funções *wavelets* ψ que formem uma base capaz de prover uma representação unívoca e eficiente de muitos tipos de texturas (Kim & Kang, 2007). Assim, utilizando *wavelets*, poderíamos trabalhar com texturas como se fossem veto-

res, inclusive, utilizar todas as propriedades e operações usuais da álgebra vetorial. Wavelets podem ser utilizadas, também, como uma ferramenta de compressão de dados. Ao decompormos qualquer função sobre uma base *wavelets* podemos avaliar a "energia" de cada uma das componentes da base, as componentes com menor energia são menos significantes para uma reconstrução da função e podem ser descartadas. O processo é semelhante a uma análise de sinal no espectro de potência utilizando transformada de Fourier. A Figura 2.1.1 ilustra um comparativo de compactação de imagem utilizando *wavelets* (padrão JPEG2000) e a transformada do cosseno discreto (padrão JPEG).



(a) Original (b) Compressão standard jpeg (c) Compressão com Wavelets

Figura 2.2 As imagens 2.2(a), 2.2(b) e 2.2(c) apresentam um comparativo de resultados entre a compressão de imagem através do método utilizado no padrão JPEG, utilizando a transformada de cosseno discreto, e utilizando *wavelets* (padrão para o JPEG2000). Tanto as imagens 2.2(b) como a 2.2(c) tem um fator de qualidade de 60% e uma taxa de compressão de 98.86%. Note que a compressão com *wavelets* produz uma imagem comprimida mais "suave", sem os artefatos em forma de blocos produzidos na transformada do cosseno discreta. Esta característica torna o processamento da imagem, do ponto de vista da visão computacional, menos problemática.

2.2 Matriz de Co-ocorrência de Níveis de Cinza

Assumindo que uma imagem digital monocromática pode ser interpretada como uma matriz $I_{N\times M}$ de N linhas e M colunas, como descrito na Seção 1.2, e que o valor de cada elemento a_{ij} representa o nível de cinza do pixel e que este nível é quantizado em N_g valores diferentes, pode-se calcular uma estatística de segunda ordem sobre os valores dos níveis de cinza con-

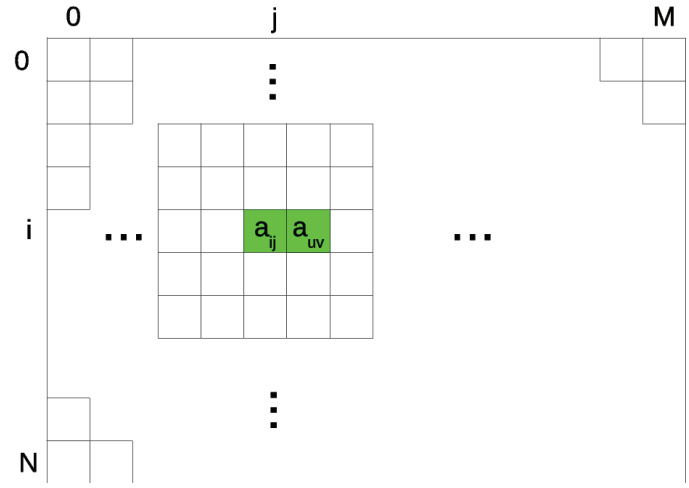
tando o número de ocorrências de pares (a_{ij}, a_{uv}) com níveis de cinza e distância δ de separação predeterminados. Os números de ocorrências dos pares para todos os valores possíveis de nível de cinza $(0..Ng)$ podem ser organizados, assim como uma imagem digital, também em forma matricial. A matriz resultante deste processo é denominada *Gray-Tone Spatial-Dependence Matrix* e foi primeiramente proposta, especificamente para processamento de imagens, por (Haralick et al., 1973) e alguns autores (Weszka et al., 1976) (Zheng et al., 2006) também utilizam o nome *Gray Level Co-Occurrence Matrix* (GLCM). Apesar deste método proposto por Haralick para discriminação de texturas já possuir mais de 35 anos, ele continua sendo largamente utilizado na íntegra ou com pequenas modificações até hoje em muitos sistemas (Zheng et al., 2006). Nas Equações 2.18 e 2.19 e nas Restrições 2.20 têm-se as expressões matemáticas para a matriz de co-ocorrência de níveis de cinza para uma determinada imagem. Onde δ_i e δ_j são os parâmetros que definem a separação δ e a orientação de comparação do par (a_{ij}, a_{uv}) da imagem. Um exemplo ilustrativo pode ser visualizado na Figura 2.3. Em 2.3(a) é representado como são avaliadas as contagens sobre um par com separação $(\delta_i = 0, \delta_j = 1)$, em 2.3(b) um exemplo de "imagem" é apresentado e em 2.3(c) a respectiva matriz de co-ocorrência calculada é mostrada utilizando a mesma separação ilustrada em 2.3.

$$GLCM_{Ng \times N_g}(I_{N \times M}) = \{c_{kl} | c_{kl} = \sum_{i=1}^{N} \sum_{j=1}^{M} f_c(i, j, \delta_i, \delta_j, k, l)\} \tag{2.18}$$

$$f_c(i, j, \delta_i, \delta_j, k, l) = \begin{cases} 1 & \text{Se } a_{ij} = k \text{ e } a_{uv} = l \\ 0 & \text{Caso contrário} \end{cases} \tag{2.19}$$

$$\begin{gathered} u = i + \delta_i \\ \therefore \quad v = j + \delta_j \\ a_{ij}, a_{uv} \in I_{N \times M} \end{gathered} \tag{2.20}$$

Quando o objetivo é utilizar a GLCM para discriminação de texturas e padrões em imagens



(a) Ilustração de como são contados os pares de níveis de cinza

0	1	1	2	3
0	0	2	3	3
0	1	2	2	3
1	2	3	2	2
2	2	3	3	2

(b) Imagem exemplo

	0	1	2	3
0	1	2	1	0
1	0	1	3	0
2	0	0	3	5
3	0	0	2	2

(c) Matriz com as ocorrências dos pares de níveis de cinza

Figura 2.3 Exemplo ilustrativo da utilização da matriz de co-ocorrência definido por Haralick para discriminar padrões de texturas em imagens.

digitais o calculo deste matriz é realizado sobre janela móvel, como a utilizada em um filtro convolutivo. Pra cada posição da janela móvel sobre a imagem original é gerada uma GLCM e as informações provenientes deste processo são utilizadas para realizar a discriminação de texturas. Entretanto, normalmente as matrizes GLCM são altamente redundantes e, por isso, normalmente não são utilizadas diretamente. Haralick propôs a utilização quatro métricas sobre as GLCM, elas são:

- *Contraste*: $CON = \sum (k-l)^2 c_{kl}$. Este é essencialmente o momento de inércia da matriz de co-ocorrência entorno de sua diagonal principal. E é uma medida natural do grau de espalhamento dos elementos da matriz;

- *Segundo Momento Angular*: $ASM = \sum c_{kl}^2$. Esta medida é pequena quando todos os valores da matriz são próximos e grande quando há valores muito grandes e muito pequenos;

- *Entropia*: $Ent = -\sum c_{kl} \log_2 c_{kl}$. Esta medida mede a quantidade de informação presente na matriz de co-ocorrência e, também, não deixa de ser um modo de se avaliar a dispersão da matriz;

- *Correlação*: $Cor = \sum (klc_{kl} - \mu_k \mu_l) \diagup (\sigma_k \sigma_l)$. Onde μ e σ são, respectivamente, a média e o desvio padrão para as linhas k e colunas l. Esta é uma medida que avalia qual é o grau de auto-semelhança das linhas e colunas da matriz de co-ocorrência.

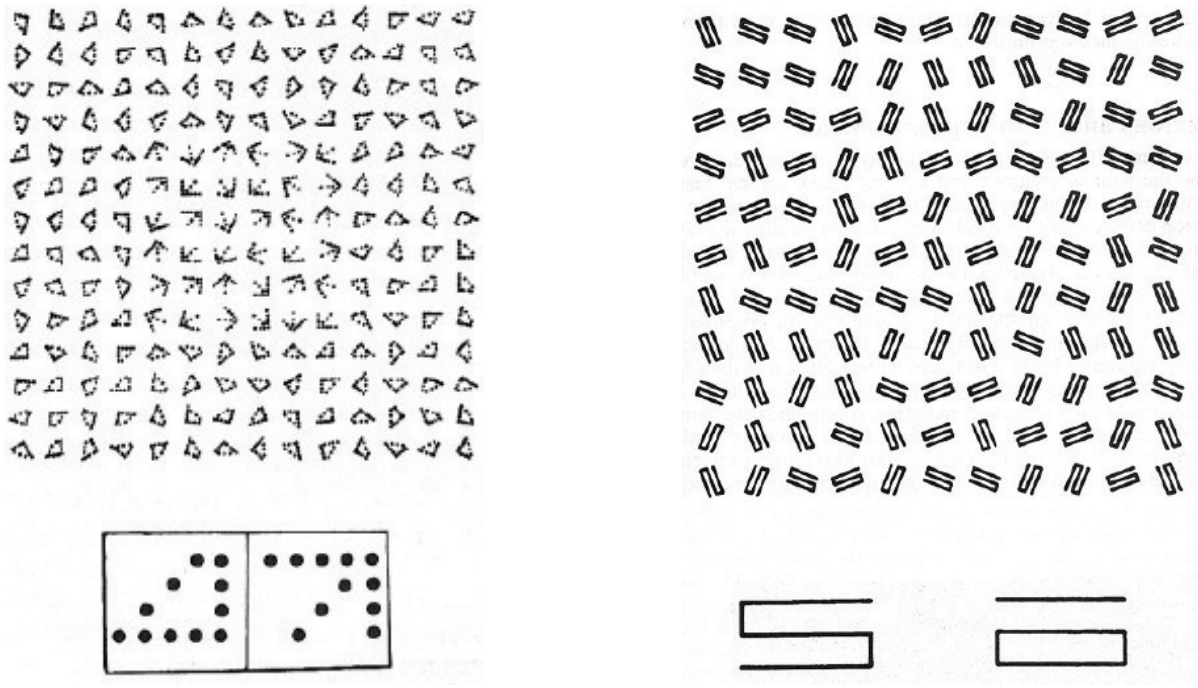
Outras métricas sobre a matriz de co-ocorrência, propostas por outros autores, podem também ser utilizadas (Gonzalez & Woods, 2000), mas este conjunto é o mais utilizado até hoje e é suficiente para muitas aplicações.

2.3 Teoria dos Textons

Análise de textura tem sido estudada por um longo período de tempo usando várias abordagens distintas. Seguindo uma linha com base em psicologia humana, Julesz (1981) publicou um estudo sobre como o sistema visual humano processa informações de textura. Ele mostrou que o ser humano consegue identificar diferenças de texturas em imagens formadas por "padrões primitivos" de modo espontâneo e sem, necessariamente, ter conhecimento do "padrão primitivo". Esta capacidade de discriminação, inerente do homem, de conseguir dividir as imagens em partes menores sem ter a consciência em si do seu conteúdo foi denominada de sistema visual de pré-atenção (*preattentive visual system*) e os "padrões primitivos" detectados por este sistema foram denominados *textons*. Estes seriam as "peças" fundamentais do sistema de segmentação de textura humano.

Julesz realizou alguns testes experimentais com padrões visuais apresentados por alguns segundos para voluntários humanos. Ele observou que quando determinados padrões de imagem eram apresentados, as pessoas conseguiam descrever o conteúdo geral da imagem e discriminar regiões, sem entretanto, conseguir identificar os padrões presentes na imagem. Julesz conjecturou, ainda, que os textons deveriam ter uma forma alongada em uma determinada direção e que o sistema visual humano seria capaz de detectar estatísticas de primeira ordem baseadas nestes componentes através de informações como tamanho, espessura, cor, concentração e orientação e descreveu regras de como poderíamos "criar" padrões de *textons*. Na Figura 2.3, há um exemplo de duas imagens utilizadas nos testes de Julesz, na Figura 2.4(a) há dois padrões diferentes formados por dois tipos de textons diferentes e na Figura 2.4(b) há, também, dois padrões, mas que utilizam o mesmo *texton*. No primeiro caso, é possível, facilmente, discriminar duas regiões na imagem, já na segunda imagem a discriminação "rápida" não é mais possível. Para o sistema visual humano de pré-atenção os padrões são idênticos, neste caso, a discriminação das regiões deve ocorrer em uma etapa posterior de processamento à do sistema

de pré-atenção, exigindo uma análise minuciosa da imagem e um esforço cognitivo maior.



(a) Dois padrões, dois *texton*

(b) Dois padrões, um *texton*

Figura 2.4 Em 2.4(a) é possível realizar uma discriminação rápida entre os dois padrões presentes na imagem, porque os padrões apresentam *textons* diferentes. Em 2.4(b) há dois padrões diferentes, mas com um único *texton*. Uma discriminação rápida não é mais possível. Para discriminar os dois tipos de padrões presentes é necessário uma "análise" mais focada e demorada da imagem.

2.4 Redes Neurais

As arquiteturas de redes neurais artificiais podem, basicamente, serem divididas em três categorias, cada uma baseada em uma filosofia diferente. Na primeira categoria, as *feedforward networks* (Rumelhart et al., 1986), ou redes alimentadas para frente, transformam os conjuntos de sinais de entrada em conjuntos de sinais de saída. A transformação entrada-saída desejada é usualmente determinada por um processo de ajuste externo de parâmetros do sistema. Exemplos muito populares de redes que pertencem a esta categoria são a *Multi Layer Percep-*

tron (MLP) e as Redes de Função de Base Radial (RBFs). Na segunda categoria, as *feedback networks* (Hopfield, 1982), ou redes retroalimentadas, o sinal de entrada define o estado inicial de ativação do sistema de realimentação e, depois, o sistema passa por um processo de sucessivas transições de estado que, assintoticamente, conduzem a um estado final estável que é definido como a saída da rede. Na terceira categoria de modelos neurais, células vizinhas em uma rede neural competem entre si através de interações mútuas laterais e desenvolvem adaptativamente a capacidade de detectar diferentes padrões de sinais sem nenhum tipo de supervisão (Kohonen, 1990). Esta categoria é baseada em aprendizagem competitiva (*competitive learning*).

2.4.1 Abordagem Não Supervisionada e Mapas Auto-Organizáveis

Os Mapas Auto-organizáveis, *Self-Organizing Maps* ou, simplesmente SOMs pertencem a terceira e última categoria de redes neurais. Existem diversas aplicações para este tipo de rede neural, mas as principais aplicações são: visualização de dados com alta dimensionalidade, redução de dimensionalidade de um conjunto de dados e categorização. A operação que um mapa auto-organizável realiza sobre um determinado conjunto de dados é um tipo de projeção não-linear da função de densidade de probabilidade destes dados. Normalmente a projeção é realizada em um espaço de dimensionalidade menor que a dos dados originais, mas isso não é uma condição necessária. Os dados aprendidos pelos SOMs distribuem ao longo dos nodos do mapa numa configuração chamada de distribuição topológica. Esta característica faz com que, de modo geral, nodos próximos apresentem informações similares e, por sua vez, nodos distantes apresentam informações distintas. A estrutura básica dos modelos auto-organizáveis é formada por uma camada de entrada e uma camada de saída. A camada de entrada propaga os valores recebidos a todos os nodos da camada de saída, na qual é criada a representação do conhecimento. O processo de aprendizado dos SOMs além de não-supervisionado é competitivo e cooperativo. Isto é, os nodos da rede na camada de saída competem entre si pelo direito

de serem atualizados. Apenas o nodo vencedor (competição), e em alguns casos seus vizinhos (cooperação), são atualizados. O nodo vencedor é aquele que apresenta maior semelhança, proximidade, ao padrão de entrada em questão e métricas diferentes podem ser usadas para mensurar o grau de proximidade.

2.4.1.1 Mapas de Kohonen

O Kohonen SOM, SOM "clássico" ou, simplesmente, SOM deve-se a Teuvo Kohonen (1990). Este foi um dos primeiro pesquisadores a realizar um extenso trabalho sobre redes neurais artificiais não-supervisionadas nas décadas de 1970 e 1980. O funcionamento básico do SOM define um mapeamento partindo de um conjunto de dados de entrada no espaço $\mathbb{R}^n$ para uma malha, normalmente, bidimensional de nodos. Cada nodo i da malha possui um vetor de pesos $\mathbf{w_i} \in \mathbb{R}^n$. Um vetor de entrada $\xi \in \mathbb{R}^n$ é comparado com todos os $\mathbf{w_i}$'s e o mais próximo do vetor de entrada é definido como sendo o *best-matching unit* (BMU) e o nodo que contém este vetor é considerado como o nodo "vencedor" ou mais apto a representar o vetor de entrada ξ, desta forma é realizada uma associação entre o vetor ξ e o nodo i, ver Figura 2.5. A métrica de distância entre ξ e $\mathbf{w_i}$ pode variar, normalmente é utilizada a distância Euclidiana. A saída da rede, depende essencialmente da aplicação, pode ser o próprio nodo (i) vencedor, o peso ($\mathbf{w_i}$) associado ao nodo vencedor, ambos ou parte da informação contida no peso do nodo vencedor. O algoritmo de treinamento do modelo de Kohonen esta descrito no Algoritmo 1. Os passos principais consistem em sortia uma amostra do conjunto de treinamento (passo 2), encontrar no mapa o nodo mais próximo da amostra sorteada (passo 3) e atualizar o valor do nodo mais próximo e dos seus vizinhos de modo que a distância entre os nodos e a amostra diminua (passo 4). O modo como os nodos são atualizados varia exponencialmente de acordo com o tempo e a distância dos nodos para a amostra (Equações 2.23 a 2.26). Estes passos básicos estão presentes na maioria dos modelos de mapas auto-organizáveis. O modelo de Kohonen foi a base para a implementação de muitas outras variações de mapas auto-organizáveis.

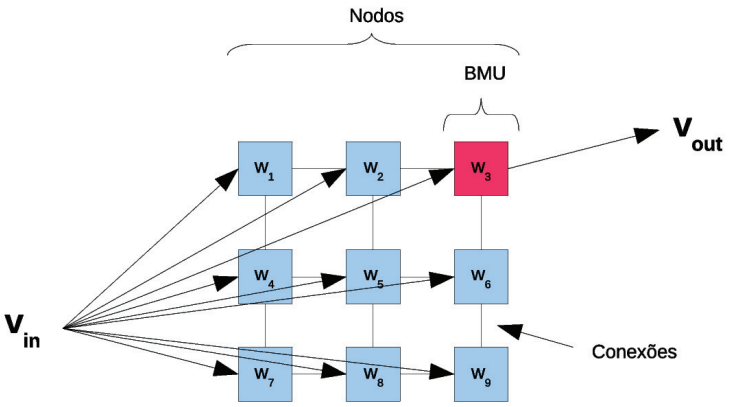


Figura 2.5 Ilustração de uma rede SOM, com um vetor de entrada $\mathbf{v_{in}}$ sendo comparado com os vetores dos nodos $\mathbf{w_n}$ e a escolha do BMU.

O SOM, apesar de ter se mostrado um potente modelo para categorização, apresenta algumas restrições importantes, algumas delas são:

- O número de nodos do mapa precisa ser conhecido previamente, sendo um parâmetro do sistema;
- O padrão de conexões entre os nodos é fixo e precisa ser determinado previamente.

Devido a estas restrições muitas abordagem foram desenvolvidas com o objetivo de implementar mapas auto-organizáveis com topologia e número de nodos dinâmicos, nas subseções seguintes serão apresentados alguns destes modelos de mapas dinâmicos.

Algoritmo 1 Kohonen SOM

1. *Inicialização:* Escolha valores aleatórios iniciais para os vetores de peso $\mathbf{w_j}(0)$. A única restrição é que os valores de inicialização sejam sempre diferentes para os $\mathbf{w_j}(0)$ pesos, onde $j = 1,2,...,N$ e N é o número de neurônios do mapa.
2. *Amostragem:* Apresenta um vetor de dados escolhido aleatoriamente, ξ, como entrada à rede.
3. *Casamento por similaridade:* Encontra o nodo mais próximo do vetor de dados escolhido utilizando o critério da menor distância Euclidiana.

$$i(\xi) = arg\,min_j \|\xi - \mathbf{w_j}\| \therefore j = 1,2,...,N \tag{2.21}$$

onde N é o número total de nodos do mapa e $i(\xi)$ é o número do nodo mais próximo do vetor apresentado.
4. *Atualização:* Atualiza os pesos de cada nodo da rede utilizando a formula:

$$\mathbf{w_j}(n+1) = \mathbf{w_j}(n) + \eta(n) h_{j,i(\xi)}(n)(\xi(n) - \mathbf{w_j}(n)) \tag{2.22}$$

onde,

$$\eta(n) = \eta_0 e^{-\frac{n}{\tau_2}} \tag{2.23}$$

$$h_{j,i(\xi)}(n) = e^{\frac{d_{j,i}^2}{2\sigma^2(n)}} \tag{2.24}$$

$$\sigma(n) = \sigma_0 e^{-\frac{n}{\tau_1}} \tag{2.25}$$

$$d_{j,i}^2 = \|\mathbf{r_j} - \mathbf{r_i}\|^2 \tag{2.26}$$

η é o parâmetro de aprendizagem, $h_{j,i(\xi)}$ é a função de vizinhança entorno do vencedor, σ é a função de decaimento exponencial ao longo do tempo e $d_{j,i}^2$ é a distância topológica entre o nodo vencedor (i) e o nodo que esta sendo atualizado (j).
5. *Continuação:* Continua a partir do passo 2 enquanto são notadas mudanças significativas nas posições dos vetores de pesos dos nodos

2.4.1.2 Growing Neural Gas

O *Growing Neural Gas* ou, simplesmente, GNG é um modelo de mapa auto-organizado incremental desenvolvido por Fritzke (Fritzke, 1995) derivado a partir dos modelos de Martinetz e Schulten. O GNG cresce e adapta as conexões dos nodos de modo incremental para se ajustar a função densidade de probabilidade definida por um determinado conjunto de treinamento. O processo de crescimento ou de inserção de novos nodos é constante e múltiplo do número (λ) de apresentações de exemplos do conjunto de treinamento. A inserção dos novos nodos é sem-

pre realizada na vizinhança do nodo com o maior "erro" acumulado. Pelo fato do desempenho do GNG ter se mostrado significativo para uma diversidade de problemas muito grande, este modelo se tornou praticamente uma referência de desempenho para realizar comparações com outros modelos dinâmicos de mapas. O processo de aprendizagem do GNG esta ilustrado no Algoritmo 2.

Algoritmo 2 GNG-SOM

0. *Inicialização do mapa:* Inicia o mapa com dois nodos com vetores de pesos aleatórios $\mathbf{w_a}$ e $\mathbf{w_b}$ no espaço $\mathbb{R}^n$

1. *Amostragem:* Selecione uma amostra ξ de acordo com uma dada probabilidade $P(\xi)$ do conjunto de dados de treinamento.

2. *Casamento por similaridade:* Encontre o nodo, s_1, com os vetores de peso mais próximo e o segundo mais próximo, s_2, da amostra ξ.

3. *Envelhecimento:* Incremente a *age* de todas as conexões emanando de s_1.

4. *Incremento do contador local:* Incremente o contador local com o quadrado da distância entre a amostra e o nodo mais próximo (s_1).

$$counter(s_1) = counter(s_1) + \Delta erro(s_1) \text{ , onde } \Delta erro(s_1) = \|\mathbf{w_{s_1}} - \xi\|^2 \tag{2.27}$$

5. *Atualização dos pesos:* Movimente o nodo s_1 e os seus vizinhos topológicos em direção da amostra ξ pela fração de ε_b e ε_n, respectivamente, da distância total.

$$\Delta \mathbf{w_{s_1}} = \varepsilon_b(\xi - \mathbf{w_{s_1}}) \tag{2.28}$$

$$\Delta \mathbf{w_n} = \varepsilon_n(\xi - \mathbf{w_n}) \tag{2.29}$$

onde,

$$\varepsilon_b > \varepsilon_n \tag{2.30}$$

6. *Conectando os nodos mais próximos:* Se s_1 e s_2 estão conectados, iguale a zero a *age* da conexão. Se a conexão não existe, crie-a.

7. *Remoção de conexões antigas:* Remova uma dada conexão se tiver uma *age* maior que age_{max}. Se isto resultar em nodos sem nenhuma conexão remova o nodo também.

8. *Inserção de novos nodos:* Se o número de amostragens realizadas for múltiplo de um inteiro λ, insere um novo nodo da seguinte maneira:

- Determine o nodo q com o maior erro acumulado.
- Insira um novo nodo r no ponto médio entre o nodo q e o nodo vizinho f com o maior erro acumulado.

$$\mathbf{w_r} = 0,5(\mathbf{w_q} + \mathbf{w_f}) \tag{2.31}$$

- Insira uma conexão nova ligando r a q e a f e remova a conexão original entre q e f.
- Decremente o a variável de erro de q e f pela multiplicação por um fator constante de α e inicialize o erro de r com o mesmo valor do erro de q, já atualizado.

9. *Decremento do erro:* Decremente todas as variáveis de erro pela multiplicação delas por uma constante β.

10. *Critério de parada:* Finalize o treinamento se o critério de parada foi atingido, se não, continue a partir de 1 (O Algoritmo do GNG não fixa nenhum critério de parada específico, pode ser o tamanho da rede, alguma medida de performance, entre outros).

2.4.1.3 Growing When Required

O *Growing When Required* ou , simplesmente, GWR é um modelo de mapa auto-organizável dinâmico semelhante ao GNG, mas que apresenta um processo de inserção de novos nodos completamente diferente. No GNG os nodos são inseridos segundo um fator λ, múltiplo do número de apresentações de exemplos, isto pode em certos casos produzir mapas com um número insuficiente de nodos ou com mais nodos que o necessário para representar um determinado conjunto de dados de treinamento. A estratégia do GWR é realizar inserções de novos nodos seguindo um critério de "erro" acumulado pelo mapa. Assim, enquanto o mapa não tiver nodos suficientes para prover uma representação do conjunto de treinamento com um "erro" satisfatório, o algoritmo de treinamento continua a inserir novos nodo. A inserção dos novos nodos, como no caso do GNG, é sempre realizada na vizinhança do nodo com o maior "erro" acumulado. O processo de treinamento do GWR esta no Algoritmo 3.

Algoritmo 3 GWR-SOM

0. Inicie o mapa com dois nodos (a e b) com vetores de pesos aleatórios $\mathbf{w_a}$ e $\mathbf{w_b}$ no espaço $\mathbb{R}^n$
1. Selecione uma amostra ξ para entrada na rede.
2. Para cada nodo i da rede calcule a distância para a amostra $\|\xi - \mathbf{w_i}\|$
3. Selecione o nodo mais próximo (BMU) e o segundo mais próximo, respectivamente $s_1, s_2 \in A$ tal que:

$$\mathbf{w_{s_1}} = arg\,min_{i \in A} \|\xi - \mathbf{w_i}\| \tag{2.32}$$

e

$$\mathbf{w_{s_2}} = arg\,min_{i \in A/\{w_{s_1}\}} \|\xi - \mathbf{w_i}\| \tag{2.33}$$

onde , $\mathbf{w_i}$ vetor de peso do nodo i.
4. Se não existe uma conexão entre s_1 e s_2 crie uma conexão, caso contrário ajuste a *age* da conexão para zero.
5. Calcule a atividade do BMU

$$a = e^{-\|\xi - \mathbf{w_{s_1}}\|} \tag{2.34}$$

6. Se a atividade de a for menor que o limiar a_T e o contador de disparo for menor que h_T então acrescente um novo nodo entre os dois nodos mais próximos da amostra s_1 e s_2.

- Adicione um novo nodo com vetor de pesos $\mathbf{w_n} = \frac{\xi + \mathbf{w_{s_1}}}{2}$
- Atualize o número de nodos $N = N + 1$
- Remova a conexão entre s_1 e s_2
- Insira uma conexão entre os nodos n e s_1

7. Caso contrário, atualize o peso do vencedor (Equação 2.35) e dos seus vizinhos (Equação 2.36).

$$\Delta \mathbf{w_{s_1}} = \varepsilon_b h_s (\xi - \mathbf{w_{s_1}}) \tag{2.35}$$

$$\Delta \mathbf{w_i} = \varepsilon_n h_i (\xi - \mathbf{w_i}) \tag{2.36}$$

onde $0 < \varepsilon_n < \varepsilon_b < 1$ e h_s é o contador de disparado nodo s_1.
8. Incremente a *age* de todas as conexões que emanam de s_1.
9. Reduza o contador de acordo com a frequência que o nodo s_1 vencedor tem sido ativado (Equação 2.37).

$$h_s(t) = h_0 - \frac{S(t)}{\alpha_b}(1 - e^{(} - \alpha_b)/\tau_b) \tag{2.37}$$

Proceda de maneira equivalente para os seus vizinhos i (Equação 2.38).

$$h_i(t) = h_0 - \frac{S(t)}{\alpha_n}(1 - e^{(} - \alpha_n)/\tau_n) \tag{2.38}$$

10. Remova todas as conexões com *age* maior que age_max e cheque se existem nodos desconectados, se existir remova-o também.
11. Finalize o treinamento se o critério de parada foi atingido, se não, continua a partir de 1 (O Algoritmo do GWR não fixa nenhum critérios de parada específico, pode ser o tamanho da rede, alguma medida de performance, entre outros).

2.4.1.4 LARFSOM

O LARFSOM é um tipo de de mapa auto-organizável projetado originalmente para trabalhar com imagens coloridas. Ele pode ser utilizado tanto para realizar redução de cores em uma imagem como para realizar segmentação por cor. O algoritmo básico de treinamento do LARFSOM tem 11 passos. Havendo, também, variações que envolvem a parte final do algoritmo

(Passo 11), uma descrição mais detalhada pode ser encontrada na dissertação (Costa, 2007) ou no artigo (Araújo & Costa, 2009). O LARFSOM esta descrito no Algoritmo 4.

Algoritmo 4 LARFSOM

1. Inicialize os parâmetros: taxa de aprendizagem final (ρ_f); modulador de taxa de aprendizagem (ε); limiar ou valor de corte de atividade (a_T); número de vitórias de cada nodo i ($d_i = 0$); número máximo de vitórias de cada nodo (d_m); iteração de tempo ($t = 0$); tempo máximo (t_{max}); erro mínimo ($deslocamento_{min}$); e o número de unidades iniciais ($N = 2$), que devem ser conectadas e cujo os pesos podem ser copiados de dois vetores, escolhido aleatoriamente, do conjunto de dados de entrada.
2. Apresente um pixel escolhido aleatoriamente da imagem original, $\xi = [rgb]^T$, como entrada à rede.
3. Calcule a distância Euclidiana entre o vetor de entrada, ξ, e o vetor de pesos, $\mathbf{w_i}$, de cada um dos nodos como a seguir:

$$d(\xi, \mathbf{w_i}) = \|\xi - \mathbf{w_i}\|^2 \therefore \|\xi - \mathbf{w_i}\|^2 = (r - w_{ir})^2 + (g - w_{ig})^2 + (b - w_{ib})^2 \tag{2.39}$$

A menor distância entre o vetor de entrada e todos os vetores de pesos de cada nodo, determinada a unidade vencedor ou BMU:

$$d(\mathbf{w_{s_1}}, \xi) \leq d(\mathbf{w_{s_2}}, \xi) \leq d(\mathbf{w_i}, \xi), ... \forall i \in NodeSet \tag{2.40}$$

onde , $NodeSet$ é o conjunto dos nodos.
Incremente o contador de vitórias de unidade vencedora $d_{s_1} = d_{s_1} + 1$
4. Insira uma conexão entre s_1 e s_2 se não existir.
5. Calcule o campo receptivo de s_1:

$$r_{s_1} = \sqrt{(w_{s_1 r} - w_{s_2 r})^2 + (w_{s_1 g} - w_{s_2 g})^2 + (w_{s_1 b} - w_{s_2 b})^2} \tag{2.41}$$

6. Calcule a atividade de s_1 por meio de uma função de base radial:

$$a_{s_1} = \frac{e^{-\|\xi - \mathbf{w_{s_1}}\|}}{r_{s_1}} \tag{2.42}$$

7. Insira um novo nodo se a ativação da unidade vencedora estiver abaixo do limiar(a_T), caso contrário atualize os pesos da BMU Se $a_{s1} < a_T$

- Adicione um novo nodo com vetor de pesos $\mathbf{w_n} = \xi$
- Atualize o número de nodos $N = N + 1$
- Remova a conexão entre s_1 e s_2
- Calcule as distâncias $d(\mathbf{w_n}, \mathbf{w_{s_1}})$, $d(\mathbf{w_n}, \mathbf{w_{s_2}})$, $d(\mathbf{w_{s_1}}, \mathbf{w_{s_2}})$
- Insira conexão entre os nodos com as duas menores distâncias

Caso contrário atualize os pesos do vencedor:

$$\Delta \mathbf{w_{s_1}} = \rho(\xi - \mathbf{w_n}) \therefore \rho = \begin{cases} \varepsilon \rho_f^{\left(\frac{d_i}{d_m}\right)}, d_i \leq d_m \\ \varepsilon \rho_f, d_i > d_m \end{cases} \tag{2.43}$$

8. Atualize o número de interações ($t = t + 1$) e retorne ao passo 2, a não ser que o critério de parada da Equação 2.44 ou o número máximo de iterações tenha sido atingido.

$$deslocamento = \frac{1}{N} \sum_{i=0}^{N-1} \|\mathbf{w_i}(t) - \mathbf{w_i}(t+1)\|^2 \leq deslocamento_{min} \tag{2.44}$$

9. Remova os nodos desconectados
10. Após o processo de treinamento, atribua a cor representada por cada vetor de pesos dos nodos a uma cor da paleta de cores.
11. Na imagem final, substituir cada cor original dos pixels pela cor mais próxima encontrada na paleta de cores.

2.5 Discussão Final

Neste capítulo foram apresentados os fundamentos necessários à elaboração de um novo modelo de segmentação de imagens por discriminação de textura utilizando computação inteligente. O problema de segmentação de imagens, sua formalização e métricas específicas serão apresentados no Capítulo 3. A apresentação de um novo modelo de segmentação será realizada no Capítulo 4.

Os principais mapas auto-organizáveis encontrados na literatura e que se mostram adequados para segmentação de textura foram abordados, são eles SOM, GNG, GWR e o LARFSOM. As características fundamentais destes SOMs, úteis ao processo de segmentação, são: aprendizado não-supervisionado; capacidade de adaptação à distribuições de densidade; capacidade de realizar agrupamentos; possibilidade de treinamento on-line. Estas são as características principais que motivaram a utilização de SOMs para segmentação de imagens por textura em detrimento de outras técnicas baseadas puramente em processamento de imagens.

CAPÍTULO 3

Descrição do Problema

Segmentação é uma parte fundamental do processo de particionamento do espaço de dados de imagens em regiões salientes e é um pré-requisito para processos subsequentes mais complexos como reconhecimento de objetos e interpretação de cena. A etapa de segmentação de imagens e a eficiência com que é realizada afeta diretamente a performance de um sistema automático de análise de imagem (Rui et al., 1999). Neste processo, uma imagem pode ser dividida em regiões homogêneas em relação a uma dada característica de textura ou cor para que, posteriormente, as diversas regiões possam ser analisadas de forma individual ou combinadas em macrorregiões. Existem mais de 1000 métodos de segmentação diferentes já publicados (Y. Zhang, 1997) e este problema ainda continua longe de ser resolvido em definitivo (Haindl, 1998). Os vários métodos de segmentação podem, usualmente, ser categorizados em três tipos básicos de abordagens (Reed & Dubuf, 1993): Baseados em região, baseados em fronteira ou híbridos (diversos tipos de combinações dos dois primeiros tipos anteriores). O tipo de abordagem de segmentação depende, essencialmente, do problema, mas normalmente métodos baseados em regiões apresentam maior robustez em relação à delimitação de regiões em imagens com muito ruído. Além disto, se desejado, uma vez obtidas as áreas das regiões segmentadas, o processo para identificar as bordas das regiões é trivial. Assim, pode-se realizar as seguintes diferenciações básicas entre os algoritmos de segmentação em relação ao critério de agrupamento.

- Segmentação por Área (por similaridade) - O processo de segmentação por área é baseado no agrupamento de pixels com características de vizinhança semelhantes. Assim, a segmentação por área pode ser interpretado como um processo de agrupamento de

características semelhantes nas áreas da imagem. Nesta abordagem, o número de agrupamentos é o número de regiões diferentes presentes na imagem observada. Para obter as bordas das regiões um operador de derivada, como o filtro sobel, pode ser utilizado.

- Segmentação por Borda (por dissimilaridade) - O processo de segmentação por borda é baseado no agrupamento de regiões que apresentam um mesmo grau de "dissemelhança local". Ou seja, regiões da imagem que contêm um mesmo nível ou quantidade de transições abruptas. Nesta abordagem o número de agrupamentos utilizados define o número de "curvas de nível"[1] que a imagem segmentada terá e não o número de regiões. Para obter o número de regiões pode ser utilizada alguma técnica de crescimento de regiões.

3.1 Métodos Clássicos de Segmentação de Imagens

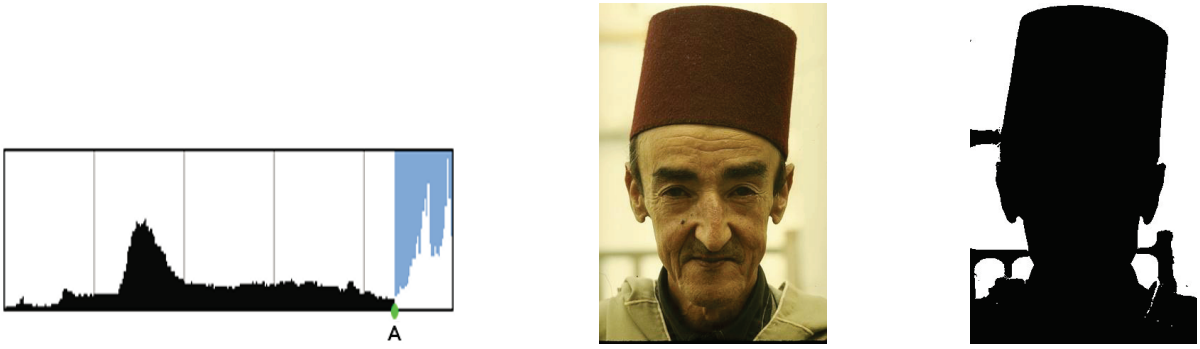
Embora existam muitos tipos diferentes de algoritmos de segmentação de imagens, o panorama geral dos métodos de segmentação utilizados em processamento de imagens pode ser bem representado por três abordagens principais: Limiarização, *Watershed* e Crescimento de Regiões. Estes podem ser considerados os métodos "clássicos" (Gonzalez & Woods, 2000) (Beucher & Mathmatique, 1991). Uma breve descrição sobre cada um destes métodos será realizada a seguir.

3.1.1 Segmentação por Limiarização de Histograma

Em muitas aplicações de processamento de imagens, os níveis de cinza pertencentes a região de interesse são substancialmente diferentes dos níveis de cinza pertencentes ao "fundo" da imagem ou a região onde não há informações de interesse. Métodos baseados em limiar (*th-*

[1] Um tipo de representação gráfica em duas dimensões de uma vista superior de um relevo, comumente utilizada em topografia. Neste tipo de representação, a altura do relevo (eixo Z) é quantizada em um determinado número de camadas que são representadas por curvas fechadas no gráfico.

reshold) são as ferramentas mais simples para separar objetos da região de fundo da imagem (*background*) (Sezgin & Sankur, 2004). Estes métodos são frequentemente utilizados, por exemplo, para análise de imagens de documento, com o objetivo de extrair caracteres, logos ou símbolos. A imagem resultante de uma segmentação por limiar é uma imagem binária, o limiar divide a imagem em uma região pertencente ao fundo (*background*) e outra região pertencente à "frente" (*foreground*). Existem muitas variações de limiarização, mas a grande maioria utiliza apenas as informações presentes no histograma dos níveis de cinza da imagem para realizar a limiarização. A Figura 3.1 ilustra como esta técnica é utilizada para segmentar uma imagem. A Figura 3.1(b) é a imagem original que se deseja segmentar. A Figura 3.1(a) mostra o histograma dos níveis de cinza presentes na imagem analisada, através deste histograma pode-se, de modo *ad-hoc*, escolher um valor na escala de cinza para a segmentação (ex. ponto "A" na Figura 3.1(a)). Existem outros métodos, que também podem realizar a "escolha" do nível de cinza de modo automático, através de técnicas de aglomeração. A Figura 3.1(c), mostra o resultado final da limiarização que foi aplicado sobre a imagem original.



(a) (b) (c)

Figura 3.1 Em 3.1(a) o histograma dos níveis de cinza da imagem original 3.1(b). O ponto "A" no histograma mostra o limiar utilizado para produzir a imagem da Figura 3.1(c).

3.1.2 Watershed

O *Watershed* (Beucher & Mathmatique, 1991) é uma técnica de segmentação que utiliza gradiente descendente para identificar regiões de fronteira e de convergência do gradiente. O método tem inspiração geográfica, a imagem a ser segmentada é tratada como um relevo, onde o valor de intensidade do nível de cinza é a altura do relevo. Gradualmente o relevo vai sendo "inundado" por um líquido que tende a fluir para as regiões mais baixas do relevo, formando bacias hidrográficas. A formação destas "bacias" acaba por particionar a imagem em regiões, concluindo o processo de segmentação. A técnica *Watershed* trás duas vantagens, em relação as técnicas de limiarização. A primeira é a possibilidade de se obter imagens segmentadas em mais de dois tipos regiões, o que não é possível com limiarização que divide a imagem, apenas, em regiões de "frente" e de "fundo". A segunda vantagem é que controlando o modo como o "fluido" imaginário será "despejado" sobre a imagem pode-se ter algum controle sobre como as regiões irão se formar e segmentar a imagem. A desvantagem desta técnica é que normalmente precisa-se conhecer os pontos onde serão colocadas as "fontes de água"(ou sementes). Realizar um posicionamento aleatório do "fluido", como em uma "chuva", também é possível, mas neste caso a segmentação resultante se torna muito fragmentada, característica que é normalmente indesejada.

3.1.3 Crescimento de Regiões

O abordagem básica de crescimento de regiões (Gonzalez & Woods, 2000) inicia a partir de uma região semente na imagem, que é assumida estar dentro da área que se deseja segmentar do resto da imagem. As vizinhanças de todos os pixels sementes são avaliadas para decidir se são pontos que podem ser considerados pertencentes ao mesmo grupo das sementes ou não. Os pixels vizinhos reconhecidos como pertencentes ao mesmo grupo das sementes são rotulados como sementes para uma próxima interação e o processo se repete até que não haja

mais pixels vizinhos das sementes que sejam reconhecidos como pertencentes ao mesmo grupo. O critério de vizinhança adotado (por exemplo, 4 ou 8 vizinhos) e a métrica de semelhança entre pixels varia bastante e depende, essencialmente, da finalidade da segmentação. Em algumas implementações as métricas de distância utilizadas para o crescimento de regiões levam em consideração a textura nas vizinhança dos pixels. Este técnica de segmentação, assim como o *Watershed*, também apresenta um maior número de possibilidades que a limiarização e pode, também, produzir imagens com mais de dois segmentos. Entretanto, parte do pressuposto que o sistema conhece a localização das sementes na imagem.

3.2 Informações Relevantes para Segmentação

Do ponto de vista das informações presentes em uma imagem ou vídeo que podem ser utilizadas em algoritmos de segmentação existem, essencialmente, cinco características relevantes, são elas:

- Intensidade de Luz ou Brilho - Talvez essa seja a característica mais básica e elementar que possa ser utilizada para segmentar imagens. Consiste, basicamente, em dividir a imagem em regiões que apresentam um determinado intervalo de brilho. Existem muitos métodos de segmentação que utilizam exclusivamente esta característica (Sezgin & Sankur, 2004);

- Cor - É uma extensão natural do processo de segmentação por brilho, só que aplicado a cores. Os métodos de segmentação, muitas vezes, são os mesmos que os utilizados para segmentação por brilho;

- Textura - O ser humano tem capacidade nata de perceber e conseguir discriminar certos padrões visuais complexos. Embora não exista uma definição formal do que é textura, é relativamente fácil para um ser humano perceber e conseguir discriminar uma grande

quantidade de micro-padrões em imagens, formados por combinações variadas de cor e luminosidade. Devido a isto, textura é uma característica de extrema importância, porque pode conter uma quantidade de informação muito significativa a respeito de um determinado objeto. Muitas vezes é possível não só realizar segmentação, mas também realizar identificação de objeto utilizando apenas as informações sobre textura (Manjunath & Ma, 1996). Normalmente objetos naturais como árvores, pedras ou água não apresentam uma forma bem definida, mas possuem características texturais bem definidas podendo ser identificados por esta característica;

- Movimento - É uma característica também muito utilizada em processamento de imagens e pode destacar facilmente objetos móveis em uma determinada cena;

- Estereoscopia - Estereoscopia é a capacidade de percepção espacial, de profundidade, em imagem. É utilizada para segmentar uma imagem em planos de profundidade. Assim, mesmo regiões que apresentem cores e brilho semelhantes, mas em distâncias diferentes em relação à câmera, podem ser facilmente percebidas e segmentadas.

Dentre estas características a segmentação por discriminação de textura em imagem tem um papel importante em muitos algoritmos (Reed & Dubuf, 1993) (Cremers et al., 2007) (J. Zhang & Tan, 2002). Uma imagem, frequentemente, tem muitos níveis de cinza e cores diferentes e, por consequência, as técnicas de segmentação baseadas nestas características nem sempre são suficientes para realizar as discriminações desejadas entre as diversas regiões da imagem (Reed & Dubuf, 1993) (Cremers et al., 2007). A detecção de movimento e a visão estereoscópica são duas características muito poderosas para segmentação e discriminação de objetos em uma cena, mas não são muito utilizadas, quando comparada às outras característica, pois precisam de condições especiais para serem utilizadas. A percepção de movimento só pode ser utilizada, por questões óbvias, em imagens em movimento e a estereoscopia necessita da aquisição de pelo menos duas imagens para realizar o registro de um única cena. Em contrapartida, a carac-

terística textural presente em uma cena tem a vantagem de ser altamente específica para muitos objetos (Luo & Crandall, 2006). Além disso, ao contrário dos níveis de cinza e das cores, uma cena, frequentemente, não apresenta uma grande quantidade de texturas. Assim, uma segmentação que utiliza essa característica em seu processo de discriminação das diversas regiões presentes em uma imagem conduz a uma imagem segmentada em poucas regiões e, ainda, dependendo das características dos objetos presentes na cena há uma boa probabilidade de haver regiões inteiras que correspondem a um único objeto na cena. Devido a estas vantagens, a segmentação de imagens baseada em textura é uma área de intensa atividade e desenvolvimento nos últimos trinta anos tendo sido publicados muitos algoritmos como consequência deste esforço de diferentes graus de complexidade (Reed & Dubuf, 1993). Métodos de segmentação por textura supervisionado ou não constituem um requisito básico para buscar de imagens por conteúdo, controle de qualidade, segurança, aplicações médicas e muitas outras.

Métodos não-supervisionados, os quais não assumem qualquer conhecimento a *priori* sobre a cena, são os mais interessantes para as aplicações da segmentação de imagens, mas são, obviamente, mais desafiadores que os métodos supervisionados. O principal problema envolvendo os métodos supervisionados de segmentação é que a rotulação de dados é uma tarefa custosa e/ou complexa em vários sentidos. Sistemas não-supervisionados, não necessitam de rotulação, mas precisam utilizar transformações não-lineares nos dados de treinamento que garantam que os dados sejam categorizados da maneira desejada. Em processamento de imagens essas transformações são normalmente chamadas de descritores de características da imagem. A vantagem de se trabalhar com descritores de características e aprendizagem não-supervisionada é que os descritores podem ser determinados baseados em conhecimento teórico e propriedades estatísticas do problema que esta sendo tratado.

Existem vários métodos (Materka et al., 1998)(J. Zhang & Tan, 2002) para reconhecimento e identificação de texturas em imagens digitais. As principais abordagens são:

- Abordagem Estatística (momentos, estatística de segunda ordem, etc)(Tuceryan, 1994)(J.

Zhang & Tan, 2002)(Cremers et al., 2007);

- Abordagem baseada em modelos (morfologia matemática, dimensão fractal, etc) (J. Zhang & Tan, 2002);

- Abordagem baseada em filtros (Fourier, Wavelets)(J. Zhang & Tan, 2002)(Kim & Kang, 2007).

Nos modelos propostos neste trabalho para resolver o problema de segmentação por textura será considerada uma abordagem de métodos estatísticos de discriminação de textura.

3.3 Formalização do Problema

A formalização do problema de segmentação de imagens por discriminação de texturas pode ser desdobrada em dois casos distintos, dependendo se o número de segmentos é dado. Assim, têm-se as seguintes possibilidades para os enunciados:

1. Dados uma imagem $I_{N\times M}$, um conjunto de descritores de características Ψ e um número K de regiões desejadas, obter de modo não-supervisionado uma imagem discriminada em exatamente K regiões de texturas diferentes;

2. Dados uma imagem $I_{N\times M}$ e um conjunto de descritores de características Ψ, obter de modo não-supervisionado uma imagem discriminada em K regiões de texturas diferentes, onde, K não é conhecido a *priori*.

Este trabalho propõe uma abordagem capaz de atacar as duas variações do problema de segmentação por discriminação de textura.

3.4 Métodos de Avaliação

Um dos principais problemas dos algoritmos de segmentação não-supervisionados é determinação da métrica de comparação. Em algoritmos supervisionados, o tutor pode simplesmente treinar uma rede neural para reconhecer, por exemplo, telha de uma casa. Se ao apresentar a imagem de uma casa as telhas desta não forem identificadas prontamente pelo algoritmo de segmentação então isto pode ser considerado um erro cometido pelo sistema. Em um sistema não-supervisionado a informação do que deve ser considerado cabe, essencialmente, ao descritor dos dados que esta sendo utilizado. Por isso, a escolha de bons descritores é essencial e deve seguir alguns critérios que levem a crer que a informação desejada será destacada na imagem. Entretanto, frequentemente os descritores são funções com dimensão maior que 3 e são difíceis de serem completamente visualizados. Ou seja, muitas vezes certos comportamentos ficam ocultos. Assim, ao se escolher um determinado conjunto de descritores a serem utilizados em conjunto com métodos de agrupamento é possível ter uma previsão do que ocorrerá, mas não uma garantia. As vezes, ocorrem alguns resultados inesperados, mas que não podem ser considerados simplesmente errados. Este é um procedimento muito delicado, porque não necessariamente há um erro na segmentação. Dois segmentadores diferentes podem produzir duas segmentações distintas de uma mesma imagem sem que necessariamente um esteja errado ou seja menos eficiente. Este problema de interpretação ocorre com o próprio homem. Se for entregue, por exemplo, uma mesma fotografia para duas pessoas diferentes recortarem a imagem em regiões interessantes muito provavelmente o resultado dos cortes será diferente. Com a finalidade de reduzir o problema da subjetividade da questão, foram selecionadas três métricas de comparação distintas baseadas nos trabalhos de Mehmet Sezgin e Bulent Sankur (2004) e Hoover et al. (1996).

3.4.1 Erro de Classificação

O erro de classificação (*Misclassification error, ME*) reflete o percentual de imagem de *background* classificada erradamente como *foreground* e a quantidade de *foreground* classificado erradamente como *background*. Para um problema de segmentação com duas regiões o ME pode ser expresso pela Equação 3.1.

$$ME = 1 - \frac{|B_O \cap B_T| + |F_O \cap F_T|}{|B_O| + |F_O|} \tag{3.1}$$

onde B_O e F_O denotam o *background* e o *foreground* verdadeiros de uma imagem de referência. Esta imagem é, normalmente, denominada de *groundtruth*, por definir a verdadeira segmentação da imagem. O termos B_T e F_T denotam as áreas de *background* e o *foreground* discriminadas na imagem de teste e $|\cdot|$ é a cardinalidade do conjunto (a quantidade de pixels). O ME varia de 0, para uma segmentação perfeitamente de acordo com o *groundtruth*, a 1 para uma imagem segmentada de modo completamente errado em relação a imagem de *groundtruth*. Este trabalho, considera o caso com n regiões R_i ($i = 1..n$) realizando a média da aplicação da Equação 3.1 para cada par $\{F_O = R_i, B_O = \bigcup_{j \in P} R_j \mid i = 1..n;\ P = \{1,..,N\} - i\}$.

3.4.2 Não Uniformidade de Região

A métrica de não uniformidade de região (*Region Nonuniformity*, NU) avalia o grau de homogeneidade com que as regiões são segmentadas, de modo semelhante a uma medida de dispersão. Por isso, neste caso não é necessário o conhecimento do *groundtruth* para sua avaliação, ela esta definida na Equação 3.2

$$NU = \frac{|F_T|}{|F_T + B_T|} \frac{\sigma_f^2}{\sigma^2} \tag{3.2}$$

onde, σ^2 representa a variância da imagem inteira e σ_f^2 representa a variância do *foreground*, F_T e B_T são, respectivamente, o conjunto de pontos pertencentes ao *foreground* e ao *background*. É esperado que em uma imagem bem segmentada tenha regiões uniformes e contínuas e, neste caso, esta métrica apresentará um valor próximo de zero. Enquanto que, em segmentações muito fragmentadas, o valor se torna próximo de 1. Este trabalho, considera o caso com n regiões R_i $(i = 1..n)$ como o maior valor obtido utilizando a Equação 3.2 para cada par $\{F_T = R_i, B_T = \bigcup_{j \in P} R_j \mid i = 1..n;\ P = \{1,..,N\} - i\}$.

Dadas duas regiões R_i e R_j da imagem teste e duas regiões $\bar{R}_u$ e $\bar{R}_v$ da imagem *groundtruth*, a determinação das respectivas regiões de *foreground* (F_T e F_O) e *background* (B_T e B_O) para as métricas ME e NU é realizada de acordo as Regras 3.3 à 3.6.

$$F_O \equiv \bar{R}_u \tag{3.3}$$

$$B_O \equiv \bar{R}_v \tag{3.4}$$

$$F_T = \begin{cases} R_i \ ; \ \text{se } |R_i \cap F_O| \geq |R_j \cap F_O| \\ R_j \ ; \ \text{caso contrário} \end{cases} \tag{3.5}$$

$$B_T = \begin{cases} R_i \ ; \ \text{se } |R_i \cap F_O| < |R_j \cap F_O| \\ R_j \ ; \ \text{caso contrário.} \end{cases} \tag{3.6}$$

3.4.3 Detecções Corretas

A métrica de detecções corretas (*Correct Detection*, CS) avalia o total de regiões segmentadas corretamente de acordo com a imagem do *groundtruth*. Para uma imagem segmentada em n regiões R_i ($i = 1..n$) e a correspondente imagem de *groundtruth* com m regiões $\bar{R_u}$ ($u = 1..m$) , uma detecção correta precisa satisfazer à Regra 3.7.

$$D_{iu} = \begin{cases} 1 \ ; \ \text{Se} \ |R_i \cap \bar{R_u}| \geq k|R_i| \ \text{e} \ |R_i \cap \bar{R_u}| \geq k|\bar{R_u}| \\ 0 \ ; \ \text{caso contrário.} \end{cases} \tag{3.7}$$

onde k é uma constante que controla qual a taxa aceitável de sobreposição entre as regiões na imagem de teste e na imagem do *groundtruth*. O valor de k é escolhido entre 0,5 e 1. Nesta dissertação $k = 0,6$.

O valor das detecções corretas (CS) é dado pela Equação 3.8.

$$CS = \frac{1}{nm} \sum_{i=1}^{n} \sum_{u=1}^{m} D_{iu} \tag{3.8}$$

CAPÍTULO 4

Abordagem Proposta

Análise de textura e segmentação de imagens têm sido estudadas por um longo período de tempo usando várias abordagens distintas (Zheng et al., 2006). Vários métodos realizam a análise de textura diretamente sobre os tons de cinza de uma imagem. Exemplos de alguns destes métodos incluem *grey level co-occurrence matrix* (GLCM) (Haralick et al., 1973) e filtragem espacial (Weszka et al., 1976), métodos baseados em propriedades estatísticas e de frequência presentes em imagens. Em sua maioria, estes métodos se originaram da necessidade prática de segmentar imagens aéreas fornecidas por satélites ou por aviões de reconhecimento aéreo. Em muitos trabalhos envolvendo estes métodos não há uma diferenciação nítida do conceito de padrão e de textura. Seguindo uma linha de pensamento distinta dos modelos anteriores, baseando-se em psicologia humana e considerando textura como um padrão identificável pelo visão humana, Julesz (1981) publicou um estudo sobre como o sistema visual humano consegue realizar discriminação de texturas. Ele mostrou que o homem consegue identificar diferenças de texturas em imagens formadas por certos padrões primitivos de modo espontâneo, não consciente, e sem, necessariamente, ter consciência do padrão primitivo. Esta capacidade de segmentação, inerente do ser humano, de conseguir dividir as imagens em partes menores sem ter a consciência em si do seu conteúdo foi denominada de sistema visual de pré-atenção (*pre-attentive visual system*) e os padrões primitivos detectados por este sistema foram denominados *textons*. Estes seriam as peças fundamentais do sistema de segmentação de textura humano.

Julesz conjecturou, baseado em alguns testes práticos com padrões visuais, que os textons deveriam ter uma forma alongada em uma determinada direção e que o sistema visual humano seria capaz de detectar estatísticas de primeira ordem baseadas nestes componentes através de

informações como tamanho, espessura, cor, concentração e orientação.

É interessante notar que parte das definições e conceitos desenvolvidos por Julesz conflitam diretamente com alguns métodos já bem estabelecidos de segmentação por textura, como, por exemplo, as técnicas que utilizavam o espectro de potência de Fourier para segmentar imagens com texturas de diferentes graus de granulosidade (Weszka et al., 1976). Julesz mostrou em um trabalho (Julesz & Bergen, 1983) que é possível construir padrões, analiticamente, que tenham espectro de potência idênticos e sejam facilmente distinguíveis visualmente pelo ser humano. Comprovando que o sistema visual humano de pré-atenção não distingue padrões por análise de Fourier. Este e outros conflitos existentes entre as várias abordagem de segmentação de imagens por textura podem ser, em parte, explicados pela definição vaga e subjetiva do que vem a ser propriamente uma textura (Zheng et al., 2006). A análise de Julesz é antropocêntrica e muitos outros métodos, principalmente os mais antigos, que derivam de uma abordagem mais técnica e baseada em teoria de Sinais e Sistemas não compartilham o mesma ponto de vista. Nota-se claramente que a segmentação por textura pode variar muito não só no que diz respeito às técnicas utilizadas, mas também na interpretação do que é uma textura em si.

Tendo em mente tais estudos, problemas e conflitos envolvendo a segmentação por textura, este trabalho propõe a implementação de um sistema de segmentação de imagens que utiliza conceitos inspirados no trabalho de Julesz, mas que também se estende como um sistema mais geral de segmentação não se restringindo a um conceito antropocêntrico de textura. Principalmente tendo em vista que, apesar da construção de um sistema automático de segmentação segundo uma visão humana possa ser muito interessante, muitas vezes é necessário seguir por outro caminho. Isto é, um sistema que seja capaz de fornecer uma interpretação de uma determinada imagem que não nos seja óbvia ou visualmente distinguíveis, algo como uma visão espectral de uma imagem. Assim, na proposição desta dissertação, os padrões primitivos formadores de textura, ou o que seria equivalente aos textons de Julesz, são funções arbitrárias que realizam um mapeamento entre cada pixel de uma janela da imagem para um espaço de

características. Este trabalho denomina tais funções arbitrárias de descritores de características. Um mapa auto-organizável dinâmico é utilizado para capturar as características mais gerais dos agrupamentos formados nos espaço de característica e, posteriormente, realizar o mapeamento inverso, do espaço de características para os pixels para obter a imagem segmentada em regiões. O modelo proposto visa funcionar como um *preattentive visual system* humano (Julesz, 1981)(1983)(1988)(Tuceryan, 1994)(Kovacs & Julesz, 1994)(Serre et al., 2007) que segmenta uma imagem utilizando padrões primitivos sem a necessidade do conhecimento de nenhum rótulo, isto é, utilizando uma abordagem não-supervisionada.

A proposta apresenta duas abordagens distintas para treinamento de padrões e duas formas de segmentação. Em relação ao treinamento, a percepção de características gerais pode ser adquirida com respeito à imagem que está sendo segmentada ou em relação a um grupo de imagens de referência. Em relação ao tipo de segmentação, o sistema pode atuar de modo a ser sensível à borda ou a áreas. Estas abordagens produzem resultados semelhantes e a partir de uma segmentação por área é possível se obter uma segmentação por borda e vice-versa. A utilização da opção de segmentação por borda ou por área depende especificamente da finalidade da aplicação da segmentação.

O diagrama do modelo proposto (Abordagem 1), focando apenas nas características gerais da imagem a ser segmentada, pode ser visto na Figura 4.1. Neste modelo o sistema recebe como parâmetros de entrada uma imagem e um descritor de características (o análogo funcional a um *texton*). O tipo do mapa auto-organizável, empregado como categorizador não-supervisionado, utilizado pelo modelo depende, essencialmente, se o número de segmentos da imagem será pré-determinado. Caso não haja nenhuma informação a *priori* sobre o número de segmentos um mapa com crescimento controlado pelo erro não-supervisionado, como o GWR ou LARF-SOM, pode ser utilizado. Caso se deseje um número fixo de segmentos, pode-se utilizar um mapa como o GNG, ou mesmo, o mapa original de Kohonen. O sistema utiliza um filtro gaussiano passa-baixa (FPB) sobre a imagem original, com a finalidade de remover variações muito

grandes nas cores dos pixels de imagem e de servir como ferramenta de seleção do nível de detalhes presentes na segmentação. A amostragem das características presentes na imagem a ser segmentada é realizada utilizando-se uma janela móvel, transformando as informações contidas em cada pixel de cor ($\mathbf{p} = [p_r, p_g, p_b]$) da janela em vetores $\mathbf{v_{D_x}}$ (representando *textons*, considerando a analogia com o trabalho de Julesz). A partir da lista com os vetores de textons (A) é composto um conjunto de treinamento, com dados normalizados, que será apresentado ao mapa auto-organizável. Após realizar o agrupamento do espaço de características, o mapeamento inverso dos centros do mapa no espaço de características para o espaço dos pixels produz uma nova imagem que corresponde a uma segmentação por discriminação de características da imagem original. Dependendo do descritor de característica utilizado, o sistema pode segmentar as regiões da imagem de diferentes maneiras. No sistema proposto nesta dissertação, o processo de composição das regiões se dá baseado exclusivamente nas informações presentes na imagem que está sendo analisada. O modelo não utiliza nenhum tipo de dicionário de texturas para realizar associações de padrões com a imagem teste, como empregado em alguns trabalhos (Raghu & Yegnanarayana, 1998)(Kwang In Kim & Kim, 2002)(Batista et al., 2005)(Kim & Kang, 2007).

Uma segunda variação do modelo (Abordagem 2) pode ser implementada, focando em características gerais presentes em várias imagens de referência, como mostrado no diagrama da Figura 4.2. Esta abordagem visa tirar proveito da capacidade dos mapas auto-organizáveis de conseguirem aprender funções de densidade de probabilidade. Assim, características muito presentes em um determinado conjunto de imagens podem ser destacadas mesmo que estas características não estejam presentes de maneira significativa na imagem que esta sendo segmentada. Isto é importante quando a parte que se deseja segmentar ou destacar é pequena em termos de área em relação a área total da imagem. Nesta segunda abordagem, o modelo não utiliza um "dicionário" ou banco de dados de texturas conhecidas. O banco de imagens de referência não possui nenhum tipo de rotulação de textura e não necessariamente cada imagem

do banco de imagens precisa conter uma única textura.

As diferenças entre as Abordagens 1 e 2 estão unicamente no modo de treinamento do mapa auto-organizável. Todas as demais etapas e parâmetros do modelo são idênticos para ambas as abordagens.

O método de segmentação de imagens por área pode ser visto no Algoritmo 8 e o método proposto para segmentação por borda pode ser visto no Algoritmo 9, ambas na seção 4.3.

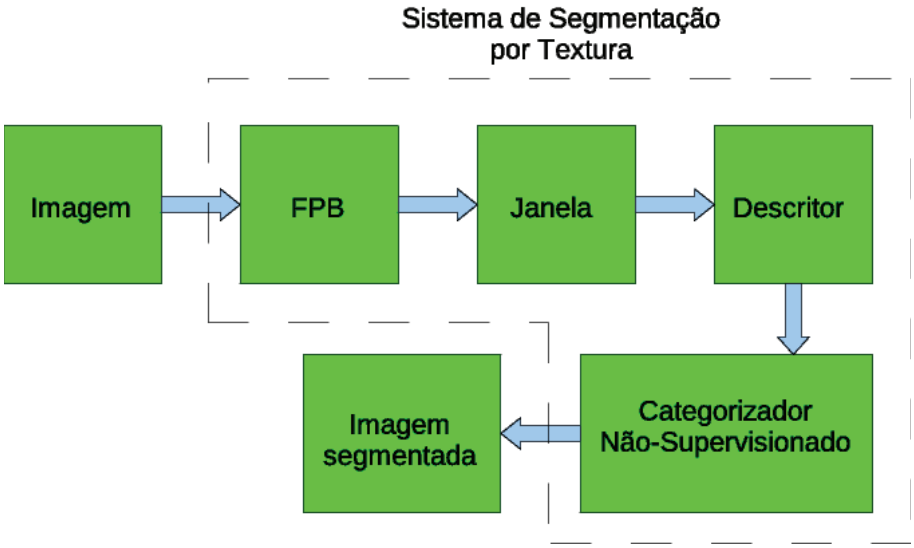


Figura 4.1 Diagrama em bloco do Abordagem 1 proposta para segmentação por textura. Neste, o sistema realiza a segmentação baseado no treinamento com amostras de características provenientes unicamente da imagem teste.

4.1 Requisitos e Restrições da Solução e do Problema Proposto

Na primeira abordagem o modelo proposto precisa como parâmetros de entrada: uma imagem para segmentar $im_{M\times N}$, um tipo de janela (para detectar área ou borda), um tamanho de janela de inspeção (L_x e L_y) , um período de amostragem para cada dimensão (δ_i e δ_j), um parâmetro de escala s para um filtro gaussiano e descritor de características D_x.

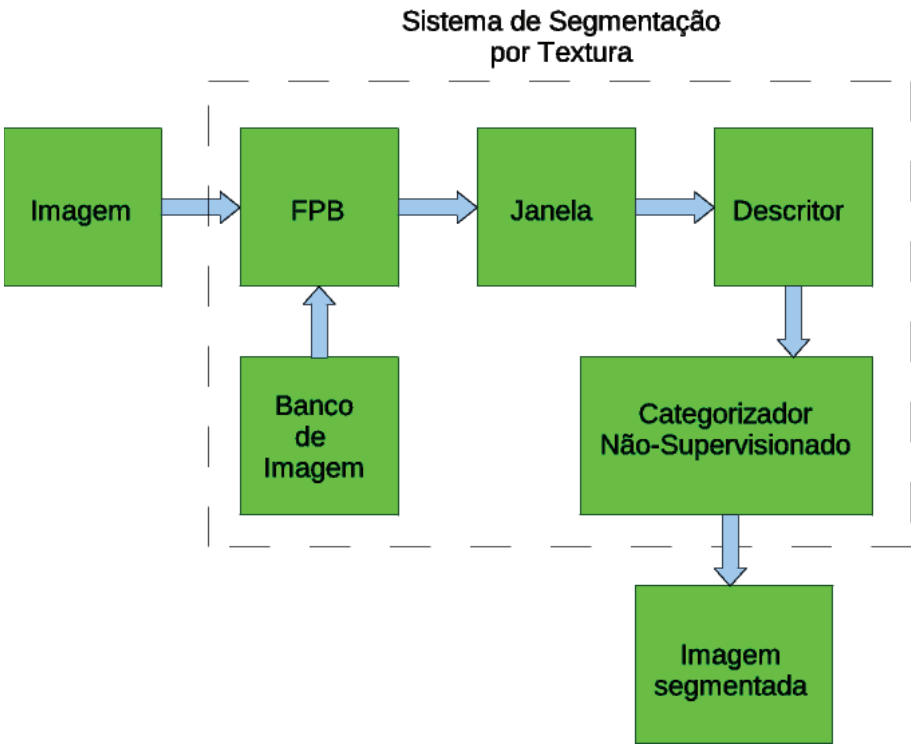


Figura 4.2 Diagrama em bloco do Abordagem 2 proposta para reconhecimento de textura. Neste, o sistema é treinado utilizando amostras provenientes da imagem teste e de um conjunto de outras imagens de referência. O sistema segmenta a imagem teste baseado nas características gerais presentes em todas as imagens utilizada no treinamento e não somente na imagem teste. Este modo de treinamento é especialmente interessante se a característica que se deseja segmentar não muito frequente na imagem teste.

O sistema pode utilizar dois tipos básicos de janelas, uma retangular, mais convencional, utilizada para segmentação de área e uma em forma de estrela (ver Figura 4.5) para segmentação por borda. A principal diferença entre as janelas é a forma como capturam os valores dos pixels. Na retangular todos os pixels contidos na janela formam um único vetor que é utilizado como entrada para o descritor de características e posteriormente para realizar o agrupamento por semelhança. A janela em forma de estrela captura um par de vetores a cada 45 graus e estes pares são utilizados individualmente pelo descritor de características e, posteriormente, para realizar um agrupamento por dissemelhança.

Os parâmetros L_x e L_y correspondem na janela retangular as suas arestas, respectivamente,

nos eixos x e y. No caso da janela em forma de estrela o parâmetro L_x corresponde ao tamanho dos vetores dispostos radialmente em relação ao centro da janela e o parâmetro L_y não é utilizado neste caso. O tamanho da janela (parâmetros L_x e L_y) são críticos para o bom funcionamento do sistema e precisam ter um tamanho suficiente para capturar toda a informação necessária para realizar a discriminação das texturas presentes na imagem. O ajuste deste parâmetro deve ser feito através de tentativa e erro e de conhecimento experimental. Infelizmente, apesar do uso de janelas ser utilizado praticamente universalmente na análise de textura em imagens, não existe um "método" padrão conhecido para o ajuste de tal parâmetro (Gonzalez & Woods, 2000). Esse é um problema em aberto que afeta, como um todo, a área de análise de textura. Entretanto, seguindo a linha de pensamento proposta por Julesz, que preconiza que todas as texturas são formadas por pequenas "peças" (os textons), pode-se, através de uma inspeção visual em algumas imagens de teste, definir qual é o tamanho aproximado destas janelas para que sejam suficientes para capturar as "peças" que formam as texturas. Isto pode ser obtido pela resposta a seguinte pergunta: *Qual é a maior janela que pode ser utilizada para inspecionar uma determinada imagem teste de modo que se consiga perceber os padrões de textura sem, contudo, ser suficiente para identificar os objetos e a cena presentes na imagem?* É uma pergunta subjetiva e que trás consigo o referencial de apenas um indivíduo. Utilizando este critério é possível obter resultados satisfatórios para a definição do tamanho da janela. A limitação desta heurística é que só pode ser empregada quando o sistema realiza uma segmentação seguindo padrões "humanos" de reconhecimento de textura. Se for desejado utilizar o sistema para reconhecer padrões não perceptíveis para o homem, a determinação deste parâmetro terá que ser realizada apenas com tentativa e erro.

O período de amostragem corresponde, respectivamente, aos espaçamentos em x e y que são utilizados para deslizar a janela de amostragem por toda a área da imagem a ser segmentada. O parâmetro s é o raio da janela de um filtro gaussiano utilizado para realizar um suavização preliminar na imagem de teste. Este é referido como "parâmetro de escala" porque através

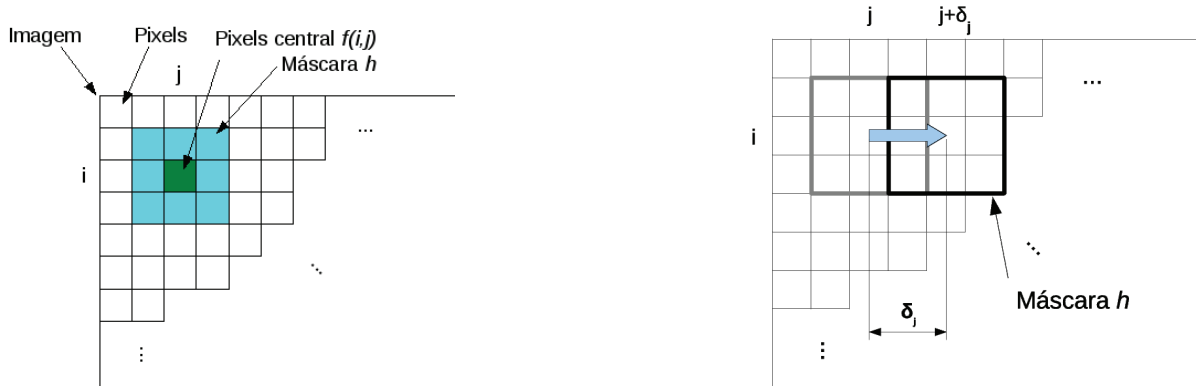
deste é possível ajustar qual é o nível de detalhes, exatamente como um filtro passa baixa, que se deseja observar na imagem, sem que para isto seja necessário realizar um reescalonamento desta imagem. É um parâmetro particularmente interessante e útil quando se deseja realizar uma segmentação estratificada, onde você inicia o processo com um valor elevado de s, para segmentar apenas as características mais "marcantes" da imagem, e posteriormente realizando refinamentos em cada uma das áreas iniciais até obter o resultado desejado. Embora esta característica não necessariamente precise estar "incluída" em um sistema de segmentação ele torna mais conveniente realizar análise multiresolução. Um ajuste simples para este parâmetro é considerar s uma ordem de grandeza inferior ao comprimento, em pixels, da maior diagonal dos objetos ou texturas que se deseja segmentar.

O último parâmetro, o descritor de características, atua como uma função dos pixels contidos na janela de inspeção que realiza uma transformação do espaço dos vetores de pixels para um outro espaço de características. No sistema, o descritor tem o papel de atuar como a função que define cada *texton*, mas não necessariamente precisa ser uma função que atue com a mesma percepção humana. O sistema proposto visa ser uma generalização do sistema de pré-atenção humano e não um modelo para o sistema humano. Deste modo, a escolha do tipo de descritor pode modificar a percepção do sistema em relação à detecção dos "padrões naturais" contidos nas imagens analisadas. Utilizando descritores capazes de identificar padrões semelhantes aos *textons* de Julesz, a segmentação é aproximada a um padrão "humano" e, utilizando outros tipos, pode-se ter um sistema capaz de detectar padrões não capturados pelo processamento visual de humanos. Com este conceito, o sistema se torna bastante geral e flexível.

4.1.1 Tipos de Janelas Utilizadas no Modelo e seu Funcionamento

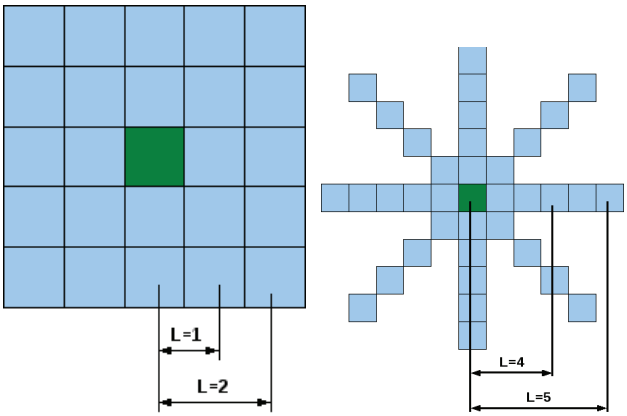
O modelo proposto pode utilizar dois tipo de janelas para delimitar o espaço de análise e extrair as características necessárias a segmentação. As janelas realizam um varredura pixel a pixel sobre as imagens analisadas. A Figura 4.3(a) mostra uma representação geral de uma janela

utilizada em processamento de imagem, ela é centrada no pixel $im(i, j)$ da inspeção. Neste sistema, toda a informação que poderá ser utilizada para decidir a qual segmento pertence o pixel $im(i, j)$ está contida na janela, área em cinza claro na Figura 4.4. Como as informações da janela são tratadas dependerá, essencialmente, do descritor utilizado, a discussão sobre descritores de imagem está na Seção 4.1.2. O deslocamento da janela sobre a imagem é realizado em passos de amostragem (δ_i, δ_j), como ilustrado na Figura 4.3(b). Uma janela quadrada (ver Figura 4.4(a)) é utilizada para a segmentação por área. Neste processo, todos os pixels contidos na área da janela são utilizados como entrada de dados do descritor de textura e apenas um vetor de características é produzido e associado ao pixel central da janela. Para a segmentação por borda, um tipo diferente de janela é utilizada. Um nova janela em forma de estrela (ver Figura 4.4(b)) é proposta e utilizada para que o processo de captura de textura possa ocorrer segundo uma determinada direcionalidade. Um total de 8 vetores (indicados na Figura 4.5) de características são produzidos por pixel central. Uma borda é detectada quando, no mapa auto-organizável, a soma das distância entre os nodos vencedores em cada uma das direções da janela exceder um limiar th. Exemplo, dado um limiar th e os vetores horizontais da janela $\mathbf{v_0}$ e $\mathbf{v_4}$, Figura 4.5, o sistema detecta uma borda na vertical quando a distância entre os nodos vencedores no mapa associados aos respectivos vetores $\mathbf{v_0}$ e $\mathbf{v_4}$ for maior que th. O mesmo procedimento se aplica para os outros pares de vetores contidos na janela de detecção de bordas. Ou seja, os pares $\mathbf{v_1}$ e $\mathbf{v_5}$, $\mathbf{v_2}$ e $\mathbf{v_6}$ e o par $\mathbf{v_3}$ e $\mathbf{v_7}$ estão relacionados, respectivamente, com a detecção de bordas em 45 graus, 90 graus e 135 graus em relação a horizontal (vetores $\mathbf{v_0}$ e $\mathbf{v_4}$). Os mesmos descritores utilizados para a discriminação de áreas de textura diferentes podem ser utilizados em relação a detecção de bordas, entretanto os descritores são aplicados a cada um dos oito vetores direcionais da máscara de detecção de borda.



(a) (b)

Figura 4.3 A Figura 4.3(a) ilustra o posicionamento da janela de amostragem em relação ao pixel investigado $f(i, j)$ e a imagem teste. A Figura 4.3(b) ilustra a movimentação da janela de amostragem de acordo com o ajuste dos passos (δ_i, δ_j).



(a) Segmentação por área (b) Segmentação por borda

Figura 4.4 A janela 4.4(a) é utilizada para segmentação por área e a janela 4.4(b) é para segmentação por borda, nesta é possível ter informação sobre a direção que um determinado padrão de tons de cinza ou cor se apresenta. Independente do tipo da janela o seu tamanho é sempre regulável e é um parâmetro do sistema.

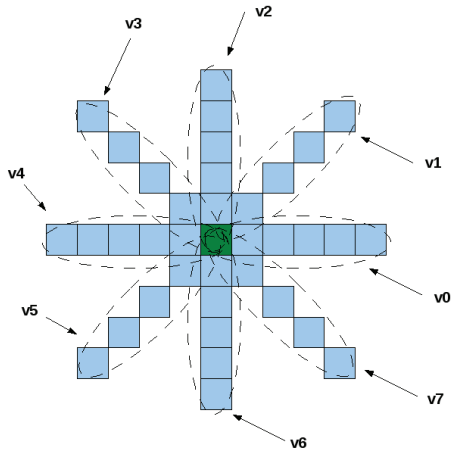


Figura 4.5 A janela utilizada para detecção de bordas produz um total de 8 vetores no espaço de características($\mathbf{v_0}, \mathbf{v_1}, \mathbf{v_2}, \mathbf{v_3}, \mathbf{v_4}, \mathbf{v_5}, \mathbf{v_6}, \mathbf{v_7}$) por pixel central investigado.

4.1.2 Descritores Implementados

Descritores de imagem são funções que realizam uma transformação do espaço de cores (ou dos níveis de cinza) para um espaço de características (Gonzalez & Woods, 2000). Devido ao alto grau de redundância de informação presente no espaço de cor, frequentemente, para representar as informações deste espaço, são utilizas funções não unívocas e não lineares. Ou seja, muitas vezes o descritor exerce um papel semelhante a de uma função *hash*. Matematicamente, um descritor pode ser interpretado como expresso na equação 4.1.

$$D_x(J_{M^* \times N^*}) = \mathbf{v_{D_x}} \therefore M^* < M; N^* < N \tag{4.1}$$

onde, D_x é uma função qualquer que recebe como parâmetro de entrada uma submatriz $J_{M^* \times N^*}$ (a janela de amostragem, ver Seção 4.1.1) com M^* linhas e N^* colunas retiradas da imagem inspecionada $im_{M \times N}$ e retorna um vetor k-dimensional $\mathbf{v_{D_x}}$. No sistema proposto, os vetores $\mathbf{v_{D_x}}$ são utilizados como entrada para a rede neural do sistema.

Quatro descritores de textura foram implementados para testar o modelo proposto de segmentação de imagens. Todos os descritores são baseados em propriedades estatísticas e são aplicados diretamente sobre os valores dos pixels da imagem em RGB, imagens em tons de cinza são consideradas possuindo as três componentes de cor (R, G e B) idênticas. Com essa abordagem o sistema é capaz de tratar de maneira indistinta tanto imagens coloridas como imagens em tons de cinza. A apresentação resumida dos descritores implementados e sua motivação, são tratados a seguir:

- *Descritor de histograma* – Este é o tipo mais geral possível de descritor baseado em estatística de primeiro ordem de cor (Zheng et al., 2006). O seu único parâmetro é o número N de intervalos para construir o histograma. Com um valor definido para N[1], um histograma dos níveis de intensidade de cada componente de cor presentes na janela utilizada é montado. Os histogramas para os canais R, G e B da imagem são concatenados formando um vetor com dimensão $3N$ que é utilizado como padrão de entrada para o treinamento de um mapa auto-organizável. Apesar de ser uma forma geral de caracterizar uma distribuição de cor presente em uma determinada parte de uma imagem, este descritor apresenta, em geral, uma dimensionalidade relativamente alta. Utilizando, por exemplo, histogramas com 10 divisões para cada cor o resultado da composição final do vetor de características terá dimensão 30. Isto é custoso tanto em termos de processamento, pois, os mapas precisam de mais interações no treinamento, quanto em termos de memória;

- *Descritor de Média e Variância* – Supondo uma distribuição normal no histograma de cada componente de cor da janela utilizada, representá-la simplesmente pela sua média e variância é uma maneira natural, eficiente e compacta. Logicamente que, apesar do teorema do limite central[2], é uma pressuposto forte espera que uma determinada distri-

[1] Nos testes realizados utilizando este descritor $N = 8$.

[2] O teorema do limite central, em probabilidade, expressa o fato de a soma de muitas variáveis aleatórias independentes e com mesma distribuição de probabilidade tender à distribuição normal.

buição de cor calculada sobre uma determinada parte de uma imagem seja normal ou mesmo que seja normal na maioria da vezes. Entretanto, apesar disto, este descritor simples tem se mostrado adequado para muitas aplicações (Hoogs et al., 2003)(Zheng et al., 2006). É particularmente interessante por ser em duas dimensões e permitir a visualização dos conjuntos de pontos (média,variância) para uma dada imagem, muito útil para uma compreensão inicial do processo (ver Figura 5.6.1);

- *Descritor MVSK* – É uma descritor semelhante ao descritor de momentos de Hu (Hu, 1962), só que não apresenta invariância a transformações afins e possui dimensão 4. As siglas "MVSK" correspondente às medidas de média, variância, *skewness* e *kurtosis*. Todas estas métricas são formas de parametrização do histograma de cor. A média e a variância são medidas bem conhecidas da localização do "pico" do histograma e de sua dispersão. O *skewness* mede a simetria do histograma podendo assumir valores positivos, negativos ou zero. O *kurtosis* é uma medida de quão agudo é o pico do histograma. Com este conjunto de quadro medidas é possível se caracterizar relativamente bem um histograma utilizando apenas 4 dimensões.

- *Descritor de Haralick* – O artigo que propõe este descritor é de 1973 (Haralick et al., 1973), apesar disto este descritor ainda é válido, utilizado e citado em muitos trabalhos atuais (Zheng et al., 2006). É, talvez, o descritor de textura mais antigo ainda utilizado. Ele se baseia em estatística de segunda ordem de níveis de cinza. A descrição do seu funcionamento está na Seção 2.2.

Os descritores implementados são apenas algumas possibilidades, uma variedade enorme de métodos e ferramentas podem ser utilizadas para se construir descritores de textura (Zheng et al., 2006). Entretanto as ferramentas mais utilizadas estão entre cadeia Markov (Perez, 1998), Wavelets (Kim & Kang, 2007), estatística (Haralick, 1979), análise de Fourier (Weszka et al., 1976) e combinações variadas destas. Vários estudos sobre outras possibilidades de descritores

de textons foram iniciados e ainda estão em curso.

4.1.3 Mapas Auto-Organizáveis

O modelo proposto tem em seu cerne um mapa auto-organizável dinâmico para realizar a categorização dos dados fornecidos pelos descritores, sendo um dos parâmetros principais do modelo o número de neurônios utilizados no processo de segmentação. O fato de contar com um número de neurônios regulável para segmentação não necessariamente implica no uso de um mapa dinâmico, um mapa simples de Kohonen poderia ser utilizado ou até mesmo um algoritmo mais simples como o *k-means*. Entretanto, mapas dinâmicos oferecem algumas vantagens adicionais. Uma primeira vantagem é que utilizando um mapa dinâmico não é necessário se preocupar com a topologia, preocupação esta existente no modelo de Kohonen original. Uma segunda vantagem é que certos tipos de mapas como o GWR e o LARFSOM não têm crescimento baseado em inserções fixas de nodos novos, crescem em função de um parâmetro de erro de aprendizagem não-supervisionada. Assim, poderia haver um sistema de segmentação onde é imposto o número de nodos ao sistema, utilizando, por exemplo, um mapa como o GNG ou poder-se-ia deixar o sistema se "adaptar" e criar um número adequado de nodos até atingir um determinado critério de erro. Este tipo de capacidade "adaptativa" é inerente nos mapas auto-organizáveis dinâmicos. Para os teste, com o modelo proposto foram implementados o mapa de Kohonen, o GNG, o GWR, o LARFSOM e mais duas variações baseadas na estrutura do LARFSOM. As variações implementadas do LARFSOM foram propostas nesta dissertação para estudo adicional e para conhecer algumas características de comportamento deste mapa.

4.2 Revisitando o Algoritmo do LARFSOM

Como comentado na Seção 2.4.1.4 o LARFSOM foi projetado especificamente para trabalhar com imagens coloridas. Entretanto o algoritmo pode ser facilmente reescrito, sem impacto sobre o comportamento em si, para atuar sobre dados genéricos. Para tal, basta considerar os dados de entrada como vetores e não como pixel (com componentes R,G,B). Realizando esta consideração e eliminando algumas etapas do algoritmo do LARFSOM específicas para imagens (Passos 10 e 11 Algoritmo 4) resulta no que foi denominado de "LARFSOM Generalizado", descrito no Algoritmo 5. A única mudança real do LARFSOM para o "LARFSOM Generalizado" foi a inserção, por motivos práticos, de um parâmetro adicional (N_{max}) que impõe um limite superior para o crescimento do mapa. Enquanto o número de nodos do mapa for menor que o valor máximo de nodos permitidos (N_{max}) o funcionamento é idêntico ao do LARFSOM, quando o número máximo de nodos permitidos for atingido o passo 7 do Algoritmo 5, que produz a inserção de novos nodos, não é mais executado. O passo 9, do algoritmo original, também foi modificado na generalização do LARFSOM, onde havia uma etapa de remoção de nodos desconectados. Entretanto, essa etapa não é necessária, pois de acordo com o passos 7 (o único que pode desconectar nodos) sempre são inseridas duas conexões entre três nodos, não havendo a possibilidade de um nodo ficar sem nenhuma conexão. Tomando-se este algoritmo como base, algumas variações deste algoritmo foram realizadas com a finalidade de compreender melhor o comportamento do LARFSOM e identificar alguns pontos que podem ser modificados. Os resultados de alguns testes envolvendo o LARFSOM e suas variações estão na Seção 5.1.

Algoritmo 5 LARFSOM Generalizado

1. Inicialização de parâmetros: taxa de aprendizagem final (ρ_f); modulador de taxa de aprendizagem (ε); limiar ou valor de corte de atividade (a_T); número de vitórias de cada nodo i ($d_i = 0$); número máximo de vitórias de cada nodo (d_m); número máximo de nodos (N_{max}); iteração de tempo ($t = 0$); tempo máximo (t_{max}); erro mínimo ($deslocamento_{min}$); e o número de unidades iniciais ($N = 2$), que devem ser conectadas e cujo os pesos podem ser copiados de dois vetores, escolhido aleatoriamente, do conjunto de dados de entrada.
2. Apresente um vetor de dados escolhido aleatoriamente, ξ, como entrada à rede.
3. Encontre os dois nodos $\{\mathbf{w_{s_1}}, \mathbf{w_{s_2}}\}$ com as menores distância Euclidianas em relação ao vetor de entrada:

$$d(\mathbf{w_{s_1}}, \xi) \leq d(\mathbf{w_{s_2}}, \xi) \leq d(\mathbf{w_i}, \xi), ... \forall i \in NodeSet \therefore d(\xi, \mathbf{w_i}) = \|\xi - \mathbf{w_i}\|^2 \tag{4.2}$$

onde, *NodeSet* é o conjunto dos nodos.
Incremente o contador de vitórias de unidade vencedora $d_{s_1} = d_{s_1} + 1$
4. Insira uma conexão entre s_1 e s_2 se não existir.
5. Calcule o campo receptivo de s_1:

$$r_{s_1} = \|\mathbf{w_{s_1}} - \mathbf{w_{s_2}}\| \tag{4.3}$$

6. Calcule a atividade de s_1 por meio de uma função de base radial:

$$a_{s_1} = \frac{e^{-\|\xi - \mathbf{w_{s_1}}\|}}{r_{s_1}} \tag{4.4}$$

7. Se $a_{s_1} < a_T$ e número de nodos do mapa < N_{max}

- Adicione um novo nodo com vetor de pesos $\mathbf{w_n} = \xi$
- Atualize o número de nodos $N = N + 1$
- Remova a conexão entre s_1 e s_2
- Calcule as distâncias $d(\mathbf{w_n}, \mathbf{w_{s_1}})$, $d(\mathbf{w_n}, \mathbf{w_{s_2}})$, $d(\mathbf{w_{s_1}}, \mathbf{w_{s_2}})$
- Insira conexão entre os nodos com as duas menores distâncias

Caso contrário, atualize os pesos do vencedor:

$$\Delta \mathbf{w_{s_1}} = \rho(\xi - \mathbf{w_n}) \therefore \rho = \begin{cases} \varepsilon \rho_f^{\left(\frac{d_i}{d_m}\right)}, d_i \leq d_m \\ \varepsilon \rho_f, d_i > d_m \end{cases} \tag{4.5}$$

8. Atualize o número de interações ($t = t + 1$) e se um dos critério de parada da Equação 4.6 ou o número máximo de iterações for satisfeito termine o programa.

$$deslocamento = \frac{1}{N} \sum_{i=0}^{N-1} \|\mathbf{w_i}(t) - \mathbf{w_i}(t+1)\|^2 \leq deslocamento_{min} \tag{4.6}$$

9. retornar ao passo 2

No processo de quantização de cores, sair de alguns centenas de milhares de cores para algo entorno de um centena é um resultado inquestionavelmente bom e nesta função o LARFSOM atinge seus objetivos (Costa, 2007). É capaz de prover uma redução de cores de ordens de grandeza e ainda assim manter uma imagem com uma aparência aceitável. Outra característica muito boa do LARFSOM é seu número reduzido de parâmetros a serem ajustados. Praticamente o seu único parâmetro crítico é o a_T, os demais parâmetros não produzem forte impacto sobre os resultados do agrupamento. O a_T é um limiar que define se um novo nodo deve ser inserido, para um determinado ajuste de ordem de grandeza dos nodos presentes no mapa. Entretanto, o controle se torna mais complicado se desejarmos ajustar o mapa de modo a possuir poucos nodos, da ordem da dezena. Isto se deve ao fato que o a_T é um limiar que atua sobre um termo que varia exponencialmente com o erro da distância do nodo vencedor para a amostra e com o inverso da distância entre o primeiro e o segundo vencedor (Equação 4.4). Em nuvens de pontos muito irregulares em relação a distribuição dos nodos, o que normalmente acontece quando há poucos nodos, as distâncias das amostras para os nodos são muito maiores ou muito menores que a unidade o que faz o termo exponencial dominar a Equação 4.4, propiciar a inserção de um número muito grande de nodos, ou produzir uma inserção de um número pequeno de nodos. Devido a esta característica indesejável do LARFSOM, a utilização deste mapa com o modelo proposto deve ser acompanhada de um ajuste cuidadoso do a_T. Duas variações do LARFSOM foram propostas para reduzir a limitações e possibilitar a utilização do mapa tanto no modelo proposto para segmentação como em qualquer outro problema de categorização que necessite de baixo número de nodos. As variações propostas do LARFSOM podem ser analisadas nos Algoritmos 6 e no 7.

A primeira variação do LARFSOM, o LARF2, é muito semelhante à implementação original. A diferença nesta variação é que o calculo da atividade dos nodos do mapa (a_{s_1}) não é ponderada pelo campo receptivo entre os nodos s_1 e s_2. Desta forma o limiar de atividade (a_T) se torna uma métrica exponencial da distância admissível entre nodos e amostras de trei-

namento. Com esta modificação e com algum conhecimento a priori sobre as distâncias médias entre os dados de treinamento se torna mais fácil estimar o termo a_T que controla o crescimento do mapa. Se for desejado, por exemplo, que o erro máxima entre um nodo e uma amostra de treinamento ($\|\xi - \mathbf{w_{s_1}}\|$) seja d, pode-se utilizar um a_T tal que $a_T = e^{-d}$. Deste modo, sempre que uma amostra (ξ) estiver afastado a uma distância maior que d do nodo mais próximo (s_1) será inserido um novo nodo no local da amostra. No caso do LARFSOM original esta previsão para a máxima distância d admissível entre um nodo e uma amostra não é possível por causa da ponderação do campo receptivo (r_{s_1}, no Algoritmo 5).

Algoritmo 6 LARFSOM Generalizado Variação LARF2

1. Inicialização de parâmetros: modulador de taxa de aprendizagem do vencedor (ε_b); modulador de taxa de aprendizagem da vizinhança (ε_n); limiar de atividade (a_T); número máximo de nodos (N_{max}); Idade máxima de uma aresta (age_{max}); iteração de tempo ($t = 0$);tempo máximo (t_{max}); deslocamento mínimo ($deslocamento_{min}$); e o número de unidades iniciais ($N = 2$), que devem ser conectadas e cujo os pesos podem ser copiados de dois vetores, escolhido aleatoriamente, do conjunto de dados de entrada.
2. Apresente um vetor de dados escolhido aleatoriamente, ξ, como entrada à rede.
3. Encontre os dois nodos $\{\mathbf{w_{s_1}}, \mathbf{w_{s_2}}\}$ com as menores distância euclidianas em relação ao vetor de entrada:

$$d(\mathbf{w_{s_1}}, \xi) \leq d(\mathbf{w_{s_2}}, \xi) \leq d(\mathbf{w_i}, \xi), ... \forall i \in NodeSet \therefore d(\xi, \mathbf{w_i}) = \|\xi - \mathbf{w_i}\|^2 \tag{4.7}$$

onde, *NodeSet* é o conjunto dos nodos.
4. Insira uma conexão entre s_1 e s_2 se não existir.
5. Calcule a atividade de s_1 por meio de uma função de base radial:

$$a_{s_1} = e^{-\|\xi - \mathbf{w_{s_1}}\|} \tag{4.8}$$

6. Se a_{s_1} for maior que a_T e o número de nodos do mapa $< N_{max}$ criar um novo nodo s_n, onde $\mathbf{w_{s_n}} = \xi$, e conecte-o a s_1 e s_2 caso contrário atualize a posição do nodo vencedor por um fator e_b(Equação 4.9), dos seus vizinho por um fator de e_n (Equação 4.10) e incremente a *age* em 1 de todos as arestas que emanam do nodo vencedor.

$$\mathbf{w_{s_1}}(t+1) = \mathbf{w_{s_1}}(t) + \varepsilon_b(\xi - \mathbf{w_{s_1}}(t)) \tag{4.9}$$

$$\mathbf{w_n}(t+1) = \mathbf{w_n}(t) + \varepsilon_n(\xi - \mathbf{w_n}(t)) \tag{4.10}$$

7. Se houver alguma aresta com *age* maior que age_{max} remova esta aresta.
8. Incremente t em 1 e se um dos critério de parada da Equação 4.11 ou o número máximo de iterações for satisfeito termine o programa.

$$deslocamento = \frac{1}{N}\sum_{i=0}^{N-1} \|\mathbf{w_i}(t) - \mathbf{w_i}(t+1)\|^2 \leq deslocamento_{min} \tag{4.11}$$

9. Remova os nodos desconectados e retornar ao passo 2.

A segunda variação do LARFSOM, LARF22, contem menos semelhanças como o modelo original. Neste, o mapa não calcula nenhum tipo de atividade para cada nodo ou campo receptivo, ao invés é calculada a distância Euclidiana (d_{s_1}) diretamente entre o primeiro nodo mais próximo (s_1) e a amostra (ξ). Caso esta distância seja superior a uma distância máxima admissível (d_{max}) o mapa insere um novo nodo s_n no local da amostra, caso contrário o primeiro e o segundo nodos mais próximos da amostra são deslocados em relação a esta seguindo uma regra de atualização de pesos semelhante a utilizada no algoritmo do mapa GNG. A proposta desta

variação é forçar o mapa a manter os nodos a distâncias relativamente constantes entre si e não muito superiores a d_{max}, forçando o mapa a ter uma distribuição mais uniforme dos centros.

Algoritmo 7 LARFSOM Generalizado Variação LARF22

1. Inicialização de parâmetros: modulador de taxa de aprendizagem do vencedor (ε_b); modulador de taxa de aprendizagem da vizinhança (ε_n); distância máxima entre nodos (d_{max}); número máximo de nodos (N_{max}); Idade máxima de uma aresta (age_{max}); iteração de tempo ($t = 0$);tempo máximo (t_{max}); deslocamento mínimo ($deslocamento_{min}$); e o número de unidades iniciais ($N = 2$), que devem ser conectadas e cujo os pesos podem ser copiados de dois vetores, escolhido aleatoriamente, do conjunto de dados de entrada.
2. Apresente um vetor de dados escolhido aleatoriamente, ξ, como entrada à rede.
3. Encontre os dois nodos $\{\mathbf{w_{s_1}}, \mathbf{w_{s_2}}\}$ com as menores distância euclidianas em relação ao vetor de entrada:

$$d(\mathbf{w_{s_1}}, \xi) \leq d(\mathbf{w_{s_2}}, \xi) \leq d(\mathbf{w_i}, \xi), ... \forall i \in NodeSet \therefore d(\xi, \mathbf{w_i}) = \|\xi - \mathbf{w_i}\|^2 \tag{4.12}$$

Onde, *NodeSet* é o conjunto dos nodos.
4. Insira uma conexão entre s_1 e s_2 se não existir.
5. Calcule a distância Euclidiana entre o primeiro vencedor s_1 e a amostra ξ:

$$d_{s_1} = \|\mathbf{w_{s_1}} - \xi\| \tag{4.13}$$

6. Se d_{s_1} for maior que d_{max} e o número de nodos do mapa $< N_{max}$ criar um novo nodo s_n, onde $\mathbf{w_{s_n}} = \xi$, e conecte-o a s_1 e s_2 caso contrário atualize a posição do nodo vencedor por um fator e_b(Equação 4.14), dos seus vizinho por um fator de e_n (Equação 4.15) e incremente a *age* em 1 de todos as arestas que emanam do nodo vencedor.

$$\mathbf{w_{s_1}}(t+1) = \mathbf{w_{s_1}}(t) + \varepsilon_b(\xi - \mathbf{w_{s_1}}(t)) \tag{4.14}$$

$$\mathbf{w_n}(t+1) = \mathbf{w_n}(t) + \varepsilon_n(\xi - \mathbf{w_n}(t)) \tag{4.15}$$

7. Se houver alguma aresta com *age* maior que age_{max} remova esta aresta.
8. Incrementar t em 1 e se um dos critério de parada da Equação 4.16 ou o número máximo de iterações for satisfeito termina o programa.

$$deslocamento = \frac{1}{N}\sum_{i=0}^{N-1} \|\mathbf{w_i}(t) - \mathbf{w_i}(t+1)\|^2 \leq deslocamento_{min} \tag{4.16}$$

9. retornar ao passo 2

4.3 Apresentação Formal da Solução

O modelo proposto para segmentação de imagens por discriminação de áreas está mostrado no Algoritmo 8. O passo 1, determina o nível de detalhes desejados para a segmentação. É uma operação realizada por um filtro gaussiano passa-baixa. O passo 2, realiza a amostragem de características presentes na imagem teste ($im_{M\times N}$) utilizando o descritor de características D_x e a janela de segmentação de área com dimensões L_x e L_y deslocada de intervalos regulares δ_j no eixo x e δ_i no eixo y da imagem teste. Os vetores de amostra $\mathbf{v_{D_x}}$ são armazenadas em uma lista A. No passo 3, os vetores de amostras têm sua média removida e são reescalonados pelo desvio padrão em relação a cada dimensão dos vetores $\mathbf{v_{D_x}}$. Este é um procedimento "padrão" de pré-processamento de dados muito recomendado por alguns autores (Haykin, 1999). Sua principal finalidade é tornar o desempenho da rede neural menos susceptível a variações numéricas do conjunto de treinamento. Embora este pré-processamento possa ser dispensado quando utilizado, por exemplo, um SOM ou um GNG como mapa do sistema, este procedimento é importante para mapas como o LARFSOM. O passo 4, consiste no treinamento propriamente dito do mapa utilizando os vetores de características $\mathbf{v_{D_x}}$ já pré-processados. O critério de parada para o treinamento dependerá do tipo de mapa. Para um GNG, por exemplo, o critério de parada é atingir o número máximo de nodos (N_{max}). Para um mapa como o LARFSOM, o critério de parada pode ser o deslocamento dos centros abaixo de um mínimo (Algoritmo 5, Equação 4.6). Finalmente, no passo 5, todos os nodos do mapa são rotulados aleatoriamente com um valor numérico que identifica cada nodo univocamente. A partir destes rótulos numéricos o mapeamento inverso é realizado com a finalidade de projetar os agrupamentos realizados pelo mapa no espaço de características para o espaço dos pixels na imagem teste. Um dado pixel $im(i,j)$ da imagem teste receberá o mesmo rótulo numérico que possui o seu nodo correspondente mais próximo no espaço de características. Na imagem final, cada rótulo numérico (agrupamento) corresponde a uma "textura" vista por um sistema visual de pré-atenção artificial que utiliza o

descritor D_x para detectar um tipo de "texton". Para representação gráfica da segmentação os "rótulos numéricos" são substituídos por cores.

Algoritmo 8 Algoritmo de Segmentação de Imagens por Área

0. *Inicialização:* Entre com uma imagem $im_{M\times N}$, um descritor de imagem D_x, o número máximo de nodos N_{max}, passo de amostragem δ_i para o eixo y, passo de amostragem δ_j para o eixo x, tamanho da janela L_x ao longo do eixo x, tamanha da janela L_y ao longo do eixo y e uma escala s.
1. *Ajuste de escala:* Aplique um filtro gaussiano passa-baixa com parâmetro s sobre a imagem $im_{M\times N}$.
2. *Amostragem:* Aplique o descritor D_x sobre a imagem $im_{M\times N}$ com passo (δ_j, δ_i) e armazene o resultado $\mathbf{v_{D_x}}$ de cada amostragem em uma lista de vetores A.
3. *Homogeneização dos dados:* remova a média dos vetores contidos em A e divida os elementos dos vetores pelos seus respectivos desvios padrão.
4. *Treinamento:* Treine o *SOM* com os dados contidos na lista A.
5. *Coloração:* Atribua a cada nodo do mapa uma rotulação numérica única.
6. *Segmentação:* Atribua para cada um dos pares de pixel-texton presentes em A o rotulo numérico do nodo mais próximo.

O modelo proposto para segmentação de imagens por discriminação de bordas esta mostrado no Algoritmo 9. Seu funcionamento e parâmetros são semelhantes ao proposto para trabalhar com áreas. As principais diferenças residem na forma como é feita a amostragem dos vetores de características ($\mathbf{v_{D_x}}$), no passo 2, e na forma como o mapeamento inverso é realizado, do espaço de características para o espaço dos pixels, passo 5. Os passos 1, 3 e 4 são exatamente os mesmos realizados para a detecção de áreas. No passo 2, o descritor de características (D_x) é aplicado sobre cada um dos 8 vetores de direção da janela de detecção de borda (Figura 4.5). No passo 5, cada pixel $im(i, j)$ da imagem teste é substituído por nível de cinza de intensidade I. Onde o valor de I é a soma das distâncias entre os nodos vencedores em cada uma das direções da janela (ver Figura 4.5). Ou seja, nas direções $\{\mathbf{v_0}, \mathbf{v_4}\}$, $\{\mathbf{v_1}, \mathbf{v_5}\}$, $\{\mathbf{v_2}, \mathbf{v_6}\}$, $\{\mathbf{v_3}, \mathbf{v_7}\}$. Quanto maior forem as distâncias em cada uma das direções da janela maiores serão as diferenças de borda em torno do pixel $im(i, j)$. Ao final do passo 5, a imagem resultante está em tons de cinza, onde o nível de cinza está relacionado com a "intensidade da borda" ou quão acentuada é a borda. O passo 6, consiste na utilização de uma limiarização simples em um nível de cinza th sobre imagem em tons de cinza produzida no passo 5. Resultando em uma representação binária das fronteiras das regiões segmentadas na imagem teste.

Algoritmo 9 Algoritmo de Segmentação de Imagens por Borda

0. *Inicialização:* Entre com uma imagem $im_{M\times N}$, um descritor de texton D_x, o número máximo de neurônios N_{max}, passo de amostragem δ_i para o eixo y, passo de amostragem δ_j para o eixo x, tamanho da janela L e uma escala s, limiar de detecção de borda th.
1. *Ajuste de escala:* Aplique um filtro gaussiano passa-baixa com parâmetro s sobre a imagem $im_{M\times N}$.
2. *Amostragem:* Aplique o descritor D_x sobre cada um dos 8 vetores de pixels produzidos pela janela de borda com passos de amostragem (δ_j, δ_i) e armazenar o resultado de cada aplicação em uma lista de pontos A
3. *Homogeneização dos dados:* remova a média dos vetores contidos em A e dividir os elementos dos vetores pelos seus respectivos desvios padrão.
4. *Treinamento:* Treine o mapa *som* com os dados contidos na lista A.
5. *Segmentação:* Atribua para cada um dos pares de pixel-texton presentes em A a soma das distâncias entre os nodos vencedores em cada uma das direções (ver Figura 4.5) da janela.
6. *Limiarização:* Aplique uma limiarização na imagem final com limiar igual a th.

4.4 Resultados Esperados

O modelo proposto de segmentação visa atender a uma ampla gama de necessidade de segmentação, quer seja por área, quer seja por borda, utilizando descritores de texturas simples ou complexos. Entretanto, o comportamento e a eficiência com que o modelo irá segmentar uma dada imagem dependerá radicalmente do descritor utilizado. Os descritores atuam como um elemento chave no processo de segmentação do modelo proposto. As expectativas para este modelo são atender com sucesso os seguintes desafio de segmentação de imagens:

- Realizar segmentação de textura com um número pré-determinado de regiões (utilizando o GNG);
- Realizar segmentação de textura sem nenhuma restrição em relação ao número de regiões a segmentar (utilizando o LARFSOM);
- Realizar segmentação de textura por borda ou por área.

CAPÍTULO 5

Experimentos Realizados

Este capítulo apresenta na Seção 5.1 os resultados de testes comparativos de agrupamento com cinco modelos de mapas auto-organizáveis (GNG, GWR, LARFSOM, LARF2, LARF22). Os testes com os mapas foram realizados com o objetivo de identificar características relevantes para o modelo proposto de segmentação de textura. Na Seção 5.2, são descritos os objetivos dos testes com o modelo de segmentação proposto e algumas questões relevantes para a análise dos resultados. Na Seção 5.3, é descrito o arranjo experimental dos testes comparativos do modelo proposto. Nas Seções 5.6.1 e 5.6.2 são apresentados, respectivamente, uma análise visual dos resultados dos experimentos e uma análise quantitativa do desempenho do modelo em segmentação de imagens. Os experimentos utilizaram quatro descritores (ver Seção 4.1.2), quatro tipos de mapas (dos cinco analisados) e três métricas de comparação de desempenho. A título de comparação com o desempenho dos mapas, o k-means (Richard O. Duda & Stork, 2000) foi incluído no conjunto de categorizadores testados com o sistema proposto de segmentação.

O modelo de segmentação, os mapas auto-organizáveis, o k-means, os tipos de janelas de amostragem e os descritores de imagem utilizados nesta dissertação foram implementados utilizando orientação a objeto em linguagem C++ e em ambiente GNU/Linux. A implementação dos mapas e dos descritores foi realizada como módulos do sistema de segmentação proposto. A implementação reproduz os diagramas de bloco das Figuras 4.1 e 4.2.

5.1 Estudo Comparativo de Modelos de Mapas Auto-organizáveis

Neste trabalho foram implementados cinco diferentes modelos de mapas auto-organizáveis e o algoritmo do k-means, todos podem ser utilizados com o modelo proposto de segmentação. Com o objetivo de testar o funcionamento e observar características relevantes presentes nos mapas, simulações *online* foram realizadas, permitindo observar o crescimento dos mapas sobre nuvens bidimensionais de dados (ver Figura 5.1). Os modelos também foram testados em relação ao erro não-supervisionado e em relação ao crescimento do número de nodos, ambos, em função do número de apresentações de amostras. As nuvens de pontos utilizadas nos testes foram formadas a partir de 12 distribuições gaussianas bidimensionais geradas artificialmente com média e desvio selecionados aleatoriamente (Figura 5.1). Cada nuvem utilizada nos testes possui 600 pontos, cada distribuição gaussiana contribuindo com 50 pontos. Os parâmetros dos mapas foram ajustados por um processo interativo composto por dez tentativas de ajustes, sendo selecionado o melhor conjunto de parâmetros para cada tipo de mapa. Na Figura 5.2(a), o gráfico mostra os valores médios de 10 instâncias (em cada instância uma nuvem diferente é gerada para cada mapa) do erro não-supervisionado em função da apresentação de novas amostras. Neste gráfico nota-se que todos os modelos apresentam um desempenho bastante próximo, mas as variações LARF2 e LARF22 apresentam um desempenho um pouco melhor. Uma maior diferenciação entre o comportamento dos mapas pode ser notada no gráfico 5.2(b) (também, com dez instâncias), que apresenta o crescimento do mapa em nodos em relação ao número de exemplos apresentados. O LARF2, que apresentou o menor erro (Figura 5.2(a)), alocou, praticamente, um nodo para cada exemplo apresentado. Revelando que o seu "bom" desempenho é decorrente de uma memorização dos pontos e não propriamente de um aprendizado da distribuição de probabilidade dos pontos. O LARF22 apresentou realmente um bom desempenho sendo "econômico" na alocação de novos nodos à medida que mais exemplos são apresentados. Uma afirmação definitiva sobre o melhor desempenho é difícil de ser feita,

principalmente, se for levado em consideração que os gráficos representam os desempenhos para apenas uma das várias possibilidades de configuração dos parâmetros de cada mapa. Entretanto, é possível observar no gráfico 5.2(b) um comportamento peculiar entre o GNG e os outros mapas. Os modelos de mapas que apresentam inserção de nodos baseados em métrica de erro não-supervisionado (GWR, LARFSOM, LARF2, LARF22) são mais "econômicos" em relação à inserção de novos nodos no mapa. A inserção de novos nodos nestes mapas apresenta uma forma tipo logarítmica em relação à apresentação de exemplos. Esta característica pode ser altamente desejável em um sistema de segmentação, pois, normalmente é desejado um número pequeno de agrupamentos (divisões das regiões da imagem). Logo, no caso da segmentação por área, se o número de classes a serem segmentadas em uma imagem não for conhecido a *priori* a melhor opção é deixar a escolha do número de classes a critério de um mapa como o GWR, LARFSOM ou suas variações. No caso do processo de segmentação por detecção de borda, estes mapas podem ser utilizados de modo irrestrito, já que neste modo de segmentação não há uma correspondência direta entre o número de nodos e o número de segmentos gerados. Em termos práticos, os resultados de segmentação por borda utilizando mapas com mais de 20 nodos sugerem não haver diferenças visuais significativas.

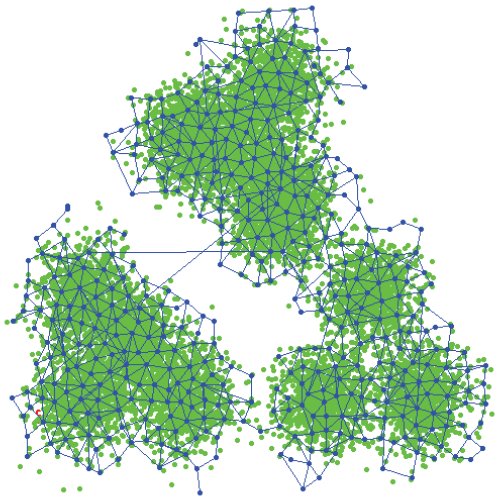
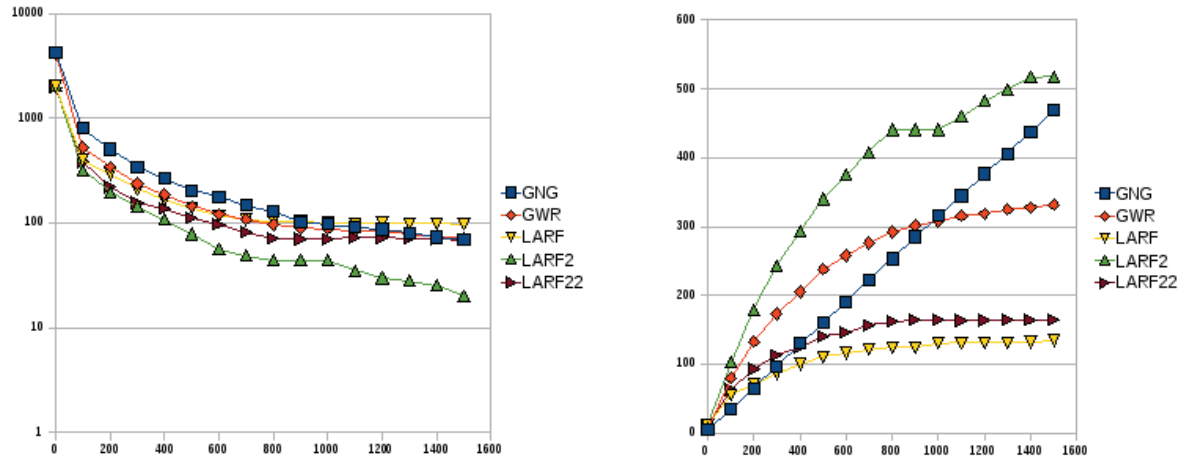


Figura 5.1 LARFSOM "cobrindo" a nuvem de pontos utilizada nos testes de desempenho dos os SOM's. A nuvem de pontos é formada por 600 pontos originados de 12 distribuições gaussianas com média e desvio selecionados aleatoriamente.



(a) Amostras × Erro

(b) Amostras × Nodos

Figura 5.2 Desempenho dos mapas implementados para utilização no modelo de segmentação em uma execução de teste. O gráfico 5.2(a) mostra o Erro não-supervisionado em função do número de amostras apresentadas ao mapa. O gráfico 5.2(b) mostra o número de nodos em função do número de amostras apresentadas ao mapa.

5.2 Objetivos dos Experimentos de Segmentação

Os testes experimentais realizados com o modelo proposto têm por principal finalidade mostrar a utilização dos seus vários modos de funcionamento, permitir comparações entre os vários descritores propostos e os mapas dinâmicos utilizados e permitir subsídios para traçar comparações objetivas e comparações visuais.

Como o modelo proposto é bastante geral e apresenta vários modos de funcionamento, a realização dos experimentos foi focada nas características mais relevantes do modelo. Se for levado em consideração os cinco SOMs apresentados e o k-means, o modo de segmentação por área ou borda, o modo de treinamento (abordagem 1 ou 2) e os quatro tipos de descritores haveria 96 possibilidades de configurações diferentes. Isto, sem incluir variações de parâmetros "menores" como os parâmetros de aprendizagem de cada mapa, tamanho das janelas, passo de amostragem e os parâmetros de alguns descritores, como o de Haralick. Devido a este grande número de possibilidades, testes com todas as possibilidades se tornam inviáveis. Assim, as principais características consideradas nos experimentos de segmentação foram divididas em duas fases de testes da seguinte forma:

- Primeira fase de testes:

  Realizar segmentação de textura com um número pré-determinado de regiões (utilizando o GNG);

  Realizar segmentação de textura sem nenhuma restrição em relação ao número de regiões a segmentar (utilizando o LARFSOM);

  Realizar segmentação de textura por borda ou por área utilizando as janelas propostas para esta finalidade (Seção 4.1.1).

- Segunda fase de testes:

Comparar o desempenho do sistema de segmentação utilizando categorizadores diferentes.

Comparar o desempenho do sistema de segmentação utilizando descritores diferentes.

A primeira fase, apresenta resultados quantitativos e qualitativos e se propõe a apresentar as diferenças de comportamento do sistema em relação ao tipo de categorizador escolhido e serve, também, de base para a segunda fase, um estudo mais detalhado do desempenho do sistema proposto em relação aos categorizadores e descritores utilizados. Na primeira fase foram utilizados apenas os mapas GNG e LARFSOM, que de acordo com os testes comparativos realizados na Seção 5.1 são os que apresentam comportamentos mais distintos entre si. As diferenças mais significativas entre os SOMs residiram na forma de crescimento ao longo do treinamento. O GNG apresentou um crescimento linear dos nodos ao longo do treinamento. O GWR, LARFSOM, LARF2 e LARF22 apresentaram crescimento logarítmico, alocando muitos nodos no início do treinamento e, posteriormente, tendendo para um determinado valor constante de nodos. Entre os descritores, foram utilizados na primeira fase de testes: MV (Média e Variância), MVSK (Média, Variância, *skewness* e *kurtosis*) e o descritor de Haralick. A segunda fase dos testes, apenas quantitativa, visou estudar o desempenho do sistema em relação a utilização dos categorizadores, independentemente dos descritores utilizados, e a utilização dos descritores, independentemente dos categorizadores utilizados. Nesta fase foram utilizados os mapas GNG, LARFSOM, LARF2 e LARF22 e o k-means. Os descritores da segunda fase de testes foram: MV, MVSK, histograma, Haralick em uma direção(HaD) e em quatro direções(Ha)[1].

[1] Ver a seção 5.5

5.3 Procedimentos Experimentais

No geral, independentemente da segmentação ser por borda ou área, utilizando brilho, cor ou textura, um problema que existe com a comparação de algoritmos de segmentação diz respeito às métricas de avaliação que variam muito e são pouco consensuais (Haindl, 1998). Um dos métodos de comparação de imagens segmentadas mais utilizados é a avaliação visual subjetiva (H. Zhang et al., 2008). Em função da subjetividade inerente da avaliação do que vem a ser uma boa ou má segmentação de imagem, a avaliação do modelo proposto, foi realizada utilizando comparações visuais e, também, comparações quantitativas. Nas avaliações quantitativas, foram utilizadas as métricas de erro de classificação (ME), não uniformidade (NU) e detecções corretas (CS). A métrica ME varia entre 0 e 1 e corresponde a uma estimativa do grau de erro na superposição das regiões segmentadas na imagem teste com as correspondentes regiões na imagem *groundtruth*. Quanto menor o seu valor melhor é a segmentação. A métrica NU é uma estimativa de homogeneidade das regiões segmentadas, varia de 0 a um valor próximo de 1. Normalmente segmentações com poucas regiões contínuas correspondem a um valor baixo com esta métrica. A métrica CS varia entre 0 e 1 e corresponde a quantidade de área da imagem teste que esta corretamente segmentada em relação ao *groundtruth*. Quanto maior o valor do CS melhor a segmentação. Estas três métricas distintas de comparação estão discutidas na Seção 3.4.

Nas análises visuais foram utilizadas imagens coloridas de cenários naturais do banco de imagens BSDS300 de Berkeley (Martin et al., 2001) e composições artificiais geradas a partir das imagens em tons de cinza do banco de imagens Brodatz (1966). Para as comparações quantitativas foram utilizadas apenas composições de imagens do Brodatz. Esta opção é em decorrência das imagens serem em tons de cinza e possuírem histograma equalizado, o que as tornam mais indicadas para realizar testes de segmentação por textura, onde características como, cor média e brilho não são desejadas, para não interferirem nos resultados da segmenta-

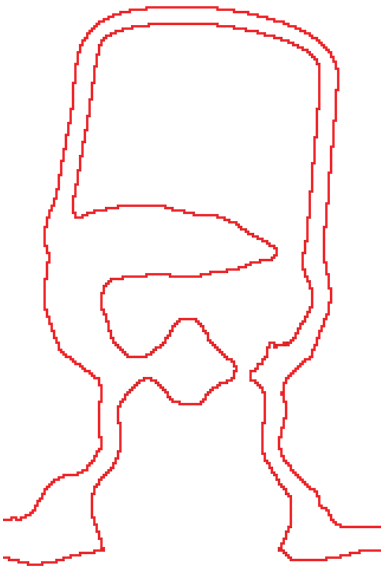
ção.

Levando-se em consideração que as segmentações de imagens por borda ou por área, podem frequentemente obter resultados semelhantes (ver Figuras 5.3(a), 5.3(b), 5.3(c) e 5.3(d)), os testes quantitativos foram realizados apenas com segmentações por área.

(a) Imagem Teste.

(b) Segmentação por borda.

(c) Segmentação por área.



(d) bordas da segmentação por área.

Figura 5.3 Em 5.3(a), a imagem número 189080 do banco de imagens BSDS300. A Figura 5.3(b) mostra o resultado de uma segmentação por borda aplicada sobre a Figura 5.3(a) utilizando o descritor MVSK e o mapa GNG com 3 nodos. A Figura 5.3(c) mostra o resultado de uma segmentação por área aplicada sobre a Figura 5.3(a) utilizando descritor de média e variância e o mapa GNG com 3 nodos. A Figura 5.3(d) são as bordas da Figura 5.3(c) obtidas utilizando o filtro de detecção de bordas Sobel. É possível notar claramente que as imagens 5.3(b) e 5.3(d) são muito semelhantes, embora tenham sido obtidas por processos completamente diferentes.

5.4 Parâmetros dos Mapas

Os parâmetros dos mapas GNG, LARFSOM, LARF2 e LARF22 utilizados no sistema de segmentação foram determinados através de um processo de tentativa e erro. Foram realizados dez testes de desempenho, utilizando as métricas da Seção 3.4, preliminares com cada mapa auto-organizável. Cada um dos dez testes com cada mapa auto-organizável possuía um conjunto próprio de parâmetros. Ao final dos teste, o melhor conjunto de parâmetros para cada mapas auto-organizável foi escolhido e fixado. Os parâmetros finais adotados para os mapas auto-organizável GNG, LARFSOM, LARF2 e LARF22 estão nas, respectivas, Tabelas 5.1, 5.2, 5.3 e 5.4. Todos as imagens segmentadas apresentadas nesta dissertação, assim como, todos os testes de desempenho quantitativos utilizaram estes parâmetros.

Tabela 5.1 Parâmetros do GNG utilizado no modelo proposto de segmentação.

Parâmetro	Valor
age_{max}	88
λ	200
α	0,05
β	0,00005
ε_n	0,05
ε_b	0,2

Tabela 5.2 Parâmetros do LARFSOM utilizado no modelo proposto de segmentação.

Parâmetro	Valor
a_T	0,0025
d_m	100
ε	0,2
ρ_f	0,1

Tabela 5.3 Parâmetros do LARF2 utilizado no modelo proposto de segmentação.

Parâmetro	Valor
age_{max}	200
a_T	0,1
ε_n	0,05
ε_b	0,005

Tabela 5.4 Parâmetros do LARF22 utilizado no modelo proposto de segmentação.

Parâmetro	Valor
age_{max}	200
a_T	0,1
ε_n	0,05
ε_b	0,005

5.5 Parâmetros do Descritor de Haralick

Nos testes de segmentação o descritor de Haralick foi utilizado com duas configurações de parâmetros. Na primeira configuração a matriz de co-ocorrência foi construída a partir dos quatro vizinhos mais próximos do pixel central (0,0), ou seja, utilizando os pontos vizinhos: (0,-1), (0,1), (1,0) e (-1,0). Uma ilustração da disposição destes pontos pode ser visualizada na Figura 5.4(a). A segunda configuração de parâmetros para o descritor de Haralick tem um arranjo assimétrica e utiliza apenas o ponto (1,1) para construir a matriz de co-ocorrência. Uma ilustração da segunda configuração utilizada pode ser visualizada na Figura 5.4(b). Na primeira fase de testes de segmentação foi utilizada apenas a primeira configuração(Ha), na segunda fase de testes tanto a configuração Ha como a HaD foram utilizadas para o descritor de Haralick.

(a) Ha (b) HaD

Figura 5.4 Em 5.4(a) e 5.4(b), representações das configurações utilizadas nos testes de segmentação envolvendo o descritor de Haralick. O valor (0,0) representa o pixel centra da máscara que esta sendo inspecionado, os demais pontos são referenciados em relação ao pixel central.

5.6 Primeira Fase de Testes

As composições de imagens utilizadas para avaliar o desempenho do sistema foram geradas artificialmente através de uma montagem em mosaico de quatro imagens sorteadas sem repetição do banco Brodatz. A Figura 5.5(a) exemplifica a configuração de uma composição de teste. A Figura 5.5(b) corresponde à segmentação "ideal" (*groundtruth*) da Figura 5.5(a). Cada cor é corresponde a região de "uma" textura.

(a) Imagem teste. (b) *Groundtruth* da imagem teste.

Figura 5.5 Em 5.5(a), um mosaico montado a partir de imagens de textura do banco Brodatz para testar o modelo de segmentação. A Figura 5.5(b) mostra o *groundtruth* corresponde da Figura 5.5(a).

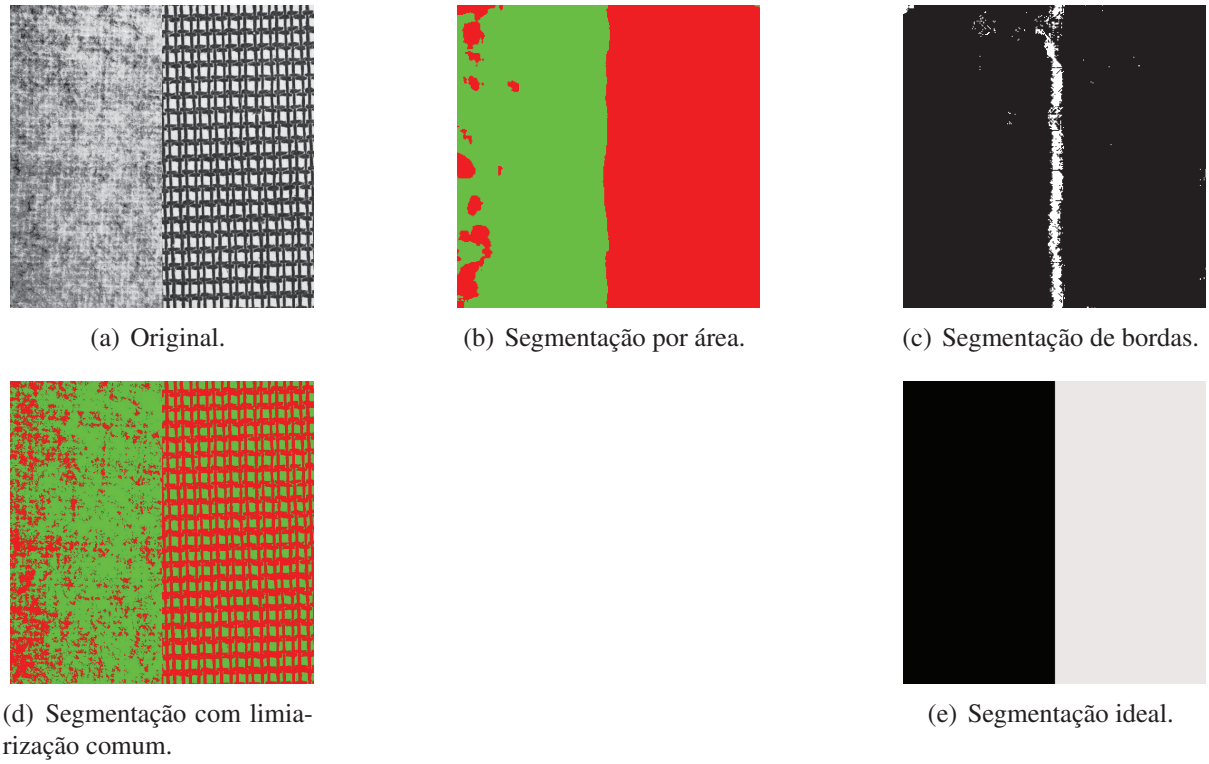
Cada uma das quatro subregiões do mosaico (Figura 5.5(a)) foram retiradas a partir de janelas de 256x256 pixels de uma posição aleatória no interior das imagens originais de 512x512. Este procedimento reduzir o tempo de processamento do sistema e aumenta o erro nas segmentações das imagens devido à redução na quantidade de informação total que o sistema utiliza para discriminar os padrões de texturas presentes nas imagens. Entretanto, a redução na resolução das imagens é realizada em todos os testes, não havendo nenhum conjunto de descritores e mapas auto-organizáveis beneficiado e, consequentemente, não gera prejuízo nas comparações.

O tamanho da janela utilizada nos experimentos foi de 50x50 pixels, um pouco abaixo de um quarto do tamanho de cada imagem de textura (256x256) nos mosaicos. Para as imagens coloridas do banco BSDS300, que não têm dimensões e resolução fixas, o tamanho das janelas foi ajustado interativamente para produzir um bom resultado em cada caso.

5.6.1 Análise Visual dos Experimentos

As Figuras de 5.6(a) a 5.6(e) mostram um exemplo de segmentação por área realizada pelo sistema proposto utilizando o descritor MV em conjunto com o mapa GNG com dois nodos. A Figura 5.6(a) mostra a imagem teste do sistema, que é um mosaico formado por duas texturas

(números d19 e d20) do Brodatz. A Figura 5.6(b) mostra a imagem teste segmentada por área na qual cada cor corresponde a uma "textura". Claramente nota-se que o sistema só discriminou duas "texturas" na imagem teste e que as regiões pintadas estão próximas das áreas correspondentes a das texturas originais, mostrada na Figura 5.6(e). A Figura 5.6(c) mostra a segmentação por borda das texturas presentes na Figura 5.6(a) utilizando-se, também, o mapa GNG e o descritor de MV. A Figura 5.6(e) é normalmente referenciada como "*groundtruth*". Esta imagem é o resultado da segmentação considerada ideal por algum referencial, normalmente o referencial humano. Logicamente que a Figura do *groundtruth* não esta disponível em um cenário geral. Na realidade, o *groundtruth* só tem a finalidade de servir de referencial para realizar a avaliação do grau de acerto do sistema. Não tem nenhuma influencia sobre o treinamento do sistema ou sobre o processo de discriminação. O sistema é, realmente, não-supervisionado. Um método de segmentação tradicional, que só leve em consideração a informação de brilho da imagem não conseguiria produzir uma separação das regiões da Figura 5.6(a). Como comparação, a Figura 5.6(d) é resultado de uma segmentação por limiarização realizada utilizando o software livre de edição de imagens GIMP. Nesta segmentação só foi considerada a informação de brilho da imagem. Assim, o resultado da segmentação divide a imagem em regiões claras e escuras e não em regiões de texturas diferentes. Comparando as segmentações das Figuras 5.6(d) e 5.6(b) nota-se que só as regiões da Figura 5.6(b) divide a imagem em praticamente duas áreas contínuas, de modo semelhante ao representado na imagem de *groundtruth*.



(a) Original. (b) Segmentação por área. (c) Segmentação de bordas.

(d) Segmentação com limiarização comum. (e) Segmentação ideal.

Figura 5.6 Comparativo de segmentação. A Figura 5.6(a) mostra a imagem teste. Na Figura 5.6(b), o resultado de uma segmentação de textura por área utilizando o descritor MV e o mapa auto-organizável GNG. Na Figura 5.6(c), o resultado de uma segmentação por borda utilizando o mesmo descritor da segmentação por área e o mesmo mapa auto-organizável. Na Figura 5.6(d), uma segmentação por limiarização tradicional realizado no programa de edição de imagens GIMP e colorizada com as mesmas cores utilizadas na Figura 5.6(b). A Figura 5.6(e) mostra a segmentação por área considerada "ideal" (*groundtruth*).

Comparando as Figuras 5.6(b) e 5.6(e), verifica-se algumas pequenas regiões interpretadas pelo sistema de maneira diferente em relação ao *groundtruth* (pequenas "ilhas" de cores diferentes da vizinhança). Como o descritor utilizado na segmentação da imagem em questão é bidimensional (média em x e variância em y), pode-se plotar o espaço dos dados do descritor em um gráfico $X \times Y$ simples para analisar visualmente a forma da nuvem de pontos. A Figura 5.6.1 mostra o espaço dos dados do descritor MV para a Figura 5.6(a), nota-se que existem dois agrupamentos predominantes em forma de "^". Cada um destes agrupamentos esta relacionado com uma das texturas presentes na imagem teste. O descritor de média e variância (MV) não foi suficientemente "potente" para produzir uma separação total entre os agrupamentos relaci-

onados com as texturas (há uma pequena superposição dos agrupamento na parte mais central da imagem). Esta superposição de agrupamentos é a responsável pelas pequenas "ilhas" de textura divergentes do *groundtruth* da imagem. É interessante notar que se fosse utilizada apenas a informação do valor médio do nível de cinza (eixo x na Figura 5.6.1), que essencialmente corresponde ao brilho da imagem, teríamos uma grande superposição dos agrupamentos e, consequentemente, um resultado ruim de segmentação. Isto está coerente com o resultado obtido na Figura 5.6(e), que utiliza um método que só leva em consideração o brilho da imagem. Este exemplo, ilustra bem a importância que tem a seleção de descritores adequados para realizar segmentação de textura. O caso também é análogo para a correspondente segmentação por borda.

No Apêndice A estão algumas imagens do banco BSDS300 segmentadas por borda e por área pelo modelo proposto. Em conjunto com as imagens segmentadas pelo sistema há, também, uma correspondente imagem segmentada por um humano e uma por limiarização utilizando o método de Otsu para prover alguns elementos de comparação visual.

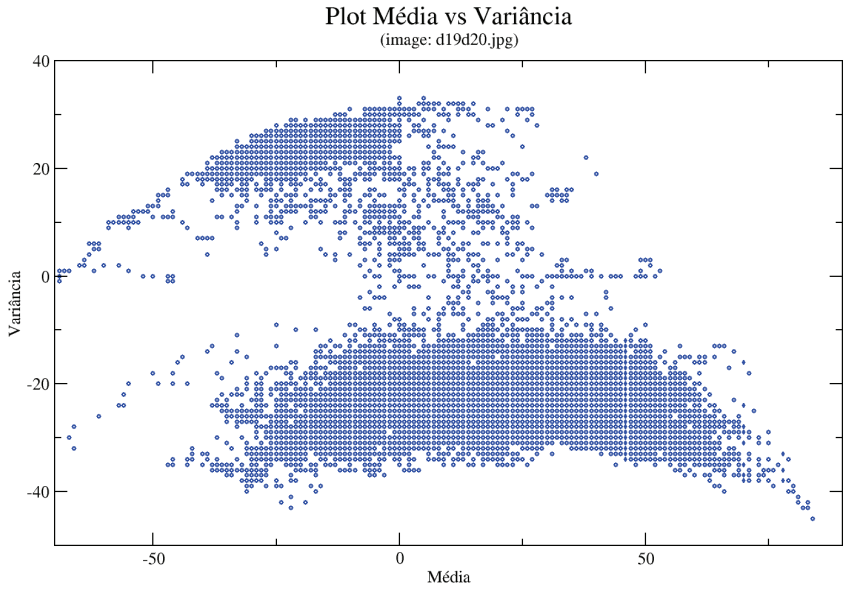


Figura 5.7 Gráfico da distribuição de pontos (m,v) amostrados da Figura 5.6(a) formando claramente dois agrupamentos. Nota-se, também, que há uma pequena superposição dos agrupamentos, sendo a causa da segmentação da Figura 5.6(e) não ter sido completamente bem sucedida.

5.6.2 Análise Quantitativa dos Experimentos

Na avaliação quantitativa da primeira fase, foram realizados um total de 10 testes com as combinações de descritores e mapas apresentados na Tabela 5.5. Em cada teste o sistema realizou a segmentação sobre um total de 56 mosaicos formados por quatro imagens diferentes selecionadas aleatoriamente e sem reposição do banco de imagens Brodatz, que possui 129 imagens de textura. Para o teste e para a finalidade de comparação, cada imagem do Brodatz foi considerada como sendo uma imagem composta por uma "única textura". Este pressuposto, apesar de ser muito forte, é adotado em muitos trabalhos. Os resultados finais para cada combinação de descritor-mapa são formados pelas médias e desvios dos resultados individuais calculados com as métricas selecionadas.

Tabela 5.5 Combinações de descritores e mapas utilizados nos testes de segmentação da primeira fase.

Tipo do Mapa Utilizado	Tipo de descritor utilizado	Variação da escala de cinza
GNG	descritor de Média e Variância	256
	descritor de MVSK	256
	descritor de Média e Variância	10
	descritor de MVSK	10
	descritor de Haralick	10
LARFSOM	descritor de Média e Variância	256
	descritor de MVSK	256
	descritor de Média e Variância	10
	descritor de MVSK	10
	descritor de Haralick	10

Os testes quantitativos da primeira fase tiveram uma subdivisão em dois grupos: um primeiro utilizando imagens com 256 níveis de cinza e um outro com imagens com níveis de cinza requantizados para 10 níveis. No primeiro subgrupo foram utilizados todos os descritores propostos exceto o descritor de Haralick, que exige um imagem com poucos níveis de cinza para permitir uma computação eficiente. No segundo grupo, com as imagens já requantizadas para

10 níveis de cinza, foram utilizados todos os descritores. A requantização dos níveis de cinza foi realizada por um mapa GNG. O primeiro subgrupo de teste teve como finalidade comparar os desempenhos dos descritores propostos utilizando a escala de cinza completa, ou seja 256 níveis. No segundo subgrupo, o objetivo foi permitir a comparação do descritor de Haralick com os demais descritores propostos utilizando a mesma quantidade de níveis de cinza.

Um resultado típico de segmentação produzido sobre, por exemplo, a Figura 5.5(a) esta nas Figuras 5.8(a) e 5.8(b). Partindo do resultado de várias segmentações como estas, com diferentes conjuntos de imagens, diferentes descritores e utilizando o *groundtruth* mostrado na Figura 5.5(b) foram construídas as Tabelas de 5.6 a 5.15 que contêm os valores médios, os respectivos desvios e os intervalos de confiança a 90% das métricas de avaliação de segmentação definidas na Seção 3.4).

(a) GNG (b) LARFSOM

Figura 5.8 Em 5.8(a), o resultado de uma segmentação por área aplicada sobre a Figura 5.5(a) utilizando descritor de média e variância e o mapa GNG com 4 nodos. A Figura 5.8(b) mostra o resultado de uma segmentação por área aplicada sobre a Figura 5.5(a) utilizando descritor de média e variância e o mapa LARFSOM.

Nas Tabelas de 5.6 a 5.15, o valor de N é o número de tons de cinza utilizando (256 ou 10), *nodos* é o número de nodos do mapa (4 para o GNG e sem restrições de nodos para o LARFSOM), período de amostragem utilizado nos testes foi fixado em δ_i=4 e δ_j=4 e o *Número de amostras* é a quantidade de composições de imagens teste que foram segmentadas para realizar os cálculos de média e desvio das métrica utilizadas. O $\overline{m}$ é o valor médio das métricas obtido após 56 segmentações em imagens diferentes e $\hat{s}$ é o valor do desvio padrão. O valor $\Delta_{90\%}$ corresponde a variação do intervalo de confiança de 90% em torno da média $\overline{m}$ e $\overline{m}_{Sup}$ e $\overline{m}_{Inf}$ correspondem aos respectivos limites superior e inferior do intervalo de confiança de 90% com respeito à média *overlinem* das métricas. Todos os mapas auto-organizáveis GNG

utilizaram 4 nodos em todos os testes quantitativos. O número de nodos para o mapa auto-organizável LARFSOM é dado em termos médios ($\overline{m}_{nodos}$) com o seu respectivo desvio padrão ($\hat{s}_{nodos}$), visto que não há um valor fixado máximo para o número de nodos deste mapa. Nestes testes quantitativos o parâmetro de escala s foi ajustado para zero, ou seja, foi ajustado de modo que a máxima quantidade de informação fosse utilizada pelos descritores. O tamanho da janela de amostragem foi ajustado para um valor um pouco abaixo da metade de cada padrão (com 256x256 pixel cada) de textura presente no mosaico, utilizando-se $L_x = 50$ e $L_y = 50$.

Os intervalos com 90% de confiança para as métricas de comparação utilizando a distribuição t-Student com um $t_{n-1,1-\frac{\alpha}{2}} = 1,67$ com $n = 56$ (número de medidas realizadas) e $\alpha = 0,1$ foram construídos para os resultados de cada teste. A expressão do intervalo de confiança é dada pela Equação 5.1.

$$(\overline{m} - n^{-1/2}\hat{s}t_{n-1,1-\frac{\alpha}{2}}, \overline{m} + n^{-1/2}\hat{s}t_{n-1,1-\frac{\alpha}{2}}) \tag{5.1}$$

onde $\overline{m}$ é a estimativa da média calculada a partir das amostras dos dados e $\hat{s}$ é a estimativa do desvio calculado, também, a partir das amostras dos dados.

Tabela 5.6 Desempenho de segmentação segundo as métricas da Seção 3.4 utilizando o mapa GNG com um número pré-determinado de regiões e o descritor MV (de média e variância).

Parâmetros Map=GNG MV N=256 nodos=4			
	ME	NU	CS
$\overline{m}$	0,154577	0,0965107	0,432947
$\hat{s}$	0,0381131	0,0538749	0,21515
$\Delta_{90\%}$	0,008516	0,012037	0,048071
$\overline{m}_{Sup}$	0,163093	0,108548	0,481018
$\overline{m}_{Inf}$	0,46061	0,084473	0,384876
Número de amostras			56

Tabela 5.7 Desempenho de segmentação segundo as métricas da Seção 3.4 utilizando o mapa LARFSOM e descritor MV (de média e variância).

Parâmetros Map=LARFSOM MV N=256 $\overline{m}_{nodos} = 5.07$ $\hat{s}_{nodos} = 1.40$			
	ME	NU	CS
$\overline{m}$	0,132365	0,0843681	0,492804
$\hat{s}$	0,0485201	0,0774225	0,225003
$\Delta_{90\%}$	0,010841	0,017299	0,050273
$\overline{m}_{Sup}$	0,143206	0,101667	0,543077
$\overline{m}_{Inf}$	0,121524	0,067070	0,442531
Número de amostras			56

Tabela 5.8 Desempenho de segmentação segundo as métricas da Seção 3.4 utilizando o mapa GNG com um número pré-determinado de regiões e descritor MVSK.

Parâmetros Map=GNG MVSK N=256 nodos=4			
	ME	NU	CS
$\overline{m}$	0,154043	0,0954578	0,416666
$\hat{s}$	0,0367227	0,0483396	0,213567
$\Delta_{90\%}$	0,008205	0,010801	0,047717
$\overline{m}_{Sup}$	0,162248	0,106258	0,464383
$\overline{m}_{Inf}$	0,145838	0,084657	0,368949
Número de amostras			56

Tabela 5.9 Desempenho de segmentação segundo as métricas da Seção 3.4 utilizando o mapa LARFSOM e descritor MVSK.

Parâmetros Map=LARFSOM MVSK N=256 $\overline{m}_{nodos}$=5.14 $\hat{s}_{nodos}$=1.51			
	ME	NU	CS
$\overline{m}$	0,135018	0,0825511	0,467789
$\hat{s}$	0,0468169	0,0774714	0,212294
$\Delta_{90\%}$	0,010460	0,017309	0,047433
$\overline{m}_{Sup}$	0,145478	0,099861	0,515222
$\overline{m}_{Inf}$	0,124558	0,065242	0,420356
Número de amostras			56

Tabela 5.10 Desempenho de segmentação segundo as métricas da Seção 3.4 utilizando o mapa GNG com um número pré-determinado de regiões e descritor MV.

Parâmetros Map=GNG MV-N N=10 nodos=4			
	ME	NU	CS
$\overline{m}$	0,160527	0,0999265	0,426287
$\hat{s}$	0,0368314	0,059265	0,205065
$\Delta_{90\%}$	0,008229	0,013242	0,045818
$\overline{m}_{Sup}$	0,168756	0,113168	0,472105
$\overline{m}_{Inf}$	0,152298	0,086685	0,380469
Número de amostras			56

Tabela 5.11 Desempenho de segmentação segundo as métricas da Seção 3.4 utilizando o mapa LARFSOM e descritor MV.

Parâmetros Map=LARFSOM MV-N N=10 $\overline{m}_{nodos}$=5.02 $\hat{s}_{nodos}$=1.52			
	ME	NU	CS
$\overline{m}$	0,137061	0,0874342	0,460273
$\hat{s}$	0,0456536	0,0753241	0,218115
$\Delta_{90\%}$	0,010200	0,016830	0,048734
$\overline{m}_{Sup}$	0,147261	0,104264	0,509007
$\overline{m}_{Inf}$	0,126861	0,070605	0,411539
Número de amostras			56

Tabela 5.12 Desempenho de segmentação segundo as métricas da Seção 3.4 utilizando o mapa GNG com um número pré-determinado de regiões e descritor MVSK.

Parâmetros Map=GNG MVSK-N N=10 nodos=4			
	ME	NU	CS
$\overline{m}$	0,146488	0,0894612	0,458808
$\hat{s}$	0,0398941	0,0385704	0,213868
$\Delta_{90\%}$	0,008914	0,008618	0,047785
$\overline{m}_{Sup}$	0,155402	0,098079	0,506593
$\overline{m}_{Inf}$	0,137574	0,080843	0,411023
Número de amostras			56

Tabela 5.13 Desempenho de segmentação segundo as métricas da Seção 3.4 utilizando o mapa LARFSOM e descritor MVSK.

Parâmetros Map=LARFSOM MVSK-N N=10 $\overline{m}_{nodos}$=4.89 $\hat{s}_{nodos}$=1.33			
	ME	NU	CS
$\overline{m}$	0,138959	0,091898	0,456896
$\hat{s}$	0,0477497	0,079257	0,227881
$\Delta_{90\%}$	0,010669	0,017708	0,050916
$\overline{m}_{Sup}$	0,149628	0,109606	0,507812
$\overline{m}_{Inf}$	0,128290	0,074190	0,405980
Número de amostras			56

Tabela 5.14 Desempenho de segmentação segundo as métricas da Seção 3.4 utilizando o mapa GNG com um número pré-determinado de regiões e descritor de Haralick.

Parâmetros Map=GNG Haralick N=10 nodos=4			
	ME	NU	CS
$\overline{m}$	0,151587	0,096354	0,454169
$\hat{s}$	0,0374887	0,0466917	0,199066
$\Delta_{90\%}$	0,008376	0,010432	0,044477
$\overline{m}_{Sup}$	0,159963	0,106786	0,498646
$\overline{m}_{Inf}$	0,143211	0,085922	0,409692
Número de amostras			56

Tabela 5.15 Desempenho de segmentação segundo as métricas da Seção 3.4 utilizando o mapa LARFSOM e descritor de Haralick.

Parâmetros Map=LARFSOM Haralick N=10 $\overline{m}_{nodos}$=4.21 $\hat{s}_{nodos}$=1.38			
	ME	NU	CS
$\overline{m}$	0,149284	0,137601	0,41428
$\hat{s}$	0,034534	0,0832688	0,166395
$\Delta_{90\%}$	0,007716	0,018605	0,037178
$\overline{m}_{Sup}$	0,157000	0,156206	0,451458
$\overline{m}_{Inf}$	0,141568	0,118996	0,377102
Número de amostras			56

Analisando-se os resultados desta primeira fase de testes pode-se notar que, no geral, o desempenho da segmentação esta mais relacionada aos descritores utilizados que aos mapas ou a quantização de níveis de cinza (256 ou 10 níveis de cinza). O sistema de segmentação, quando utilizando o LARFSOM, conseguiu realizar uma segmentação melhor (pelo critério de segmentações corretas, CS) que utilizando o GNG em 3 casos dos 5 analisados (Tabelas 5.6 e 5.7, 5.10 e 5.11 e 5.8 e 5.9). Entretanto, quando considerado o desempenho de segmentação com os intervalos de confiança para as médias, os mapas mostram-se estatisticamente equivalentes. Assim, independentemente do descritor utilizado, o desempenho geral do sistema utilizando-se o mapa LARFSOM foi equivalente ao utilizando o mapa GNG. Este é um resultado interessante se for levado em consideração que, a *priori*, o sistema de segmentação utilizando o mapa GNG ajustado para 4 nodos, possuía mais informações que o sistema utilizando o LARFSOM, pois o GNG já possuía o número correto de regiões a serem segmentadas e o LARFSOM teve que encontrar esse número.

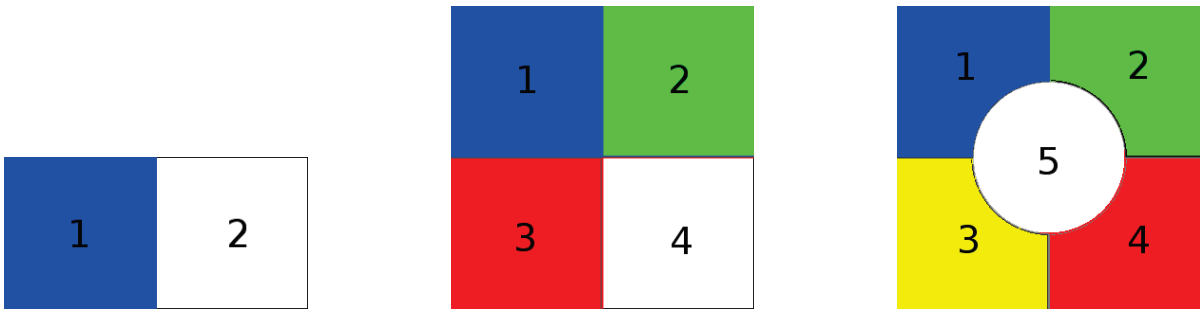
O sistema de segmentação utilizando a combinação de GNG-Haralick[2] conseguiu um resultado melhor, considerando os intervalos de confiança, que a combinação LARFSOM-Haralick[3] em relação a métrica de NU (não uniformidade).

Outro aspecto interessante revelado nos experimentos quantitativos, foi o descritor de Haralick ter apresentado uma performace semelhante a dos descritores MV e MVSK, considerados mais simples e menos eficientes. Uma explicação para estes resultados é que o descritor de Haralick têm parâmetros ajustáveis, vizinhança utilizada para construir a matriz de co-ocorrência, e isto pode torná-lo muito específico, sendo eficiente para detectar alguns tipos de padrões em detrimento de outros. Nos testes, a configuração do descritor de Haralick foi a Ha (ver Seções 5.5 e 2.2). O resultado com este descritor sugere que mais estudos sobre o desempenho e a avaliação do descritor deva ser realizado incluindo variações nos seus respectivos parâmetros de distância e ângulo para avaliar se os resultados se mantém com outros parâmetros δ_i e δ_j.

[2]Mapa GNG e descritor de Haralick.
[3]Mapa LARFSOM e descritor de Haralick.

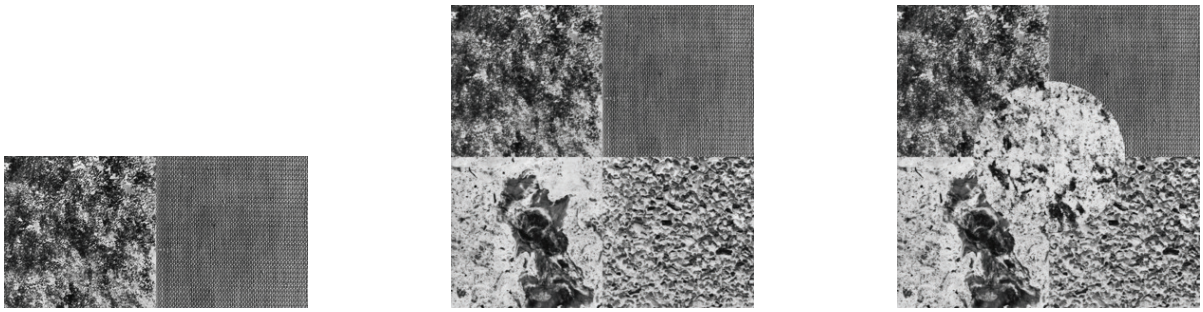
5.7 Segunda Fase de Testes

Na segunda fase de testes também foram utilizados mosaicos de texturas gerados artificialmente a partir de imagens do Brodatz, de modo análogo aos procedimentos da primeira fase. Entretanto foram utilizados três tipos diferentes de mosaicos, com duas, quatro e cinco texturas. As Figuras 5.9(a), 5.9(b) e 5.9(c) ilustram os groundtruths dos mosaicos e as Figuras 5.10(a), 5.10(b) e 5.10(c) apresentam alguns exemplos dos respectivos padrões utilizados. Na primeira fase dos testes as imagens do Brodatz utilizadas para compor os mosaicos eram escolhidas de modo completamente aleatório e sem reposição cada vez que o sistema utilizava um novo descritor, nesta segunda fase o procedimento de escolha destas imagens foi modificado para permitir uma reprodutibilidade maior. Assim, foi gerada uma lista de 59 amostras de imagens do Brodatz selecionadas aleatoriamente e, apartir desta lista, foram produzidos de modo sequencial os padrões com duas, quatro e cinco texturas. Os mesmos padrões foram utilizados nos testes com todos os descritores e todos os mapas. As combinações das texturas do Brodatz utilizadas para compor cada elemento dos conjuntos de imagens podem ser verificada nas tabelas B.1, B.2 e B.3 no apêndice B. O conjunto das imagens com duas texturas do Brodatz foi denominado de conjunto P2, do mesmo modo, foram criados os conjuntos P4 e P5, respectivamente, apartir das imagens com quatro e cinco padrões de textura do Brodatz.



(a) P2 (b) P4 (c) P5

Figura 5.9 Em 5.9(a), 5.9(b) e 5.9(c), as configurações do groundtruth dos mosaicos utilizados na segunda fase de testes.



(a) P2 (b) P4 (c) P5

Figura 5.10 Em 5.10(a), 5.10(b) e 5.10(c), exemplos dos mosaicos utilizados na segunda fase de testes.

5.7.1 Análise Quantitativa dos Experimentos

Na segunda fase dos testes de segmentação foram realizados todos os passos da primeira fase, mas utilizando uma quantidade maior de categorizadores, imagens de teste e descritores. A tabela 5.16 resume as combinações de categorizadores, conjuntos de teste e descritores utilizados nesta fase.

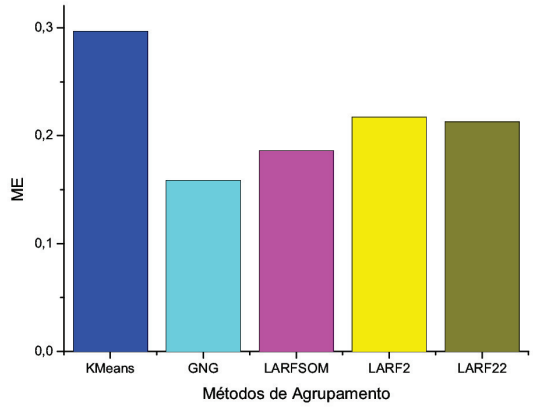
Tabela 5.16 Combinações de descritores e mapas utilizados na segunda fase de testes de segmentação.

Categorizadores	Tipo de descritor utilizado	Níveis de cinza	Conjuntos de Imagens
k-means	MV	256	P2, P4, P5
	MVSK	256	P2, P4, P5
	Histograma	256	P2, P4, P5
	Haralick	10	P2, P4, P5
	HaralickD	10	P2, P4, P5
GNG	MV	256	P2, P4, P5
	MVSK	256	P2, P4, P5
	Histograma	256	P2, P4, P5
	Haralick	10	P2, P4, P5
	HaralickD	10	P2, P4, P5
LARFSOM	MV	256	P2, P4, P5
	MVSK	256	P2, P4, P5
	Histograma	256	P2, P4, P5
	Haralick	10	P2, P4, P5
	HaralickD	10	P2, P4, P5
LARF2	MV	256	P2, P4, P5
	MVSK	256	P2, P4, P5
	Histograma	256	P2, P4, P5
	Haralick	10	P2, P4, P5
	HaralickD	10	P2, P4, P5
LARF22	MV	256	P2, P4, P5
	MVSK	256	P2, P4, P5
	Histograma	256	P2, P4, P5
	Haralick	10	P2, P4, P5
	HaralickD	10	P2, P4, P5

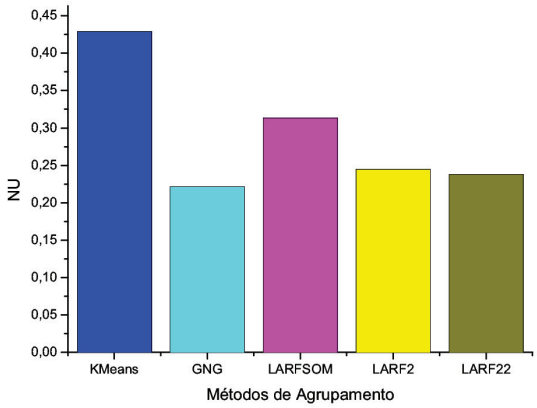
Nesta fase foi acrescentado um novo conjunto de parâmetros para o descritor de Haralick, além do conjunto de parâmetro utilizado na primeira fase (Ha). A motivação foi verificar se ocorreria alguma mudança significativa do desempenho do descritor de Haralick com a variação dos seus parâmetros. Na 5.16, onde há Haralick apenas, foi utilizado os mesmos parâmetros utilizados na primeira fase de testes (Ha). O HaralickD é o descritor de Haralick com um

conjunto novo de parâmetros (HaD), não utilizados na primeira fase de testes.

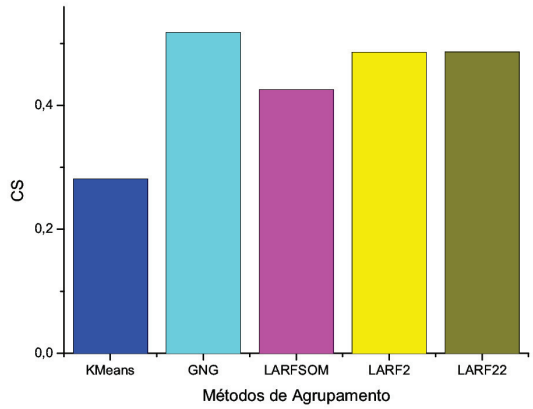
O desempenho geral do sistema de segmentação utilizando os diferentes métodos de categorização pode ser verificado nos gráficos das Figuras 5.11(a), 5.11(b), 5.11(c) considerando o resultado médio apresentado com todos os descritores utilizadas nos testes, a média de 845 casos para cada categorizador. O categorizador k-means apresenta o pior desempenho entre os categorizadores.



(a) ME



(b) NU



(c) CS

Figura 5.11 Em 5.11(a), 5.11(b) e 5.11(c), os resultados médios, considerando todos os descritores, utilizando, respectivamente, as métricas ME, NU e CS.

O tempo médio de treinamento dos categorizadores utilizados na segunda fase de testes para os padrões com 5 grupos de textura pode ser observado no gráfico da Figura 5.12. Todos os métodos utilizaram conjuntos de treinamento de 19600 amostras apresentadas até duas vezes, ou seja, 39200 apresentações por treinamento com 5 nodos. O categorizador k-means apresentou o melhor resultado em termos de tempo de execução.

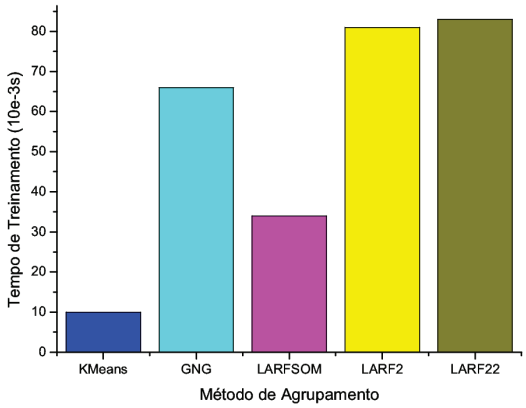
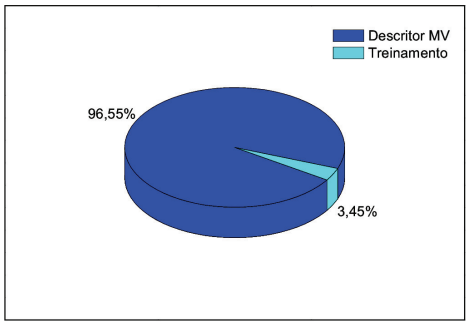
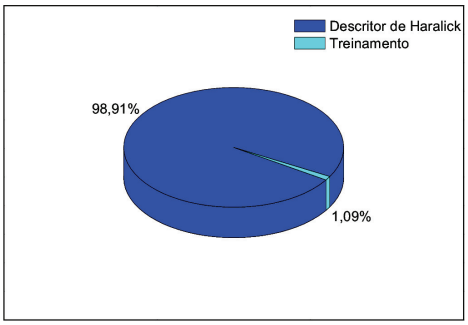


Figura 5.12 O tempo médio dos treinamentos dos cinco tipos de categorizadores utilizados na segunda fase de testes.

De acordo com a Figura 5.12,na pior das hipóteses o tempo de treinamento dos categorizadores é da ordem de 100ms, as Figuras 5.13(a) e 5.13(b) apresentam um comparativo entre o tempo de treinamento e o tempo total de processamento, respectivamente, do descritor mais rápido(ME) e do mais lento(Haralick) para a categorização de uma determinada imagem com 5 padrões de textura. Analisando estes gráficos nota-se que o tempo de treinamento é sempre menor que 5% do tempo total para realizar a categorização. Isto se deve ao fato das rotinas utilizadas pelos descritores, executadas pixel-a-pixel, dominarem em relação ao tempo de treinamento dos categorizadores. Deste modo, mesmo o categorizador k-means tendo apresentado um desempenho médio quase dez vezes mais rápido que o mapa auto-organizável mais lento, não contribui de maneira significativa no desempenho final do sistema de discriminação de textura.



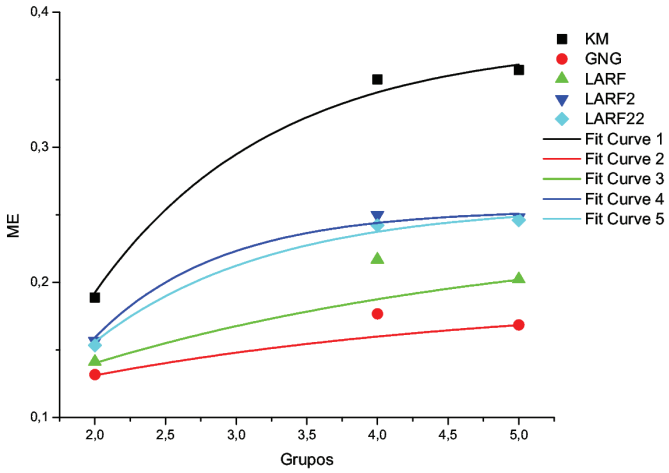
(a) MV



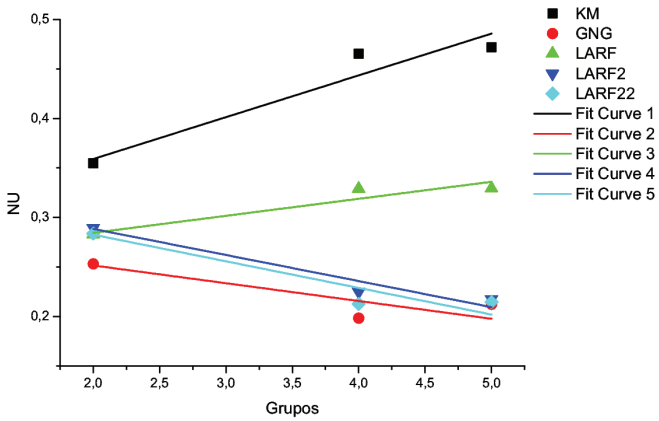
(b) Haralick

Figura 5.13 Em 5.13(a) e 5.13(b), os resultados médios dos tempos de treinamento em relação os tempos de processamento total utilizando, respectivamente, o descritor mais rápido e o mais lento.

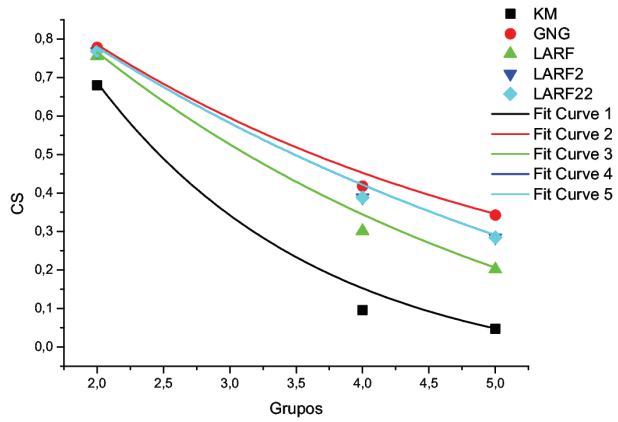
O desempenho do sistema também foi avaliado em relação ao número de texturas presentes em cada imagem segmentada. Os resultados médios, utilizando os cinco descritores propostos, dos categorizadores segundo as métricas de ME, NU e CS para padrões de teste com 2, 4 e 5 texturas diferentes podem ser visualizados nos gráficos das Figuras 5.14(a), 5.14(b) e 5.14(c).



(a) ME



(b) NU



(c) CS

Figura 5.14 Em 5.14(a), 5.14(b) e 5.14(c), os resultados médios, considerando todos os descritores, utilizando, respectivamente, as métricas ME, NU e CS para padrões de 2, 4 e 5 texturas.

Embora não seja possível observar uma tendência em relação ao critério de não uniformidade (NU, Figura 5.14(b)), de acordo com os gráficos das Figuras 5.14(a) e 5.14(c) é possível notar que existe um decaimento tipo exponencial do desempenho de categorização em relação ao número de texturas presentes nas imagens inspecionadas. A queda no desempenho de segmentação é esperada, pois o número de amostragens por imagem é fixo (19200 pontos de amostragem) e aumentando-se o número de regiões distintas em uma imagem acaba-se diminuindo o número de amostras por região e consequentemente degradando o conjunto de pontos utilizado para treinar os categorizadores.

Analisando o desempenho de segmentações corretas no gráfico da Figura 5.15 dos descritores utilizados, independentemente do método de categorização, é possível notar que o descritor que apresentou o melhor resultado foi o descritor de histograma. Curiosamente, o descritor de Haralick, nas duas configurações utilizadas nesta segunda fase de teste, apresentou os piores resultados, reproduzindo o desempenho obtido na primeira fase dos testes. Como dito anteriormente, este resultado pode ser creditado ao alto grau de especificidade do descritor de Haralick. Entretanto, com os resultados destes testes fica claro que este descritor não é uma boa escolha para sistemas de segmentação não-supervisionados que não utilizam um dicionário de texturas.

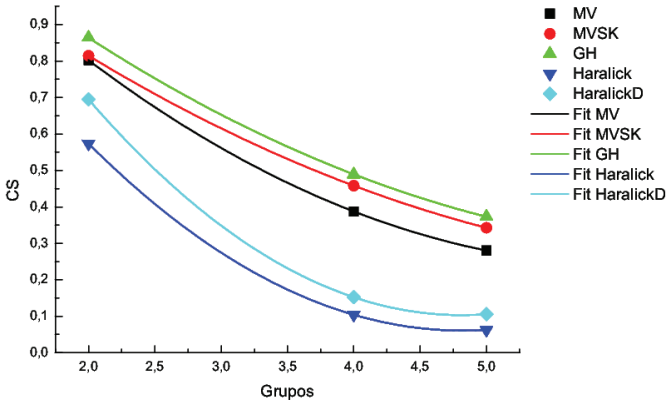


Figura 5.15 Comparativo das segmentações corretas em função dos descritores utilizados.

5.8 Discussão Final

A principal finalidade dos experimentos realizados neste seção foi mostrar os resultados de alguns modos de segmentação permitidos pelo sistema proposto, fornecer elementos visuais de comparação e realizar comparativos quantitativos. Em relação as comparações quantitativas os resultados mostraram que o sistema utilizando o mapas auto-organizáveis dinâmicos obtiveram um desempenho geral melhor que utilizando um categorizador clássico como o k-means, também foi mostrado que o desempenho do LARFSOM foi estatisticamente equivalente ao do GNG, mesmo sem fornecer o número de regiões corretas para o LARFSOM. Em relação aos descritores os resultados indicaram que, para o modelo proposto de segmentação, o descritor de Haralick não oferece um bom desempenho, principalmente quando lavado em consideração o seu tempo de processamento e sua necessidade de memória.

CAPÍTULO 6

Conclusões e Trabalhos Futuros

A utilização de algoritmos inteligentes para segmentação é um campo rico em pesquisas nos dias atuais. Diferentes sistemas são propostos na literatura, alguns apresentando resultados interessantes, contudo, entre as várias técnicas de segmentação por textura existentes não há uma proposta com uma estrutura padronizada capaz de modelar uma ampla faixa de necessidades diferentes em segmentação de imagens. Esta dissertação propõe uma estrutura geral, modular e bastante flexível para realizar segmentação de texturas em imagens.

Esta dissertação realiza uma revisão dos trabalhos envolvendo segmentação de imagens, reconhecimento e discriminação de textura e mapas auto-organizáveis. Entre as principais contribuições para a Ciência, propõe um sistema de segmentação de imagens por discriminação de texturas com inspiração no sistema visual humano de preatenção descrito por Julesz (1981). O sistema proposto possui uma arquitetura modular e pode ser configurado de diversos modos diferentes. O sistema se utiliza de características específicas de mapas auto-organizáveis como o GNG (Fritzke, 1995) e o LARFSOM (Costa, 2007) para realizar tipos diferentes de segmentações. O sistema pode ser utilizado para dividir uma determinada imagem em um número fixo de regiões ou em um número variável. Dependendo da configuração escolhida, o sistema proposto é capaz de realizar segmentação de imagem por área ou por borda. A segmentação realizada pelo sistema proposto pode, ainda, ter seu nível de detalhes ajustado, podendo prover imagens segmentadas em poucas regiões suaves e com características mais gerais ou prover imagens com muitas regiões e muitos detalhes.

Um novo tipo de janela em formato de estrela para detecção de borda em imagens é proposto para a utilização em sistemas de segmentação não-supervisionados, permitindo que um

sistema de segmentação, originalmente projetado para segmentar áreas, possa, com poucas modificações, ser utilizado para segmentar bordas. A utilização de tal janela dá flexibilidade ao sistema e permite realizar a detecção de bordas delimitadas por texturas e não apenas por brilho, como a maioria dos métodos de detecção de bordas utilizados.

Esta dissertação faz, também, uma revisão do algoritmo do LARFSOM e apresenta um algoritmo generalizado do LARFSOM que inclui limitação no crescimento de nodos e remove a regra de remoção de nodos do algoritmo original que não apresenta efeitos.

Através de um estudo preliminar de mapas auto-organizáveis para a utilização no sistema de segmentação proposto, foram criadas e testadas duas variações no algoritmo original do LARFSOM. As variações surgiram de um processo de análise do algoritmo do LARFSOM. Uma das variações propostas apresentou resultados preliminares promissores e será objeto de estudos futuros.

O sistema de segmentação de texturas proposto produziu resultados visuais de segmentação de imagens de cenários naturais, quando comparadas a segmentações realizadas por seres humanos, que se mostraram promissores. Diversas imagens do banco BSDS300, como exemplo de utilização do sistema de segmentação, foram segmentadas por borda e por área. Em alguns casos os resultados se aproximaram bastante das segmentações humanas.

Diversos experimentos foram apresentados e testes de desempenho foram realizados nos bancos de imagem Brodatz e BSDS300 utilizando diversas configurações do modelo proposto. Resultados com combinações de mapas e descritores diferentes são comparados entre si utilizando critérios estatísticos de avaliação.

Os resultados dos experimentos revelam algumas características não previstas entre os descritores implementados e os mapas. As expectativas que o descritor de Haralick apresentasse uma performasse média superior a dos descritores MV e MVSK não se confirmou. Nos experimentos, embora as segmentações com descritores diferentes tenham produzido resultados visuais diferentes, o desempenho médio dos descritores foi bastante semelhante. A utiliza-

ção do descritor de Haralick não conduziu a resultados melhores que descritores considerados mais simples e menos eficientes, como o MV. O descritor de Haralick tem parâmetros ajustáveis em ângulo e distância e isto pode torná-lo muito específico, sendo eficiente para detectar alguns tipos de padrões em detrimento de outros. Outro resultado interessante, é o fato do sistema de segmentação utilizando o LARFSOM ter apresentado resultados médios equivalentes aos utilizando o mapa GNG com o número correto de regiões já pré-ajustado. O LARFSOM apresentou um número variado de nodos (entre 3 e 10 nodos) para representar as regiões segmentadas, na maioria dos casos onde o número de nodos foi superior a 4 (o número de regiões do *groundtruth*) as regiões excedentes correspondiam a pequenas áreas das imagens testada. Foi constatado, também, que mais de 90% do tempo utilizado na segmentação é destinado ao processamento dos descritores, mesmo no caso de descritores simples como o de média e variância. Em consequência disto, a velocidade de treinamento dos categorizadores não se torna um fator crítico no desempenho do sistema. O sistema foi testado utilizando o categorizador k-means e mapas auto-organizáveis dinâmicos. Na comparação dos resultados, para imagens com duas regiões de texturas distintas o desempenho do k-means foi semelhante ao desempenho obtido com os mapas, mas a medida que o número de texturas por imagens aumenta o desempenho do k-means cai rapidamente se tornando inferior aos resultados obtidos com os mapas.

6.1 Trabalhos Futuros

Em trabalhos futuros os testes de segmentação de textura serão estendidos, utilizando os descritores implementados e utilizando novos descritores. A questão sobre o desempenho do descritor de Haralick será analisada novamente, realizando mais testes com diferentes parâmetros de ângulo e distância. Novas métricas de distância também serão implementadas para complementar as já utilizadas.

A base de imagens utilizadas nos testes de segmentação será ampliada e outras formas de mosaicos serão estudados para utilização nos teste, mosaicos com outras formas além da composição com duas, quatro e cinco texturas. Formas arredondadas, mais suaves ou mesmo de imagens naturais podem ser utilizadas.

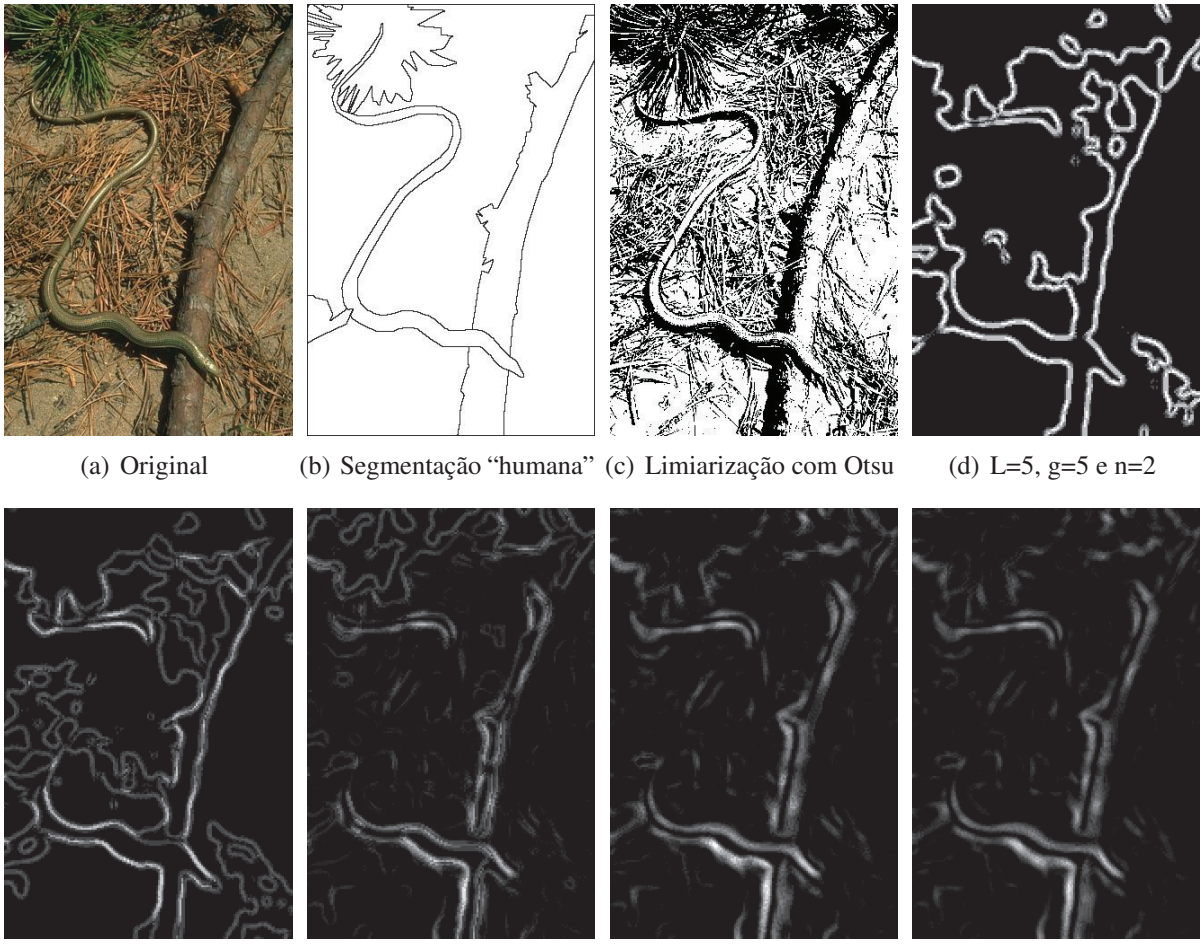
Um estudo preliminar para a implementação de descritores de textura utilizando *wavelets* para codificar textons específicos para determinados tipos de padrões mais complexos foi iniciado e espera-se ser terminado em trabalhos futuros. Estes trabalhos incluem implementar novos descritores utilizando *wavelets* bi-ortogonais que possam forma uma base para um espaço de texturas. Estes descritores *wavelets* não implicariam em nenhuma mudança na estrutura do sistema proposto de segmentação e trariam mais opções para sua utilização.

APÊNDICE A

Testes de Segmentação de Textura em Imagens Naturais

As imagens a seguir são alguns dos resultados de segmentação obtidos utilizando-se o sistema proposto para segmentar bordas e áreas em imagens naturais do banco BSDS300. Como elemento de referência visual em cada caso há um *groundtruth* referenciado por um humano e uma segmentação por limiarização utilizando o método de Otsu (1979).

A finalidade das imagens deste apêndice é complementar a Seção 5.6.1 e servir de referência visual para comparação e fornecer um melhor entendimento do modelo de segmentação de textura proposto e seus vários modos de funcionamento.

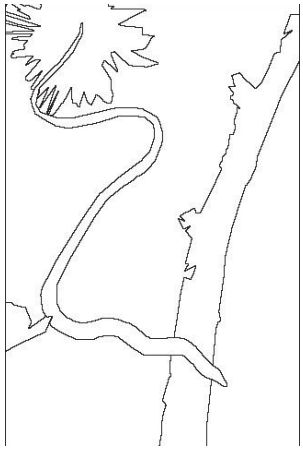


(a) Original (b) Segmentação “humana” (c) Limiarização com Otsu (d) L=5, g=5 e n=2

(e) L=5, g=5 e n=3 (f) L=5, g=5 e n=10 (g) L=5, g=5 e n=40 (h) L=5, g=5 e n=80

Figura A.1 As Figuras de A.1(a) à A.1(c) mostram, respectivamente, a imagem teste (número 175032, do banco BSDS300), a segmentação humana e a limiarização de Otsu. As Figuras de A.1(d) à A.1(h) mostram o resultado de diversas segmentações utilizando o Mapa GNG e o descritor MVSK. Ainda em relação as Figuras de A.1(d) à A.1(h), em sua legenda, L é o tamanho da janela de segmentação de bordas, g é a largura do filtro gaussiano, n é o número de nodos. Observa-se que quanto maior o número de nodos mais específico é o destaque sobre as bordas mais salientes, formando uma espécie de curva de nível. Onde, quanto maior for a intensidade do nível de cinza mais saliente é a “borda”.

(a) Original

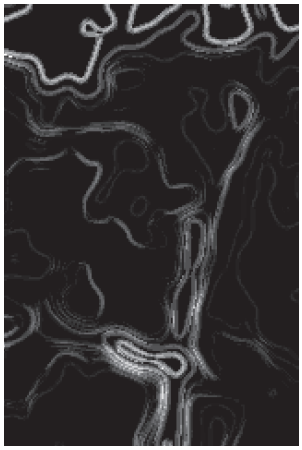


(b) Segmentação “humana”

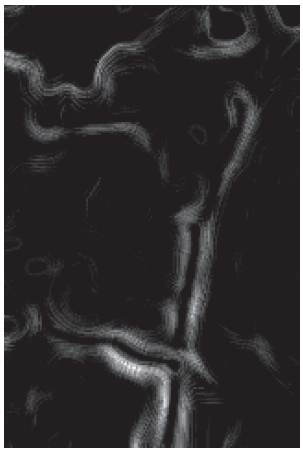
(c) Limiarização com Otsu

(d) L=5, g=10, n=2

(e) L=5, g=10, n=3



(f) L=5, g=10, n=10

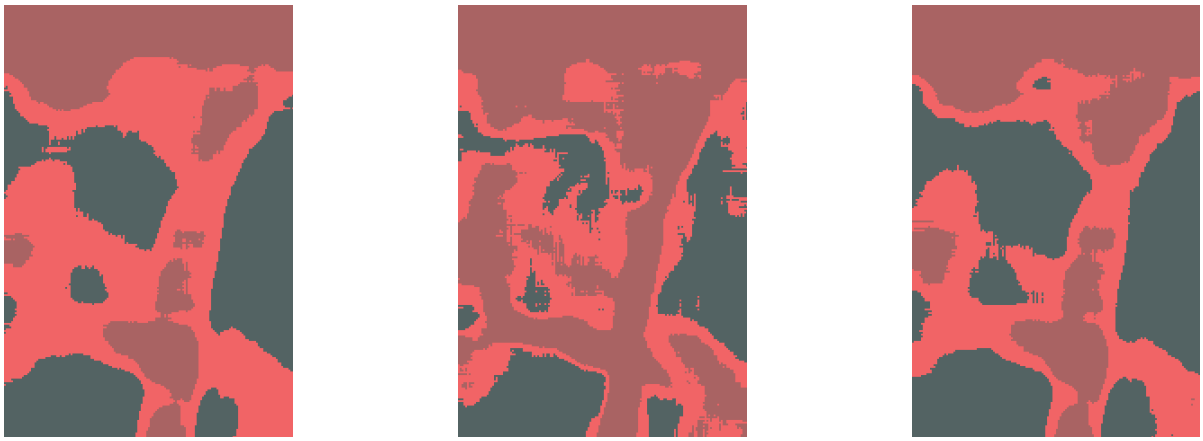


(g) L=5, g=10, n=40

(h) L=5, g=10, n=80

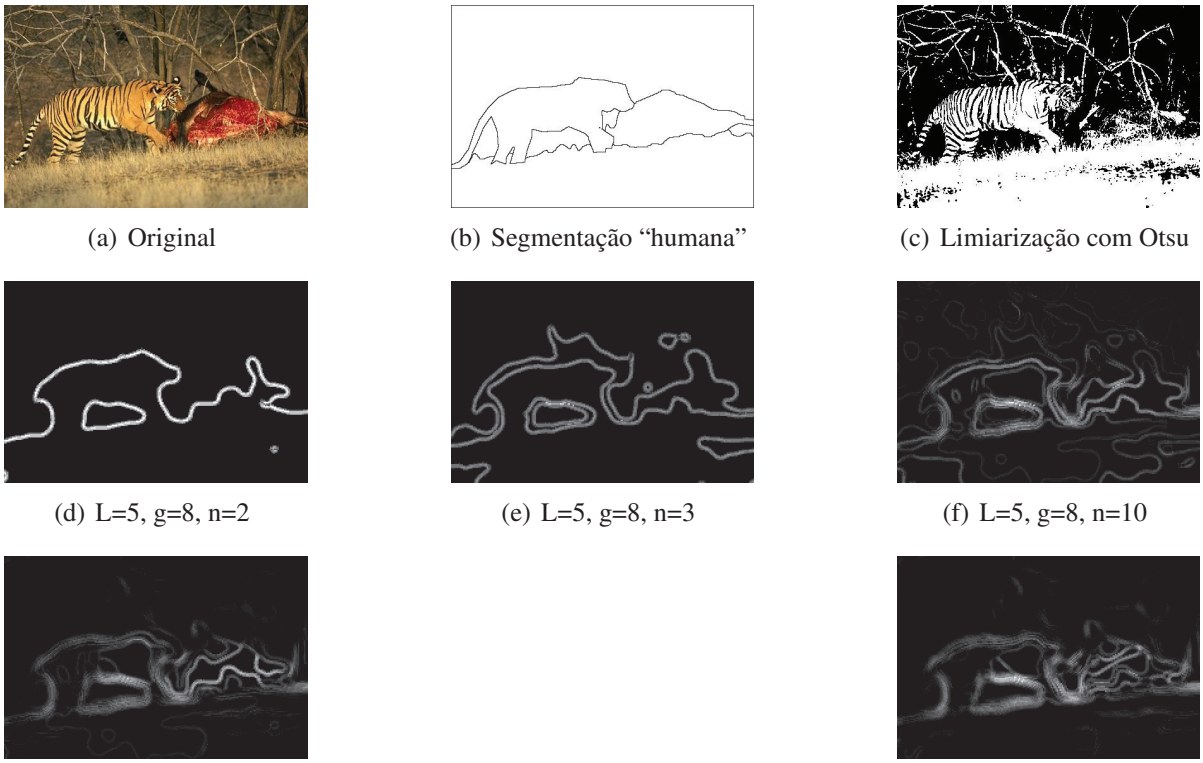
Figura A.2 As Figuras A.2(a) a A.2(h) são semelhantes ao conjunto de Figuras A.1(a) a A.1(h), mas o parâmetros do filtro de passa baixa (g) esta ajustado, neste caso, para 10.

(a) Original (b) Segmentação “humana” (c) Limiarização com Otsu (d) MV L=15, g=5, n=3



(e) MV L=30, g=5, n=3 (f) MVSK L=15, g=5, n=3 (g) MVSK L=30, g=5, n=3

Figura A.3 As Figuras de A.3(a) a A.3(c) mostram, respectivamente, a imagem teste, a segmentação humana e a limiarização de Otsu (1979). As Figuras de A.3(d) a A.3(g) mostram segmentações de textura por área da imagem teste com dois tipos diferentes de descritores (MV e MVSK) e dois tamanhos de janelas (15x15 e 30x30). Nas legendas das Figuras de A.3(d) à A.3(g), L é o tamanho da janela de segmentação de bordas, g é a largura do filtro gaussiano, n é o número de nodos.

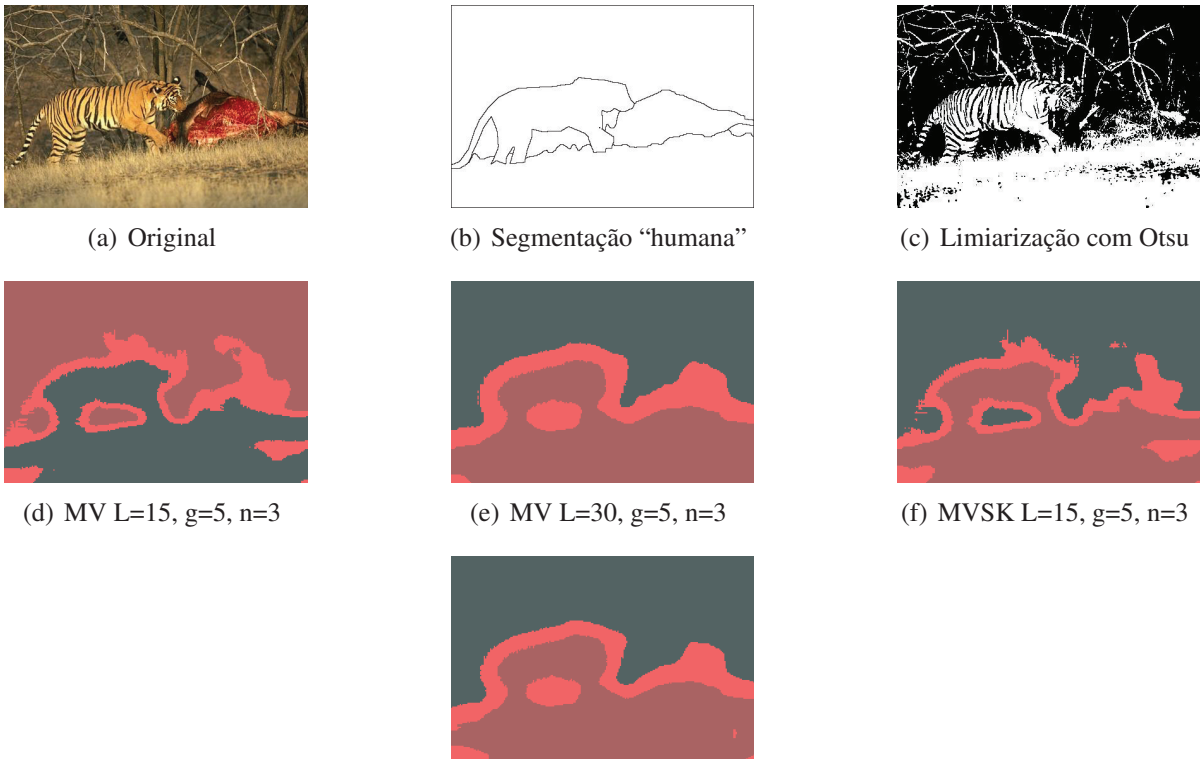


(a) Original (b) Segmentação “humana” (c) Limiarização com Otsu

(d) L=5, g=8, n=2 (e) L=5, g=8, n=3 (f) L=5, g=8, n=10

(g) L=5, g=8, n=40 (h) L=5, g=8, n=80

Figura A.4 A Figura A.4(a) é a imagem 187039 do banco BSDS300. As Figuras A.4(d) à A.4(h) são o resultado da segmentação por borda utilizando o descritor MVSK e o mapa GNG. Nas legendas das Figuras de A.4(d) à A.4(h), L é o tamanho da janelas de segmentação de bordas, g é a largura do filtro gaussiano, n é o número de neurônios. Os mesmos passos utilizado nas Figuras A.1(a) a A.1(c), utilizando outra imagem teste.

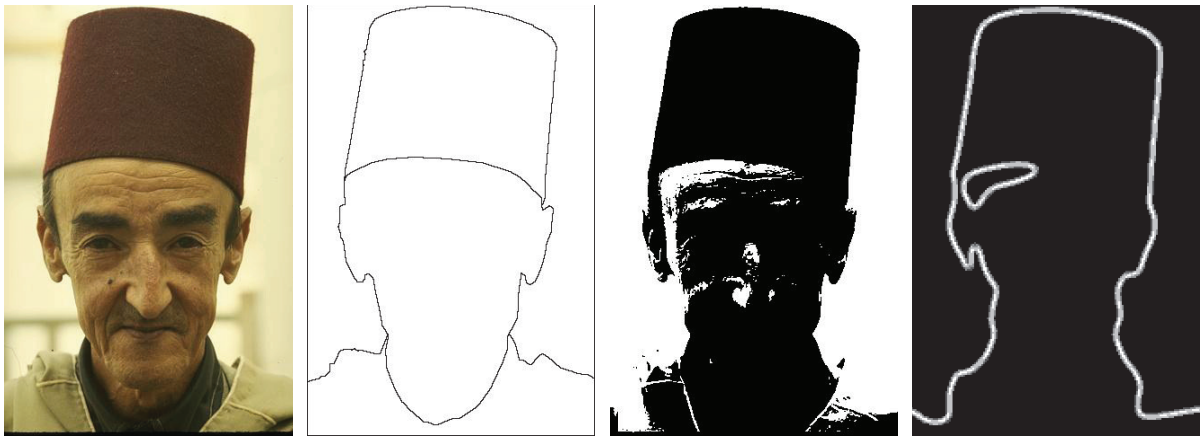


(a) Original (b) Segmentação “humana” (c) Limiarização com Otsu

(d) MV L=15, g=5, n=3 (e) MV L=30, g=5, n=3 (f) MVSK L=15, g=5, n=3

(g) MVSK L=30, g=5, n=3

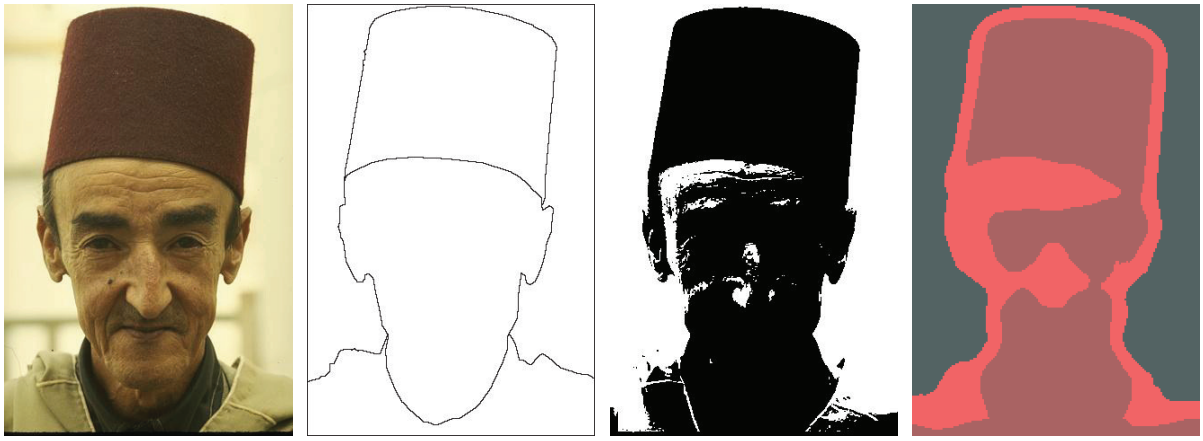
Figura A.5 Nas legendas das Figuras de A.5(d) à A.5(g), *L* é o tamanho da janelas de segmentação de bordas, *g* é a largura do filtro gaussiano, *n* é o número de neurônios. As Figuras de A.5(d) a A.5(g) mostram segmentações de textura por área da imagem teste com dois tipos diferentes de descritores (MV e MVSK) e dois tamanhos de janela (15x15 e 30x30) e utilizando o mapa GNG.



(a) Original (b) Segmentação “humana” (c) Limiarização com Otsu (d) L=5, g=8, n=2

(e) L=5, g=8, n=3 (f) L=5, g=8, n=10 (g) L=5, g=8, n=40 (h) L=5, g=8, n=80

Figura A.6 Nas legendas das Figuras de A.6(d) à A.6(h), *L* é o tamanho da janelas de segmentação de bordas, *g* é a largura do filtro gaussiano, *n* é o número de neurônios. Os mesmos passos utilizado nas Figuras A.1(a) a A.1(c), mas com outra imagem teste.



(a) Original (b) Segmentação “humana” (c) Limiarização com Otsu (d) MV L=15, g=5, n=3

(e) MV L=30, g=5, n=3 (f) MVSK L=15, g=5, n=3 (g) MVSK L=30, g=5, n=3

Figura A.7 As Figuras de A.7(d) a A.7(g) mostram segmentações de textura por área da imagem teste com dois tipos diferentes de descritores (MV e MVSK), dois tamanhos de janela (15x15 e 30x30) e utilizando o mapa GNG. Nas legendas das Figuras de A.7(d) à A.7(g), *L* é o tamanho da janela de segmentação de bordas, *g* é a largura do filtro gaussiano, *n* é o número de nodos.

APÊNDICE B

Padrões do Brodatz Utilizados nos Mosaicos de Teste

As tabelas B.1, B.2 e B.3 contem os padrões de textura do Brodatz utilizados na montagem, respectivamente, dos mosaicos com 2, 4 e 5 texturas diferentes. Estes mosaicos foram utilizados nos testes quantitativos.

Tabela B.1 Combinações de padrões do Brodatz utilizadas para formar o conjunto de teste com dois padrões de textura.

Rótulo do mosaico	Rótulo do Brodatz	
P2_1	d100	d53
P2_2	d53	d58
P2_3	d58	d5
P2_4	d5	d60
P2_5	d60	d112
P2_6	d112	d51
P2_7	d51	d62
P2_8	d62	d11
P2_9	d11	d13
P2_10	d13	d1
P2_11	d1	d20
P2_12	d20	d66
P2_13	d66	d24
P2_14	d24	d25
P2_15	d25	d26
P2_16	d26	d32
P2_17	d32	d16
P2_18	d16	d35
P2_19	d35	d3
P2_20	d3	d102
P2_21	d102	d103
P2_22	d103	d89
P2_23	d89	d21
P2_24	d21	d23
P2_25	d23	d63
P2_26	d63	d104
P2_27	d104	d8
P2_28	d8	d90
P2_29	d90	d92
P2_30	d92	d105
P2_31	d105	d88
P2_32	d88	d94
P2_33	d94	d95
P2_34	d95	d107
P2_35	d107	d108
P2_36	d108	d40
P2_37	d40	d42
P2_38	d42	d45
P2_39	d45	d72
P2_40	d72	d73
P2_41	d73	d109
P2_42	d109	d75
P2_43	d75	d76
P2_44	d76	d80
P2_45	d80	d82
P2_46	d82	d46
P2_47	d46	d48
P2_48	d48	d68
P2_49	d68	d6
P2_50	d6	d71
P2_51	d71	d83
P2_52	d83	d77
P2_53	d77	d78
P2_54	d78	d84
P2_55	d84	d85
P2_56	d85	d96
P2_57	d96	d97
P2_58	d97	d99
P2_59	d99	d86

Tabela B.2 Combinações de padrões do Brodatz utilizadas para formar o conjunto de teste com quatro padrões de textura.

Rótulo do mosaico	Rótulo do Brodatz			
P4_1	d100	d53	d58	d5
P4_2	d53	d58	d5	d60
P4_3	d58	d5	d60	d112
P4_4	d5	d60	d112	d51
P4_5	d60	d112	d51	d62
P4_6	d112	d51	d62	d11
P4_7	d51	d62	d11	d13
P4_8	d62	d11	d13	d1
P4_9	d11	d13	d1	d20
P4_10	d13	d1	d20	d66
P4_11	d1	d20	d66	d24
P4_12	d20	d66	d24	d25
P4_13	d66	d24	d25	d26
P4_14	d24	d25	d26	d32
P4_15	d25	d26	d32	d16
P4_16	d26	d32	d16	d35
P4_17	d32	d16	d35	d3
P4_18	d16	d35	d3	d102
P4_19	d35	d3	d102	d103
P4_20	d3	d102	d103	d89
P4_21	d102	d103	d89	d21
P4_22	d103	d89	d21	d23
P4_23	d89	d21	d23	d63
P4_24	d21	d23	d63	d104
P4_25	d23	d63	d104	d8
P4_26	d63	d104	d8	d90
P4_27	d104	d8	d90	d92
P4_28	d8	d90	d92	d105
P4_29	d90	d92	d105	d88
P4_30	d92	d105	d88	d94
P4_31	d105	d88	d94	d95
P4_32	d88	d94	d95	d107
P4_33	d94	d95	d107	d108
P4_34	d95	d107	d108	d40
P4_35	d107	d108	d40	d42
P4_36	d108	d40	d42	d45
P4_37	d40	d42	d45	d72
P4_38	d42	d45	d72	d73
P4_39	d45	d72	d73	d109
P4_40	d72	d73	d109	d75
P4_41	d73	d109	d75	d76
P4_42	d109	d75	d76	d80
P4_43	d75	d76	d80	d82
P4_44	d76	d80	d82	d46
P4_45	d80	d82	d46	d48
P4_46	d82	d46	d48	d68
P4_47	d46	d48	d68	d6
P4_48	d48	d68	d6	d71
P4_49	d68	d6	d71	d83
P4_50	d6	d71	d83	d77
P4_51	d71	d83	d77	d78
P4_52	d83	d77	d78	d84
P4_53	d77	d78	d84	d85
P4_54	d78	d84	d85	d96
P4_55	d84	d85	d96	d97
P4_56	d85	d96	d97	d99
P4_57	d96	d97	d99	d86

Tabela B.3 Combinações de padrões do Brodatz utilizadas para formar o conjunto de teste com cinco padrões de textura.

Rótulo do mosaico	Rótulo do Brodatz				
P5_1	d100	d53	d58	d5	d60
P5_2	d53	d58	d5	d60	d112
P5_3	d58	d5	d60	d112	d51
P5_4	d5	d60	d112	d51	d62
P5_5	d60	d112	d51	d62	d11
P5_6	d112	d51	d62	d11	d13
P5_7	d51	d62	d11	d13	d1
P5_8	d62	d11	d13	d1	d20
P5_9	d11	d13	d1	d20	d66
P5_10	d13	d1	d20	d66	d24
P5_11	d1	d20	d66	d24	d25
P5_12	d20	d66	d24	d25	d26
P5_13	d66	d24	d25	d26	d32
P5_14	d24	d25	d26	d32	d16
P5_15	d25	d26	d32	d16	d35
P5_16	d26	d32	d16	d35	d3
P5_17	d32	d16	d35	d3	d102
P5_18	d16	d35	d3	d102	d103
P5_19	d35	d3	d102	d103	d89
P5_20	d3	d102	d103	d89	d21
P5_21	d102	d103	d89	d21	d23
P5_22	d103	d89	d21	d23	d63
P5_23	d89	d21	d23	d63	d104
P5_24	d21	d23	d63	d104	d8
P5_25	d23	d63	d104	d8	d90
P5_26	d63	d104	d8	d90	d92
P5_27	d104	d8	d90	d92	d105
P5_28	d8	d90	d92	d105	d88
P5_29	d90	d92	d105	d88	d94
P5_30	d92	d105	d88	d94	d95
P5_31	d105	d88	d94	d95	d107
P5_32	d88	d94	d95	d107	d108
P5_33	d94	d95	d107	d108	d40
P5_34	d95	d107	d108	d40	d42
P5_35	d107	d108	d40	d42	d45
P5_36	d108	d40	d42	d45	d72
P5_37	d40	d42	d45	d72	d73
P5_38	d42	d45	d72	d73	d109
P5_39	d45	d72	d73	d109	d75
P5_40	d72	d73	d109	d75	d76
P5_41	d73	d109	d75	d76	d80
P5_42	d109	d75	d76	d80	d82
P5_43	d75	d76	d80	d82	d46
P5_44	d76	d80	d82	d46	d48
P5_45	d80	d82	d46	d48	d68
P5_46	d82	d46	d48	d68	d6
P5_47	d46	d48	d68	d6	d71
P5_48	d48	d68	d6	d71	d83
P5_49	d68	d6	d71	d83	d77
P5_50	d6	d71	d83	d77	d78
P5_51	d71	d83	d77	d78	d84
P5_52	d83	d77	d78	d84	d85
P5_53	d77	d78	d84	d85	d96
P5_54	d78	d84	d85	d96	d97
P5_55	d84	d85	d96	d97	d99

Referências Bibliográficas

Araújo, A. R., & Costa, D. C. (2009). Local adaptive receptive field self-organizing map for image color segmentation. *Image and Vision Computing*, *27*(9), 1229 - 1239.

Bashar, M., Ohnishi, N., Matsumoto, T., Takeuchi, Y., Kudo, H., & Agusa, K. (2005, NOV). Image retrieval by pattern categorization using wavelet domain perceptual features with lvq neural network. *Pattern Recognition Letters*, *26*(15), 2315-2335.

Batista, L. V., Meira, M. M., & Cavalcanti, N. L. J. (2005, 6–9 Nov.). Texture classification using local and global histogram equalization and the lempel-ziv-welch algorithm. In *Proc. fifth international conference on hybrid intelligent systems his '05* (p. 6).

Beucher, S., & Mathmatique, C. D. M. (1991). The watershed transformation applied to image segmentation. In *Scanning microscopy international* (pp. 299–314).

Brodatz, P. (1966). *Textures: A photographic album for artists and designers*. New York: Dover Publications.

Costa, D. C. (2007). *Mapa auto-organizável com campo receptivo adaptativo local para segmentação de imagens*. Unpublished master's thesis, Universidade Federal de Pernambuco Centro de Informática Pós-graduação em Ciência da Computação.

Cremers, D., Rousson, M., & Deriche, R. (2007). A review of statistical approaches to level set segmentation: Integrating color, texture, motion and shape. *International Journal of Computer Vision*, *72*, 215.

Fritzke, B. (1995). A growing neural gas network learns topologies. 625–632.

Gonzalez, R. C., & Woods, R. E. (2000). *Processamento de imagens digitais*. Edgard Blücher.

Haindl, M. (1998). Unsupervised texture segmentation. In *Advances in pattern recognition. ad-*

nan amin, dov dori, pavel pudil, herbert freeman eds., lecture notes in computer science 1451, isbn: 3-540-64858-5 (pp. 1021–1028). Springer-Verlag.

Haralick, R. (1979). Statistical and structural approaches to texture. *Proceedings of the IEEE*, *67*(5), 786-804.

Haralick, R., Shanmuga.k, & Dinstein, I. (1973). Textural features for image classification. *IEEE Transactions on Systems Man and Cybernetics*, *SMC3*(6), 610-621.

Haykin, S. (1999). *Neural networks a comprehensive foundation* (second ed.). Prentice Hall.

Hoogs, A., Collins, R., Kaucic, R., & Mundy, J. (2003, APR). A common set of perceptual observables for grouping, figure-ground discrimination, and texture classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *25*(4), 458–474.

Hoover, A., JeanBaptiste, G., Jiang, X. Y., Flynn, P. J., Bunke, H., Goldgof, D. B., et al. (1996, July). An experimental comparison of range image segmentation algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *18*(7), 673–689.

Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences of the United States of America-biological Sciences*, *79*(8), 2554–2558.

Hu, M. (1962). Visual pattern recognition by moment invariants. *IRE Trans. Info. Theory*, *IT-8*, 179-187.

Julesz, B. (1981). Textons, the elements of texture-perception, and their interactions. *Nature*, *290*(5802), 91-97.

Julesz, B., & Bergen, J. (1983). Textons, the fundamental elements in preattentive vision and perception of textures. *Bell System Technical Journal*, *62*(6), 1619-1645.

Julesz, B., & Krose, B. (1988, MAY 26). Visual texture-perception - features and spatial filters. *Nature*, *333*(6171), 302-303.

Kim, S. C., & Kang, T. J. (2007, APR). Texture classification and segmentation using wavelet packet frame and gaussian mixture model. *Pattern Recognition*, *40*(4), 1207-1221.

Kohonen, T. (1990, SEP). The self-organizing maps. *Proceedings of the IEEE*, *78*(9), 1464-1480.

Kovacs, I., & Julesz, B. (1994, AUG 25). Perceptual sensitivity maps within globally defined visual shapes. *Nature*, *370*(6491), 644-646.

Kwang In Kim, S. H. P., Keechul Jung, & Kim, H. J. (2002). Support vector machines for texture classification. *IEEE Transactions on Pattern Analysis And Machine Intelligence*, *24*(11), 1542–1550.

Lathi, B. P. (2007). *Sinais e sistemas lineares*. Bookman.

Luo, J., & Crandall, D. (2006). Color object detection using spatial-color joint probability functions. *IEEE Transactions on Image Processing*, *15*.

Mallat, S. (1999). *A wavelet tour of signal processing*. Academic Press.

Manjunath, B., & Ma, W. (1996, AUG). Texture features for browsing and retrieval of image data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *18*(8), 837-842.

Martin, D., Fowlkes, C., Tal, D., & Malik, J. (2001). A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proc. 8th int'l conf. computer vision* (pp. 416–423).

Materka, A., Strzelecki, M., Analysis, T., Review, M. A., Materka, A., & Strzelecki, M. (1998). *Texture analysis methods - a review* (Tech. Rep.). Institute of Electronics, Technical University of Lodz.

Otsu, N. (1979). Threshold selection method from gray-level histograms. *IEEE Transactions On Systems Man and Cybernetics*, *9*(1), 62–66.

Palm, C. (2004). Color texture classication by integrative co-occurrence matrices. *Pattern Recognition*, *37*, 965–976.

Perez, P. (1998). *Markov random fields and images.*

Raghu, P., & Yegnanarayana, B. (1998, MAY). Supervised texture classification using a probabilistic neural network and constraint satisfaction model. *IEEE Transactions on Neural*

Networks, *9*(3), 516-522.

Reed, T., & Dubuf, J. (1993, MAY). A review of recent texture segmentation and feature-extraction techniques. *Cvgip-image Understanding*, *57*(3), 359-372.

Richard O. Duda, P. E. H., & Stork, D. G. (2000). *Pattern classification*. Wiley-Interscience.

Rui, Y., Huang, T., & Chang, S. (1999, MAR). Image retrieval: Current techniques, promising directions, and open issues. *Journal of Visual Communication and Image Representation*, *10*(1), 39-62.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986, October). Learning representations by back-propagating errors. *Nature*, *323*(6088), 533–536.

Serre, T., Oliva, A., & Poggio, T. (2007, April). A feedforward architecture accounts for rapid categorization. *Proceedings of the National Academy of Sciences of the United States of America*, *104*(15), 6424–6429.

Sezgin, M., & Sankur, B. (2004). Survey over image thresholding techniques and quantitative performance evaluation. *Journal of Electronic Imaging*, *13(1)*, 146-165.

Tuceryan, M. (1994). Moment based texture segmentation. *Pattern Recognition Letters*, *15*, 659–668.

Weszka, J., Dyer, C., & Rosenfeld, A. (1976). Comparative-study of texture measures for terrain classification. *IEEE Transactions on Systems Man and Cybernetics*, *6*(4), 269-285.

Zhang, H., Fritts, J. E., & Goldman, S. A. (2008, May). Image segmentation evaluation: A survey of unsupervised methods. *Computer Vision and Image Understanding*, *110*(2), 260–280.

Zhang, J., & Tan, T. (2002, MAR). Brief review of invariant texture analysis methods. *Pattern Recognition*, *35*(3), 735-747.

Zhang, J., Walter, G., Miao, Y., & Lee, W. (1995, JUN). Wavelet neural networks for function learning. *IEEE Transactions on Signal Processing*, *43*(6), 1485-1497.

Zhang, Y. (1997). Evaluation and comparison of different segmentation algorithms. *Pattern Recognition Letters*, *18*, 963-974.

Zheng, C., Sun, D., & Zheng, L. (2006). Recent applications of image texture for evaluation of food qualities - a review. *Trends in Food Science & Technology*, *17*(3), 113-128.

Zhu, S., Guo, C., Wang, Y., & Xu, Z. (2005, APR-MAY). What are textons? *International Journal of Computer Vision*, *62*(1-2), 121-143.