

Viviane de Almeida Ventura

***Representação de Imagens Através de Grafos
Utilizando o Algoritmo Split-And-Merge Combinado
Com Descritores de Cor e Textura***

Vitória - ES, Brasil

27 de agosto de 2009

Viviane de Almeida Ventura

***Representação de Imagens Através de Grafos
Utilizando o Algoritmo Split-And-Merge Combinado
Com Descritores de Cor e Textura***

Dissertação apresentada como requisito parcial
para obtenção do Grau de Mestre em Ciência da
Computação pela Universidade Federal do Es-
pírito Santo.

Orientador:

Thomas Walter Rauber

Co-orientador:

Maria Claudia Silva Boeres

DEPARTAMENTO DE INFORMÁTICA
CENTRO TECNOLÓGICO
UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO

Vitória - ES, Brasil

27 de agosto de 2009

Dissertação sob o título “*Representação de Imagens Através de Grafos Utilizando o Algoritmo Split-And-Merge Combinado Com Descritores de Cor e Textura*”, defendida por Viviane de Almeida Ventura e aprovada em 27 de agosto de 2009, em Vitória, Estado do Espírito Santo, pela banca examinadora constituída pelos professores:

Prof. Ph.D. Thomas Walter Rauber
Orientador
Universidade Federal do Espírito Santo

Prof. Dra. Maria Claudia Silva Boeres
Coorientadora
Universidade Federal do Espírito Santo

Prof. Dr. Evandro Ottoni Teatini Salles
Universidade Federal do Espírito Santo

Prof. Dr. Elias Silva de Oliveira
Universidade Federal do Espírito Santo

Prof. Dra. Aura Conci
Universidade Federal Fluminense

Abstract

This master thesis presents the use of graphs to image representation. There are several approaches based on graphs in computer imaging, such as image retrieval based on content, tumor detection in medical imaging and image recognition, which is one of the most important area of the image processing research.

When an image is recognized by its comparison to a model and both are represented by graphs, the matching graph problem is then characterized, where the criteria for correspondence is defined as a set of values to measure the similarity between the model and the image to be recognized. An interface for the manipulation of images and their corresponding graphs, called ImGraph, was implemented as a undergraduation computer science course final project. This tool is able to create attributed graphs that represent the images (graphs) and their features (attributes).

This dissertation consists of an extension of an known automatic segmentation algorithm investigation with color and texture descriptors. The combination of color and texture characteristics is analysed for the segmentation algorithm chosed with the objective of improving the image representation by attributed graphs. The split-and-merge segmentation approach is used, recursively splitting the image into smaller homogeneous regions and then joining them according to some similarity measure, considered in this work as probabilistic distances. It was also analyzed the combination of texture descriptor LBP (Local Binary Pattern) with the color descriptor, on the premise that some regions of the image can be better described not only by a specific descriptor, but also by a combination of them. Also, to reduce the size of the RGB color space of three dimensions to one dimension the PCA (Principal Components Analysis) method was used.

As another contribution of this thesis, two mathematical functions defined from the attributes of the graphs for the similarity computing were proposed and investigated. All the cited contributions were added to the ImGraph, for its improvement. Experiments were conducted using a high resolution aerial space image of the Jucu River, in the state of Espírito Santo.

Resumo

Este trabalho apresenta a utilização de grafos para a representação de imagens. Há várias abordagens que utilizam a teoria dos grafos na computação de imagens, tais como recuperação de imagens baseadas em conteúdo, detecção de tumores em imagens médicas e reconhecimento de cenas, uma das linhas de pesquisa mais importantes da área de processamento de imagens.

Quando uma imagem é reconhecida através da comparação com um modelo e ambas são representadas por grafos, é caracterizado o problema de correspondência de grafos, onde o critério para a comparação é definido a partir de um conjunto de valores que medem a similaridade entre o modelo e a imagem a ser reconhecida. A construção de uma interface para a manipulação das imagens e dos grafos correspondentes, denominada *ImGraph*, foi tema de um projeto final de graduação. Essa ferramenta permite a criação de grafos de atributos que representam as imagens (grafos) e suas principais características (atributos).

Esta dissertação consiste no estudo de descritores de cor e textura aplicados em um algoritmo conhecido de segmentação automática de imagens. A combinação de características de cor e textura para segmentação automática de imagens é investigada com o objetivo de aperfeiçoar a representação da mesma através de grafos de atributos. A abordagem de segmentação *split-and-merge* é usada, dividindo a imagem recursivamente em regiões homogêneas cada vez menores, e então unindo-as de acordo com alguma medida de similaridade, computada nesta dissertação através de medidas de distâncias probabilísticas. Foram analisadas a combinação do descritor de texturas LBP (*Local Binary Pattern*) com o descritor de cor, partindo da premissa que algumas regiões da imagem avaliada podem ser melhor descritas não apenas por um descritor específico, mas por uma combinação dos mesmos. A redução da dimensão do espaço de cor RGB de três dimensões para uma dimensão foi realizada através da utilização do PCA (*Principal Components Analysis*).

Como segunda contribuição desta dissertação, duas funções matemáticas definidas a partir dos atributos dos grafos, para o cálculo das similaridades, são propostas. Todas as contribuições desta dissertação foram incorporadas ao *ImGraph*, visando o seu aprimoramento. Experimentos foram realizados utilizando-se uma imagem aérea de alta-resolução espacial do Rio Jucu, Estado do Espírito Santo.

Dedicatória

Dedico este trabalho à Deus, pela força que Ele me conferiu nos momentos de necessidade;

Aos meus pais, pelo amor, paciência e incentivo.

Agradecimentos

Agradeço aos meus pais, que trilharam comigo este caminho, pelo carinho, pelo apoio, pela força, pelo amor incondicional;

Aos meus colegas de curso, que compartilharam os prazeres e dificuldades desta jornada;

Aos meus parentes e amigos, que me apoiaram;

Ao Leonardo Tadeu Franklin de Oliveira pela paciência, companhia e ajuda fornecida;

E principalmente a Deus, sem Ele esta conquista não seria possível.

Sumário

Lista de Figuras

Lista de Tabelas

1	Introdução	p. 13
1.1	Motivação	p. 15
1.2	Trabalhos Correlatos	p. 15
1.3	Grafos e Matrizes de Similaridade	p. 20
1.4	Objetivos e Contribuições	p. 21
1.5	Organização do Texto	p. 22
2	Estado da Arte	p. 23
2.1	Técnicas de Segmentação de Imagens	p. 24
2.1.1	Limiarização	p. 25
2.1.2	Segmentação baseada em detecção de descontinuidades	p. 26
2.1.3	Segmentação baseada em regiões	p. 30
2.2	Análise de Texturas	p. 34
2.2.1	Matriz de Coocorrência de Tons de Cinza	p. 37
2.2.2	Espectro de Textura	p. 39
2.2.3	Primitivas de textura (textons)	p. 41
2.2.4	Local Binary Pattern	p. 43
2.3	Redução de Dimensionalidade	p. 44
2.3.1	Extração de Características	p. 45

2.3.2	Seleção de Características	p. 49
3	Experimentos Computacionais	p. 52
3.1	Medidas de Distâncias Utilizadas	p. 54
3.2	Processo de Segmentação das Imagens	p. 56
3.2.1	Descritor de texturas <i>Local Binary Pattern</i>	p. 56
3.2.2	Descritor de cor	p. 57
3.2.3	Compressão de Dados Através da Análise de Componentes Principais	p. 58
3.2.4	O Método <i>Split-And-Merge</i> Combinado com os Descritores de Cor e Textura	p. 60
3.3	Representação da Imagem através de Grafos de Atributos	p. 62
3.4	Funções de Agregação	p. 63
3.4.1	Escolha dos Critérios	p. 63
3.4.2	Normalização e Combinação de Critérios	p. 64
3.5	Experimentos Realizados	p. 66
3.5.1	Segmentação de Imagens	p. 68
3.5.2	Função de Agregação	p. 95
4	Conclusões e trabalhos futuros	p. 106
	Referências Bibliográficas	p. 108
	Apêndice A – Ambiente Gráfico para Representação de Imagens Através de Gra- fos	p. 115
A.1	Funcionalidades	p. 116
A.2	Protótipo Implementado	p. 121

Lista de Figuras

2.1	2.1(a) Imagem a ser segmentada. 2.1(b) Histograma do nível de cinza da imagem. <i>Pixels</i> abaixo do limiar serão rotulados como <i>pixels</i> do objeto de interesse e aqueles acima do limiar serão rotulados como fundo.	p. 26
2.2	Máscara utilizada para detecção de pontos.	p. 27
2.3	Máscaras para detecção de linhas.	p. 28
2.4	Operadores de Sobel [1].	p. 29
2.5	Máscara usada para computar o Laplaciano.	p. 30
2.6	Árvore Quadrática (<i>Quadtree</i>) que representa a Região <i>R</i>	p. 32
2.7	Passos seguidos para a segmentação de uma imagem utilizando o método <i>split-and-merge</i>	p. 33
2.8	Textura colorida e sua versão em escala de cinza.	p. 35
2.9	Imagem com dimensão 3×3 <i>pixels</i> e quatro níveis de cinza.	p. 38
2.10	Exemplo demonstrando os passos necessários para o cálculo do LBP. 2.10(a) Imagem original, 2.10(b) imagem binária gerada por comparação com o <i>pixel</i> central, 2.10(c) máscara LBP e 2.10(d) resultado da operação.	p. 44
2.11	Mapeamento de componente não-linear usando Redes Neurais.	p. 48
3.1	Etapas do método proposto.	p. 53
3.2	Imagem original, suas três componentes e as três componentes do PCA.	p. 60
3.3	Comparação dos métodos de redução de cor. A imagem em escala de cinza obtida pela soma ponderada dos canais R,G,B à esquerda e a imagem em escala de cinza obtida pelo método PCA à direita.	p. 61
3.4	Imagens de 500×408 <i>pixels</i> , obtidas através de procedimentos sobre a imagem colorida.	p. 69

3.5	Imagens de 2740×2440 <i>pixels</i> , obtidas através de procedimentos sobre a imagem colorida.	p. 70
3.6	$Test_{LBP}$: $tm = ts = 1.1, D = L_D, ms = 8$	p. 72
3.7	Imagens Segmentadas pelo método $Test_{LBP}$	p. 74
3.8	$Test_{GRAY}$: $tm = ts = 1.1, D = L_D, M = 8$	p. 75
3.9	Imagens Segmentadas pelo método $Test_{GRAY}$	p. 77
3.10	$Test_{GRAY+LBP}$: $tm = ts = 1.1, D = L_D, M = 8$	p. 78
3.11	Imagens Segmentadas pelo método $Test_{GRAY+LBP}$	p. 80
3.12	$Test_{PCA}$: $tm = ts = 1.1, D = D_J, M = 8$	p. 81
3.13	Imagens Segmentadas pelo método $Test_{PCA}$	p. 83
3.14	$Test_{PCA+LBP}$: $tm = ts = 1.1, D = L_D, M = 8$	p. 84
3.15	Imagens Segmentadas pelo método $Test_{PCA+LBP}$	p. 86
3.16	$Test_{RGB}$: $tm = ts = 1.1, D = L_D, M = 8$	p. 87
3.17	Imagens Segmentadas pelo método $Test_{RGB}$	p. 89
3.18	$Test_{RGB+LBP}$: $tm = ts = 1.1, D = L_D, M = 8$	p. 90
3.19	Imagens Segmentadas pelo método $Test_{RGB+LBP}$	p. 92
A.1	Janela principal do ImGraph	p. 121

Lista de Tabelas

2.1	Matriz de coocorrência, cada elemento da matriz diz respeito às transições de níveis de cinza.	p. 38
3.1	Uma seleção de medidas de distância probabilísticas em reconhecimento de padrões [2], [3], [4]. P e Q são dois histogramas com n valores diferentes. . .	p. 55
3.2	Expressões analíticas de distâncias probabilísticas entre duas densidades normais.	p. 55
3.3	Autovalores obtidos através da aplicação do PCA na imagem selecionada. . .	p. 59
3.4	$Test_{LBP}$: Imagens segmentadas	p. 73
3.5	$Test_{GRAY}$: Imagens segmentadas.	p. 76
3.6	$Test_{GRAY+LBP}$: Imagens segmentadas.	p. 79
3.7	$Test_{PCA}$: Imagens segmentadas.	p. 82
3.8	$Test_{PCA+LBP}$: Imagens segmentadas.	p. 85
3.9	$Test_{RGB}$: Imagens segmentadas.	p. 88
3.10	$Test_{RGB+LBP}$: Imagens segmentadas.	p. 91
3.11	Tempo de execução médio do método proposto em milissegundos.	p. 94
3.12	À esquerda, os números que denotam as arestas do modelo. À direita, a imagem modelo.	p. 96
3.13	À esquerda, a correspondência entre a imagem modelo e a imagem alvo. À direita, a imagem segmentada utilizando-se a 1ª componente do PCA, com os parâmetros $ts = tm = 1.15$, $ms = 4$, $F = N$, $D = L_J$	p. 97
3.14	À esquerda, correspondência entre a imagem modelo e a imagem alvo. À direita, imagem segmentada utilizando-se a 1ª componente do PCA combinada com o descritor LBP, com os parâmetros $ts = tm = 1.15$, $ms = 4$, $F = N$, $\alpha = 0.5$, $D = L_J$	p. 98

3.15	Matriz de similaridade entre vértices PCA. $\beta = 0.9$ e $D = L_M$	p. 100
3.16	Similaridade entre a aresta 1 (0,2) do grafo a ser reconhecido e as arestas do grafo modelo	p. 101
3.17	Similaridade entre a aresta 2 (0,3) do grafo a ser reconhecido e as arestas do grafo modelo	p. 101
3.18	Similaridade entre a aresta 5 (2,3) do grafo a ser reconhecido e as arestas do grafo modelo	p. 101
3.19	Matriz de similaridade entre vértices PCA+LBP. $\beta = 0.7$, $\alpha = 0.5$ e $D = L_M$.	p. 103
3.20	Similaridade entre a aresta 0 (0,1) do grafo a ser reconhecido e as arestas do grafo modelo	p. 104
3.21	Similaridade entre a aresta 2 (0,5) do grafo a ser reconhecido e as arestas do grafo modelo	p. 104
3.22	Similaridade entre a aresta 4 (1,5) do grafo a ser reconhecido e as arestas do grafo modelo	p. 104

1 Introdução

O reconhecimento estrutural de uma cena com muitos objetos requer o uso de modelos complexos. Uma boa caracterização dos objetos presentes na cena e das relações espaciais existentes entre eles se faz necessária para que se possa reconhecer a cena como um todo. A própria especificação de um determinado modelo que venha a representar as características relevantes para o reconhecimento de cenas é foco de extensivos estudos. O reconhecimento baseia-se na identificação das estruturas dos objetos, ao invés da identificação de detalhes dos mesmos, também presentes na cena. Esse tipo de reconhecimento torna-se difícil devido a similaridades entre diferentes objetos e a grande variabilidade de um mesmo objeto em diferentes instâncias da cena.

Cada vez mais, a tecnologia atual e os avanços computacionais oferecem a possibilidade de pesquisadores investirem na busca de soluções eficientes para tal problema. O reconhecimento de uma imagem pode ser efetuado através de sua comparação a um modelo da mesma (por exemplo, pode-se comparar imagens em ressonância magnética de um cérebro humano com o seu Atlas). Para realizar a análise automática de imagens de forma completamente independente, ou seja, não depender da interação humana nem mesmo indiretamente, um sistema deveria ser dotado da capacidade de extrair informação pertinente da imagem (para sua adequada representação via modelos matemáticos), generalizar conhecimento a partir de instâncias particulares e fazer inferências a partir de informações incompletas.

Modelos baseados em grafos são frequentemente utilizados para representar cenas em processamento de imagens [5, 6, 7, 8]. Em geral, vértices desses grafos representam objetos e arestas (ou arcos) representam relações existentes entre esses objetos. A informação relevante para o processo de reconhecimento pode ser extraída da cena e representada através de grafos relacionais de atributos (GRA). A estrutura da cena é mapeada pelo grafo e os atributos representam características da mesma, importantes para o reconhecimento.

Com o intuito de validar os métodos propostos para efetuar o reconhecimento de imagens representadas por grafos, é de fundamental importância ter acesso a um conjunto de exemplos

obtidos a partir de imagens reais. A representação de uma imagem via grafo requer que a imagem seja dividida em regiões (que serão representadas pelos vértices do grafo) e que as relações entre essas regiões sejam determinadas (o que definirá o método de construção das arestas do grafo). Outra necessidade importante para a descrição de uma imagem via grafos, é a determinação dos atributos que representam as características da mesma. Identificar o conjunto de atributos relevantes para a descrição da imagem é uma tarefa difícil. O ideal é identificar aqueles que permitem uma fácil discriminação da imagem, quando comparada a um modelo.

Para suprir essa carência, uma interface para a manipulação das imagens e o desenho dos grafos correspondentes, denominada *ImGraph*, foi construída como projeto final de graduação [9]. A ferramenta permite a criação de grafos de atributos que representam imagens fornecidas como entrada. Dentre suas funcionalidades, destaca-se a geração de grafos para representação de imagens usadas como modelos e imagens a serem reconhecidas a partir dos modelos. Para isso, a ferramenta permite a segmentação manual (no caso do modelo) e automática (no caso da imagem a ser reconhecida) das imagens, além de utilizar funções que combinem os atributos em valores de similaridade.

Portanto, esta dissertação consiste em uma extensão da interface apresentada em [9]. Como aperfeiçoamento da ferramenta, foi sugerida uma nova abordagem para a segmentação automática de imagens, incluindo uma análise mais detalhada da fase de pré-processamento, que consiste na segmentação e extração de características da imagem, ao qual a mesma é submetida antes de ser convertida em um grafo. É investigada a contribuição dos descritores de cor e textura no resultado final da segmentação, bem como as medidas de distâncias utilizadas para medir a similaridade entre duas regiões. Também foi analisada a utilização do PCA (*Principal Components Analysis* - Análise de Componentes Principais) [10], em que o espaço de cor RGB foi reduzido de três dimensões para uma dimensão através do método.

Segmentada a imagem, a mesma será representada através de grafos de atributos que serão utilizados para a criação de matrizes, chamadas matrizes de similaridade, que deverão conter informações acerca da similaridade entre vértices (resp. arestas) entre dois grafos de entrada: o grafo modelo e o grafo que se pretende reconhecer neste modelo. Para a construção das matrizes, foram criadas e estudadas duas funções para se verificar a similaridade entre vértices e arestas. Ao final, todas as novas contribuições foram acrescentadas ao ambiente *ImGraph*.

1.1 Motivação

Esta dissertação tem por motivação o aprimoramento do ambiente *ImGraph* para descrição de imagens por grafos, proposto em [9] utilizado para representar imagens de entrada fornecidas como grafos de atributos. Assim, como objetivo principal, realizou-se a implementação e o estudo sobre segmentação automática de imagens utilizando o algoritmo *split-and-merge* [1] combinado com descritores de cor e textura, visando um resultado tal que a imagem segmentada cognitivamente remete à divisão da imagem em regiões delimitadas pelos contornos existentes na imagem original. Uma segmentação bem realizada fornece um pré-processamento da imagem para extração de vértices, arestas e atributos que irão compor o grafo de atributos que a representa. Também foi realizada a implementação e o estudo acerca da função de agregação, responsável por medir a similaridade entre os atributos de vértices e/ou arestas. Essas matrizes são utilizadas em algumas abordagens que propõem a resolução do problema de reconhecimento de cenas.

1.2 Trabalhos Correlatos

Um dos exemplos do uso de grafos na representação de imagens é no problema de reconhecimento de faces. Uma importante demanda que aparece em diferentes problemas subjacentes a esse tipo de reconhecimento é a localização e segmentação das regiões com características faciais. Em [11] propõe-se um procedimento de segmentação de regiões com características faciais baseadas em um modelo, onde a intenção é modelar, através de uma supersegmentação, o conhecimento básico sobre as características de uma face como um grafo. Assim, os grafos relacionais de atributos foram utilizados para a representação das características extraídas de faces humanas. Para a extração apenas das partes que serão utilizadas no processo é utilizado um método para a delimitação das informações relevantes, como olhos, nariz e boca. Este método mostrou-se rápido e eficiente mesmo nas ocorrências de olhos fechados e sorrisos. O método desenvolvido utiliza um algoritmo chamado *watershed* para segmentação baseada no gradiente da imagem. Nesta fase, também são extraídos os atributos que serão utilizados nos grafos para representar vértices e arestas. Os atributos utilizados nos vértices foram a média do nível de cinza e a distância do centroide de cada região ao limite pré-definido na fase da localização das partes relevantes da imagem. Nas arestas foram utilizadas as informações de distância entre centroides de regiões adjacentes.

Grafos também podem ser utilizados para representar cenas complexas. No trabalho descrito em [8], uma metodologia para reconhecimento e monitoramento de objetos em sequência

de imagens digitais é apresentada. Assim como em [11] e [12], o homomorfismo é resolvido através de um algoritmo de otimização de busca em árvore. Assim, objetos são representados por grafos relacionais de atributos, cuja estrutura de dados de vértices e arestas representam, respectivamente, objetos e relações (estrutural, temporal, etc.) entre eles. O modelo adotado para descrever objetos alvo, ou seja, objetos que se pretendem reconhecer no modelo, é representado por suas partes componentes e o reconhecimento é feito pelo casamento inexato de grafos, que consiste em casar de forma aproximada os grafos derivados de um vídeo com um grafo proveniente da representação de uma imagem modelo. Este casamento é realizado através de um algoritmo de otimização de pesquisa em árvore e minimização de uma função de custo pré-definida. Finalmente, a monitoração do objeto é realizada de acordo com uma transformação afim derivada de parâmetros extraídos da fase de reconhecimento.

Outra aplicação inclui o problema de recuperação de imagens baseados em conteúdo (CBIR - *Content Based Image Retrieval*). Em [13] foi apresentada uma nova abordagem para recuperação de imagens. A técnica é baseada em redes neurais recursivas [14, 15], onde os critérios de recuperação são derivados do próprio aprendizado da rede. A imagem é representada através de grafos e combina características estruturais com simbólicas, enquanto uma rede neural recursiva descobre a representação ótima para a busca na base de imagens. Usa-se grafos dirigidos para representar a imagem, sendo que esta técnica pode ser estendida para reconhecimento de padrões invariantes, podendo ser explorada a característica de isomorfismo que apresenta os grafos, atingindo invariância à translação, escala e rotação. O problema deste tipo de representação é o alto custo computacional.

Em [16] foi implementado um sistema de recuperação de imagens por conteúdo. Foram utilizados como atributos dos vértices a área e o raio de cada região, bem como a orientação, definida pelo ângulo entre a direção horizontal e a coordenada de comprimento. Para as arestas foram utilizadas a distância (menor distância entre as bordas de duas regiões) e o ângulo (ângulo da direção horizontal da linha que conecta os centros de massa de duas regiões). Neste método o usuário informa uma imagem como exemplo, adicionando um valor de tolerância para a localização de imagens semelhantes. Este valor é utilizado pela estrutura de indexação na busca das imagens armazenadas no banco de dados.

Um protótipo de um sistema de recuperação de imagens com base na cor foi apresentado em [17]. Este armazena informações de regiões com a mesma cor da imagem (regiões cromáticas), além do tamanho, posição e limites com outras regiões. Para isso, são utilizados os seguintes grafos: o *Modified Color Adjacency Graph* (MCAG) e o *Spatial Variance Graph* (SVG). A recuperação da imagem é feita através de uma medida de similaridade baseada na comparação

dos grafos mencionados. Para a indexação utiliza-se uma R-Tree, uma estrutura capaz de indexar dados multidimensionais. O protótipo analisa as regiões cromáticas de uma imagem, suas adjacências e variações espaciais. A extração de atributos é feita primeiro removendo ruídos que existem próximos às adjacências das regiões cromáticas, tornando as regiões uniformes e as transições entre as regiões mais aguçadas, proporcionando a geração de um grafo mais preciso. Depois, são descritas as regiões cromáticas da imagem e o tamanho das adjacências entre as mesmas, bem como a distribuição de cor no domínio espacial da imagem. As informações necessárias para elaborar este grafo são as cores da imagem, a contagem de cor da imagem e coordenadas de cada *pixel*, para cada cor. Cada nó contém a própria variância espacial da cor e a variância espacial relativa de duas cores. Cada borda tem como seus atributos a variância espacial entre as cores denotadas pelos nós ligados. A partir da obtenção do vetor de atributos desta imagem, a recuperação é feita através da procura na base de assinaturas, imagens que correspondem ao vetor de atributos da imagem consulta. Depois de obtidas as imagens, é efetuada a classificação para ordenar os resultados, através da aplicação de uma métrica de similaridade.

Outro trabalho que representa imagens através de grafos para recuperação de imagens baseados em conteúdo é proposto em [18]. Depois de segmentada, características de textura, usando a transformada *wavelet* [19], e características de cor, usando histograma, são extraídas da imagem. Assim, essas características são comparadas com as características das imagens segmentadas da base de dados. A medida de similaridade usada para comparar características de textura é a distância euclidiana, e para a cor é a distância quadrática.

Uma abordagem apresentada em mineração de imagens foi apresentada em [20], que trata da extração de padrões e características singulares a partir de grandes coleções de imagens de um domínio específico. O objetivo é identificar as melhores características e assim extrair as informações relevantes a partir das imagens. Vários métodos de extração, assim como as características extraídas e as técnicas utilizadas, são avaliados para a contribuição da resolução do problema. Resultados experimentais apresentados mostram que os recursos utilizados são suficientes para identificar os padrões a partir das imagens.

O problema de detecção e localização de objetos duplicados foi tratado em [21]. Diversas aplicações requerem métodos eficazes de detecção de objetos duplicados, como codificação automática de vídeo e de imagem, vídeo vigilância, ou vídeo pesquisa. Neste trabalho, uma abordagem baseada em grafos para detecção de objetos 3D duplicados em imagens fixas é proposto. Um grafo modelo é usado para representar a informação espacial do objeto, a fim de evitar o uso explícito do modelo do objeto 3D. O trabalho apresentado propõe a detecção da presença de um objeto-alvo, prevendo a sua caixa delimitadora, baseado em um conjunto de

imagens que contenham esse objeto. Características das regiões de interesse são extraídas, e a posição, dimensão e orientação para cada região de interesse são calculados.

Grafos também podem ser utilizados para extrair informações textuais. O trabalho descrito em [22] utiliza imagens representadas por grafos e árvore geradora mínima para extrair textos de *outdoor*. Os atributos do grafo incluem cores representativas da região, posições, tamanho e aspecto da área e forma do contorno.

Em [23] é proposta uma nova abordagem para segmentação de imagens representadas por grafos, aplicadas a cenas urbanas para sensoriamento remoto. O método extrai os resultados de uma imagem previamente supersegmentada através de algoritmos conhecidos como crescimento de regiões ou *watershed*. Esses objetos de entrada são conectados através de um Grafo Relacional de Atributos. Através da análise das conexões, é possível procurar por áreas retangulares, dentro de nós conectados. Áreas retangulares são muito comuns em um ambiente urbano. Os objetos, ou vértices do grafo, cuja união forma mais regiões retangulares, são unidos resultando em uma região nova e maior, com formas mais adequadas ao ambiente urbano. O procedimento começa a partir de simples segmentos, aplicando os estágios de pré-processamento e pré-classificação. O resultado final é obtido através da união de regiões similares conectadas na estrutura do grafo. O estágio de pré-classificação realiza uma classificação não supervisionada através dos mapas auto-organizáveis de Kohonen (SOM) [24], clusterizando regiões com classes similares. Para cada região, o algoritmo extrai um conjunto de formas e atributos espectrais, tais como relação perímetro/área, compacidade e média dos *pixels*. Esse conjunto de atributos é utilizado para separar regiões dentro da classe principal.

Uma característica comum às abordagens que realizam reconhecimento de cenas baseadas em grafos é a representação das imagens comparadas através de Grafos Relacionais de Atributos, extraindo as características primordiais que a compõem ao mesmo tempo que se mantém a coerência estrutural dos objetos presentes na mesma. Para isso, é fundamental que a segmentação seja bem sucedida, evitando supersegmentação desnecessária e resíduos, que costumam gerar grafos com muitas informações (muitos vértices e/ou arestas) a serem analisadas. Para se obter uma segmentação o mais livre possível dessas características, nesta dissertação foram consideradas a cor e a textura da imagem para investigar a possível melhoria da qualidade da segmentação realizada através do uso destes descritores. Embora não haja definição concreta sobre o conceito de textura, a mesma é facilmente percebida e é rica em informação visual. Geralmente texturas são padrões visuais complexos compostos por entidades, ou subpadrões, que têm brilho, cor, inclinação e tamanho característicos [25]. Os seres humanos podem facilmente captar as texturas sobre uma superfície, mesmo com nenhuma informação de cor. A cor

atua apenas como uma ajuda para interpretações mais ricas. Mesmo quando a informação de cor é distorcida, por exemplo, devido ao daltonismo, o sistema visual ainda funciona. Intuitivamente, isso sugere que, pelo menos para o nosso sistema visual, a cor e a textura são fenômenos distintos. No entanto, a utilização conjunta de características de cor e textura tem sido uma abordagem popular para análise de texturas coloridas [26, 27].

A extração de informações como a textura e cor de uma imagem é de grande relevância em muitas aplicações de tratamento e reconhecimento de imagens. Uma série de abordagens distintas utilizando a cor e a informação de textura têm sido propostos. O problema de classificação de granito foi utilizado como teste para o método, no trabalho descrito por [28], que propôs um método de segmentação de imagem aérea, no qual são computados independentemente espaços de cor e textura. A segmentação final é obtida através da decisão feita por cada classificador, utilizando cor ou textura isoladamente. Também é possível citar o trabalho de [29], que descreve um método de extração de características em imagens de alta-resolução aplicadas sobre o algoritmo de segmentação *split-and-merge* com o descritor de texturas LBP (*Local Binary Pattern*) [30, 31, 32].

Outra vantagem de utilizar informações como cor e textura é a independência em relação à posição, orientação, tamanho, forma e brilho do objeto. Entretanto, para a definição de quais atributos serão usados, deve-se analisar o objetivo da aplicação em que eles serão utilizados, pois em uma mesma imagem, a importância das características varia de acordo com este objetivo. No processo de busca de imagens similares, por exemplo, não existe a característica ideal, mas sim, a característica que atende ao objetivo da consulta. A importância de uma determinada característica varia de acordo com o tipo de imagem utilizado, por exemplo, a característica cor não é muito aplicada em métodos baseados em imagens médicas [33].

São muitas as informações que podem ser extraídas de uma imagem após o seu processamento. Algumas destas informações podem não possuir grande importância, pois seus valores não possuem destaque, tornando as informações redundantes em relação às demais. Para que sejam descartadas informações que não são importantes dentro do conjunto dos dados extraídos da imagem, utilizam-se métodos de redução de dimensão. Em [34] é utilizado um método para encontrar as características homogêneas dentro da imagem. Esta informação é de suma importância para definir o processamento que será utilizado. Nesta dissertação, foi utilizado o método PCA [10] para reduzir a dimensão do espaço de cores. O processo de extração das características das imagens produz um conjunto de n características que, juntas, formam o vetor de características. Os vetores de características gerados formam um espaço multidimensional. As imagens são manipuladas como objetos dentro deste espaço. Métodos como a utilização

de algoritmo genético [35] podem ser usados para a extração de características. A utilização deste tipo de algoritmo é demonstrada em [36], onde a extração de característica é aplicada em imagens de fotografias aéreas, mais precisamente, em imagens de lagos e rios em uma floresta.

Em [37], o cálculo de textura em imagens de satélite utiliza matrizes de coocorrência [1] e vetores de características. Estes vetores são calculados através de descritores estatísticos. Os descritores são utilizados para extrair as informações contidas nas matrizes. A matriz de coocorrência responde pela quantidade de diferentes combinações, referentes a valores de intensidade dos *pixels* existentes em uma imagem.

Outros trabalhos que utilizam imagens representadas através de grafos, incluem: reconhecimento de formas [38], procura de objetos de interesse em cenas [39], reconhecimento de objetos [40, 41] e detecção de imagens duplicadas [42]. Na área médica, a utilização das informações adquiridas através das imagens tem sido fundamental [16, 43]. Estas técnicas são capazes de encontrar tumores baseando-se, exclusivamente, na interpretação das imagens obtidas por meio do seu processamento, auxiliando os médicos no diagnóstico precoce de doenças.

Os conceitos envolvidos na criação da representação das imagens através de grafos e matrizes de similaridade serão apresentados na próxima seção.

1.3 Grafos e Matrizes de Similaridade

Um grafo é uma estrutura $G = (V, E)$, onde V é um conjunto finito de vértices e $E \in V \times V$, um conjunto finito de arcos (se o grafo é orientado) ou arestas (caso contrário) [44]. Neste trabalho são considerados apenas os grafos sem orientação. As cardinalidades dos conjunto V e E são representadas respectivamente por $|V|$ e $|E|$. Um grafo pode ser visualizado como uma representação gráfica das relações existentes entre elementos de dados. Cada aresta $e \in E$ será denotada pelo par de vértices $e = \{v, w\}$ que a forma. Os vértices v e w são os extremos da aresta e são ditos adjacentes ou vizinhos, e a aresta e é dita incidente a ambos os vértices v e w .

Vértices e arestas podem conter informações. Quando essas informações são simples rótulos ou pesos, tais como números ou letras, o grafo é chamado grafo rotulado. Entretanto, quando vértices e arestas apresentam alguma informação extra, então o grafo é chamado grafo de atributos, sendo que os atributos (informação extra) podem estar vinculados a vértices, arestas ou a ambos.

Um grafo também pode ser representado computacionalmente por uma matriz de adjacências: um grafo G com $|V|$ vértices pode ser representado por uma matriz de adjacências M de

dimensão $|V|^2$. Para representar um grafo não direcionado, simples e sem pesos nas arestas, basta que as entradas (i, j) da matriz M contenham o valor 1 se os vértices v_i e v_j são adjacentes e 0 caso contrário. Se as arestas do grafo tiverem pesos, a entrada (i, j) de M contém o peso da aresta $\{i, j\}$. Em grafos simples e não direcionados, as matrizes de adjacência são simétricas e possuem diagonal nula. Os grafos são muito úteis na representação de conhecimento [45] e de problemas da vida real, em vários campos profissionais. Por exemplo, pode-se representar um mapa de estradas através dos grafos e usar algoritmos específicos de percurso em um grafo para determinar o caminho mais curto entre dois pontos, ou o caminho mais econômico, a partir dos pesos (ou custos) das arestas ou dos vértices. O custo total do caminho será calculado a partir destes pesos. Em diversos trabalhos, grafos de atributos são utilizados para representar imagens [8, 23, 11, 46]. Assim, nesta dissertação, grafos de atributos representarão imagens, sendo que os vértices designam regiões da imagem analisada, contendo informações acerca da região, como cor e centros de gravidade da região, e as arestas designam as relações existentes entre essas regiões, contendo informações acerca dessas relações, como distância entre os centros de gravidade da região, ou seja, o comprimento da aresta, e sua angulação. Esses grafos serão representados computacionalmente através de matrizes de adjacências.

Algumas abordagens que utilizam grafos para resolver o problema de reconhecimento de cenas por correspondência de grafos utilizam, também, para auxiliar a tarefa do reconhecimento, informações armazenadas em matrizes de similaridade, que medem o grau de semelhança entre cada par de vértices e de arestas dos grafos comparados.

Entende-se por matriz de similaridade entre vértices (resp. arestas) uma matriz M de dimensão $|V| \times |V|$ (resp. $|E| \times |E|$), onde cada m_{ij} corresponde à similaridade existente entre o vértice (resp. aresta) i do modelo e o vértice (resp. aresta) j da imagem alvo a ser reconhecida. A similaridade é calculada de acordo com as funções descritas no Capítulo 3.

1.4 Objetivos e Contribuições

Este trabalho propõe um método para representação de imagens através de grafos de atributos utilizando uma segmentação com base na análise da textura e/ou da cor. A textura é descrita por um conjunto de medidas calculadas pelo descritor *Local Binary Pattern* (LBP), e a cor é descrita através de histogramas. A similaridade entre os grafos gerados é medida por funções criadas a partir das medidas de distâncias utilizadas, a saber distâncias probabilísticas na forma discreta e analítica.

Dentre os objetivos gerais a serem alcançados pelo trabalho, podem-se citar a escolha das

amostras de imagens para testes; a extração e seleção das medidas mais representativas dos objetos ou regiões das imagens, armazenando-as como atributos de vértices e/ou arestas; a criação das matrizes de similaridade entre vértices e arestas; a aplicação e avaliação da metodologia proposta. O processo de análise e segmentação de imagens requer a investigação de diversos parâmetros que afetam seu desempenho, tais como o tipo de propriedade utilizado para separar ou unir regiões, número de *pixels* utilizados para compor o descritor LBP, valores de limiar, tamanho da janela de vizinhança entre *pixels*, dimensão do vetor de características e muitos outros. Essa avaliação deve seguir critérios rigorosos para permitir a redução dos custos computacionais e a geração de resultados mais precisos, possibilitando sua aplicação em uma maior variedade de imagens.

1.5 Organização do Texto

No Capítulo 2 será apresentado o estado da arte em segmentação de imagens, análise de texturas e redução e análise de dimensionalidade, fazendo uma revisão dos principais fundamentos e abordagens relacionados. O Capítulo 3 descreve o método proposto para representação da imagem através de grafos, explicando as fases de segmentação e extração de imagens, e as funções de agregação utilizadas para a criação das matrizes de similaridade. Também são descritos nesse capítulo os experimentos computacionais realizados, utilizando o ortofotomosaico do Rio Jucu. Finalmente, no Capítulo 4 serão discutidos os resultados obtidos, as possíveis contribuições, vantagens ou desvantagens e limitações deste trabalho, além de se tecerem comentários e conclusões. Também serão citados alguns pontos que poderiam ser alterados e algumas opções e sugestões para trabalhos futuros.

2 *Estado da Arte*

O primeiro passo para proceder a análise de imagens é a segmentação, que consiste em definir, na imagem, recortes automáticos para identificar objetos ou regiões de interesse, a fim de se obter uma interpretação semântica da cena de acordo com a aplicação. A segmentação subdivide uma imagem em suas partes ou objetos constituintes. O nível até o qual essa subdivisão deve ser realizada, assim como a técnica utilizada, depende do problema que está sendo resolvido. Talvez a característica mais importante de um método de segmentação seja a definição do que é uma região. Esta pode ser definida como um conjunto de *pixels* conectados por meio de uma condição de uniformidade. Uma característica muito utilizada na avaliação dessa condição é a análise de texturas. A textura, presente na maioria das imagens naturais, é fundamental na visão humana, pois contribui na melhoria da exatidão do processo de reconhecimento e classificação de imagens. Essa condição também pode ser estabelecida não por um, mas por vários fatores. Nesse ponto, pode ser necessária a aplicação de extratores ou seletores de características para redução da dimensionalidade do mesmo, a fim de se obter as características mais representativas da imagem.

Nesse contexto uma série de trabalhos tem sido desenvolvidos utilizando diferentes técnicas de segmentação automática, análise de texturas e definição das características mais relevantes em consideração, para representação e interpretação de imagens. Essas técnicas envolvem não somente diferentes modelos de tratamento de dados, mas também englobam diferentes métodos de representação da informação relevante para a tarefa de interpretação.

Portanto, este capítulo aborda o estado da arte das técnicas de segmentação de imagens, análise de texturas e redução de dimensionalidade. O capítulo inicia com técnicas de segmentação de imagens, seguindo para as abordagens mais utilizadas na análise de texturas. Por último uma descrição dos métodos de extração e seleção de características para redução de dimensionalidade é apresentada.

2.1 Técnicas de Segmentação de Imagens

Uma das etapas mais críticas no processo de reduzir a informação das imagens é a segmentação, que consiste em dividir a imagem em regiões que provavelmente correspondem a unidades estruturais na cena ou distinguir objetos de interesse [47], tendo por objetivo simplificar e/ou alterar a representação da imagem em algo que seja mais significativo e fácil de se analisar. Técnicas de segmentação de imagens são tipicamente utilizadas para localizar objetos e bordas (linhas, curvas, etc.) em imagens. Mais precisamente, a segmentação é um processo que classifica cada *pixel* da imagem, tal que *pixels* com o mesma classificação dividem certas propriedades, como cor, luminância ou textura [48], tendo por resultado um conjunto de segmentos que coletivamente cobrem toda a imagem, ou um conjunto de contornos extraídos da imagem. Regiões adjacentes são significativamente diferentes com respeito às mesmas propriedades, de onde se pode extrair objetos e outras entidades utilizadas na análise subsequente da imagem, como classificação, descrição e reconhecimento de objetos, extração de características, entre outros.

Segundo Gonzalez [1], os métodos de segmentação são classificados em três categorias básicas:

- Limiarização;
- Segmentação baseada em detecção de descontinuidades;
- Segmentação baseada em regiões.

Cada técnica pode ser associada a algum grau de intervenção humana, podendo ser obtida de forma manual, semiautomática e automática. A segmentação manual de uma imagem tem como principal vantagem a possibilidade de contar com outros dados (percepção humana, pesquisa de campo, etc.) além dos contidos na imagem. A segmentação manual de uma imagem não é única, e diferentes indivíduos podem gerar diferentes resultados e um mesmo indivíduo pode gerar resultados diferentes em tempos diferentes. Nos algoritmos de segmentação semiautomáticos, a intervenção manual é utilizada para fornecer pontos característicos da estrutura a ser segmentada ou para delimitar uma região onde ela é encontrada. Na segmentação automática não há intervenção humana. Entretanto, pode haver intervenção para eventuais correções da segmentação obtida. A segmentação feita automaticamente (a partir de algoritmos desenvolvidos para este propósito) baseia-se primariamente em técnicas de reconhecimento de similaridade ou de diferenças mensuráveis entre regiões adjacentes.

Há inúmeras técnicas disponíveis para segmentação, sendo que a natureza do problema define qual delas é a mais adequada. Assim, existem diversas técnicas de segmentação de imagens, mas não existe nenhum método único que seja capaz de segmentar todos os tipos de imagem [1, 49]. Esta área representa até hoje uma linha de pesquisa importante do processamento de imagens, principalmente por ela estar na base de todo o processamento da informação em uma imagem.

Neste trabalho, o algoritmo utilizado na fase de segmentação automática foi o *split-and-merge*. Uma vantagem deste método é a independência do descritor da região e medida de similaridade que permite a realização de experimentos com uma grande variedade de diferentes critérios. A seguir, as três classes de segmentação que compõem a abordagem adotada por [1] são brevemente apresentadas.

2.1.1 Limiarização

A limiarização é o método mais simples e intuitivo de segmentação de imagens. Basicamente, é estabelecido um limiar para separar alvos distintos de uma imagem e todos os *pixels* que estão dentro de uma faixa de intensidade são classificados como pertencentes a uma mesma região.

A limiarização pode ser entendida como uma operação que envolve testes em uma função da forma $T = T[x, y, p(x, y), f(x, y)]$, onde $f(x, y)$ é o nível de cinza no ponto (x, y) e $p(x, y)$ denota alguma propriedade local deste ponto, por exemplo, a média dos níveis de cinza de uma vizinhança centrada no ponto (x, y) . Deste modo, uma imagem limiarizada $g(x, y)$ pode ser definida como:

$$g(x, y) = \begin{cases} 1, & \text{se } f(x, y) > T; \\ 0, & \text{se } f(x, y) \leq T \end{cases} \quad (2.1)$$

Desse modo, *pixels* rotulados como 1 correspondem a alvos (áreas de objetos de interesse) enquanto os rotulados como 0 correspondem ao fundo (*background*). A segmentação é, então, conduzida analisando a imagem *pixel* por *pixel* e rotulando cada um como alvo ou fundo em função do limiar selecionado [1].

Apesar de possuir a desvantagem de ser sensível ao ruído (imperfeições inesperadas em uma imagem) e não levar em consideração informações espaciais ou conhecimento *a priori*, técnicas baseadas em limiarização são muito utilizadas para segmentação de imagens por serem computacionalmente simples em relação a outras técnicas, podendo ser empregada como o

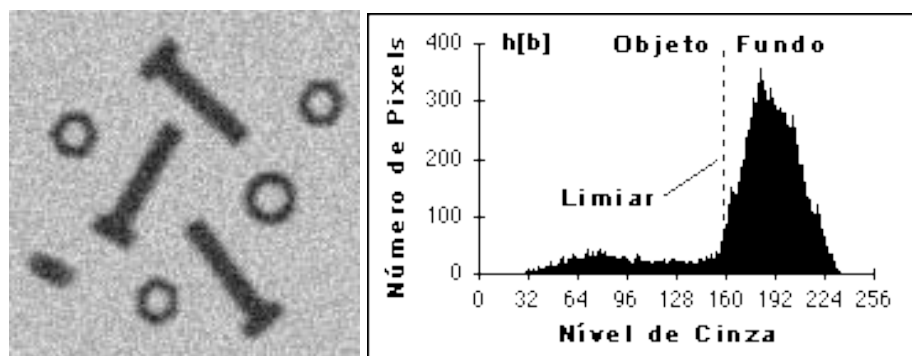


Figura 2.1: 2.1(a) Imagem a ser segmentada. 2.1(b) Histograma do nível de cinza da imagem. *Pixels* abaixo do limiar serão rotulados como *pixels* do objeto de interesse e aqueles acima do limiar serão rotulados como fundo.

primeiro passo de um processo mais elaborado de segmentação [50]. Uma imagem, composta por dois objetos distintos, isto é, cada objeto possui uma imagem com níveis de cinza bem diferenciados, pode ser facilmente segmentada pela utilização de um simples valor de limiar. Este limiar teria o valor intermediário entre as concentrações de nível de cinza de cada objeto.

A técnica de limiarização está intimamente relacionada ao conceito de histograma. Um histograma consiste em uma manifestação discreta de uma função de densidade de probabilidade sendo representado por um vetor $h(n)$ cujos valores para cada índice n correspondem à quantidade de vezes que o nível de cinza n ocorre na imagem. As informações contidas no histograma podem auxiliar na definição de um limiar. Por exemplo, picos no histograma frequentemente identificam várias regiões homogêneas e um limiar pode ser aplicado entre esse picos [47]. Este fato é ilustrado pela Figura 2.1, onde o pico encontra-se entre os valores 160 e 224, isto é, há vários *pixels* cujos níveis de cinza estão concentrados nessa faixa de valores. Assim, tomando-se o valor 160 como limiar, será possível identificar a grande maioria desses objetos.

2.1.2 Segmentação baseada em detecção de descontinuidades

Os métodos de segmentação por detecção de descontinuidades envolvem a localização de regiões da imagem onde a variação dos tons de cinza ocorre de maneira relativamente abrupta e podem ocorrer na forma de pontos isolados, linhas, segmentos ou curvas. A partir delas são formados os contornos ou bordas dos objetos contidos na imagem. Sua detecção pode ser feita automaticamente por operadores baseados em derivadas locais, os quais são implementados usando-se janelas (máscaras) de tamanhos variados.

A existência de descontinuidades é característica de um conjunto limitado de imagens. Em

muitas delas, a transição de uma região para outra ocorre de maneira tão sutil que torna a aplicação dos métodos de detecção de bordas uma opção inviável. Por vezes, estas imagens necessitam de operadores específicos que realcem as bordas antes da segmentação [1].

A seguir serão apresentados três métodos de detecção de discontinuidades: detecção de pontos, linhas e bordas.

Detecção de Pontos

A detecção e segmentação de pontos isolados é aplicada principalmente na análise de partículas [1]. A formulação mede as diferenças ponderadas entre o ponto central e seus vizinhos. A ideia é que o nível de cinza de um ponto isolado seja completamente diferente do nível de cinza de seus vizinhos. Basicamente, isso consiste na convolução espacial de uma máscara 3×3 (ver Figura 2.2) com a imagem, de modo que a intensidade de *pixel* resultante é dada pela soma dos produtos mostrada na Equação 2.2, que define a convolução espacial de uma máscara com uma imagem.

$$R = w_1z_1 + w_2z_2 + \dots + w_9z_9 = \sum_{i=1}^9 w_i z_i \quad (2.2)$$

onde z_i é o nível de cinza do *pixel* associado com o coeficiente w_i da máscara.

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

(a) Uma máscara 3×3 genérica.

-1	-1	-1
-1	8	-1
-1	-1	-1

(b) Máscara utilizada para detecção de pontos isolados a partir de um fundo constante.

Figura 2.2: Máscara utilizada para detecção de pontos.

Através da máscara utilizada na Figura 2.2(a), é possível dizer que um ponto foi detectado se $|R| > T$, em que T é um limiar não negativo e R é dado pela Equação 2.2. Como é possível deduzir, a partir da Equação 2.2 e do exemplo da Figura 2.2, numa área de nível de cinza constante o resultado dessa operação será 0. Por outro lado, se a máscara está centrada num ponto isolado, cuja intensidade é maior que a intensidade de pano de fundo, o resultado será diferente de 0.

Detecção de Linhas

O processo para detecção de linhas é similar ao processo de detecção de pontos. Considere a máscara mostrada na Figura 2.3. Se a primeira máscara fosse movida por todas as partes da imagem, ou seja, se fosse realizada a convolução da mesma como descrito na Equação 2.2, ela deveria responder mais fortemente a linhas (largura em *pixel*) orientadas horizontalmente. Com o fundo constante, uma resposta máxima deveria ocorrer quando a linha passar pela linha do meio da máscara (2, 2, 2)..

-1	-1	-1	-1	-1	2	-1	2	-1	2	-1	-1
2	2	2	-1	2	-1	-1	2	-1	-1	2	-1
-1	-1	-1	2	-1	-1	-1	2	-1	-1	-1	2
(a) Horizontal			(b) +45°			(c) Vertical			(d) -45°		

Figura 2.3: Máscaras para detecção de linhas.

Da mesma forma, a segunda máscara responde melhor a linhas orientadas a 45°, a terceira máscara a linhas verticais e a quarta máscara a linhas na direção -45° [1]. Para determinar a direção de determinada linha, basta convoluir as quatro máscaras e verificar qual delas ofereceu a maior intensidade em módulo.

Detecção de Bordas

Embora a detecção de pontos e linhas sejam elementos de qualquer discussão sobre segmentação, a detecção de bordas é a abordagem mais comum para detectar discontinuidades significativas em nível de cinza. A razão disso é que pontos isolados e linhas finas não são ocorrências frequentes na maioria das aplicações de interesse prático [1].

Basicamente, os métodos de detecção de bordas se baseiam nos conceitos matemáticos de Gradiente e Laplaciano, que são operadores de derivação local. Considerando a imagem digital como uma função de duas variáveis x e y , pode-se a partir da análise da primeira derivada desta função encontrar transições, visto que, para as regiões da imagem onde o valor de intensidade dos *pixels* é constante, a derivada será zero e, para as bordas, a derivada deverá assumir um valor positivo ou negativo. O valor da segunda derivada pode ser utilizado para se distinguir em que sentido está ocorrendo a variação, que pode ser de altos valores de nível de cinza para baixos ou o contrário.

Operadores de Gradiente

Alguns métodos para detectar bordas se apoiam no conceito matemático de *gradiente*. Dada uma função $g(x, y)$ que seja contínua e admita variáveis parciais em (x_0, y_0) , o vetor gradiente

da função g (intensidade) no ponto citado é dado pela seguinte equação:

$$\vec{\nabla} g(x_0, y_0) = \left(\frac{\partial g}{\partial x}(x_0, y_0), \frac{\partial g}{\partial y}(x_0, y_0) \right) \quad (2.3)$$

O vetor $\vec{\nabla} g(x_0, y_0)$ aponta para a direção de taxa máxima de variação da função g no ponto (x_0, y_0) . Para detecção de bordas, o interesse se concentra na magnitude desse vetor, muitas vezes denotado simplesmente como *gradiente*, dado pela seguinte equação:

$$|\vec{\nabla} g(x_0, y_0)| = \sqrt{\left(\frac{\partial g}{\partial x}(x_0, y_0) \right)^2 + \left(\frac{\partial g}{\partial y}(x_0, y_0) \right)^2} \quad (2.4)$$

A direção do vetor gradiente é também uma medida importante que pode ser usada para a análise e orientação da borda, um importante problema em textura e análise de movimento [51]. A direção do vetor é obtida pela equação:

$$\alpha(x_0, y_0) = \arctan \left(\frac{\frac{\partial g}{\partial y}(x_0, y_0)}{\frac{\partial g}{\partial x}(x_0, y_0)} \right) \quad (2.5)$$

No domínio discreto de imagens, o cálculo do *gradiente* pode ser feito pelos operadores de Sobel. Basicamente, esse método consiste em duas máscaras de convolução que calculam o vetor gradiente na direção x e y , conforme demonstra a Figura 2.4.

-1	-2	-1	-1	0	1
0	0	0	-2	0	2
1	2	1	-1	0	1

(a) Gradiente na direção x (b) Gradiente na direção y

Figura 2.4: Operadores de Sobel [1].

Há outros métodos que detectam bordas por gradiente, tais como os operadores de Roberts [1], Prewitt [1], etc., mas estes seguem basicamente a mesma ideia dos operadores de Sobel, de modo que não serão descritos neste trabalho.

Operador Laplaciano

O operador Laplaciano [1] é um operador derivativo de segunda ordem definido para a função contínua $g(x, y)$ nos pontos (x_0, y_0) descrito pela seguinte equação:

$$\vec{\nabla}^2 g(x_0, y_0) = \left(\frac{\partial^2 g}{\partial x^2}(x_0, y_0), \frac{\partial^2 g}{\partial y^2}(x_0, y_0) \right) \quad (2.6)$$

No domínio discreto, o operador Laplaciano pode ser representado pela máscara presente na Figura 2.5. Como é possível perceber, o Laplaciano resulta no valor 0 para áreas de nível de cinza constante.

0	-1	0
-1	4	-1
0	-1	0

Figura 2.5: Máscara usada para computar o Laplaciano.

Embora esse operador seja sensível a transições de intensidade, raramente ele é usado na forma citada acima para detecção de bordas. A razão é que, por ser um operador de derivada de segunda ordem, ele é muito sensível a ruídos na imagem. Para contornar esse problema, esse operador pode ser combinado com uma função Gaussiana, que é um filtro de suavização, gerando assim um Laplaciano de Gaussiana (LoG) [1], conforme definido na equação abaixo.

$$\vec{\nabla}^2 g(x_0, y_0) = \left(1 - \frac{x_0^2 + y_0^2}{\sigma^2}\right) \exp\left(-\frac{x_0^2 + y_0^2}{2\sigma^2}\right) \quad (2.7)$$

Os algoritmos de segmentação baseados em bordas geralmente utilizam mecanismos capazes de interligar os segmentos obtidos na fase inicial de detecção. O objetivo é produzir formas e curvas significativas, de modo que, por meio destas, seja possível o estudo das características geométricas dos objetos. No entanto, esta pode não ser uma tarefa muito simples. Um pequeno ruído encontrado na imagem pode, muitas vezes, conduzir o algoritmo à produção de curvas desconexas e não significativas, impossibilitando a conexão dos pontos. Após a detecção das discontinuidades, é necessário, normalmente, a aplicação de algum método capaz de conectar tais fragmentos e gerar contornos que estejam associados com os contornos reais dos objetos. Estes métodos são denominados algoritmos de enlace e na maioria das vezes eles fazem uso das técnicas de percurso em grafos.

2.1.3 Segmentação baseada em regiões

Enquanto os métodos de limiarização e de detecção de bordas resolvem o problema de segmentação encontrando diferenças nas tonalidades dos *pixels* ou conjunto de *pixels*, os métodos baseados em regiões abordam a procura de similaridades entre elas. Neste caso, a segmentação é realizada unindo-se regiões adjacentes se as mesmas forem similares ou dividindo uma região em regiões menores se a mesma não é homogênea.

O método de segmentação orientado à regiões deve proceder da seguinte forma: cada *pixel* deve pertencer a uma região, e todos os *pixels* de uma mesma região devem ser conexos e satis-

fazer a uma propriedade. As regiões devem ser disjuntas e diferentes em relação à propriedade analisada.

Esse tipo de segmentação divide a imagem procurando por regiões que atendam a algum critério de similaridade. Em geral, uma imagem segmentada é formada por um número n de regiões da imagem, sendo que a união dessas regiões compõem a imagem completa. As técnicas mais conhecidas desta categoria são denominadas Crescimento de Região (*Region Growing*) e Divisão e Fusão (*Split-And-Merge*).

Crescimento de Região

Segmentações produzidas por algoritmos que utilizam técnicas de crescimento de regiões empregam um procedimento que consiste em agrupar *pixels* ou sub-regiões em regiões maiores. Nesta técnica, a segmentação se inicia a partir de um conjunto inicial de pontos (sementes), agregando-se a cada um deles novos *pixels* vizinhos que contenham propriedades similares, tais como cor, textura ou nível de cinza. Um critério simples de agregação para novos *pixels* é o módulo da diferença entre os tons de cinza dos *pixels* em questão, ou seja, se este valor for menor que um determinado limiar, então o novo *pixel* é agregado à região. A seleção do critério de similaridade depende da aplicação considerada e do tipo de dado disponível.

Alguns problemas desta técnica são a seleção das partições iniciais, a seleção de propriedades convenientes para aumentar (ou agrupar) essas partições e a formulação de uma regra de término para o processo. Outro problema identificado é que toda vez que um novo *pixel* é acrescentado a um segmento da imagem, a característica deste segmento é redefinida. A adição de um *pixel* a um segmento é dependente do limiar de similaridade adotado. *Pixels* situados em zonas de fronteira entre duas regiões distintas podem ter características impuras, ou seja, seus valores podem ser uma combinação dos valores que definem essas regiões. Deste modo, se o limiar estabelecido é muito alto, o crescimento de regiões vai deixar muitos desses *pixels* como segmentos diminutos. Entretanto, se o limiar é baixo, segmentos que representam classes distintas serão agrupados como pertencentes à mesma região.

Divisão e Fusão (*Split-And-Merge*)

Segundo [1], uma abordagem para segmentar uma imagem através desta técnica consiste em subdividir a imagem de entrada sucessivamente em quadrantes que satisfaçam certa propriedade. Este método é uma alternativa de segmentação baseada em crescimento de regiões que não utiliza um conjunto inicial de pontos (sementes) para a resolução do problema [1].

A imagem é inicialmente dividida em quadrantes. Para cada quadrante, verifica-se se a propriedade escolhida é válida ou não. Se um dado quadrante não satisfaz a propriedade, ele é subdividido em subquadrantes e assim por diante (fase de Divisão). Em uma segunda etapa, é realizado o agrupamento de regiões adjacentes que são similares (fase de Fusão), isto é, que satisfazem a propriedade definida. O processo termina quando não for mais possível realizar nenhuma divisão ou agrupamento.

Em [1] é apresentado um procedimento para essa técnica. Supondo que R representa a imagem inteira, o processo de segmentação consiste em particionar R em n sub-regiões, R_1, R_2, \dots, R_n , tal que:

- I. $\bigcup_{i=1}^n R_i = R$;
- II. R_i é uma região conectada, $i = 1, 2, 3, \dots, n$;
- III. $R_i \cap R_j = \emptyset$ para todo i e j , $i \neq j$;
- IV. $P(R_i) = \text{VERDADEIRO}$ para $i = 1, 2, 3, \dots, n$; e
- V. $P(R_i \cup R_j) = \text{FALSO}$ para $i \neq j$

onde $P(R_i)$ é um predicado lógico definido sobre os pontos no conjunto R_i e \emptyset representa o conjunto vazio.

O Passo I indica que a segmentação deve ser completa, isto é, todo *pixel* deve estar na região. O Passo II requer que os pontos na região devem ser conectados. O Passo III indica que as regiões devem ser distintas. O Passo IV lida com as propriedades que devem ser satisfeitas pelos *pixels* numa região segmentada, por exemplo $P(R_i) = \text{VERDADEIRO}$ se todos os *pixels* em R_i tiverem a mesma intensidade. O Passo V indica que as regiões R_i e R_j são diferentes no sentido de satisfazer o predicado P .

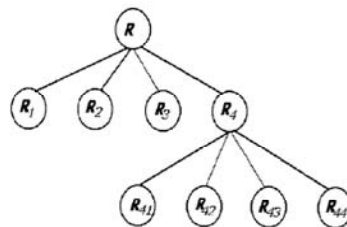


Figura 2.6: Árvore Quadrática (*Quadtree*) que representa a Região R .

Esta técnica particular de separação tem uma conveniente representação na forma *quadtree*, como representado na Figura 2.6. Nota-se que a raiz da árvore corresponde à imagem inteira e que cada nó corresponde a uma subdivisão.

0	1	0	0	7	7	7	7
1	0	2	2	7	7	7	7
0	2	2	2	7	7	7	7
4	4	2	2	7	7	7	7
0	0	1	1	3	3	7	7
1	1	2	2	3	7	7	7
2	4	3	0	5	7	7	7
2	3	3	5	5	0	7	7

(a) Imagem inteira.

0	1	0	0	7	7	7	7
1	0	2	2	7	7	7	7
0	2	2	2	7	7	7	7
4	4	2	2	7	7	7	7
0	0	1	1	3	3	7	7
1	1	2	2	3	7	7	7
2	4	3	0	5	7	7	7
2	3	3	5	5	0	7	7

(b) Divisão da imagem não homogênea em 4 regiões.

0	1	0	0	7	7	7	7
1	0	2	2	7	7	7	7
0	2	2	2	7	7	7	7
4	4	2	2	7	7	7	7
0	0	1	1	3	3	7	7
1	1	2	2	3	7	7	7
2	4	3	0	5	7	7	7
2	3	3	5	5	0	7	7

(c) Divisão de regiões não homogêneas em 4 regiões.

0	1	0	0	7	7	7	7
1	0	2	2	7	7	7	7
0	2	2	2	7	7	7	7
4	4	2	2	7	7	7	7
0	0	1	1	3	3	7	7
1	1	2	2	3	7	7	7
2	4	3	0	5	7	7	7
2	3	3	5	5	0	7	7

(d) Divisão de regiões não homogêneas em 4 regiões.

0	1	0	0	7	7	7	7
1	0	2	2	7	7	7	7
0	2	2	2	7	7	7	7
4	4	2	2	7	7	7	7
0	0	1	1	3	3	7	7
1	1	2	2	3	7	7	7
2	4	3	0	5	7	7	7
2	3	3	5	5	0	7	7

(e) Resultado da segmentação após fusão de regiões homogêneas.

Figura 2.7: Passos seguidos para a segmentação de uma imagem utilizando o método *split-and-merge*.

A Figura 2.7 ilustra o procedimento realizado pelo algoritmo *split-and-merge*. Suponha que R representa a imagem inteira, como mostra a Figura 2.7(a) e $P(R_i) = \text{VERDADEIRO}$ se todos os *pixels* em R_i possuírem a mesma intensidade. Portanto, para a completa região R tem-se que $P(R) = \text{FALSO}$, de maneira que a imagem deve ser subdividida, como mostra a Figura 2.7(b). No próximo passo, apenas a região mais acima e à direita satisfaz o predicado, não sendo alterada, enquanto as outras três regiões são divididas em subquadrantes como mostrado na Figura 2.7(c). O passo é realizado mais uma vez, como mostrado na Figura 2.7(d). Nesse ponto, várias regiões podem ser fundidas, resultando na segmentação final apresentada na Figura 2.7(e). Finalmente, todas as regiões que compõem a imagem satisfazem ao predicado P .

Na próxima seção serão apresentados brevemente alguns descritores de textura.

2.2 Análise de Texturas

Uma das tarefas mais complexas presente na análise de imagens está na definição de um conjunto de características capazes de descrever de maneira efetiva cada região contida em uma imagem. Para isso, uma abordagem natural está em recorrer às características utilizadas pelo sistema visual humano na interpretação de informações visuais. A textura encontra-se dentre essas características, e embora não tenha um conceito amplamente aceito, podemos interpretá-la como característica e aparência tátil de superfície para reconhecer a forma e do que é feito um objeto. Também pode ser formada por uma única superfície através de variações na forma, iluminação, sombras, absorção e reflectância. Dessa maneira, a utilização de informações texturais se apresenta como mais uma abordagem para a descrição de regiões da imagem. Nas imagens digitais, as características de uma textura podem ser detectados através de variações na captura de intensidade ou cor. Embora, em geral, não haja informações sobre a causa das variações, as diferenças entre os *pixels* da imagem fornecem um meio prático de analisar as propriedades texturais dos objetos.

Apesar de o sistema visual humano apresentar facilidade no reconhecimento e descrição de texturas, é extremamente difícil formalizar sua definição ou desenvolver um conjunto de descritores que possam ser utilizados para análise de imagens em diferentes domínios de aplicações. Infelizmente não há definição formal de textura digital em termos matemáticos, apesar de encontrar muitas aplicações desta característica na área de visão computacional [52]. A falta de uma teoria torna o problema de análise de texturas mal colocados do ponto de vista matemático, onde métodos de análise não podem ser provados corretos ou errados. Assim, a avaliação deve ser realizada de uma maneira empírica. Contudo, medidas que utilizam textura têm sido utilizadas com sucesso em muitas tarefas de visão computacional [53, 54, 55, 56, 57].

Gonzalez [1] cita a existência de três principais abordagens para se descrever texturas: espectral, estrutural e estatística. A abordagem espectral diz respeito a propriedades baseadas no espectro de Fourier [19], onde pode ser detectada a existência de padrões periódicos ou semiperiódicos. A abordagem estrutural representa textura como sendo formada pela repetição de padrões que obedeçam alguma regra de posicionamento para a sua geração, introduzindo o conceito de textura primitiva, muitas vezes chamados *textels* ou *textons*. Para descrever uma textura são necessários um vocabulário de *textels* e a descrição das suas relações. O objetivo é descrever a complexidade das estruturas com primitivas simples, por exemplo, através de grafos. Modelos baseados em primitivas foram amplamente usados para explicar a percepção humana de texturas [58, 59]. Já na abordagem estatística, a descrição de texturas considera o relacionamento, a distribuição dos tons de cinza e o interrelacionamento entre eles, tratando

texturas como fenômenos estatísticos. Estatísticas de Diferença de histogramas e coocorrência, pesquisadas por [60, 61], podem servir como exemplos simples de medidas de texturas estatística.

Métodos de análise de textura têm sido desenvolvidos com a escala de cinza das imagens, intuitivamente por boas razões. Os seres humanos podem facilmente captar as texturas sobre uma superfície, mesmo com nenhuma informação de cor. A Figura 2.8 mostra uma fotografia de massa tricolor e sua versão escala de cinza. A única coisa que não pode ser inferida, com base na escala de cinza, é a cor das massas. A textura em si é a mesma. O sistema visual humano é capaz de interpretar cenas acromáticas, por exemplo em baixos níveis de iluminação.

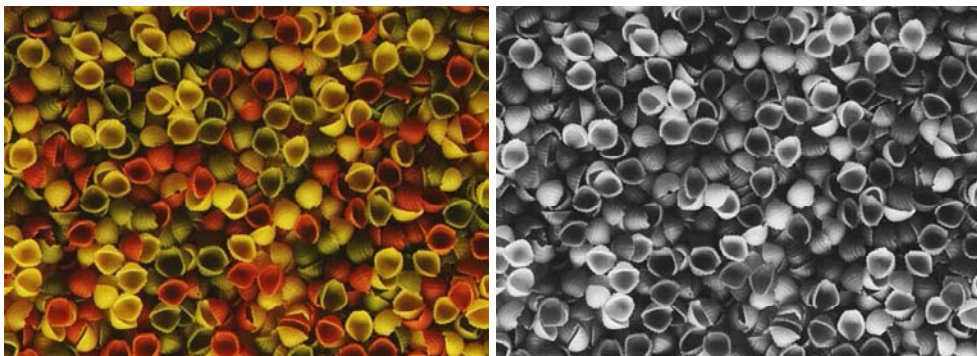


Figura 2.8: Textura colorida e sua versão em escala de cinza.

A informação de cor atua como uma ajuda para possibilitar interpretações mais ricas. Mesmo quando essa informação é distorcida, por exemplo devido ao daltonismo, o sistema visual ainda funciona. Intuitivamente, isso sugere que, pelo menos para o nosso sistema visual, a cor e a textura são fenômenos distintos. A discussão sobre separar a informação de cor e textura não é baseada em pura intuição. A investigação sobre o sistema visual humano tem proporcionado muitas evidências de que o sinal da imagem é, na verdade, composto por componentes de luminância e crominância. Ambos são processados por vias distintas [26, 62], embora existam algumas interações secundárias entre as vias. Os estudos psicofísicos de [27] também sugerem que informações de cor e textura são processadas separadamente.

Mojsilovic [63] em seu artigo sobre o vocabulário e a gramática de padrão de cores, sugere que a percepção global de padrões de cores é formada através da interação das componentes de luminância, crominância e de um padrão acromático. Os componentes de luminância e crominância são utilizados na extração de informação baseadas em cor, enquanto o componente de padrão acromático é utilizado como padrão de informação de textura. O artigo conclui que a percepção humana de padrões não está relacionada ao conteúdo de uma imagem colorida. As dimensões mais fortes são "predominância de cor"(presença/ausência de cor dominante) e "pureza de cor"(grau de intensidade), indicando que, em um primeiro momento, as pessoas

usam principalmente informações de cores para julgar similaridade.

Uma série de abordagens distintas utilizando a cor e a informação de textura tem sido propostas. Por exemplo, [64] extraiu características de textura a partir de uma imagem em nível de cinza, enquanto as medidas derivadas de histogramas de cor foram utilizadas para descrição de cor. Um problema de classificação de granito foi utilizado como teste para o método. Em [28] é proposto um método de segmentação de imagens aéreas, no qual são computados independentemente espaços de cor e textura. Então, a segmentação final é obtida pela decisão tomada por cada classificador (cor ou textura isoladamente).

Em algumas aplicações, como inspeção de papel e têxteis, a textura deve ser utilizada por não existir informação de cor [65, 66]. No entanto, em muitas aplicações, incluindo algumas tarefas de inspeção visual, a cor é a principal fonte de informação. Descritores de cor foram utilizados para inspecionar madeira, cerâmica, alimentos, etc [67, 68]. Às vezes, porém, elementos de cor por si só não fornecem informações suficientemente precisas. Imagens com condições desiguais de iluminação, similarmente coloridas, mas de superfícies texturizadas de maneira diferente, sugerem que a segmentação seria melhor conduzida com o suporte adicional dos métodos de análise de textura.

Dentre as principais aplicações que utilizam análise de texturas estão a classificação, que objetiva a criação de um mapa onde cada região homogênea é identificada como pertencente a uma determinada classe; a segmentação, visando à determinação das bordas entre as diferentes regiões texturizadas contidas em uma imagem e; finalmente, a síntese de texturas, responsável pela determinação de um modelo capaz de gerar uma dada textura.

Normalmente, os métodos que efetuam análise de texturas são obtidos através dos processos de extração e seleção de características. A extração é responsável por executar transformações nos dados de entrada, de modo a descrevê-los de maneira representativa e simplificada, e a seleção visa reduzir o número de características, bem como eliminar aquelas que apresentem redundância. Uma forma de representar as texturas é através do uso de descritores de textura, conforme apresentado a seguir. Os descritores de textura utilizam alguma informação da imagem para tentar representá-las. Alguns se apresentam na forma de matrizes, outros como histogramas modificados para tentar representar as imagens. Como exemplos, existem a Matriz de Coocorrência de Tons de Cinza (GLCM - *Grey Level Co-occurrence Matrix*), o LBP (*Local Binary Pattern*), as unidades de textura e as primitivas de texturas (textons).

2.2.1 Matriz de Coocorrência de Tons de Cinza

A Matriz de coocorrência de tons de cinza (GLCM - *Gray Level Co-Occurrence Matrix*) foi primeiramente descrita por [60], e é baseada na realização de uma varredura na imagem, mantendo-se a par do comportamento da periodicidade dos *pixels* que diferem na intensidade e que são separados por uma distância d fixa. Normalmente, a direção entre dois *pixels* também é importante para classificação a partir de textura, e esta é implementada por matrizes múltiplas, a partir das quais são extraídas medidas utilizadas para análise das texturas, sendo uma para cada região de interesse. Em geral, são utilizadas quatro direções: vertical, horizontal e duas diagonais (0, 45, 90 e 135 graus, respectivamente). Para cada valor de d temos quatro matrizes, cada uma com uma dimensão 256×256 *pixels*, para uma imagem original com 256 tons de cinza. Ou seja, ela nada mais é que uma tabulação sobre o número de combinações distintas de valores de intensidade que ocorrem em uma imagem [37].

As matrizes de coocorrência levam em consideração a relação entre dois *pixels* por vez, sendo o primeiro chamado de *pixel* de referência e o segundo chamado de *pixel* vizinho. Tem-se que o elemento $P(m,n)$ da matriz de coocorrência representa o número de transições entre os níveis de cinza m e n presentes na textura [69].

Antes da formação da matriz, são determinados os *pixels* e relacionamentos espaciais a serem considerados. Constrói-se um conjunto S , no qual cada elemento é composto de dois pares ordenados denotando cada *pixel* envolvido na relação espacial. Em seguida, é utilizada a relação para determinar o número de transições que ocorrem entre cada par de tons de cinza contidos na textura, onde $f(x,y)$ indica o tom de cinza do *pixel* localizado em (x,y) .

$$P(m,n) = \{((i,j), (k,l)) \in S \mid f(i,j) = m \text{ e } f(k,l) = n\} \quad (2.8)$$

onde (k,l) e (i,j) são as coordenadas dos respectivos *pixels*.

Como exemplo, consideremos que o *pixel* vizinho pode estar à direita de cada *pixel* de referência. Isto pode ser descrito como uma relação (1,0): 1 *pixels* na direção i , 0 *pixels* na direção j . Cada *pixel* na imagem se torna um *pixel* de referência, iniciando no canto superior esquerdo indo até o canto inferior direito. Note que os *pixels* situados na margem direita não possuem vizinhos na sua direita. Portanto, estes não serão utilizados para esta contagem. A Tabela 2.1 exemplifica uma matriz de coocorrência para 4 níveis de cinza.

Uma vez determinado o número de ocorrências de cada uma das transições de níveis de cinza, basta acrescentar o valor $P(m,n)$ na m -ésima linha e n -ésima coluna da matriz, obtendo-se

	0	1	2	3
0	P(0,0)	P(0,1)	P(0,2)	P(0,3)
1	P(1,0)	P(1,1)	P(1,2)	P(1,3)
2	P(2,0)	P(2,1)	P(2,2)	P(2,3)
3	P(3,0)	P(3,1)	P(3,2)	P(3,3)

Tabela 2.1: Matriz de coocorrência, cada elemento da matriz diz respeito às transições de níveis de cinza.

então a matriz de coocorrência. Percebe-se claramente que as dimensões de tal matriz dependem do número de níveis de cinza contidos na textura. Consideremos uma imagem I com 3×3 pixels e com 4 níveis de cinza nas direções 0 e 90 graus e distância 1, conforme mostra a Figura 2.9.

0	1	2
3	3	0
2	2	1

Figura 2.9: Imagem com dimensão 3×3 pixels e quatro níveis de cinza.

Sua matriz de coocorrência $M(I, x, d)$ para a distância e ângulos especificados, seria dada pelas seguintes matrizes:

$$M(I, 0^\circ, 1) = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 2 & 0 \\ 0 & 2 & 2 & 0 \\ 1 & 0 & 0 & 2 \end{bmatrix}, \quad M(I, 90^\circ, 1) = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 2 \\ 1 & 1 & 2 & 0 \end{bmatrix}$$

Segundo [69], uma vez que obtivemos as matrizes de coocorrência, estas precisam ser normalizadas para que seja possível extrair os descritores. Esta normalização é realizada a partir da divisão de cada elemento da matriz original pela soma dos seus componentes.

Os descritores estatísticos são responsáveis por analisar uma dada matriz de coocorrência M para categorizar a textura de uma região sobre a qual M foi calculada. Um conjunto de descritores úteis para esse propósito incluem:

1. Probabilidade máxima
2. Momento de diferença de elementos de ordem k
3. Momento inverso de diferença de ordem k
4. Entropia

5. Uniformidade

A ideia básica está em caracterizar o conteúdo de M através desses descritores. Por exemplo, o primeiro descritor fornece uma resposta mais alta ao predicado P . O segundo descritor possui um valor relativamente baixo quando os valores altos de M estiverem próximos da diagonal principal. O terceiro descritor tem o efeito oposto. O quarto descritor é uma medida de aleatoriedade, atingindo seu valor mais alto quando todos os valores de M forem iguais. Por outro lado, o quinto descritor é menor quando os valores de c_{ij} forem iguais.

O LBP, descritor de textura utilizado neste trabalho, pode ser considerado como uma especialização do método da Matriz de coocorrência de tons de cinza. Uma comparação do LBP original, LBP/C, GLCM, GLD, e muitos outros métodos foram realizados por [30] e [70]. De acordo com [70], dos 22 operadores locais de textura considerados nos experimentos, o LBP foi geralmente superior a outras medidas, proporcionando excelentes resultados especialmente para texturas determinísticas. Esta é uma das motivações para se usar LBP neste trabalho.

2.2.2 Espectro de Textura

O trabalho descrito em [71] propõe o conceito de unidade de textura, baseado na ideia de que uma imagem texturizada pode ser considerada como um conjunto de pequenas unidades essenciais, denominada unidade de textura, as quais caracterizam a informação local de um dado *pixel* em relação aos seus vizinhos. Medidas extraídas a partir de todas as unidades presentes na imagem revelam o aspecto global da textura.

Seja uma vizinhança de tamanho 3×3 , composta pelos elementos de $V = \{v_0, v_1, \dots, v_8\}$, onde v_0 representa o tom de cinza do *pixel* central, e os outros v_i , os tons de cinza de seus vizinhos. Define-se unidade de textura (TU, *texture unit*) pelo conjunto $TU = \{e_1, e_2, \dots, e_8\}$, onde cada e_i é determinado através da Equação 2.9.

$$e_i = \begin{cases} 0, & \text{se } v_i < v_0 \\ 1, & \text{se } v_i = v_0, i = 1, 2, \dots, 8 \\ 2, & \text{se } v_i > v_0 \end{cases} \quad (2.9)$$

Baseado no fato de existirem $6561 (= 3^8)$ configurações possíveis para cada unidade de textura, cria-se uma assinatura conforme define a Equação 2.10. A distribuição de frequências das unidades de uma textura é denominada espectro de textura, onde o eixo das abscissas indica o NTU , número da unidade de textura. O eixo das ordenadas representa seu número de ocorrências.

$$NTU = \sum_{i=1}^8 3^{i-1} e_i \quad (2.10)$$

A partir dessa distribuição de frequências são extraídas as características de texturas, que acrescentadas em um vetor de características são utilizadas como discriminantes em métodos de segmentação ou classificação de regiões.

A Equação 2.10 define como deve ser calculada a unidade de textura para cada conjunto de *pixels* selecionado. No entanto, exceto para v_0 , não especifica quais *pixels* correspondem a cada elemento v_i . Se considerar apenas ordenações no sentido horário, um elemento v_i , com i fixo, pode assumir oito posições distintas. Portanto, as 6561 unidades de textura podem ser rotuladas, por meio da equação 2.10, de oito maneiras distintas.

Como alternativa ao *NTU*, Ojala [30] apresenta a versão binária da unidade de textura, denominada padrões binários locais (*LBP - Local Binary Patterns*), seguindo a definição mostrada na Equação 2.11.

$$e_i = \begin{cases} 0, & \text{se } v_i < v_0 \\ 1, & \text{se } v_i \geq v_0 \end{cases}, i = 1, 2, \dots, 8 \quad (2.11)$$

Considerando a alteração na definição dos elementos e_i , a Equação 2.11 passa a ser utilizada para o cálculo da *LBP*. Dessa maneira, o valor máximo que o *LBP* pode assumir é 255, efetuando uma redução significativa no número de entradas do espectro de textura. Combinando os valores de *LBP* com medidas de contraste e covariância, o trabalho descrito em [72] obtém a distribuição conjunta das características para cada textura.

$$NTU = \sum_{i=1}^8 2^{i-1} e_i \quad (2.12)$$

Isto faz com que a distribuição seja mais compacta se comparado com o uso de 2.9 e 2.10, não existindo diferença significativa no que diz respeito à precisão da classificação. A diferença na classificação foi, porém, na maior velocidade, pois o vetor de características *LBP* é mais de 25 vezes menor do que o de *NTU*. Além disso, o vetor de características de menor dimensão tem a vantagem adicional de obter, com pequenas texturas, estatísticas mais confiáveis.

2.2.3 Primitivas de textura (textons)

A abordagem estrutural utiliza a ideia de que texturas são compostas de primitivas dispostas de forma aproximadamente regular e repetitiva, de acordo com regras bem definidas [1, 73]. Assim, um padrão complexo é então expresso em termos do relacionamento entre suas primitivas. Em outras palavras, o reconhecimento sintático formula uma descrição hierárquica de padrões complexos, construída a partir de subpadrões mais simples, sendo que no nível mais baixo se encontram os elementos mais simples, extraídos dos dados de entrada que são as primitivas. Para a fundamentação da abordagem sintática faz-se uma analogia entre as estruturas do padrão e a teoria das linguagens formais. Os padrões são vistos como sentenças pertencentes a uma linguagem, as primitivas como seu alfabeto e as sentenças são geradas de acordo com a respectiva gramática da linguagem. A ideia principal é a de que um conjunto de padrões complexos pode ser descrito por meio de um número finito de primitivas e de regras gramaticais. As regras formadas por uma gramática propiciam a descrição de como se estruturam os objetos e os padrões em termos de suas primitivas [74].

Como exemplo, pode-se citar a descrição da textura baseada em linhas paralelas regularmente espaçadas. Para entender melhor essa abordagem, pode-se utilizar uma regra do tipo $S \rightarrow aSA$. Neste tipo de regra, as letras maiúsculas A e S são variáveis e as letras minúsculas a , b e c são símbolos terminais. Uma variável é escolhida como símbolo inicial para começar a construção da cadeia de caracteres. Supondo que além da regra acima, existam ainda as seguintes regras:

- $S \rightarrow aSA$
- $S \rightarrow bA$
- $S \rightarrow a$
- $A \rightarrow cA$
- $A \rightarrow c$
- $A \rightarrow b$

Um exemplo utilizando essas regras é agora apresentado. Seja S o símbolo inicial. Desta forma, pelas regras anteriores, o sinal \rightarrow indica que S pode ser substituído por aSA (caso seja utilizada a primeira regra), bA (caso seja utilizada a segunda regra) ou a (caso a terceira regra seja utilizada), ou seja, toda vez que existir uma variável em uma regra, esta poderá ser

substituída pelo conteúdo localizado do lado direito do símbolo \rightarrow . Caso uma variável possua mais de uma regra, como ocorre neste exemplo, qualquer uma das regras poderá ser utilizada na substituição. Com essas regras, podem-se gerar vários tipos de cadeias. Uma delas poderia ser, por exemplo, *aabbcccb*, que foi obtida pelos seguintes passos:

- *S* (regra inicial)
- *aSA* (regra 1)
- *aScA* (regra 4)
- *aaSAcA* (regra 1)
- *aaScAcA* (regra 4)
- *aaScAc b* (regra 6)
- *aaScccb* (regra 5)
- *aabAcccb* (regra 2)
- *aabbcccb* (regra 6) - Sequência final

Vale observar que, dependendo das regras, uma cadeia de símbolos terminais poderá ser gerada por diferentes combinações de regras. Além disso, uma regra poderá ser do tipo $S \rightarrow \emptyset$, que indica que a variável será eliminada da cadeia sem ser substituída por um símbolo terminal. Para construir uma textura utilizando a regra da cadeia, pode-se definir uma ação para cada símbolo terminal da cadeia gerada como, por exemplo, *a* círculo à direita, *b* círculo abaixo e *c* círculo à esquerda.

É importante ressaltar que, dependendo das regras e de sua combinação, estruturas irregulares podem ser geradas. Uma primitiva de textura simples pode ser usada na formação de padrões complexos de textura através de algumas regras que limitem o número de arranjos possíveis da primitiva. A vantagem da abordagem estrutural é que ela provê uma boa descrição simbólica da imagem. Entretanto, essa característica é normalmente mais útil em tarefas de síntese do que em análise de textura.

Textons foram originalmente utilizados por [58] como as unidades básicas de pré-discriminação de textura pelo olho humano. As texturas consideradas consistiam de primitivas de textura binárias separadas - orientação de elementos, cruzamentos e terminadores. O vocabulário

texton é criado uma vez, para um conjunto de texturas, e utilizado para descrever todas as texturas do conjunto. O LBP considera, por sua vez, cada *pixel* da vizinhança separadamente, não havendo necessidade de criar um vocabulário texton específico. Uma abordagem utilizando o filtro de Gabor em textons foi comparada com o método LBP por [75], aplicando os métodos em texturas 3-D. O LBP obteve um melhor desempenho, mesmo com um conjunto de texturas mais difícil e com um número significativamente menor de *overhead* computacional.

2.2.4 Local Binary Pattern

No LBP [31], a descrição da textura é feita por um histograma representando o relacionamento local entre os *pixels*. O LBP baseou-se no conceito de unidades de textura, descrito na Subseção 2.2.2. A denominação *Local Binary Pattern* enfatiza este aspecto do operador, isto é, a vizinhança local é limiarizada por um valor em escala de cinza do *pixel* central, para formar uma imagem de padrão binário [32]. O LBP original é produzido multiplicando os valores limiarizados por pesos dados aos *pixels* correspondentes e somando os resultados. Em [32] uma forma invariante do LBP com respeito ao nível de cinza é proposto, o $LBP_{P,R}$, definido como:

$$LBP_{P,R} = \sum_{p=0}^{P-1} 2^p S_p, \quad (2.13)$$

com

$$S_p = S(g_p - g_c) \begin{cases} 0, & \text{se } g_p < g_c \\ 1, & \text{se } g_p \geq g_c \end{cases} \quad (2.14)$$

onde P é o número de vizinhos do *pixel* posicionado em um círculo concêntrico de raio R , g_c corresponde ao valor do nível de cinza do *pixel* central com respeito à informação de textura e g_p é o valor em nível de cinza de cada vizinho.

Cada valor do histograma é obtido da seguinte maneira. Uma matriz binária de dimensões definidas pelo raio R e valores calculados de (2.14) em que os *pixels* das respectivas posições são ponderados por uma máscara em que cada posição p possui o valor 2^p (a posição do bit do código binário), produzindo, assim, uma sequência de bits de tamanho P . Consequentemente, o histograma LBP varia entre 0 e $2^P - 1$ onde o máximo valor possível ocorre quando todos os valores da matriz binária são iguais a um.

Seja $R = 1$ e $P = 8$. Neste exemplo, a aplicação do LBP pode ser interpretado como uma máscara 3×3 (Figura 2.10(a)) que é limiarizada de acordo com o valor do *pixel* central (Figura

2.10(b)), e a matriz resultante é multiplicada, elemento a elemento, por uma máscara formada por potências de dois (Figura 2.10(c)). O somatório dos elementos da matriz final produz o valor desta unidade de textura (Figura 2.10(d)). Esta sequência de operações pode ser vista com maior detalhe na Figura 2.10.

6	5	2	1	0	0	1	2	4	1	0	0
7	6	1	1	X	0	8	X	16	8	X	0
9	3	7	1	0	1	32	64	128	32	0	128
(a)			(b)			(c)			(d)		

Figura 2.10: Exemplo demonstrando os passos necessários para o cálculo do LBP. 2.10(a) Imagem original, 2.10(b) imagem binária gerada por comparação com o *pixel* central, 2.10(c) máscara LBP e 2.10(d) resultado da operação.

Sendo assim, o histograma LBP varia de 0 a 255 (valor máximo alcançado quando todos os valores da matriz binária forem iguais a 1, gerando como resultado a soma dos valores da matriz de peso). No exemplo, o valor resultante será $1 + 8 + 32 + 128 = 169$. A posição 169 do histograma LBP deve, então, ser incrementada de 1. O contraste desta unidade pode ser facilmente calculado através da diferença da média simples entre os valores acima do limiar (no exemplo, $(6 + 7 + 9 + 7)/4$) e abaixo $((5 + 2 + 1 + 3)/4)$. O LBP é um modelo simples de textura e tem se mostrado ser bastante poderoso na descrição de texturas [70, 30], sendo invariante em relação a qualquer transformação monotônica na escala de cinza.

2.3 Redução de Dimensionalidade

O processo de análise de imagens normalmente utiliza uma representação mais sucinta da imagem ou de seus componentes (objetos), conhecida como vetor de características, o qual armazena os atributos mais representativos das regiões da imagem. O número de atributos ou características determina a dimensão do vetor de características e, normalmente, depende da área de aplicação e das propriedades que se deseja discriminar. De maneira geral, busca-se reduzir a dimensionalidade do vetor de características por meio da seleção dos atributos que melhor descrevem as propriedades dos componentes da imagem. A dificuldade mais evidente de se trabalhar com um espaço de características de dimensionalidade muito alta é o custo computacional, mais precisamente memória e processamento [76]. O segundo problema se refere ao fato de que muitas vezes há características presentes num vetor que não contribuem (ou contribuem muito pouco) para representar determinado objeto de uma classe. Nesse caso, há a necessidade de selecionar as medidas mais representativas para reduzir a dimensionalidade e, ao mesmo tempo, aumentar o desempenho da segmentação. Para efetuar a redução de di-

dimensionalidade, há basicamente duas abordagens: a extração e a seleção de características. Os métodos de extração criam novas características a partir de transformações ou combinações do conjunto de características original. Os métodos de seleção visam determinar, segundo critérios preestabelecidos, o melhor subconjunto de características capazes de discriminar os objetos. Nesta dissertação, foi utilizado o método de extração de características denominado Análise de Componentes Principais [10].

A seguir, serão apresentados brevemente algumas abordagens para redução de dimensionalidade.

2.3.1 Extração de Características

A extração de características envolve a transformação das bandas originais em um número reduzido de características, enquanto mantém a separação das classes tanto quanto possível. Essa transformação é geralmente linear e baseada em algum critério de otimização. A seguir serão abordados brevemente algumas metodologias para extração de características.

Análise de Componentes Principais

A análise de componentes principais (PCA - *Principal Components Analysis*) tem o propósito de derivar novas variáveis (em ordem decrescente de importância) que são combinações lineares das variáveis originais e que são linearmente descorrelacionadas [77]. Essas combinações lineares são denominadas componentes principais.

Se há p variáveis originais, é possível obter p componentes principais. No entanto, como o objetivo é a redução do número de variáveis a serem avaliadas, a informação contida nas p variáveis originais é substituída pela informação contida em k ($k < p$) componentes principais. Dessa forma, o sistema de variabilidade do vetor aleatório composto de p -variáveis originais é aproximado pelo sistema de variabilidade do vetor aleatório que contém as k componentes principais [78]. A qualidade de aproximação depende da quantidade k escolhida e pode ser medida pela proporção de variância de cada componente principal.

Para demonstrar como os componentes principais são obtidos, seja $X = (x_1, x_2, \dots, x_n)'$, um conjunto aleatório de vetores de dimensão p . O PCA é, então, descrito como uma transformação desse vetor em um vetor y , de forma que:

$$y = A(x - m_x) \quad (2.15)$$

Onde o vetor m_x da Equação 2.15 representa o vetor de valores médios de todas as variáveis de entrada e é estimado como a seguir:

$$m_x = E\{x\} = \frac{1}{K} \sum_{k=1}^K x_k \quad (2.16)$$

A matriz A da Equação 2.15 é determinada pela matriz de covariância C_x . As linhas da matriz A são formadas a partir dos autovetores de C_x ordenados em ordem decrescente de acordo com os autovalores correspondentes. É possível estimar a matriz C_x observando a seguinte relação:

$$C_x = E(x - m_x)(x - m_x)' = \frac{1}{K} \sum_{k=1}^K x_k x_k' - m_x m_x' \quad (2.17)$$

A inversão do PCA é possível com a relação:

$$x = A'y + m_x \quad (2.18)$$

A matriz de covariância C_x demonstra que as variáveis aleatórias que constituem o vetor y são descorrelacionadas. Sejam $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n$ os autovalores de C_x . Cada autovalor λ_i representa a variância de um componente principal y_j . Como os autovalores são ordenados em ordem decrescente, o primeiro componente principal é o de maior variabilidade e o p -ésimo o menor.

Essa técnica caracteriza-se por uma rotação dos eixos dos dados originais para um novo sistema (espaço). Nesse processo a correlação linear entre as sucessivas combinações lineares é eliminada. Os autovetores da matriz de variância-covariância (ou de correlações) representam os coeficientes que definem cada combinação linear. O autovalor pertinente a cada autovetor indica a variância da componente principal correspondente [79].

Uma aplicação distinta do PCA no processamento de imagem consiste na redução da cor da imagem quando os componentes da cor são reduzidos em um novo vetor de características, que contém a maior parte de informação e a maior variância entre os dados.

Análise de Variáveis Canônicas

A Análise de Variáveis Canônicas permite a redução da dimensionalidade de dados, sendo semelhante à Análise de Componentes Principais. A Análise de Variáveis Canônicas busca

posicionar um sistema de eixos de forma que as variâncias interna e entre as classes sejam minimizadas e maximizadas, respectivamente [80]. Ou seja, a análise procura, com base em um grande número de características originais correlacionadas, obter combinações lineares dessas características denominadas variáveis canônicas de tal forma que a correlação entre essas variáveis seja nula.

Segundo [81], a Análise de Variáveis Canônicas é uma técnica estatística que busca identificar e quantificar a associação entre dois grupos de variáveis, tendo por objetivo a determinação das combinações lineares c_1x e c_2y (onde x e y são as variáveis e c_1 , c_2 são os coeficientes das combinações lineares) com maior correlação possível, a fim de discernir o relacionamento entre os dois grupos de variáveis.

A correlação canônica pode ser calculada a partir da Equação 2.19:

$$\rho^* = \text{cor}(U, V) \quad (2.19)$$

onde $U = c_1x$ e $V = c_2x$. Nestas duas últimas igualdades, x e y são variáveis.

$c_1 = e_i \Sigma_i^{-1/2}$ e $c_2 = f_i \Sigma_j^{-1/2}$, onde

e_i é o autovetor correspondente ao i -ésimo autovalor de $\Sigma_i^{-1/2} \Sigma_{ij} \Sigma_j^{-1} \Sigma_{ji} \Sigma_i^{-1/2}$, e

f_i é o autovetor correspondente ao i -ésimo autovalor de $\Sigma_j^{-1/2} \Sigma_{ji} \Sigma_i^{-1} \Sigma_{ij} \Sigma_j^{-1/2}$.

Ressalta-se que os vetores de autovalores, correspondentes aos dois casos, podem ter dimensões diferentes. Finalmente,

Σ_i , Σ_j são as matrizes variância-covariância de x e y , e Σ_{ji} é a matriz de covariância cruzada [81].

Transformada Wavelet

Wavelet é uma função capaz de decompor e descrever outras funções no domínio da frequência, onde é possível analisar estas funções em diferentes escalas de frequência e de tempo. A decomposição de uma função com o uso de Wavelets é conhecida como transformada de Wavelet e tem suas variantes contínua e discreta. Graças à capacidade de decompor as funções tanto no domínio da frequência quanto no domínio do tempo, as funções Wavelet são ferramentas poderosas para a análise de sinais e compressão de dados. Para compressão de imagens, segundo [82], este método é aplicado sobre o domínio espectral de cada *pixel*. Após realizada a decomposição Wavelet sobre a assinatura de cada pixel, os coeficientes resultantes dessa

decomposição apresentarão uma redução de dimensionalidade dos dados.

A análise de Wavelet é feita pela aplicação sucessiva da transformada de Wavelet, representando a decomposição do sinal original em diversos componentes localizados no tempo e na frequência. Cada Wavelet possui melhor ou pior localização nos domínios da frequência e do tempo, por isso a análise pode ser feita com Wavelets diferentes de acordo com o resultado desejado.

Em geral a transformada contínua de Wavelet é usada na análise de sinais, enquanto a sua versão discreta é usada na compressão de dados. As transformadas de Wavelet são hoje empregadas numa vasta gama de aplicações, complementando com frequência a tradicional transformada de Fourier [19]. Um uso que vem crescendo é na compressão de dados. Como outras transformadas, as transformações de Wavelet podem ser usadas para transformar dados e codificar de forma eficiente os dados transformados, resultando em compressão.

Redes Neurais Artificiais

Redes Neurais Artificiais podem ser redutores de dimensionalidade não linear, ou seja, são algoritmos estocásticos que levam em consideração vários parâmetros de entrada (informações), fazendo uma busca aleatória no espaço n -dimensional. A redução de dimensão é possível através da Análise por Componentes Principais Não-Lineares utilizando Redes Neurais Artificiais, método denominado de C-NLPCA (*Cascaded Nonlinear Principal Component Analysis*) [83]. A Figura 2.11 ilustra como essa técnica atua sobre os dados.

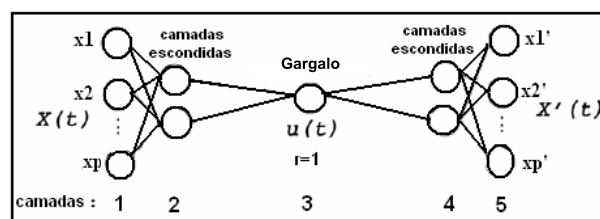


Figura 2.11: Mapeamento de componente não-linear usando Redes Neurais.

Na execução do C-NLPCA, a Rede Neural Artificial usa três camadas de neurônios, denominadas de camadas ocultas, que se posicionam entre as camadas de entrada e de saída. Estas camadas escondidas têm funções não lineares de ativação funcionando entre a camada de entrada e o neurônio de gargalo, e entre a camada de saída e o neurônio de gargalo como mostra a Figura 2.11. A função da Rede Neural, neste caso, é modelar uma composição dessas funções. Das cinco camadas da C-NLPCA, as camadas de entrada e saída (1 e 5) tem p nós, a camada 3 tem r nós, sendo $r < p$. O esperado é que os nós de saída reproduzam os sinais de entrada

apresentados à rede. Os nós das camadas 2 e 4 devem ter funções não-lineares de ativação, e os nós das camadas 1, 3 e 5 usam funções lineares de ativação [83].

Como as entradas p -dimensional devem passar através da camada r -dimensional do gargalo antes de reproduzir as entradas, a informação contida na camada de gargalo é, então, reduzida em relação àquela contida na camada de entrada, permitindo através da rede C-NLPCA a compressão e redução dos dados.

2.3.2 Seleção de Características

As técnicas para seleção de características geralmente empregam um algoritmo de busca e uma função critério de decisão. O algoritmo de busca gera e compara possíveis soluções, isto é, subgrupos de características, aplicando uma função critério de decisão como uma medida de adequacidade da solução. Inicialmente são usadas todas as características originais e, especificando-se um número desejado de características menor que a dimensão total, aquelas que não contribuem para a discriminação das classes são removidas. Um critério usualmente empregado é a separação entre as classes. Se a remoção de uma características, ou de um conjunto de características, não diminuir o valor dessa medida de separação substancialmente, então esta característica será redundante. A seguir serão abordados brevemente algumas metodologias para seleção de características.

Sequential Forward Selection - SFS

Dentre os vários métodos para seleção de características conhecidos, uma metodologia simples e amplamente utilizada é o SFS (*Sequential Forward Selection*). Essa metodologia adota a estratégia *bottom-up*, isto é, trata-se de um algoritmo que, baseado em um critério, seleciona uma variável a cada passo. Esse processo é repetido até que seja preenchido um subconjunto de interesse.

Suponhamos que se queira selecionar N características de um total de M disponíveis. Na metodologia SFS, inicialmente é selecionada entre todas as M características aquela que melhor discrimina as classes envolvidas, de acordo com alguma medida de separação. Essa primeira característica não será mais descartada e fará conjunto com a próxima característica selecionada dentre as $M - 1$ disponíveis, e assim por diante, até que sejam selecionadas as N características desejadas.

Como descrito em [84], esse algoritmo pode ser definido como a seguir. Seja X o vetor das características, d o número total de características, J como uma função critério que aponta

as melhores características no conjunto original e $Y_0 = \emptyset$ um conjunto de características. O algoritmo SFS inicia o processo iterativo com Y_0 , e, após k passos, o novo conjunto Y_k possuirá k características, tendo, portanto, uma característica selecionada em cada passo iterativo. Assim, Y_k e $B = X - Y_k$ têm tamanhos iguais a k e $d - k$, respectivamente. As características B_j do conjunto de características restantes são ordenadas de tal forma que: $J(Y_k \cup \{B_1\}) \geq J(Y_k \cup \{B_2\}) \geq \dots \geq J(Y_k \cup \{B_{d-k}\})$. Por fim, a característica B_1 é selecionada e unida ao conjunto Y_k para formar Y_{k+1} , onde $Y_{k+1} = Y_k \cup \{B_1\}$. O algoritmo é apresentado abaixo.

Algoritmo 1 Algoritmo SFS

Entrada: X - Conjunto completo de características

Entrada: $p \in \mathbb{N}$ - Número de características a descartar

Saída: Y - Conjunto de p características selecionadas de X

```

 $Y_0 \leftarrow \emptyset$  {Conjunto inicial de características escolhidas}
 $L_0 \leftarrow X$  {Conjunto inicial de características disponíveis}
para  $k = 0$  to  $p - 1$  do
  Seja  $B : J(Y_k \cup B) = \max J(Y_k \cup B)$ 
   $Y_{k+1} \leftarrow Y_k \cup B$ 
   $L_{k+1} \leftarrow L_k - B$ 
fim para
 $Y \leftarrow Y_p$  {Último conjunto de características escolhidas}
  
```

Deve-se considerar que a seleção ótima de características envolve uma busca exaustiva envolvendo todas as características, resultando em um alto custo computacional.

Sequential Backward Selection - SBS

Essa metodologia adota a estratégia *top-down*, isto é, trata-se de um algoritmo que, baseado num critério, seleciona uma variável que será eliminada, em cada passo. Esse processo é repetido até que o subconjunto restante seja do tamanho de interesse.

Como descrito em [84], esse algoritmo pode ser definido como a seguir. Seja X o vetor de características, d como o número total de características e J como uma função critério que aponta as melhores variáveis do conjunto para serem eliminadas. Assim, o algoritmo começa com o conjunto original das características, $Y_0 = X$. Após k iterações são descartadas k características e o conjunto original Y_0 é reduzido ao conjunto Y_k . Nesse caso, Y_k possui tamanho igual a $d - k$. Posteriormente ordenam-se as características B_j do conjunto de características disponíveis Y_k , de tal forma que: $J(Y_k - \{B_1\}) \geq J(Y_k - \{B_2\}) \geq \dots \geq J(Y_k - \{B_{d-k}\})$. Por fim, a característica B_1 é selecionada e eliminada Y_k para formar Y_{k+1} , onde $Y_{k+1} = Y_k - \{B_1\}$. O algoritmo é apresentado abaixo.

Algoritmo 2 Algoritmo SBS

Entrada: X - Conjunto completo de características**Entrada:** $p \in \mathbb{N}$ - Número de características a descartar**Saída:** Y - Conjunto de p variáveis características de X $Y_0 \leftarrow X$ { *Conjunto inicial de características escolhidas* }**para** $k = 0$ to $d - p - 1$ **do** Seja $B : J(Y_k - B) = \max J(Y_k - B)$ $Y_{k+1} \leftarrow Y_k - B$ **fim para** $Y \leftarrow Y_{d-p}$ { *Último conjunto de características escolhidas* }

Da mesma forma que o procedimento SFS apresentado na subseção anterior, a seleção ótima de características pelo método SBS envolve uma busca exaustiva envolvendo todas as características, resultando em um alto custo computacional.

No próximo capítulo será apresentada a metodologia desenvolvida nesta dissertação para representação de imagens através de grafos bem como os experimentos computacionais realizados.

3 *Experimentos Computacionais*

A motivação original desta dissertação surgiu da necessidade de um processo de representação de imagens através de grafos, concentrando-se, para isso, na etapa de segmentação automática da mesma, pois uma segmentação eficiente é necessária para uma boa representação da imagem através de grafos. A segmentação é efetuada a fim de se obter regiões homogêneas, cujos componentes atendem a um determinado critério de similaridade. Neste trabalho, a similaridade é calculada através de medidas de distâncias probabilísticas em sua forma discreta e analítica. O método de segmentação utilizado é o *split-and-merge*, e um estudo sobre o impacto da contribuição da textura, extraída através do descritor de textura *Local Binary Pattern*, em conjunto com o nível de luminância na análise de similaridade de regiões durante o processo de segmentação foi efetuado. A consideração de mais de uma característica para a segmentação da imagem implica na utilização de algum método para reduzir a dimensionalidade do vetor de características, com o intuito de extrair as características mais relevantes da imagem. Para isso, foi efetuada a Análise de Componentes Principais para reduzir o espaço de cor de três dimensões para uma dimensão. Depois da segmentação, a representação é gerada e as características consideradas relevantes na imagem extraídas. Posteriormente, dados dois grafos de atributos, as matrizes de similaridade são construídas utilizando-se funções de agregação, responsáveis por obter um valor único, que representa a similaridade entre as partes analisadas, a partir de um conjunto de valores.

O desenvolvimento de uma metodologia para análise e extração de informação eficiente de imagens é uma tarefa complexa e, embora diversas técnicas tenham sido propostas, não há um método genérico que apresente bom desempenho em diferentes domínios de aplicação. Um diagrama ilustrando as principais etapas do método proposto é apresentado na Figura 3.1.

Cada uma dessas etapas é descrita em mais detalhes, a seguir:

- Aquisição de dados: seleção das imagens a serem segmentadas.
- Pré-processamento e redução de dimensionalidade: preparação das imagens, convertendo para escala de cinza ou reduzindo a sua dimensionalidade através do método PCA (des-

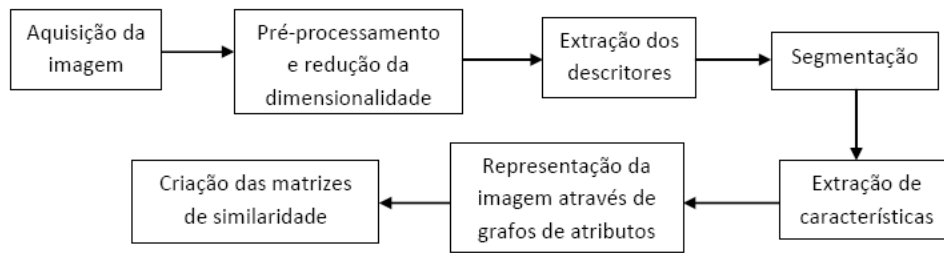


Figura 3.1: Etapas do método proposto.

critico no Capítulo 2).

- **Extração dos descritores:** utilizando as imagens selecionadas na etapa de aquisição de dados (imagem original) e pré-processamento (imagem processada para segmentação), são extraídos os descritores necessários para a segmentação. Para descrever a luminância de cada canal foram utilizados seus respectivos histogramas, e para o descritor de cor foi utilizado o histograma obtido através da aplicação do método *Local Binary Pattern* considerando 8 amostras e raio igual a 1. Assim, tanto os histogramas de luminância quanto o histograma LBP variam de 0 a 255.
- **Segmentação:** nessa etapa, a imagem é segmentada utilizando-se o método *split-and-merge*. Quanto melhor o resultado da segmentação, mais fiel será o grafo que a representa. Diversos parâmetros são considerados e avaliados, como o limiar utilizado para unir ou separar regiões, o tamanho mínimo da região considerada como critério de parada na fase de divisão, as medidas de distância utilizadas, a contribuição da textura no processo de segmentação e a avaliação de qual conjunto de características que fornece o melhor resultado (segmentação em: escala de cinza, escala de cinza e textura, RGB, RGB e textura, textura, primeira componente do PCA, primeira componente do PCA e textura). Para medir a similaridade entre as regiões, foram consideradas duas abordagens. A primeira, utilizando as medidas da Tabela 3.1 ou a distância euclidiana dada pela Equação 3.2, consiste em realizar a soma ponderada das medidas calculadas separadamente para vetor de características, quando a mesma possuir dimensão maior do que um. A segunda consiste em usar as expressões analíticas para as medidas de distâncias, como apresentadas na Tabela 3.2.
- **Extração de características:** nessa etapa, características das regiões obtidas como resultado da segmentação, geradas na etapa anterior, são extraídas. Foram consideradas como informações relevantes o(s) histograma(s) de cada região e os centros de massa, a distância entre os centros de massa das regiões e a angulação da mesma, bem como a extração

da matriz de adjacências, em que as regiões representam vértices e o relacionamento espacial entre elas representam as arestas do grafo.

- Representação da imagem através do grafo de atributos: nessa etapa, as informações obtidas na etapa anterior são organizadas de forma a representarem a imagem como um grafo de atributos.
- Criação das matrizes de similaridade: Finalmente, dados dois grafos de atributos extraídos pelos métodos acima, é construída a matriz de similaridade entre vértices e entre arestas. Essa matriz é construída utilizando-se as informações extraídas na etapa de Extração de Características e as medidas de distâncias das Tabelas 3.1 e 3.2. As medidas são aplicadas sobre os histogramas, e o resultado é combinado com a análise dos vizinhos das regiões em avaliação.

3.1 Medidas de Distâncias Utilizadas

Distância ou medidas de similaridade são de fundamental importância para problemas de reconhecimento de padrões, clusterização e recuperação de informação [85]. A escolha da medida depende do tipo de representação do objeto. A divisão da imagem em regiões onde seus *pixels* componentes atendem à uma certa propriedade pré-determinada requer que os atributos selecionados sejam representativos de cada região. Entretanto, a análise de similaridade entre as regiões necessita de medidas que possam trabalhar com espaços cuja dimensão é maior do que um. Para isso, foram utilizadas medidas probabilísticas para determinar a capacidade discriminatória de um conjunto de variáveis. Neste trabalho, diferentes medidas de distância probabilística foram utilizadas como parte complementar da fase de Divisão e Fusão do algoritmo *split-and-merge*, abordado no Capítulo 3, onde é necessário avaliar o quanto duas regiões são semelhantes para separá-las ou uni-las .

Considere duas funções de densidade de probabilidade $p_a(\mathbf{x}; \theta_a)$ e $p_b(\mathbf{x}; \theta_b)$ de d -dimensão de variáveis contínuas \mathbf{x} , definida por sua forma funcional p_a, p_b e vetores de parâmetros θ_a, θ_b respectivamente. A medida de distância probabilística D entre duas funções de densidade de probabilidade é a função que mede a diferença Δ integrada sobre o domínio \mathbb{R}^d de \mathbf{x}

$$D(p_a, p_b, \theta_a, \theta_b) = \int_{\mathbf{x}} \Delta[(p_a, p_b, \theta_a, \theta_b)] d\mathbf{x}. \quad (3.1)$$

A métrica Δ pode ser positiva, zero se o valor de duas funções coincidem, e correlacionado para o suas diferenças absolutas [2]. No contexto de classificação, as probabilidades *a priori*

$P_a := Pr[\mathbf{x} \in \omega_a], P_b := Pr[\mathbf{x} \in \omega_b]$ para duas classes ω_a, ω_b podem adicionalmente ser incorporadas dentro da distância probabilística, por isso neste caso tem-se $D = D(P_a, P_b, p_a, p_b, \theta_a, \theta_b)$. Em geral D é definido como função de densidade de probabilidade que poderia vir de famílias funcionais distintas, para uma instância de distribuição Normal univariada $p_a(x; \mu, \sigma^2)$ e uma distribuição Gama $p_b(x; k, \theta)$. Na prática, entretanto, apenas funções de distribuição de probabilidade com a mesma forma funcional $p_a = p_b$ são comparadas. No caso discreto as densidades p_a, p_b são aproximadas por histogramas $P_i, Q_i, i = 0, \dots, 255$.

Portanto, a avaliação de similaridade entre duas regiões é feita através de medidas de distâncias, aplicadas sobre o histograma das duas regiões em consideração quando o vetor de características analisado possuir apenas uma dimensão. Nesse caso, foram consideradas as seguintes medidas de distâncias probabilísticas:

Bhattacharyya	$L_B = -\ln \sum_{i=1}^n \sqrt{P_i Q_i}$
Matusita	$L_M = \sqrt{2 - 2 \sum_{i=1}^n \sqrt{P_i Q_i}}$
Kullback-Leibler	$L_K = \sum_{i=1}^n P_i \ln \frac{P_i}{Q_i}$
Divergência	$L_J = \sum_{i=1}^n (P_i - Q_i) \ln \frac{P_i}{Q_i}$

Tabela 3.1: Uma seleção de medidas de distância probabilísticas em reconhecimento de padrões [2], [3], [4]. P e Q são dois histogramas com n valores diferentes.

Quando o vetor de características analisado continha mais de uma dimensão, foram utilizadas uma especialização das medidas de distâncias probabilísticas analisadas sobre a distância gaussiana multivariada, obtendo assim sua forma analítica.

Seja $N(X; \mu, \Sigma)$ com $X \in \mathbb{R}^d$ uma densidade Gaussiana multivariada, com $p_1(x) = N(X; \mu_1, \Sigma_1)$ and $p_2(x) = N(X; \mu_2, \Sigma_2)$. A Tabela 3.2 lista expressões analíticas de algumas distâncias probabilísticas entre duas densidades Gaussianas.

Bhattacharyya	$J_B(p_1, p_2) = \frac{1}{8}(\mu_1 - \mu_2)^T [\frac{1}{2}(\Sigma_1 + \Sigma_2)]^{-1}(\mu_1 - \mu_2) + \frac{1}{2} \log \frac{ \frac{1}{2}\Sigma_1 + \Sigma_2 }{ \Sigma_1 ^{1/2} \Sigma_2 ^{1/2}}$
Matusita	$J_M(p_1, p_2) = \sqrt{2(1 - \exp(-J_B(p_1, p_2)))}$
Kullback-Leibler	$J_K(p_1 p_2) = \frac{1}{2}(\mu_1 - \mu_2)^T \Sigma_2^{-1}(\mu_1 - \mu_2) + \frac{1}{2} \log \frac{ \Sigma_2 }{ \Sigma_1 } + \frac{1}{2} tr[\Sigma_1 \Sigma_2^{-1} - I_d]$
Divergência	$J_D(p_1, p_2) = \frac{1}{2}(\mu_1 - \mu_2)^T (\Sigma_1^{-1} + \Sigma_2^{-1})(\mu_1 - \mu_2) + \frac{1}{2} tr[\Sigma_1^{-1} \Sigma_2 + \Sigma_2^{-1} \Sigma_1 - 2I_d]$

Tabela 3.2: Expressões analíticas de distâncias probabilísticas entre duas densidades normais.

onde μ e Σ são o vetor médio dos padrões e a matriz de covariância da classe (região), respectivamente. Para os objetivos deste trabalho, entretanto, foram comparados os resultados obtidos com a soma ponderada das distâncias apresentadas na Tabela 3.1 aplicadas em cada vetor separadamente e com as expressões analíticas da Tabela 3.2 aplicadas sobre o conjunto de vetores de características.

Por fim, o desempenho das distâncias probabilísticas foi comparado com o desempenho obtido através da utilização da distância euclidiana aplicados sobre os histogramas das regiões, que é dada pela fórmula:

$$L_E = \sqrt{\sum_{i=1}^n |P_i - Q_i|^2} \quad (3.2)$$

onde P e Q são dois histogramas com n valores diferentes.

A distância euclidiana é uma das medidas de dissimilaridade entre classes mais utilizadas na prática [86]. Quanto menor o valor da distância Euclidiana entre duas classes, mais próximas elas se apresentam em termos de parâmetros quantitativos por classe, logo, quanto menor a distância euclidiana, maior a eficiência do procedimento.

3.2 Processo de Segmentação das Imagens

Como descrito acima, as imagens devem passar pelo processo de segmentação para gerar o grafo que a representa. Para isso, a abordagem utilizada foi o *split-and-merge*, que consiste em subdividir recursivamente a imagem em regiões cada vez menores, considerando a homogeneidade com respeito à medida de similaridade [1] e, em um segundo estágio, unir regiões vizinhas novamente baseado no mesmo critério. A seguir são apresentadas a metodologia empregada na utilização de descritores de cor, textura e extração de características no processo de segmentação.

3.2.1 Descritor de texturas *Local Binary Pattern*

A textura fornece informações sobre a distribuição espacial das variações de tonalidade de um objeto ou de alguns grupos de objetos não identificáveis individualmente, definindo o aspecto visual de rugosidade ou suavidade de determinada superfície. Embora o sistema visual humano apresente relativa facilidade no reconhecimento e descrição de textura, definir um conjunto de descritores de textura não é uma tarefa simples. Tal dificuldade é refletida pela grande quantidade de definições e métodos de análise de textura encontrados na literatura [53, 54, 55, 56, 57].

A despeito da variedade de métodos disponíveis, este trabalho utilizou o descritor de texturas *Local Binary Pattern* devido a sua rapidez, robustez e invariância em relação a qualquer transformação monotônica na escala de cinza [70]. De acordo com a formulação apresentada

na Subseção 2.2.4, o padrão binário local foi calculado considerando o raio e a quantidade de amostras, respectivamente, iguais a $R = 1$ e $P = 8$, fazendo com que o histograma varie de 0 a 255. Esse valor foi escolhido por questões de desempenho computacional, visto que um número elevado de amostras representa o aumento da dimensão do histograma e consequentemente o aumento do tempo de processamento e memória utilizadas. Além disso, o intervalo considerado torna possível a combinação com os histogramas de cor, que possuem dimensão igual a 256.

3.2.2 Descritor de cor

As cores presentes em uma imagem desempenham um papel significativo no processo de análise realizado tanto pelos seres humanos quanto pelos computadores. Sabe-se que o olho humano pode discernir milhares de tons e intensidades de cores, comparado com apenas algumas dezenas de níveis de cinza interpretados pelo computador. A abordagem mais comum para extração de características baseadas em distribuição de cores utiliza histogramas de cor [87, 88].

Assumindo que uma imagem é uma função $f(x, y)$ de duas variáveis espaciais x e y , $x = 0, 1, \dots, N - 1$ e $y = 0, 1, \dots, M - 1$ e que só pode assumir os valores discretos $i = 0, 1, \dots, G - 1$, onde G é o número total de níveis de intensidade de uma imagem, o histograma de níveis de intensidade é uma função que mostra (para cada nível de intensidade) o número de *pixels* na imagem inteira, conforme a equação abaixo:

$$h(i) = \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} \delta(f(x, y), i) \quad (3.3)$$

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

$$\delta(j, i) = \begin{cases} 1, & j = i \\ 0, & j \neq i \end{cases} \quad (3.4)$$

O histograma de intensidade é obviamente um simples e conciso sumário da informação estatística contida na imagem. O cálculo do histograma de níveis de cinza envolve apenas *pixels* simples. O histograma, então, contém informação estatística de primeira ordem sobre a imagem (ou seu fragmento). Dividindo os valores $h(i)$ pelo número total de *pixels* de uma imagem obtém-se a densidade de probabilidade aproximada de ocorrência dos níveis de intensidade.

$$p(i) = \frac{h(i)}{NM}, \quad i = 0, 1, \dots, G - 1 \quad (3.5)$$

A forma do histograma provê muitos indícios das características da imagem. Por exemplo, um histograma de distribuição concentrada num certo intervalo indica uma imagem de pouco contraste. Por outro lado, um histograma bimodal frequentemente sugere que a imagem contém um objeto com um estreito intervalo de níveis de cinza contra um “pano de fundo” de intensidade diferente. Os histogramas de cor são invariantes quanto à translação e rotação das imagens, sendo que, com a normalização dos histogramas, pode-se obter também a invariância com respeito à escala. Entretanto, os histogramas de cor não indicam a localização espacial dos *pixels* na imagem.

Diferentes parâmetros úteis podem ser adquiridos do histograma para quantitativamente descrever as propriedades estatísticas de primeira ordem de uma imagem. Neste trabalho foram utilizadas as Equações 3.6 e 3.7 para o cálculo das expressões analíticas das medidas de distâncias probabilísticas, e representam a média e a variância, respectivamente.

$$\mu = \sum_{i=0}^{G-1} ip(i) \quad (3.6)$$

$$\sigma^2 = \sum_{i=0}^{G-1} (i - \mu)^2 p(i) \quad (3.7)$$

A média representa o valor esperado da distribuição, enquanto a variância descreve quanto os valores estão dispersos em torno da média.

Os descritores de cor para o sistema de cor RGB e do nível de cinza usados neste trabalho são descritos como histogramas de luminância, um para cada canal, cujos valores variam de 0 a 255, normalizados através da Equação 3.5.

3.2.3 Compressão de Dados Através da Análise de Componentes Principais

A redução do volume dos dados é uma tarefa comum no processamento de imagem. Há uma enorme quantidade dos algoritmos [89, 1, 90] baseados nos vários princípios que conduzem à compressão de imagem. Os algoritmos baseados na redução da cor da imagem causam, em sua maior parte, a modificação das informações contidas na imagem. Mas seus resultados são ainda aceitáveis para algumas aplicações. A transformação da imagem colorida para tons de cinza (intensidade) I faz parte dos algoritmos mais comuns. Sua execução é baseada geralmente na soma ponderada dos três componentes R , G , B de acordo com a relação:

$$I = w_1R + w_2G + w_3B \quad (3.8)$$

As matrizes R , G e B contêm componentes da cor da imagem, e os pesos w_i foram determinados levando-se em consideração a sensibilidade do sistema visual humano [1]. A análise de componentes principais fornece uma maneira alternativa a este método. A ideia é baseada na Equação 2.18, onde a matriz A é substituída pela matriz A_k em que somente os k maiores (em vez de n) autovalores são usados para sua formação. O vetor \hat{x} de variáveis reconstruídas é dado, então, pela relação:

$$\hat{x} = A'_k y + m_x \quad (3.9)$$

onde \hat{x} é uma aproximação para x .

Imagens coloridas de tamanho $M \times N$ são geralmente salvas em uma matriz tridimensional P de tamanho $M \times N \times 3$ que significa que a informação sobre a intensidade de componentes da cor está armazenada em 3 planos. O vetor de variáveis de entrada x na Equação 2.15 pode ser formado como o vetor de n -tridimensional de cada cor. Logo, são formados três vetores unidimensionais x_1 , x_2 , x_3 a partir de cada plano $P(M, N, i)$ com o comprimento de $M \times N$. A matriz de covariância C_x e a matriz correspondente A , composta pelos autovalores de C_x ordenados segundo seus autovalores, são calculadas e o vetor tridimensional \hat{x} é reconstruído de acordo com a Equação 3.9 a partir do primeiro, do segundo e do terceiro componente da imagem dada. A imagem obtida reconstruindo-se a mesma com a matriz A_k , utilizando somente o primeiro e maior autovalor, que contém a maior parte da informação, ou seja, maior variabilidade. Então, a imagem construída a partir desses valores tem o contraste máximo.

Observe uma imagem real P e seus componentes R , G , B na Figura 3.2. Seus três componentes obtidos de acordo com a Equação 3.9 para cada autovetor são apresentados. A comparação da imagem obtida através da soma ponderada da cor calculada pela Equação 3.8 com a primeira componente principal é apresentada na Figura 3.3. Os autovalores classificados em ordem decendente pertencentes à imagem selecionada são apresentados na Tabela 3.3.

λ_1	λ_2	λ_3
0.6103	0.3231	0.0418

Tabela 3.3: Autovalores obtidos através da aplicação do PCA na imagem selecionada.

Analisando a Figura 3.3 é notória a distribuição da maior variância ao longo da primeira componente principal. Percebe-se também que a segunda componente principal abrange, de

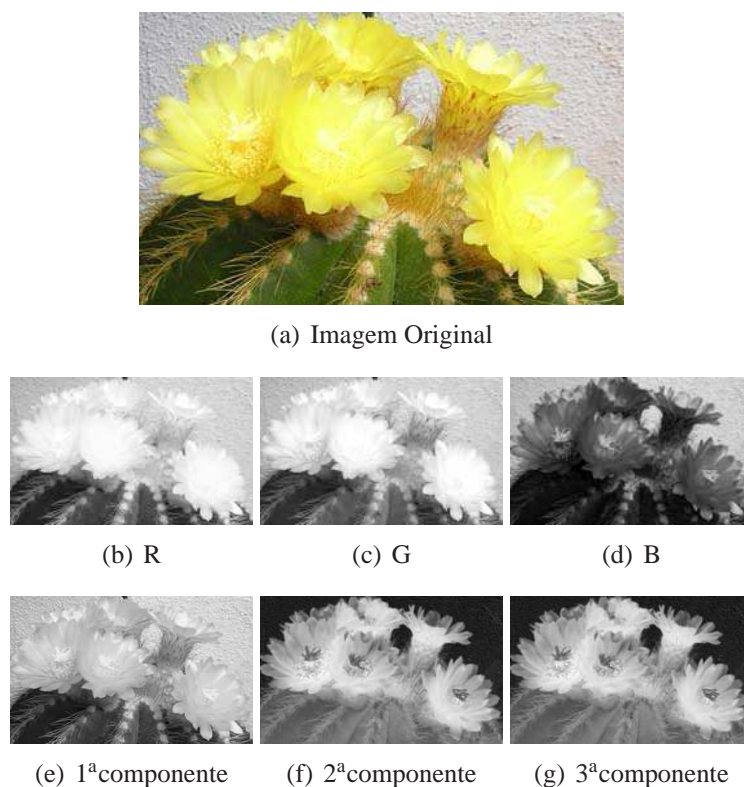


Figura 3.2: Imagem original, suas três componentes e as três componentes do PCA.

forma perpendicular à primeira componente, a segunda maior variância. O raciocínio é análogo para o espaço n -dimensional, isto é, a i -ésima componente principal possuirá a i -ésima maior variância.

Desta forma, esta dissertação considerou a redução da dimensionalidade do sistema de cor RGB utilizado aplicando-se o método de Análise de Componentes Principais para extrair uma nova característica que deverá conter uma maior discriminação das regiões da imagem em análise. Será utilizado, por questões de performance, apenas a primeira componente extraída.

3.2.4 O Método *Split-And-Merge* Combinado com os Descritores de Cor e Textura

A estratégia *split-and-merge* é uma alternativa aos métodos de crescimento de regiões baseados em um conjunto inicial de sementes. Uma vantagem deste método é a independência do descritor da região e da medida de similaridade, que permite experimentos com uma grande variedade de critérios. Para aplicar os estágios do algoritmo de segmentação *split-and-merge*, as informações dos histogramas das regiões são necessárias, sejam eles extraídos sobre as componentes de luminância dos canais R, G e B, do nível de cinza da imagem, do descritor de textura LBP ou da primeira componente resultante da aplicação do PCA na imagem.

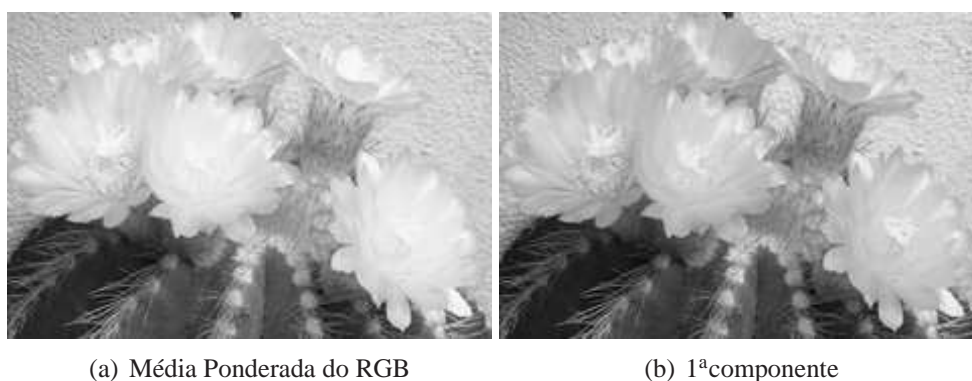


Figura 3.3: Comparação dos métodos de redução de cor. A imagem em escala de cinza obtida pela soma ponderada dos canais R,G,B à esquerda e a imagem em escala de cinza obtida pelo método PCA à direita.

Na fase de *split*, a imagem R é temporariamente dividida recursivamente em quatro regiões S_1, S_2, S_3 e S_4 , cujas dimensões são determinadas pela dimensão da imagem, para verificar a sua homogeneidade. Então, a medida de similaridade é calculada para cada par das quatro regiões criadas. Escolhendo a função de distância D para representar a medida de similaridade, denota-se $D(S_i, S_j)$ por $c_{ij}, i \neq j, i, j = 1, \dots, 4$. Dado um valor de limiar para a fase de *split* ts , se $\max c_{ij} < ts$, o algoritmo definitivamente aplica a divisão da região. Caso contrário, a região R permanece a mesma. Este processo é recursivamente realizado até que uma dada região de tamanho mínimo, ms , é atingida. Na fase de *merge*, a função de distância D é usada para comparar a região S_i com cada um dos seus 4-vizinhos S_j (ou seja, que se situam estritamente acima, abaixo, a esquerda e a direita). Similarmente à fase de *split*, ele também necessita de um valor limiar tm . Então, para S_i e S_j , se $D(S_i, S_j) < tm$, para alguma região pertencente a 4-vizinhança, o algoritmo aplica a junção. Caso contrário, a região original permanece a mesma. Este processo é recursivamente realizado para cada uma das regiões ainda não exploradas S_j dentro da 4-vizinhança até que toda a imagem seja analisada.

A medida de similaridade D é obtida de acordo com as características consideradas. Se apenas uma característica é levada em consideração, por exemplo, o histograma do nível de cinza da imagem, a distância D é calculada através da aplicação de algumas das distâncias probabilísticas da Tabela 3.1. Quando o cálculo da medida de distância exigia a análise de mais de um vetor de características, como os histogramas de cor RGB e LBP, a distância D foi calculada utilizando-se dois métodos distintos.

No primeiro método, as distâncias entre os histogramas são independentemente calculados para cada banda e então reunidos conforme o valor de luminância de cada canal RGB, que são relacionados à sensibilidade do sistema visual humano, como:

$$D = 0.3 \times D_R + 0.11 \times D_G + 0.59 \times D_B \quad (3.10)$$

onde D_R , D_G e D_B denotam a aplicação de algumas das distâncias probabilísticas da Tabela 3.1 ou da distância euclidiana, dada pela Equação 3.2. Se a análise inclui características de cor, no sistema RGB, e textura, a distância é primeiramente calculada entre os três canais e então uma combinação ponderada para a distância final entre os histogramas das regiões é obtido como:

$$D = \alpha D_{\text{RGB}} + (1 - \alpha) D_{\text{LBP}} \quad (3.11)$$

onde D_{LBP} denota a aplicação de algumas das distâncias probabilísticas da Tabela 3.1 ou da distância euclidiana, dada pela Equação 3.2, D_{LBP} a distância calculada pela Equação 3.10 e α um parâmetro usado para ponderar cada termo.

No caso da segmentação envolvendo histogramas do nível de cinza ou da primeira componente do PCA, o cálculo é efetuado como segue:

$$D = \alpha D_H + (1 - \alpha) D_{\text{LBP}} \quad (3.12)$$

onde D_H e D_{LBP} denotam a aplicação de algumas das distâncias probabilísticas da Tabela 3.1 ou da distância euclidiana, dada pela Equação 3.2 e α um parâmetro usado para ponderar cada termo.

No segundo método, os vetores de características extraídos foram considerados como dados de entrada das medidas de distâncias probabilísticas em sua forma analítica apresentadas na Tabela 3.2.

3.3 Representação da Imagem através de Grafos de Atributos

Depois de segmentada, cada região representará um vértice, e as arestas são computadas de acordo com as relações espaciais existentes entre essa região e seus vizinhos de 4-conectividade. O resultado da segmentação representará a imagem a ser reconhecida no modelo, que será obtido através da segmentação manual da imagem utilizando a ferramenta *ImGraph*. Os atributos são extraídos depois da aplicação do método de segmentação *split-and-merge*. Para os vértices, as características consideradas como atributos foram o(s) histograma(s) da respectiva região e

os centros de massa. Para as arestas, tem-se a angulação e a distância entre os centros de massa das regiões conectadas à aresta avaliada.

3.4 Funções de Agregação

Nesta seção, duas funções de agregação para criação das matrizes de similaridade são propostas e investigadas, em termos de suas propriedades e capacidade de reconhecimento. O objetivo principal consiste em identificar a função cujo valor máximo corresponda à imagem da solução que representa a correspondência mais adequada, isto é, a função que melhor representa o reconhecimento da imagem. A qualidade dos dados de entrada (matrizes de similaridade de vértices e de arestas) é primordial, pois afetam fortemente o comportamento das funções utilizadas para resolver algoritmos de reconhecimento de cenas por casamento de grafos, contribuindo na identificação da melhor correspondência.

Para a criação das matrizes de similaridade entre vértices e arestas, grafos representarão as imagens e valores numéricos serão associados a vértices e arestas de acordo com os atributos extraídos. Assim, a função de agregação tem por objetivo analisar as similaridades entre os atributos de vértices e arestas de dois grafos. Entretanto, é preciso definir os critérios necessários para a criação dessas matrizes. Estes critérios formam a base de uma decisão a partir de elementos que podem ser medidos ou avaliados. A seguir, os passos utilizados para a análise dos critérios são apresentados, e podem ser assim resumidos: escolha dos critérios, definição de pesos para os critérios, normalização e combinação dos critérios [91].

3.4.1 Escolha dos Critérios

A escolha dos critérios para a representação das imagens em forma de grafos foi feita através da relevância dos mesmos para uma melhor caracterização da imagem. Devido às características do sistema visual humano, que é capaz de interpretar cenas acromáticas, por exemplo em baixos níveis de iluminação, utilizando apenas a textura, a mesma foi levada em consideração para a discriminação das regiões. A informação de cor atua como uma ajuda para possibilitar interpretações mais ricas. E, de acordo com o trabalho apresentado em [63], em um primeiro momento, as pessoas usam principalmente informações de cor para julgar similaridade, o que faz com que essa característica também seja escolhida. Outro aspecto importante de uma região é o seu centro de massa, que, a princípio, discriminará regiões que embora tenham cor e textura similares encontram-se em regiões totalmente distintas da imagem. Assim, os critérios considerados como atributos dos vértices são o(s) histograma(s), que variam de acordo com o

tipo de segmentação escolhida, e o centro de massa de cada região.

Para as arestas, que são relações espaciais entre as regiões, intuitivamente é possível estabelecer como critério a angulação da aresta, que define onde estão posicionadas duas regiões vizinhas, e a distância entre as mesmas. Então, os critérios considerados como atributos das arestas, neste trabalho, são a angulação (que varia de 0° a 360°) e o comprimento das arestas, medido através da distância euclidiana entre os dois centros de massa.

3.4.2 Normalização e Combinação de Critérios

Um aspecto importante para a combinação de critérios é a necessidade de normalização das medidas extraídas a partir da segmentação. Como os intervalos de valores das características são, em geral, bastante distintos, uma característica não deveria predominar sobre as outras devido a essa variabilidade relativa. Todos os critérios foram normalizados para uma mesma escala, viabilizando a agregação entre eles. A maior parte dos processos de normalização utiliza o valor máximo e mínimo para a definição de uma escala. A forma mais simples é uma variação linear definida pela equação a seguir [92].

$$X_i = \frac{R_i - R_{\min}}{R_{\max} - R_{\min}} \quad (3.13)$$

onde X_i é o valor normalizado, R_i é o valor a ser normalizado, R_{\min} é o valor mínimo para o critério e R_{\max} é o valor máximo para o critério.

Depois de normalizadas, as medidas são combinadas através de uma soma ponderada, cujos pesos são fornecidos como parâmetro de entrada. Não há um método consensual para a definição de pesos [91], mas várias propostas de procedimentos para este efeito podem ser encontradas na literatura [93, 94]. Neste trabalho, como será mostrado a seguir, é atribuído um peso β para ponderar as medidas consideradas entre vértices (resp. arestas).

Função de Agregação para Vértices

A função Sim_v é composta de dois termos, que representam respectivamente as contribuições das associações entre vértices de G_1 e G_2 e as contribuições na análise dos vizinhos do vértice em consideração. Depois de normalizados, a similaridade entre dois vértices i e j é medida de acordo com a função Sim_v , a ser minimizada, que é definida como:

$$Sim_v(i, j) = \beta D_C(i, j)D(i, j) + (1 - \beta) \min(D_C(v_i, v_j)D(v_i, v_j)) \quad (3.14)$$

onde β é um parâmetro usado para ponderar cada termo de Sim_v , $D_C(i, j)$ representa a distância euclidiana entre os centros de massa dos vértices i e j , $D(i, j)$ representa a distância entre os vértices i e j como mostrado na Subseção 3.2.4, e v_i e v_j representam os vizinhos de 4-conectividade de i e j , respectivamente.

O primeiro termo da Equação 3.14 representa a contribuição média dos vértices para a correspondência. O segundo termo representa a contribuição média de seus vizinhos. Assim, para cada vértice i de G_1 e para cada vértice j de G_2 , a similaridade é analisada avaliando-se a distância D entre os histogramas das respectivas regiões. Duas regiões com histogramas similares podem estar em locais divergentes quando comparadas a imagem modelo e a imagem alvo. Por isso, essa distância é ponderada por D_C , que calcula a distância euclidiana entre o centro de massa de i e j . Assim, distância de histogramas similares serão regulados pelos centros, evitando um alta-similaridade tanto para regiões distintas com histogramas similares, tanto para regiões próximas com histogramas divergentes. Também se avaliou a contribuição dos vizinhos dos vértices i e j . Para cada vizinho de 4-conectividade de i é avaliada a similaridade com os vizinhos de 4-conectividade de j . A menor distância representará uma possível correspondência entre esses vizinhos, e se os vizinhos de i e j têm uma alta similaridade, isso provavelmente aumenta as chances de existir uma correspondência entre i e j . Logo, quanto menor o valor retornado pela Equação $Sim_v(i, j)$, mais similares são os vértices em avaliação.

Função de Agregação para Arestas

A função Sim_a é composta de dois termos, que representam respectivamente as contribuições das associações entre arestas de G_1 e G_2 e as contribuições na análise dos vértices conectados aos mesmos. Depois de normalizados, a similaridade entre dois vértices i e j é medida de acordo com a função Sim_a a ser minimizada, que é definida como:

$$Sim_a(i, j) = \beta D_M(i, j) D_N(i, j) + (1 - \beta) \min(D_C(v_i, v_j) D(v_i, v_j)) \quad (3.15)$$

onde β é um parâmetro usado para ponderar cada termo de Sim_a , $D_N(i, j)$ representa a distância euclidiana entre o comprimento e a angulação das arestas i e j , $D_M(i, j)$ representa a distância euclidiana entre os pontos médios das arestas i e j , $D(v_i, v_j)$ representa a distância entre os vértices v_i e v_j como mostrado na Subseção 3.2.4, e v_i e v_j representam os vértices ligados à i e à j , respectivamente.

O primeiro termo da Equação 3.15 representa a contribuição média das arestas para a correspondência. O segundo termo representa a contribuição média dos vértices ligados às arestas

em avaliação. Assim, para cada aresta i de G_1 e para cada aresta j de G_2 , a similaridade é analisada avaliando-se a distância euclidiana D_N entre o comprimento e a angulação das arestas que representam a ligação entre duas regiões. Duas arestas com comprimento e angulação similares podem estar em locais divergentes quando comparadas a imagem modelo e a imagem alvo. Por isso, essa distância é ponderada por D_M , que calcula a distância entre o centro de massa de i e j . Assim, distância D_N será regulada pelos centros, evitando uma alta-similaridade tanto para arestas distintas com comprimento e angulação similares, quanto para arestas próximas com comprimento e angulação divergentes. Também se avaliou a contribuição dos vértices ligados às arestas i e j . Para cada vértice de i é avaliada a similaridade com os vértices de j . A menor distância representará uma possível correspondência entre esses vértices, e se os vértices de i e j têm uma alta similaridade, isso provavelmente aumenta as chances de existir uma correspondência entre i e j .

3.5 Experimentos Realizados

Os experimentos têm por objetivo avaliar a qualidade do algoritmo *split-and-merge* aplicado em imagens em escala de cinza, coloridas (utilizando o sistema de cor RGB), usando unicamente o descritor de texturas ou combinando o descritor de cor com o descritor de texturas LBP. Além disso, foi analisado o efeito da variação dos parâmetros, tais como o limiar da fase de *split* e *merge*, as medidas de distâncias e o peso dado à contribuição da textura no processo de segmentação. Dessa forma, podemos verificar qual conjunto dos testes considerados retorna uma melhor segmentação da imagem de forma que regiões visivelmente homogêneas mantenham-se unidas.

Os testes foram realizados em um computador com processador AMD Athlon 64 3000+ 1.8 GHz, 1536 MB RAM, sistema operacional Ubuntu 6.06. O algoritmo foi implementado em linguagem C++, usando o compilador gcc versão 4.0.3.

O método foi avaliado com uma imagem de 500×408 pixels, obtida através da redução da imagem equivalente ao ortofotomosaico do Rio Jucu. O Ortofotomosaico é um produto cartográfico digital de escala 1:15.000 PEC “A”¹, de resolução espacial de 1m, elaborado a partir de um Levantamento Aerofotogramétrico na escala 1:35.000. É formado pela articulação de cerca de 540 blocos de imagens de 10×10 km. Por ser constituído de imagens de alta qualidade, é

¹PEC “A” : Padrão de Exatidão Cartográfica classe “A”, onde o erro de posicionamento de 90% dos pontos amostrados deve ser no máximo de 0,5mm na escala da carta. Para a escala de 1/15.000, corresponde a um erro de posicionamento de no máximo 7,5m. O Ortofotomosaico utilizado é um documento cartográfico georreferenciado no Sistema de Projeção UTM, Datum WGS84, zona 24s.

uma ferramenta muito útil para gestão e monitoramento ambiental, tais como identificação e mapeamento de feições geográficas e do uso do solo. Testes também foram efetuados com a imagem original do ortofotomosaico, de dimensão 2740×2440 pixels.

Sete tipos de testes foram considerados com o objetivo de examinar os impactos dos descritores de textura e/ou cor na qualidade da solução obtida pelo algoritmo:

- Test_{LBP}: Algoritmo *split-and-merge* com histograma LBP;
- Test_{GRAY}: Algoritmo *split-and-merge* com histograma em nível de cinza;
- Test_{PCA}: Algoritmo *split-and-merge* com histograma da primeira componente do PCA aplicado aos canais RGB para redução de dimensão;
- Test_{RGB}: Algoritmo *split-and-merge* com histogramas de cor RGB;
- Test_{LBP+Gray}: Algoritmo *split-and-merge* com histograma LBP e nível de cinza;
- Test_{LBP+RGB}: Algoritmo *split-and-merge* com histograma LBP e RGB;
- Test_{LBP+PCA}: Algoritmo *split-and-merge* com histograma LBP e primeira componente do PCA.

A variação dos valores dos parâmetros considerados em cada teste são $\{0.9, 1.0, \dots, 1.9\}$ para o limiar da fase de *split*, ts , e para o limiar da fase de *merge*, tm , $\{0.1, 0.2, \dots, 0.9\}$ para o parâmetro de ponderação α e $\{4, 8, \dots, 12\}$ para o tamanho mínimo das regiões analisadas na fase *split*, ms . Também foi avaliado o desempenho do algoritmo sem o teste do limiar na fase de *split*, forçando a imagem a ser segmentada até atingir o tamanho mínimo possível da região, considerando $F = N$ quando o limiar é considerado ou $F = S$ caso contrário.

Os testes foram construídos como a seguir. Para cada tamanho mínimo da região na fase de divisão do algoritmo de segmentação, ms , são variados os limiares das fases de *split* e *merge*, ts e tm , respectivamente. Para cada um desses testes, são variados as medidas de distâncias para o cálculo da distância D que avalia a similaridade entre duas regiões. E, por fim, para cada medida de distância, foi considerada a divisão forçada da região, sem o teste do limiar ts . O parâmetro α só é considerado para os testes que combinam cor e textura, e não foi utilizado quando consideradas as distâncias probabilísticas em sua forma analítica.

Depois de segmentada, a imagem tem seus atributos extraídos, sendo posteriormente convertida em um grafo de atributos. Testes com as funções de agregação são também efetuados e avaliados, dado um grafo que representa a imagen modelo, obtido manualmente através da

ferramenta *ImGraph*, e um grafo alvo que representa a imagem a ser reconhecida neste modelo, obtido pelo método de segmentação estudado. Os teste foram construídos de modo que, para cada medida de distância, fosse variado o fator β de ponderação das funções, cujo valor variou de $\{0.1, 0.2, \dots, 0.9\}$.

3.5.1 Segmentação de Imagens

Os testes de segmentação automática foram realizados com o objetivo de verificar se a segmentação feita pelo algoritmo *split-and-merge* produz uma imagem apta a ser utilizada em algoritmos de reconhecimento de cenas, e principalmente se o grafo extraído da imagem através de rotinas acrescentadas no algoritmo de segmentação fazem uma boa representação da imagem a ser reconhecida. O tempo de execução do método, apresentado na Tabela 3.11, foi obtido através da média da execução de 10 iterações, para cada teste, considerando parâmetros fixos para cada um. O tempo foi computado em milissegundos, e a primeira coluna da tabela representa os testes sobre os quais o tempo foi avaliado, a segunda coluna representa a medida de distância considerada para avaliar a similaridade D entre as regiões, como apresentado na Subseção 3.2.4, e as demais colunas representam o tempo para cada tamanho mínimo da região na fase de *split* considerado, ou seja, até que ponto o algoritmo de segmentação deverá dividir as regiões recursivamente em quatro.

Os testes de segmentação foram aplicados tanto na imagem original de 2740×2440 *pixels*, apresentada na Figura 3.5, quanto nas imagens reduzidas da mesma, de 500×408 *pixels*, apresentadas na Figura 3.4. Em ambos os casos, foram consideradas, também, as imagens pré-processadas das imagens avaliadas, que incluem a imagem resultante da aplicação do descritor de texturas LBP (Figuras 3.5(c) e 3.4(c)), a imagem em escala de cinza (Figura 3.5(b) e 3.4(b)), e a imagem resultante da aplicação do PCA (Figura 3.5(d) e 3.4(d)), considerando apenas a primeira componente por questões de economia de memória e velocidade de processamento.

As imagens escolhidas apresentam detalhes finos, devido a alta resolução da imagem original, e grande homogeneidade em termos de cor e textura, tornando complexo o processo de segmentação. A seguir são apresentados alguns dos experimentos realizados.

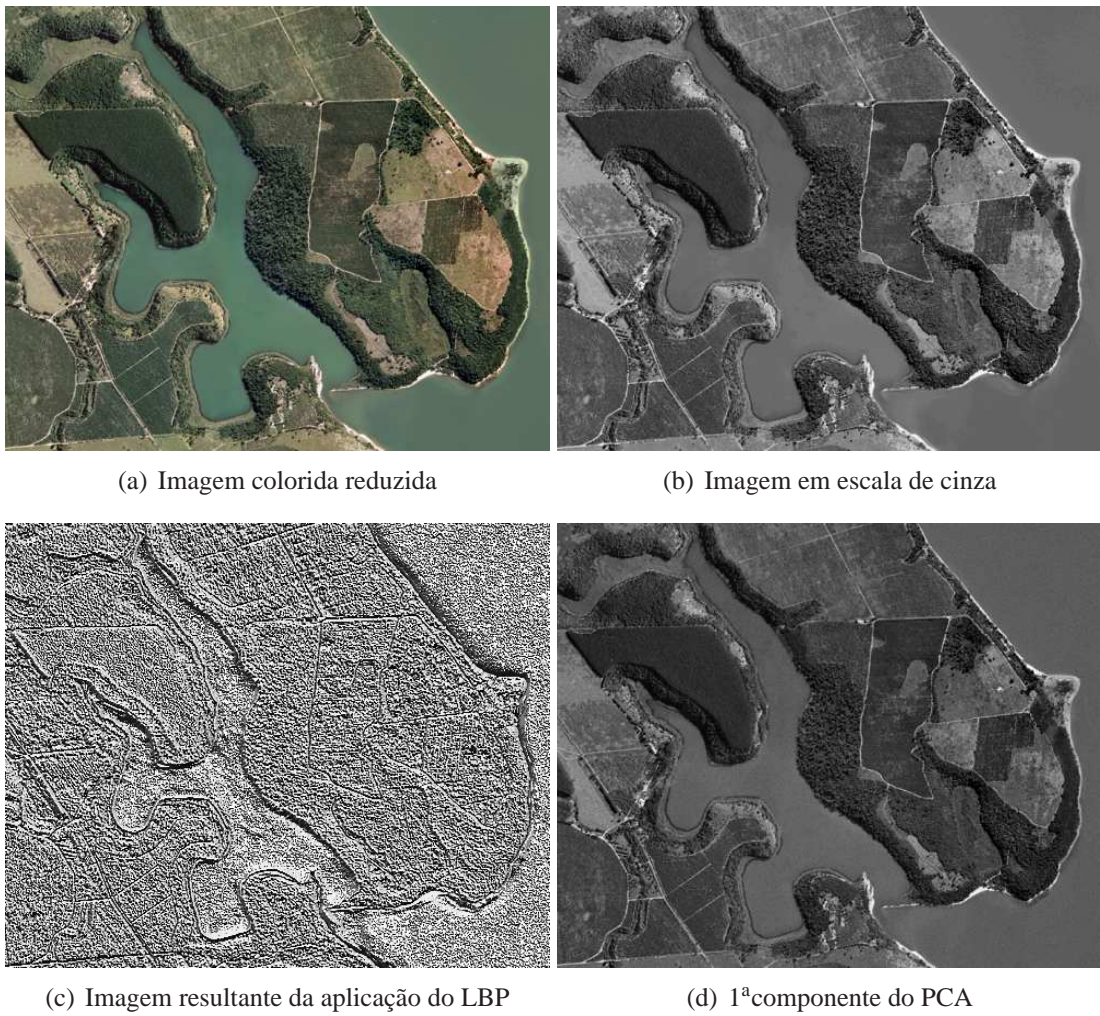


Figura 3.4: Imagens de 500×408 pixels, obtidas através de procedimentos sobre a imagem colorida.

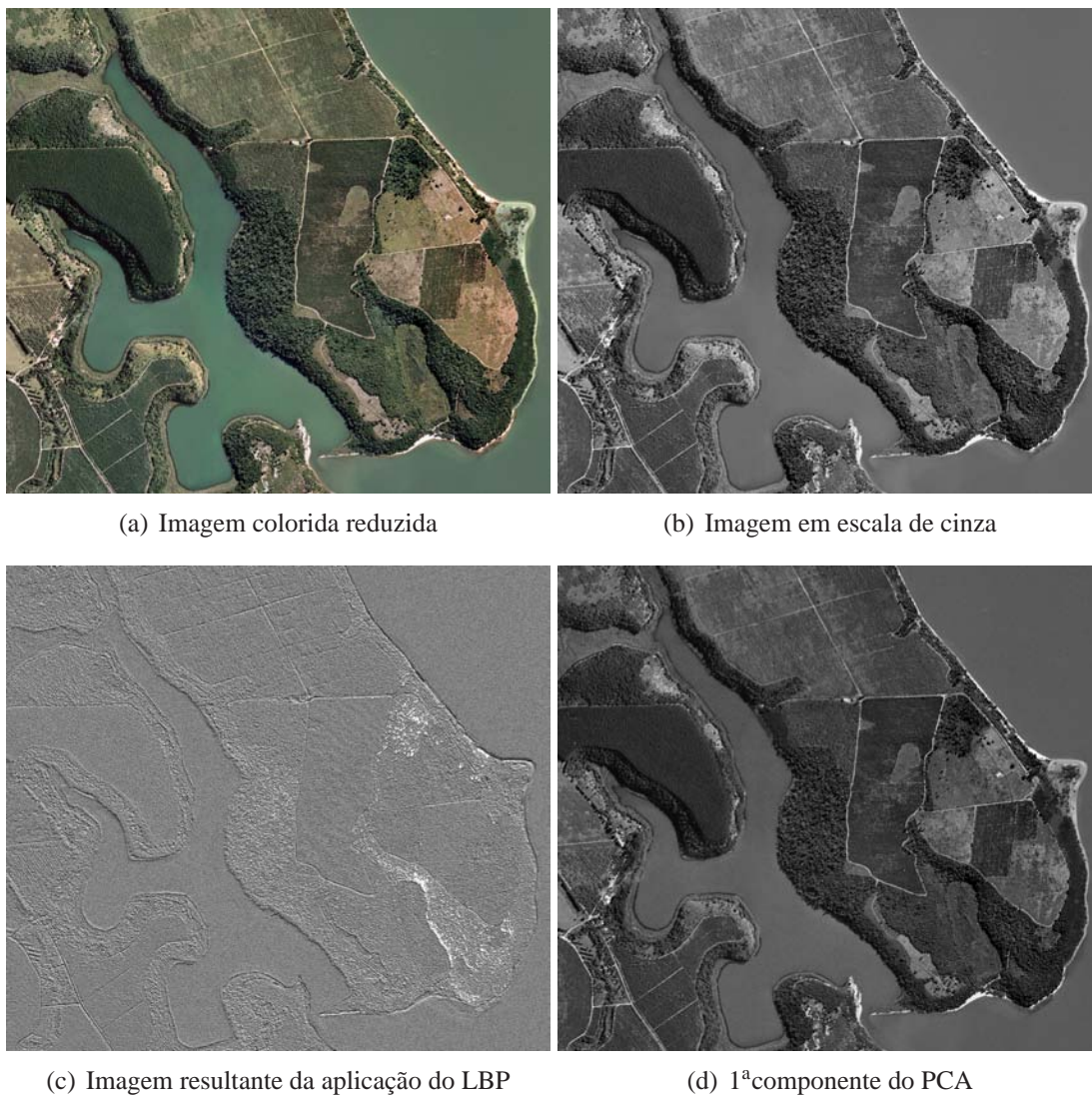


Figura 3.5: Imagens de 2740×2440 pixels, obtidas através de procedimentos sobre a imagem colorida.

Experimento 1 - LBP

Este experimento foi realizado considerando a segmentação sobre o histograma LBP da imagem. A Tabela 3.4 apresenta os resultados obtidos. A primeira coluna da tabela indica a medida de distância utilizada para mensurar a similaridade das regiões em avaliação. As colunas seguintes indicam o valor do parâmetro ms . Também foi avaliado o desempenho do método em relação à imagem original (ver Figura 3.5), cujo resultado é apresentado na Figura 3.6. Outros resultados são apresentados na Figura 3.7.

Neste teste, é possível perceber que a redução da imagem de alta-resolução da Figura 3.5 influenciou consideravelmente no resultado da segmentação. A redução diminuiu a quantidade de detalhes da imagem, como observado nas imagens 3.4(c) e 3.5(c), e a segmentação baseada apenas no critério de textura mostrou-se grosseiro, segmentando a imagem apenas entre duas regiões: terra e água, como observado na Tabela 3.4. Entretanto, a segmentação da imagem com as dimensões originais apenas pelo critério de textura mostrou-se competente (ver Figura 3.6), sugerindo que a textura é um método eficiente para segmentação de imagens de alta-resolução. Outros resultados podem ser vistos na Figura 3.7, e apresentam a segmentação sobre imagens com muita textura. Para tornar mais claro a análise da imagem, por esta ter muitos detalhes, as imagens resultantes da segmentação não foram sobrepostas às imagens originais.



Figura 3.6: $Test_{LBP}$: $tm = ts = 1.1, D = L_D, ms = 8$.








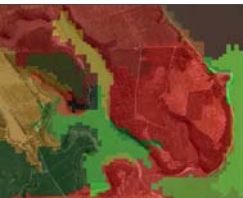







	4	8	12
$D = L_K$	 ts = 1.0, tm = 1.1 F = N	 ts = 1.0, tm = 1.2 F = N	 ts = 1.2, tm = 1.2 F = N
$D = L_B$	 ts = 1.2, tm = 1.2 F = S	 ts = 1.0, tm = 1.2 F = S	 ts = 1.0, tm = 1.1 F = N
$D = L_D$	 ts = 1.0, tm = 1.2 F = N	 ts = 1.0, tm = 1.1 F = N	 ts = 1.0, tm = 1.1 F = S
$D = L_M$	 ts = 1.0, tm = 1.1 F = N	 ts = 1.0, tm = 1.2 F = N	 ts = 1.1, tm = 1.1 F = N
$D = L_E$	 ts = 1.1, tm = 1.1 F = N	 ts = 1.1, tm = 1.2 F = N	 ts = 1.1, tm = 1.1 F = N

Tabela 3.4: $Test_{LBP}$: Imagens segmentadas

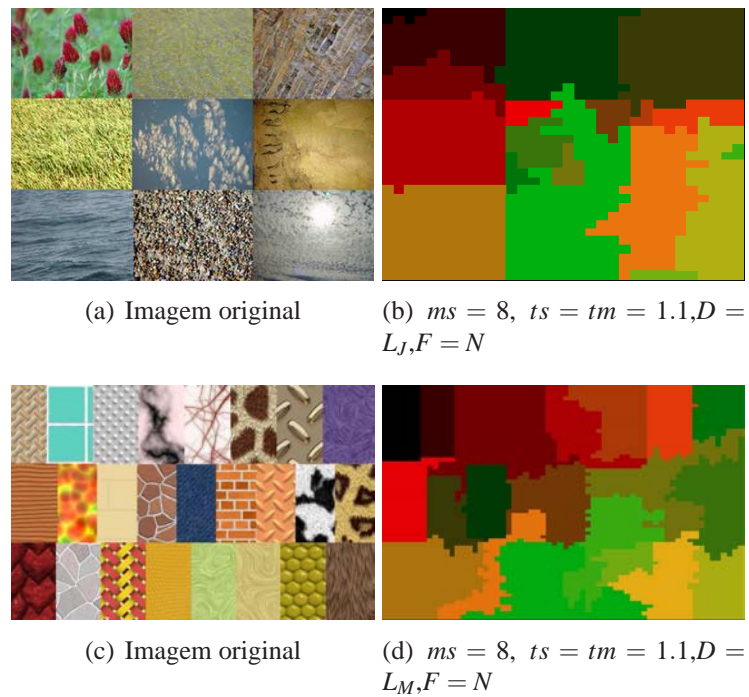


Figura 3.7: Imagens Segmentadas pelo método $Test_{LBP}$.

Experimento 2 - GRAY

Este experimento foi realizado considerando a segmentação sobre do histograma do nível de cinza da imagem. A Tabela 3.5 apresenta os resultados obtidos. A primeira coluna da tabela indica a medida de distância utilizada para mensurar a similaridade das regiões em avaliação. As colunas seguintes indicam o valor do parâmetro ms . Também foi avaliado o desempenho do método em relação à imagem original (ver Figura 3.5), cujo resultado é apresentado na Figura 3.8. Outros resultados são apresentados na Figura 3.9.

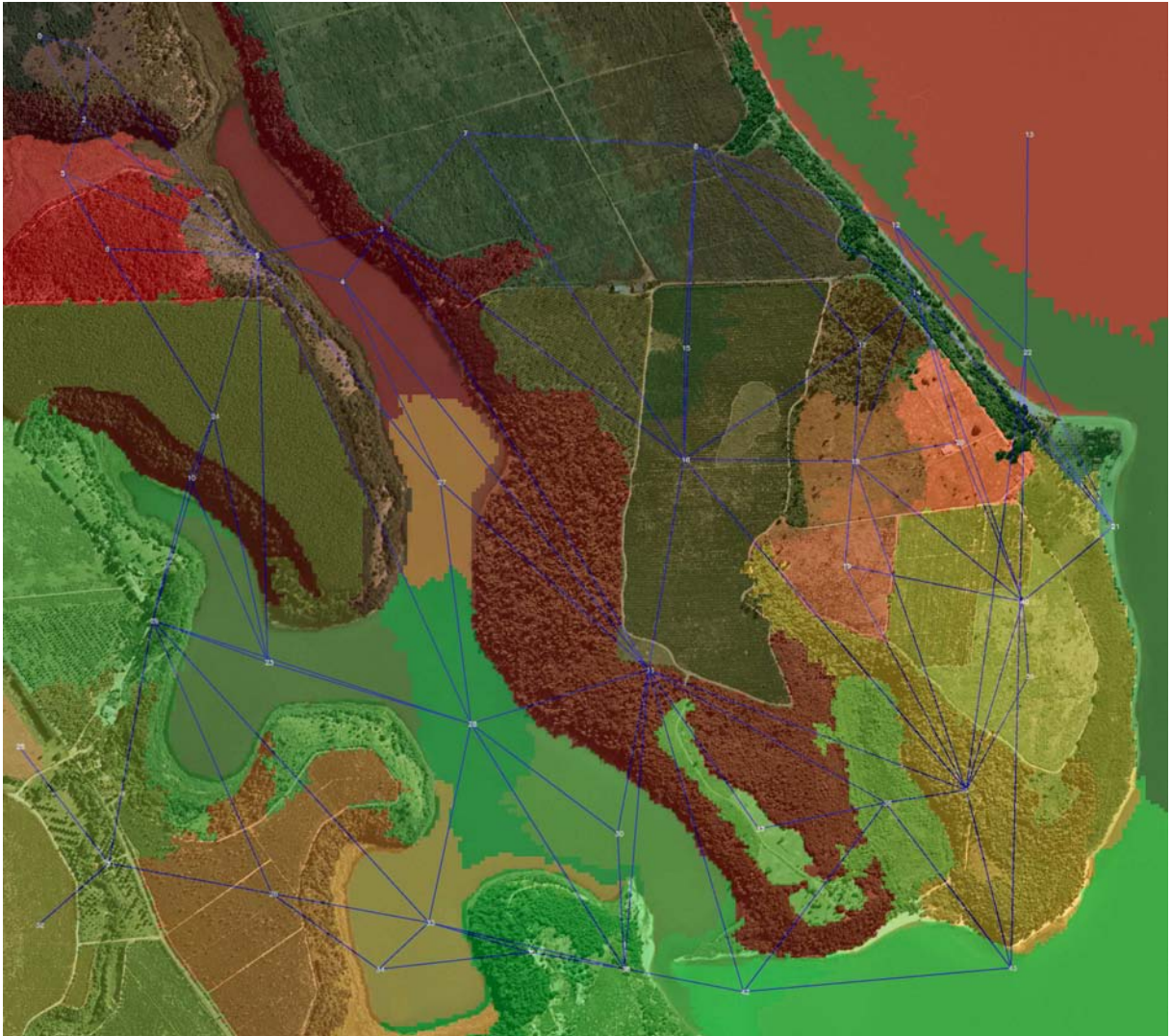


Figura 3.8: $Test_{GRAY}$: $tm = ts = 1.1, D = L_D, M = 8$.
















	4	8	12
$D = L_K$	 ts = 1.0, tm = 1.1 F = N	 ts = 1.2, tm = 1.1 F = N	 ts = 1.1, tm = 1.1 F = N
$D = L_B$	 ts = 1.1, tm = 1.1 F = S	 ts = 1.0, tm = 1.1 F = N	 ts = 1.0, tm = 1.1 F = N
$D = L_D$	 ts = 1.0, tm = 1.1 F = N	 ts = 1.2, tm = 1.1 F = N	 ts = 1.1, tm = 1.1 F = N
$D = L_M$	 ts = 1.2, tm = 1.1 F = S	 ts = 1.0, tm = 1.1 F = S	 ts = 1.0, tm = 1.1 F = N
$D = L_E$	 ts = 1.1, tm = 1.0 F = S	 ts = 1.1, tm = 1.1 F = S	 ts = 1.1, tm = 1.1 F = N

Tabela 3.5: $Test_{GRAY}$: Imagens segmentadas.

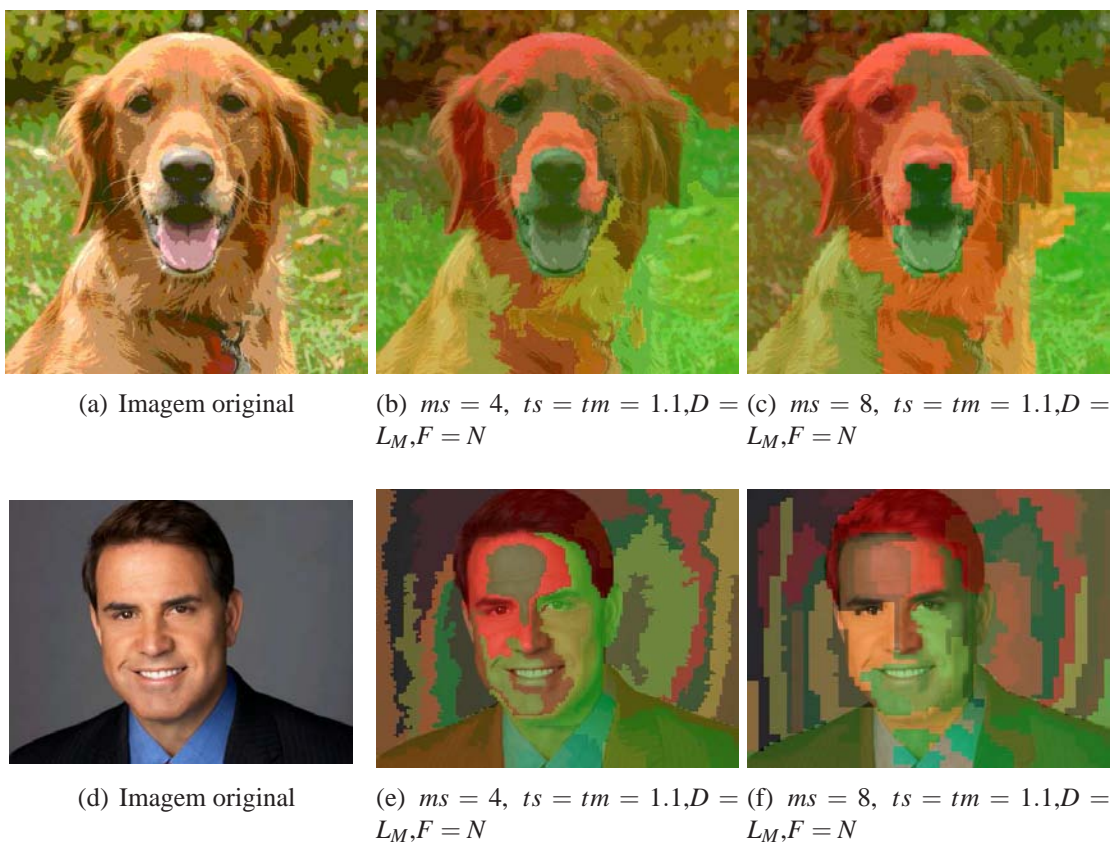


Figura 3.9: Imagens Segmentadas pelo método $Test_{GRAY}$.

Neste teste, é possível perceber que a qualidade da segmentação nas imagens reduzidas, apresentadas na Tabela 3.5, melhoraram consideravelmente em relação a segmentação utilizando apenas o descritor de textura LBP. Entretanto, ao se avaliar as Figuras 3.6 e 3.8, percebe-se que a segmentação da primeira é mais refinada do que a segunda, que uniu regiões com vegetação distintas. Isto sugere que o critério para avaliar a similaridade pode ser melhorado combinando ambas as características, como será apresentado no próximo experimento.

Experimento 3 - GRAY + LBP

Este experimento foi realizado considerando a segmentação sobre o histograma LBP e nível de cinza da imagem. A Tabela 3.6 apresenta os resultados obtidos. A primeira coluna da tabela indica a medida de distância utilizada para medir a similaridade das regiões em avaliação. As colunas seguintes indicam o valor do parâmetro ms . Também foi avaliado o desempenho do método em relação à imagem original (ver Figura 3.5), cujo resultado é apresentado na Figura 3.10. Outros resultados são apresentados na Figura 3.11.



Figura 3.10: $Test_{GRAY+LBP}$: $tm = ts = 1.1, D = L_D, M = 8$.



















	4	8	12
$D = L_K$	 ts = 1.1, tm = 1.1 F = N, $\alpha = 0.90$	 ts = 1.1, tm = 1.1 F = N, $\alpha = 0.70$	 ts = 1.1, tm = 1.1 F = N, $\alpha = 0.70$
$D = L_B$	 ts = 1.1, tm = 1.1 F = N, $\alpha = 0.90$	 ts = 1.1, tm = 1.1 F = N, $\alpha = 0.50$	 ts = 1.1, tm = 1.1 F = N, $\alpha = 0.90$
$D = L_D$	 ts = 1.0, tm = 1.1 F = N, $\alpha = 0.90$	 ts = 1.0, tm = 1.2 F = S, $\alpha = 0.70$	 ts = 1.1, tm = 1.1 F = N, $\alpha = 0.80$
$D = L_M$	 ts = 1.1, tm = 1.1 F = N, $\alpha = 0.50$	 ts = 1.1, tm = 1.1 F = N, $\alpha = 0.50$	 ts = 1.1, tm = 1.2 F = N, $\alpha = 0.90$
$D = L_E$	 ts = 1.1, tm = 1.1 F = N	 ts = 1.1, tm = 1.1 F = N	 ts = 1.1, tm = 1.2 F = N
$D = J_M$	 ts = 1.1, tm = 1.1 F = N	 ts = 1.1, tm = 1.0 F = S	 ts = 1.1, tm = 1.1 F = S

Tabela 3.6: $Test_{GRAY+LBP}$: Imagens segmentadas.

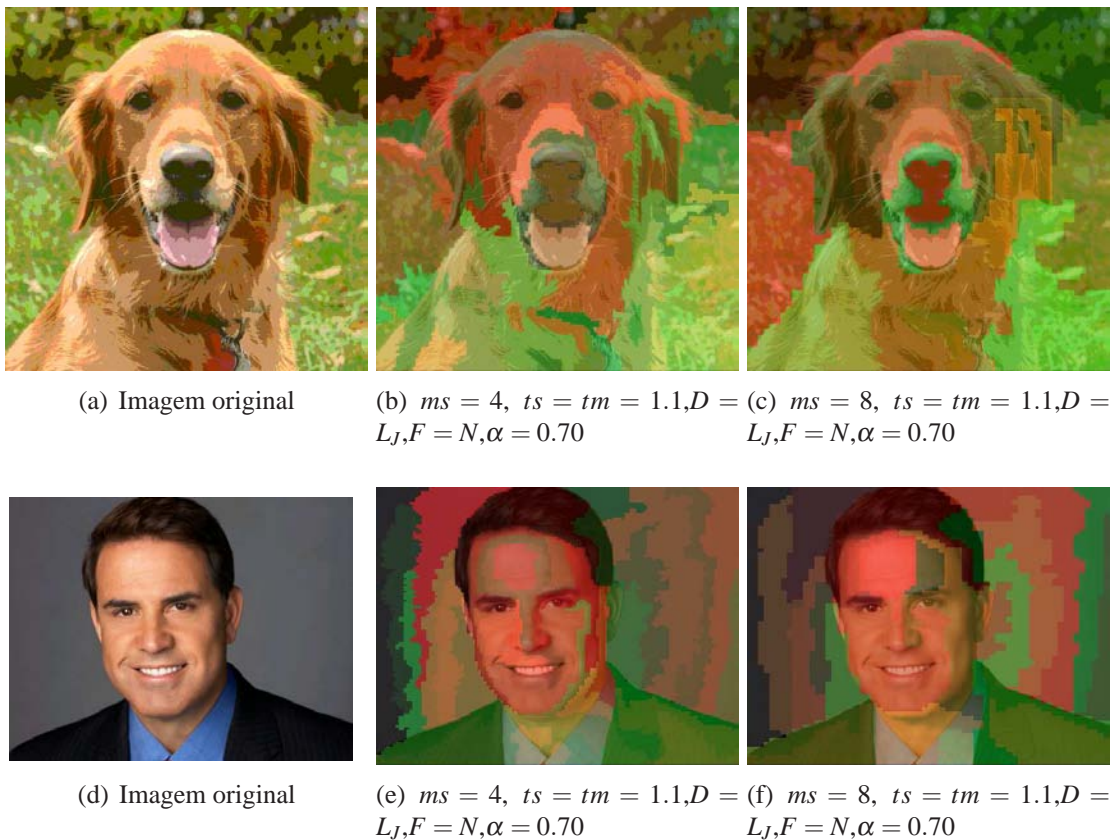


Figura 3.11: Imagens Segmentadas pelo método $Test_{GRAY+LBP}$.

Neste teste, é possível perceber que a combinação de critérios influenciou positivamente no resultado da segmentação, como observado nas imagens 3.10 e 3.6. As distâncias probabilísticas na forma analítica obtiveram um resultado bastante equivocado. Por este motivo, apenas o melhor resultado será mostrado, conseguido através da medida de distância Matusita.

Experimento 4 - PCA

Este experimento foi realizado considerando a segmentação sobre o histograma da imagem que representa a primeira componente do PCA, ou seja, a componente com maior variabilidade entre as características consideradas. A Tabela 3.7 apresenta os resultados obtidos. A primeira coluna da tabela indica a medida de distância utilizada para mensurar a similaridade das regiões em avaliação. As colunas seguintes indicam o valor do parâmetro ms . Também foi avaliado o desempenho do método em relação à imagem original (ver Figura 3.5), cujo resultado é apresentado na Figura 3.12. Outros resultados são apresentados na Figura 3.13.

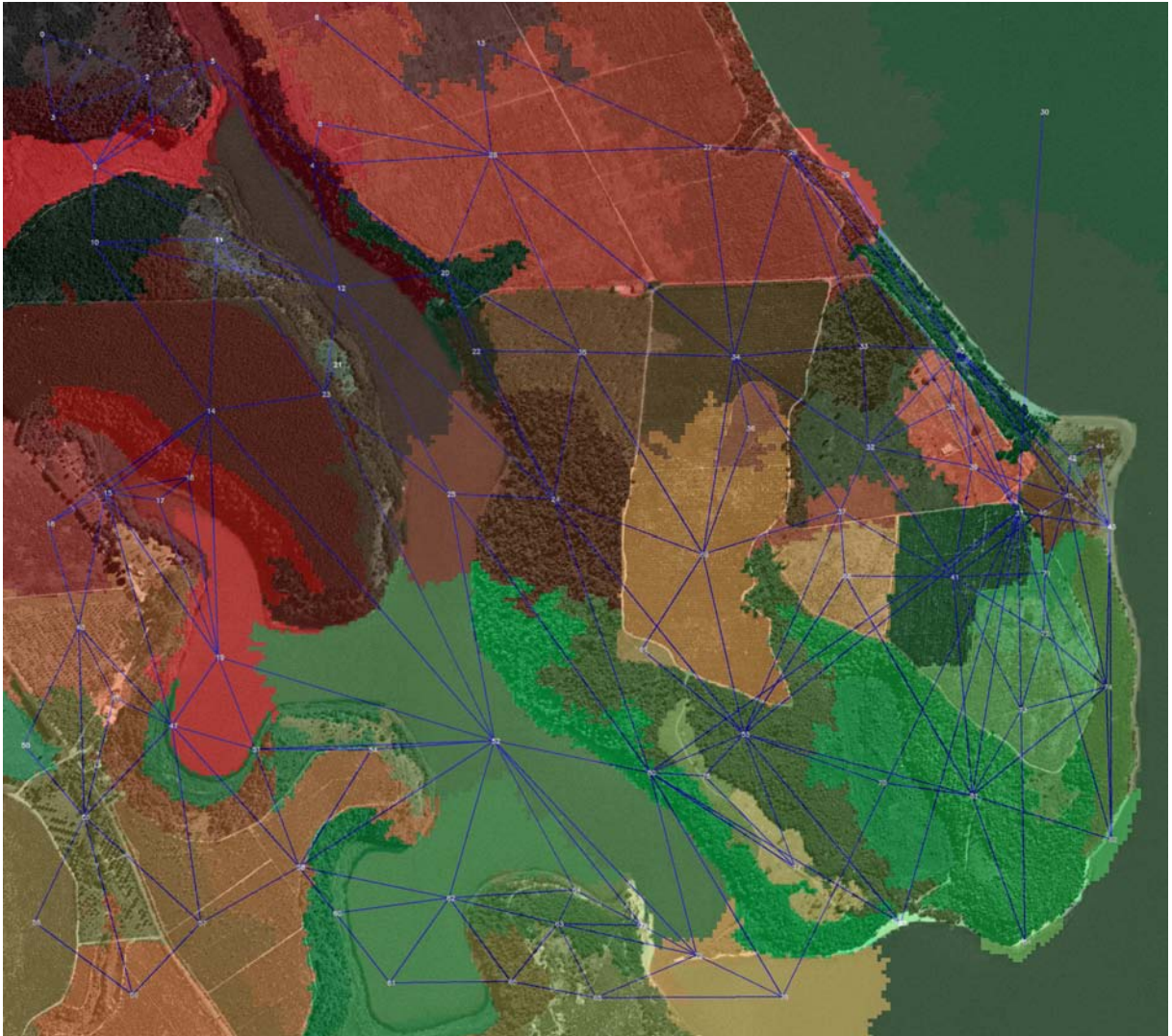


Figura 3.12: $Test_{PCA}$: $t_m = t_s = 1.1, D = D_J, M = 8$.


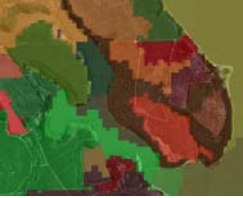













	4	8	12
$D = L_K$	 ts = 1.0, tm = 1.1 F = N	 ts = 1.0, tm = 1.1 F = N	 ts = 1.2, tm = 1.1 F = N
$D = L_B$	 ts = 1.0, tm = 1.1 F = N	 ts = 1.0, tm = 1.1 F = N	 ts = 1.0, tm = 1.1 F = S
$D = L_D$	 ts = 1.0, tm = 1.2 F = N	 ts = 1.2, tm = 1.2 F = N	 ts = 1.0, tm = 1.1 F = N
$D = L_M$	 ts = 1.2, tm = 1.2 F = N	 ts = 1.0, tm = 1.1 F = S	 ts = 1.0, tm = 1.2 F = N
$D = L_E$	 ts = 1.2, tm = 1.1 F = N	 ts = 1.0, tm = 1.1 F = S	 ts = 1.1, tm = 1.1 F = N

Tabela 3.7: $Test_{PCA}$: Imagens segmentadas.

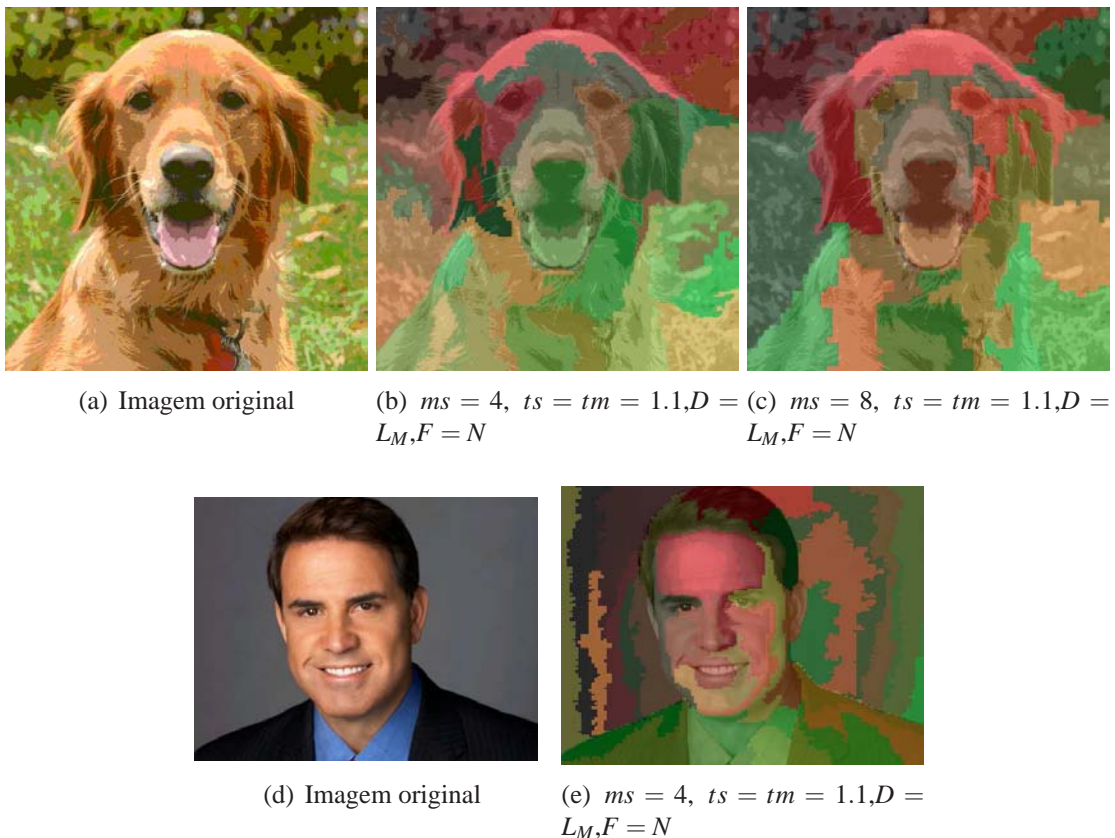


Figura 3.13: Imagens Segmentadas pelo método $Test_{PCA}$.

Neste teste, é possível perceber que a nova característica obtida pela redução da dimensionalidade da imagem melhorou consideravelmente a segmentação da mesma, em relação aos testes anteriores, principalmente se considerados os resultados apresentados na Tabela 3.7, que mostram um refinamento da segmentação, onde vegetações contínuas permaneceram como uma única região. A mesma constatação pode ser obtida através da análise da Figura 3.12. Outros resultados podem ser vistos na Figura 3.13.

Experimento 5 - PCA + LBP

Este experimento foi realizado considerando a segmentação sobre o histograma da imagem que representa a primeira componente do PCA, ou seja, a componente com maior variabilidade entre as características consideradas, em conjunto com o descritor de texturas LBP. A Tabela 3.8 apresenta os resultados obtidos. A primeira coluna da tabela indica a medida de distância utilizada para mensurar a similaridade das regiões em avaliação. As colunas seguintes indicam o valor do parâmetro ms . Também foi avaliado o desempenho do método em relação à imagem original (ver Figura 3.5), cujo resultado é apresentado na Figura 3.14. Outros resultados são apresentados na Figura 3.15.



Figura 3.14: $Test_{PCA+LBP}$: $tm = ts = 1.1, D = L_D, M = 8$.


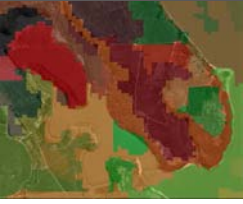














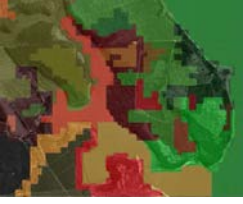
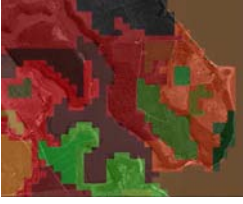
	4	8	12
$D = L_K$	 ts = 1.1, tm = 1.1 F = N, $\alpha = 0.80$	 ts = 1.2, tm = 1.1 F = N, $\alpha = 0.50$	 ts = 1.1, tm = 1.2 F = N, $\alpha = 0.80$
$D = L_B$	 ts = 1.2, tm = 1.1 F = N, $\alpha = 0.80$	 ts = 1.1, tm = 1.1 F = N, $\alpha = 0.70$	 ts = 1.2, tm = 1.1 F = N, $\alpha = 0.90$
$D = L_D$	 ts = 1.0, tm = 1.1 F = N, $\alpha = 0.50$	 ts = 1.0, tm = 1.1 F = S, $\alpha = 0.90$	 ts = 1.1, tm = 1.1 F = N, $\alpha = 0.90$
$D = L_M$	 ts = 1.1, tm = 1.1 F = N, $\alpha = 0.90$	 ts = 1.1, tm = 1.1 F = N, $\alpha = 0.60$	 ts = 1.1, tm = 1.2 F = N, $\alpha = 0.70$
$D = L_E$	 ts = 1.1, tm = 1.1 F = N	 ts = 1.0, tm = 1.1 F = N	 ts = 1.1, tm = 1.1 F = N
$D = J_B$	 ts = 1.1, tm = 1.2 F = N	 ts = 1.1, tm = 1.2 F = S	 ts = 1.1, tm = 1.2 F = S

Tabela 3.8: $Test_{PCA+LBP}$: Imagens segmentadas.

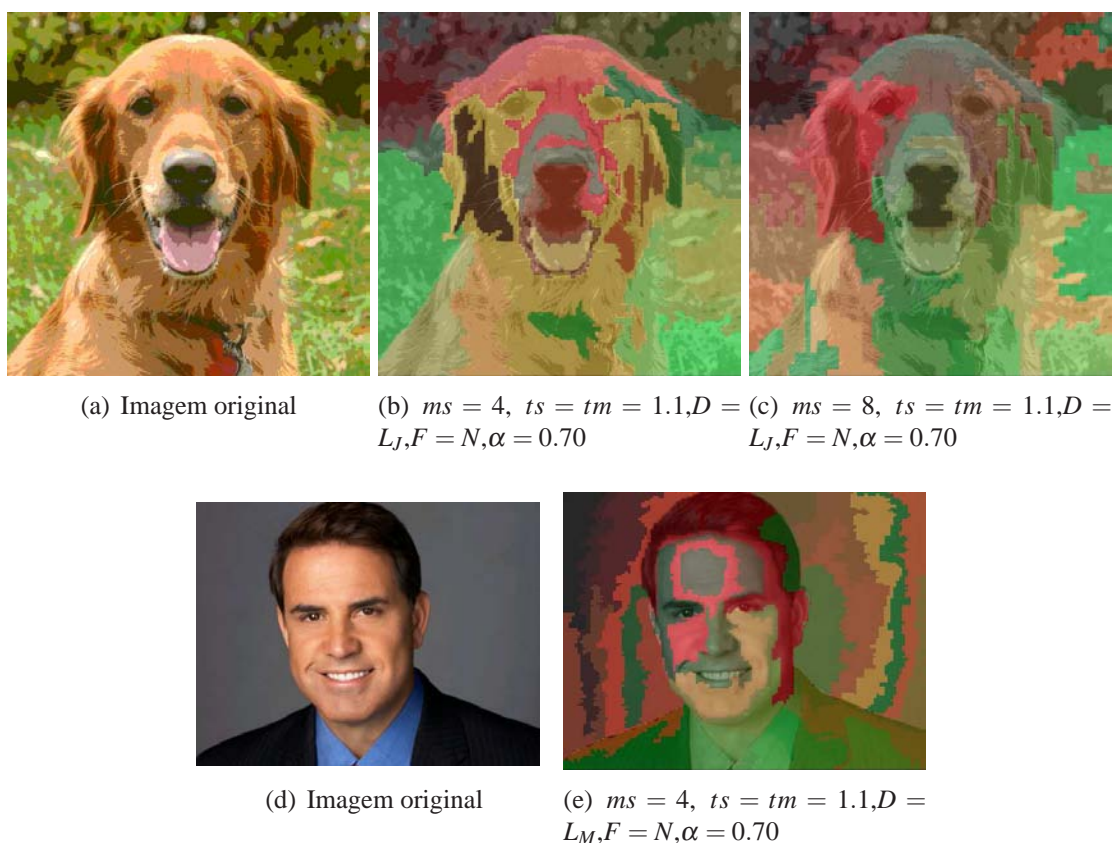


Figura 3.15: Imagens Segmentadas pelo método $Test_{PCA+LBP}$.

Neste teste, é possível perceber que a nova característica obtida pela redução da dimensionalidade da imagem combinado com o descritor de textura resultou em um aprimoramento da segmentação, mesmo nas imagens reduzidas, em relação aos testes anteriores. As distâncias probabilísticas na forma analítica obtiveram um resultado bastante equivocado. Por este motivo, apenas o melhor resultado será mostrado, conseguido através da medida de distância Bhattacharyya.

Experimento 6 - RGB

Este experimento foi realizado considerando a segmentação sobre o histograma de cada canal do sistema de cor RGB da imagem. A Tabela 3.9 apresenta os resultados obtidos. A primeira coluna da tabela indica a medida de distância utilizada para mensurar a similaridade das regiões em avaliação. As colunas seguintes indicam o valor do parâmetro ms . Também foi avaliado o desempenho do método em relação à imagem original (ver Figura 3.5), cujo resultado é apresentado na Figura 3.16. Outros resultados são apresentados na Figura 3.17.

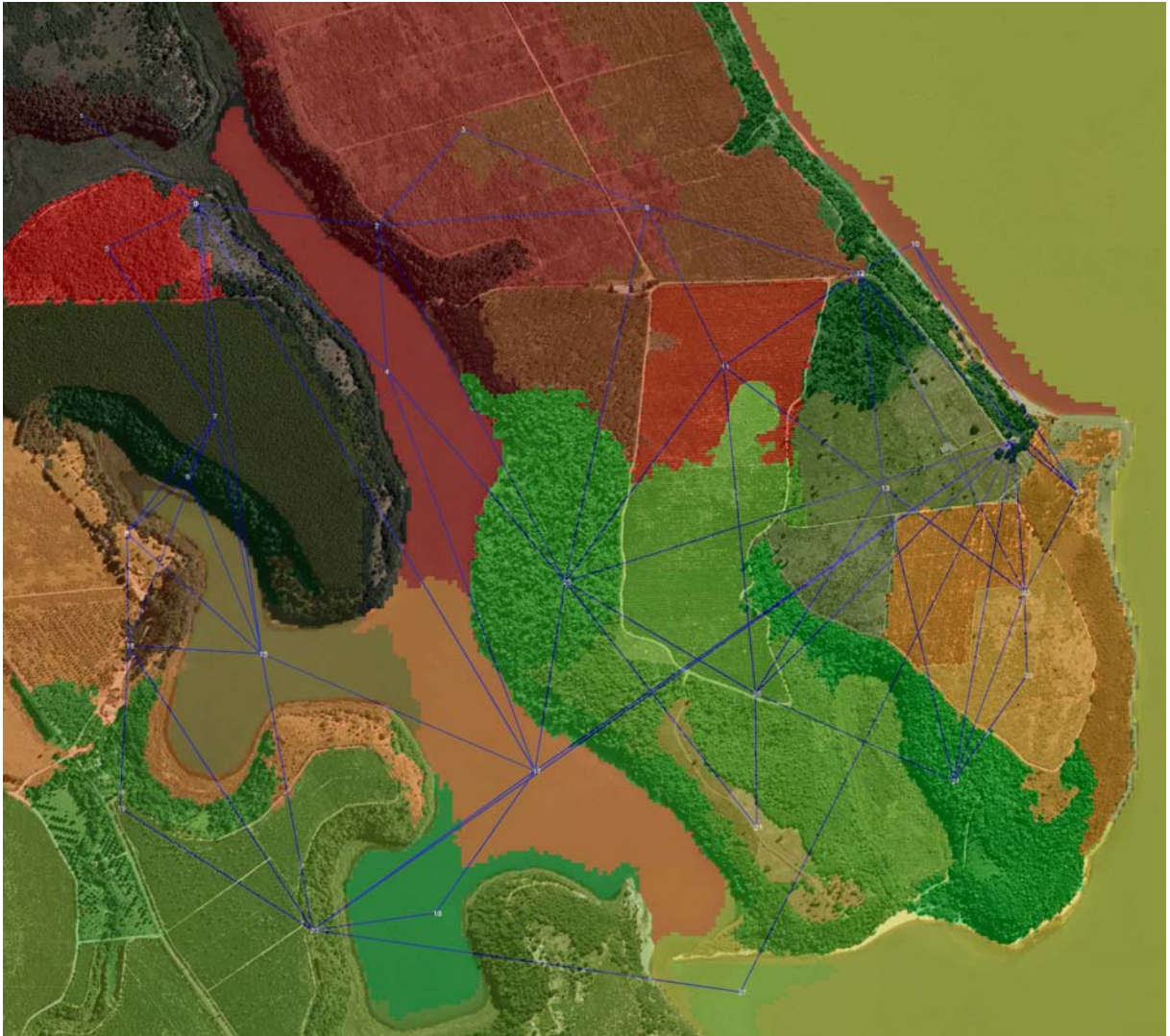


Figura 3.16: $Test_{RGB}$: $tm = ts = 1.1, D = L_D, M = 8$.









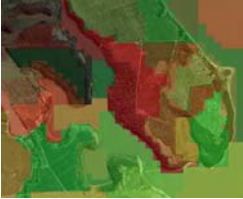









	4	8	12
$D = L_K$	 ts = 1.1, tm = 1.1 F = N	 ts = 1.1, tm = 1.1 F = N	 ts = 1.1, tm = 1.1 F = N
$D = L_B$	 ts = 1.1, tm = 1.1 F = N	 ts = 1.1, tm = 1.1 F = N	 ts = 1.1, tm = 1.1 F = N
$D = L_D$	 ts = 1.0, tm = 1.1 F = N	 ts = 1.1, tm = 1.2 F = N	 ts = 1.1, tm = 1.2 F = N
$D = L_M$	 ts = 1.0, tm = 1.1 F = N	 ts = 1.1, tm = 1.1 F = N	 ts = 1.2, tm = 1.1 F = N
$D = L_E$	 ts = 1.1, tm = 1.1 F = N	 ts = 1.0, tm = 1.1 F = N	 ts = 1.1, tm = 1.0 F = N
$D = J_M$	 ts = 1.1, tm = 1.1 F = N	 ts = 1.1, tm = 1.0 F = S	 ts = 1.1, tm = 1.1 F = N

Tabela 3.9: $Test_{RGB}$: Imagens segmentadas.

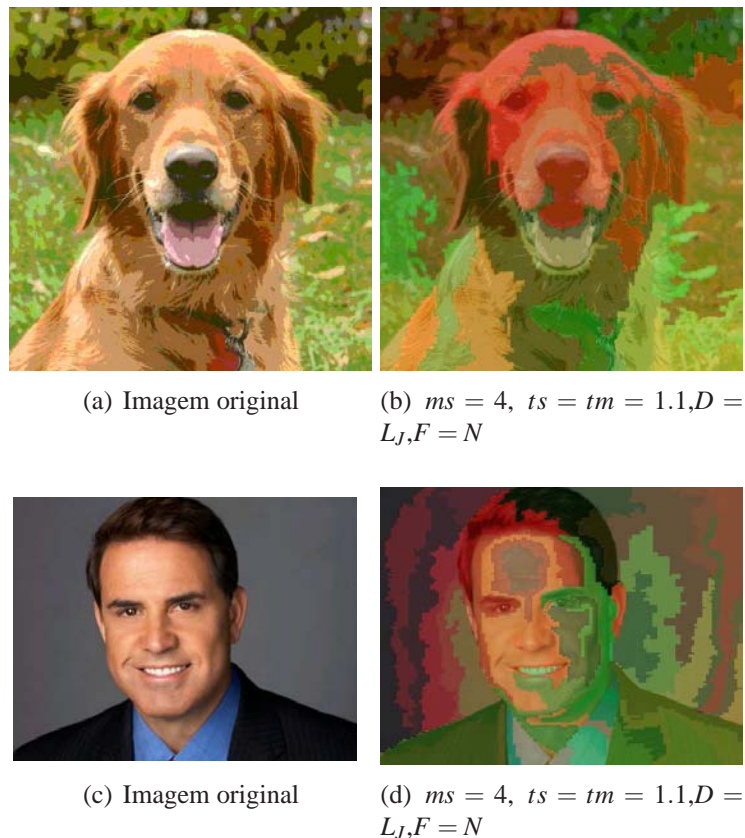


Figura 3.17: Imagens Segmentadas pelo método $Test_{RGB}$.

Neste teste, é possível perceber que os três canais de cor, combinados através da soma ponderada que considera a percepção de cor do sistema visual humano, obteve um bom resultado, mas não melhor do que o apresentado pela combinação do PCA com o descritor LBP, como pode ser constatado nas Tabelas 3.8 e 3.9 e nas Figuras 3.14 e 3.16. As distâncias probabilísticas na forma analítica obtiveram um resultado bastante equivocado. Por este motivo, apenas o melhor resultado será mostrado, conseguido através da medida de distância Matusita.

Experimento 7 - RGB + LBP

Este experimento foi realizado considerando a segmentação sobre o histograma de cada canal do sistema de cor RGB da imagem combinado com o descritor de textura LBP. A Tabela 3.10 apresenta os resultados obtidos. A primeira coluna da tabela indica a medida de distância utilizada para mensurar a similaridade das regiões em avaliação. As colunas seguintes indicam o valor do parâmetro ms . Também foi avaliado o desempenho do método em relação à imagem original (ver Figura 3.5), cujo resultado é apresentado na Figura 3.18. Outros resultados são apresentados na Figura 3.19.



Figura 3.18: $Test_{RGB+LBP}$: $tm = ts = 1.1, D = L_D, M = 8$.



















	4	8	12
$D = L_K$	 ts = 1.1, tm = 1.1 F = N, $\alpha = 0.70$	 ts = 1.1, tm = 1.1 F = N, $\alpha = 0.90$	 ts = 1.1, tm = 1.1 F = N, $\alpha = 0.70$
$D = L_B$	 ts = 1.1, tm = 1.1 F = N, $\alpha = 0.80$	 ts = 1.1, tm = 1.1 F = N, $\alpha = 0.50$	 ts = 1.0, tm = 1.1 F = N, $\alpha = 0.60$
$D = L_D$	 ts = 1.1, tm = 1.1 F = N, $\alpha = 0.80$	 ts = 1.0, tm = 1.2 F = N, $\alpha = 0.90$	 ts = 1.1, tm = 1.1 F = N, $\alpha = 0.70$
$D = L_M$	 ts = 1.1, tm = 1.1 F = N, $\alpha = 0.60$	 ts = 1.1, tm = 1.1 F = N, $\alpha = 0.50$	 ts = 1.1, tm = 1.1 F = N, $\alpha = 0.90$
$D = L_E$	 ts = 1.0, tm = 1.1 F = N	 ts = 1.1, tm = 1.1 F = N	 ts = 1.1, tm = 1.1 F = N
$D = J_D$	 ts = 1.1, tm = 1.1 F = N	 ts = 1.1, tm = 1.1 F = N	 ts = 1.1, tm = 1.1 F = N

Tabela 3.10: $Test_{RGB+LBP}$: Imagens segmentadas.

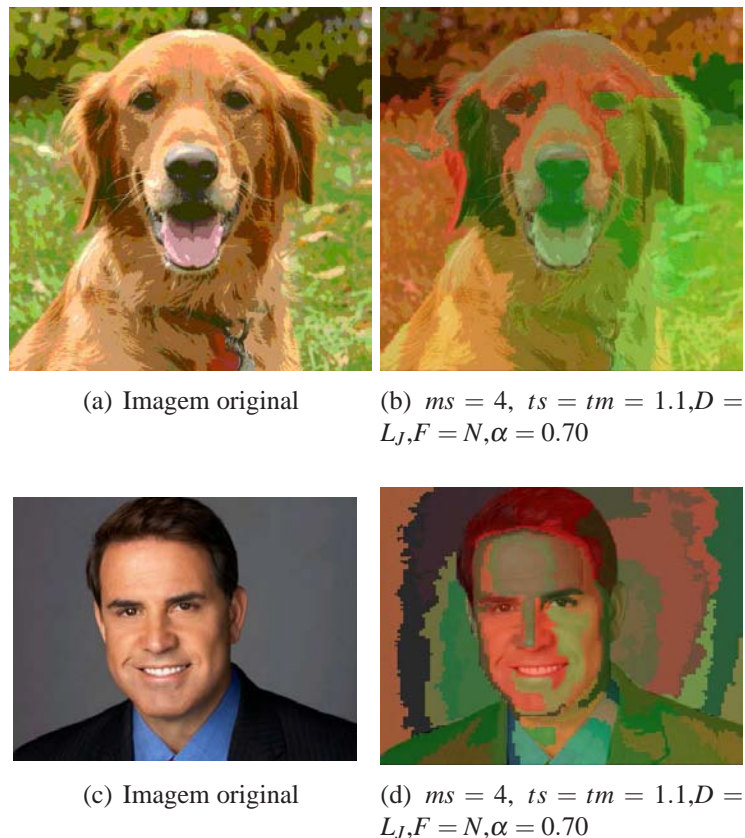


Figura 3.19: Imagens Segmentadas pelo método $Test_{RGB+LBP}$.

Neste teste, é possível perceber que a combinação de cor e textura obteve o melhor resultado até então, como pode ser constatado na Tabela 3.10 e na Figura 3.18. A combinação, novamente, refinou a segmentação em relação ao uso isolado de apenas uma característica. As distâncias probabilísticas na forma analítica obtiveram um resultado bastante equivocado. Por este motivo, apenas o melhor resultado será mostrado, conseguido através da medida de distância Divergência.

Análise dos Resultados do Processo de Segmentação

Com base nos resultados apresentados no tópico anterior, pode-se observar que existe uma extensa possibilidade de variação de parâmetros, tais como medidas de características utilizadas, tamanho mínimo da região na fase de *split*, tipo de medida de distância e segmentação utilizadas, e variação de pesos e limiares da análise de similaridade entre regiões. Os resultados mostraram que a redução da imagem provocou uma supressão das características da mesma, comprometendo o resultado da segmentação, em especial quando considerado apenas o descritor de texturas. Outro fator relevante no resultado da segmentação é o tamanho mínimo da região analisada na fase de *split*. Pela análise dos resultados, verificou-se que o tamanho ideal da região mínima está diretamente relacionado ao nível de detalhe dos objetos da imagem. Para

imagens com objetos pequenos, como as imagens de terrenos, regiões menores (4×4 pixels, por exemplo) produzem melhores resultados. Para imagens com objetos grandes ou poucos detalhes, como as imagens de mosaico, regiões maiores (por exemplo, 12×12 pixels), resultam em melhores segmentações, com a vantagem de serem efetuadas em menos tempo, como observado na Tabela 3.11.

Também foram realizados testes variando-se os limiares da fase de *split* e da fase de *merge*. É possível observar que os melhores resultados concentraram-se nos valores do intervalo [1.0, 1.2] para ambos os casos. A contribuição da textura também é um fator importante no processo de segmentação, e os melhores resultados atribuíam ao fator de ponderação α um valor do intervalo [0.5, 0.9]. Este fator deve ser escolhido de forma que a textura seja completamente descrita, pois caso contrário pode não conter todos os relacionamentos de intensidades que a caracterizam. Entretanto, para imagens com o histograma LBP comprometido pela falta de detalhes e texturas, caso apresentado na Figura 3.4(c) devido à redução da imagem original, este fator deve privilegiar o descritor de cor da imagem para um melhor resultado. Assim, foi constatado que a textura forneceu um refinamento na qualidade da segmentação, em todos os testes em que foi considerada, em comparação com a segmentação equivalente baseada apenas em um critério.

A extração de características foi outro fator que se mostrou importante na segmentação. Embora o custo computacional seja inicialmente alto, há uma compensação na redução da dimensionalidade das características analisadas. Além disso, os testes realizados com a imagem colorida utilizando-se unicamente o sistema de cor RGB tiveram tanto sucesso quanto a segmentação realizada sobre a componente principal do PCA, que possui a vantagem de economizar tempo e memória (ver Tabela 3.11).

Quanto às diferentes medidas de distância pelas quais a imagem pode ser analisada, verificou-se que a distância de Divergência, de Matusita, de Bhattacharyya e euclidiana apresentaram resultados semelhantes e satisfatórios, no sentido de dividir a imagem por tipo de vegetação, com especial destaque para as duas primeiras. Em todos os casos, a forma analítica das medidas de distância probabilísticas consideradas retornaram resultados equivocados, em parte devido a essas medidas serem utilizadas em classificadores com taxas de aprendizado, de forma que a variabilidade entre as classes analisadas sempre aumente. Os resultados observados nas Figuras 3.7, 3.9, 3.11, 3.13, 3.15, 3.17 e 3.19 ilustram que o método pode ser aplicado às mais diversas imagens e não apenas a imagens de alta-resolução. Deve-se observar, no entanto, que a qualidade dos resultados obtidos dependerá do objetivo do experimento. Por exemplo, a figura 3.9 foi supersegmentada, criando um grafo com muitos vértices e arestas, tornando inviável sua

análise em termos de equivalência de regiões com uma imagem modelo.

A Tabela 3.11 apresenta o tempo de execução do método, em milissegundos. É possível observar que, quanto menor ms , maior é o tempo de processamento. As formas analíticas das medidas de distância também demandaram um tempo maior para executarem o algoritmo, devido às sucessivas operações sobre matrizes.

Teste	D	4	8	12
Test _t LBP	LD	32.620000	3.690000	3.920000
Test _t LBP	LK	25.200000	2.080000	1.100000
Test _t LBP	LM	19.990000	0.870000	0.790000
Test _t LBP	LB	20.180000	0.940000	0.840000
Test _t LBP	LE	18.250000	0.760000	0.700000
Test _{GRAY}	LD	32.160000	3.540000	1.330000
Test _{GRAY}	LK	26.630000	2.280000	0.800000
Test _{GRAY}	LM	20.820000	0.930000	0.390000
Test _{GRAY}	LB	18.630000	0.930000	0.380000
Test _{GRAY}	LE	20.440000	0.770000	0.710000
Test _{PCA}	LD	35.620000	4.050000	1.550000
Test _{PCA}	LK	30.310000	2.920000	1.010000
Test _{PCA}	LM	21.150000	1.150000	0.600000
Test _{PCA}	LB	21.670000	1.260000	0.600000
Test _{PCA}	LE	20.440000	1.060000	0.6400000
Test _{GRAY+LBP}	LD	46.800000	6.880000	7.630000
Test _{GRAY+LBP}	LK	37.720000	4.940000	4.160000
Test _{GRAY+LBP}	LM	21.350000	1.170000	1.210000
Test _{GRAY+LBP}	LB	21.830000	1.330000	1.400000
Test _{GRAY+LBP}	LE	20.150000	1.130000	1.190000
Test _{GRAY+LBP}	J _B	30.610000	14.050000	3.23000000
Test _{GRAY+LBP}	J _K	136.020000	19.360000	8.9600000
Test _{GRAY+LBP}	J _M	118.730000	20.150000	8.400000
Test _{GRAY+LBP}	J _D	114.830000	20.950000	7.000000
Test _{PCA+LBP}	LD	47.200000	6.880000	12.230000
Test _{PCA+LBP}	LK	36.770000	3.620000	3.150000
Test _{PCA+LBP}	LM	21.560000	1.390000	2.120000
Test _{PCA+LBP}	LB	22.060000	1.590000	1.950000
Test _{PCA+LBP}	LE	21.230000	1.160000	2.010000
Test _{PCA+LBP}	J _B	21.320000	13.210000	6.800000
Test _{PCA+LBP}	J _K	137.680000	21.200000	7.290000
Test _{PCA+LBP}	J _M	121.190000	20.290000	6.740000
Test _{PCA+LBP}	J _D	120.480000	21.690000	7.240000
Test _{RGB}	LD	70.230000	10.390000	3.410000
Test _{RGB}	LK	40.040000	5.660000	1.790000
Test _{RGB}	LM	24.130000	1.640000	0.600000
Test _{RGB}	LB	26.580000	1.760000	0.620000
Test _{RGB}	LE	25.190000	1.570000	0.550000
Test _{RGB}	J _B	42.040000	18.520000	5.950000
Test _{RGB}	J _K	137.350000	22.500000	7.280000
Test _{RGB}	J _M	126.410000	20.610000	6.430000
Test _{RGB}	J _D	134.270000	22.010000	6.890000
Test _{RGB+LBP}	LD	71.780000	13.280000	14.950000
Test _{RGB+LBP}	LK	40.820000	4.920000	2.410000
Test _{RGB+LBP}	LM	20.840000	1.740000	2.010000
Test _{RGB+LBP}	LB	21.930000	2.080000	2.430000
Test _{RGB+LBP}	LE	20.450000	1.990000	1.960000
Test _{RGB+LBP}	J _B	61.040000	20.490000	6.530000
Test _{RGB+LBP}	J _K	146.650000	21.050000	8.010000
Test _{RGB+LBP}	J _M	129.730000	20.660000	6.300000
Test _{RGB+LBP}	J _D	130.890000	20.360000	6.900000

Tabela 3.11: Tempo de execução médio do método proposto em milissegundos.

A seguir serão apresentados os testes da função de agregação.

3.5.2 Função de Agregação

Os testes da função de agregação têm por objetivo verificar se os atributos escolhidos e as funções de agregação definidas descrevem fielmente as características das imagens. A figura da Tabela 3.12 mostra o grafo modelo construído com as ferramentas de desenho do *ImGraph* a partir de uma análise visual para identificar quais as regiões da imagem e seus relacionamentos representariam os vértices e arestas do grafo modelo. A imagem foi então, dividida de acordo com o tipo de vegetação local. A correspondência entre a imagem modelo e a imagem a ser reconhecida no modelo também foi obtida através dessa análise visual.

É possível perceber na figura da Tabela 3.12 o funcionamento das ferramentas de desenho do *ImGraph*. Nota-se que todas as regiões foram delimitadas utilizando-se a ferramenta de seleção poligonal. Os vértices serão denotados por ser rótulos e as arestas do modelo serão denotadas pela sequência de números mostrados na Tabela 3.12. Os grafos resultantes da segmentação automática são apresentados nas Tabelas 3.13 e 3.14.

0	(0, 1)
1	(0, 2)
2	(0, 5)
3	(6, 16)
4	(11, 16)
5	(11, 18)
6	(6, 11)
7	(7, 11)
8	(9, 11)
9	(10, 11)
10	(11, 12)
11	(12, 13)
12	(12, 17)
13	(10, 12)
14	(9, 10)
15	(7, 9)
16	(9, 14)
17	(10, 14)
18	(5, 8)
19	(8, 13)
20	(7, 8)
21	(8, 9)
22	(6, 7)
23	(5, 7)
24	(5, 13)
25	(5, 14)
26	(2, 5)
27	(2, 4)
28	(1, 4)
29	(3, 4)
30	(4, 17)
31	(2, 15)
32	(14, 15)
33	(14, 17)
34	(2, 17)
35	(0, 1)
36	(1, 2)
37	(5, 6)
38	(12, 18)
39	(13, 17)
40	(13, 14)
41	(10, 13)
42	(9, 13)



Tabela 3.12: À esquerda, os números que denotam as arestas do modelo. À direita, a imagem modelo.

0	1
1	4
2	0
3	2
4	5
5	5
6	4
7	5
8	8
9	13
10	5
11	16
12	6
13	7
14	7
15	10
16	11
17	9
18	18
19	12
20	2
21	4
22	14
23	15
24	15
25	3
26	17
27	14
28	12
29	17

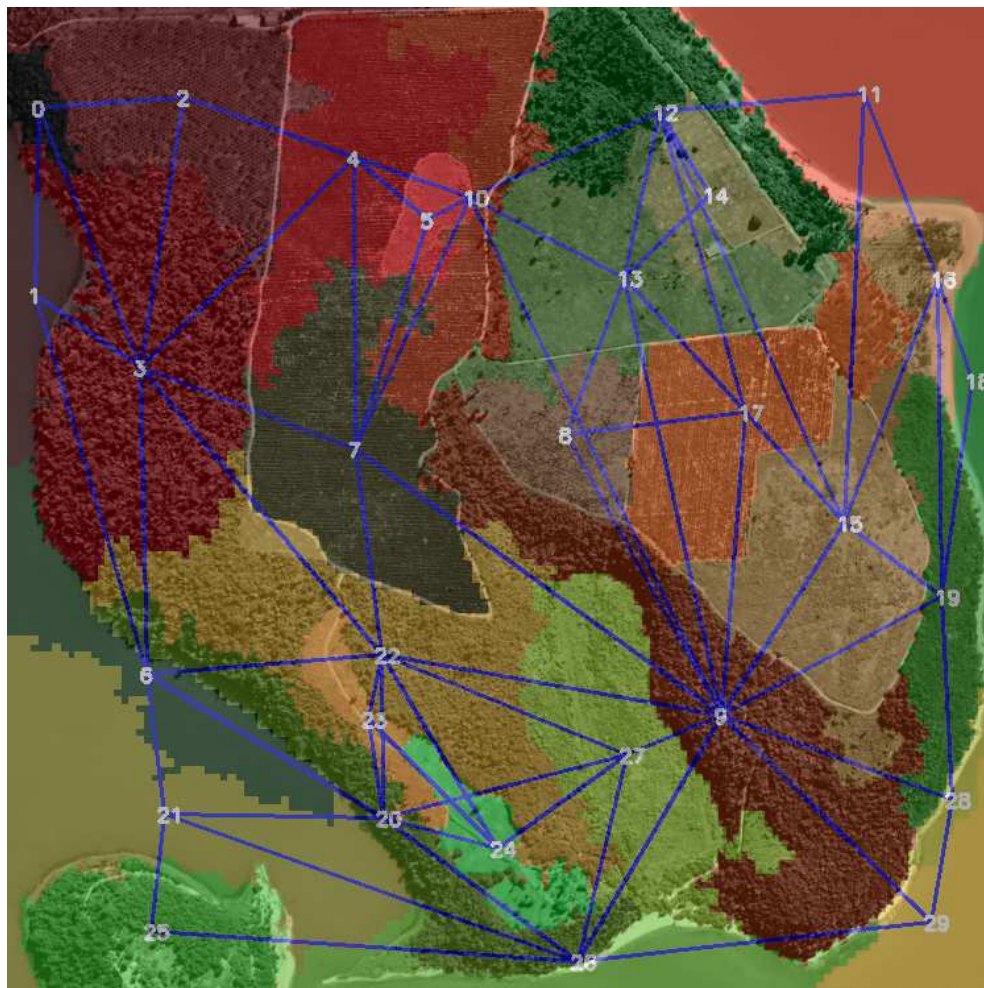


Tabela 3.13: À esquerda, a correspondência entre a imagem modelo e a imagem alvo. À direita, a imagem segmentada utilizando-se a 1ª componente do PCA, com os parâmetros $ts = tm = 1.15$, $ms = 4$, $F = N$, $D = L_J$.

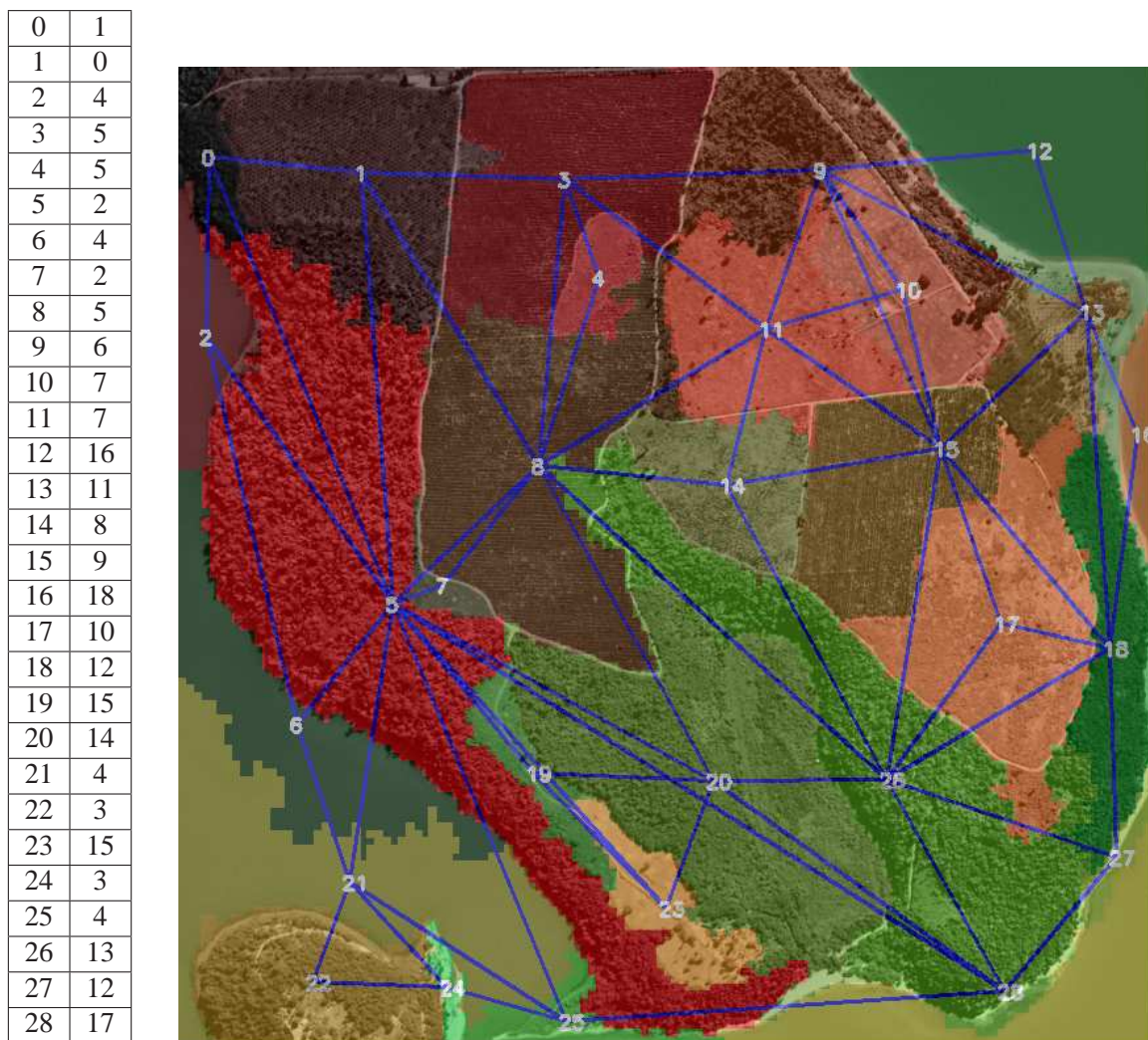


Tabela 3.14: À esquerda, correspondência entre a imagem modelo e a imagem alvo. À direita, imagem segmentada utilizando-se a 1ª componente do PCA combinada com o descritor LBP, com os parâmetros $ts = tm = 1.15$, $ms = 4$, $F = N$, $\alpha = 0.5$, $D = L_I$

As matrizes de similaridade foram impressas de forma tal que as células pintadas de amarelo representam a solução ideal encontrada pelo algoritmo, as células verdes representam a solução ideal não encontrada e as células vermelhas apresentam a solução encontrada que não correspondem a solução ideal. A seguir serão descritos dois experimentos realizados sobre essas imagens.

Experimento 1

Neste teste, foram considerados o grafo modelo e o grafo resultante da segmentação da imagem cuja dimensionalidade foi reduzida pela aplicação do algoritmo PCA, como mostrado na Tabela 3.13. O vetor solução, apresentado na mesma tabela, representa a correspondência das regiões, estabelecida entre a figura da Tabela 3.12 e a figura da Tabela 3.13. Assim, temos

que os vértices da primeira coluna representam a imagem a ser reconhecida e correspondem aos vértices da imagem modelo, presentes na segunda coluna da tabela. As matrizes de similaridade auxiliam na determinação dessa solução. É importante salientar que a melhor correspondência depende da interpretação visual das imagens, variando de acordo com o observador. O vértice 28 da imagem segmentada, por exemplo, pode ser associada tanto ao vértice 12 quanto ao vértice 17.

Para cada teste, foram variados o fator de ponderação β da função de agregação e as medidas de distância utilizadas. A Tabela 3.15 apresenta o resultado obtido para a matriz de similaridade entre vértices. Entre arestas, como a dimensão da matriz obtida foi muito alta (71×43), a mesma não será exibida em sua completude nesta dissertação, sendo apenas comentada.

Neste experimento, foram obtidas 3 correspondências incorretas para os vértices 1, 4 e 5 do grafo que representa a imagem da Tabela 3.13. Para o vértice 1, o menor valor encontrado, que corresponde a possível correspondência entre o grafo alvo e o grafo modelo, ocorreu na coluna 1, enquanto a associação correta deveria apontar para o vértice 4 do modelo. Isso se deve à proximidade dos centros de massa dos vértices 1 e 1, fazendo com que a similaridade entre essas regiões seja mais alta. Para os vértices 4 e 5, entretanto, a similaridade entre os histogramas ponderou incorretamente a função de agregação, tendo por consequência associações incorretas para os dois vértices. Vale ressaltar que o vértice 22 ocupa duas regiões na imagem modelo, e sua correspondência é possível tanto para o vértice 2 quanto para o vértice 14 do mesmo. Entretanto, se ocorrer o casamento entre os vértices 22 e 2, haveria um vértice no modelo sem correspondência. Algoritmos como o GRASP [95] impedem que essa associação seja efetuada, nesses casos, realizando o casamento de forma que nenhum vértice do modelo fique sem um vértice correspondente no grafo alvo.

Para as arestas, algumas associações são apresentadas nas Tabelas 3.16, 3.17 e 3.18.

	0	1	2	3	4	5	6	7	8	9	10	11
1	0.000420	0.009720	0.004993	0.028560	0.043408	0.039770	0.029522	0.038990	0.022343	0.041353	0.026537	0.030270
	12	13	14	15	16	17	18	19	20	21	22	23
1	0.048180	0.023072	0.025744	0.024096	0.039640	0.050964	0.013700	0.037117	0.040445	0.020489	0.028690	0.012074
	24	25	26	27	28	29	30	31	32	33	34	35
1	0.020800	0.025127	0.011458	0.026213	0.007211	0.038527	0.044791	0.026111	0.038312	0.037679	0.025621	0.000420
	36	37	38	39	40	41	42	43				
1	0.006042	0.011889	0.021995	0.046560	0.042006	0.037098	0.041907	0.032533				

Tabela 3.16: Similaridade entre a aresta 1 (0,2) do grafo a ser reconhecido e as arestas do grafo modelo

	0	1	2	3	4	5	6	7	8	9	10	11
2	0.001357	0.004216	0.010968	0.044827	0.061950	0.064670	0.045621	0.054519	0.039058	0.059045	0.045587	0.047158
	12	13	14	15	16	17	18	19	20	21	22	23
2	0.024709	0.041886	0.041797	0.037568	0.022414	0.032312	0.022597	0.048787	0.051401	0.033149	0.041909	0.022050
	24	25	26	27	28	29	30	31	32	33	34	35
2	0.031228	0.011370	0.013877	0.016304	0.000342	0.031865	0.042576	0.017829	0.024991	0.028767	0.021273	0.001357
	36	37	38	39	40	41	42	43				
2	0.000208	0.022318	0.042026	0.025811	0.036514	0.021049	0.024007	0.055071				

Tabela 3.17: Similaridade entre a aresta 2 (0,3) do grafo a ser reconhecido e as arestas do grafo modelo

	0	1	2	3	4	5	6	7	8	9	10	11
5	0.002279	0.002266	0.007818	0.087043	0.119655	0.061692	0.087751	0.113632	0.115347	0.116991	0.125182	0.132538
	12	13	14	15	16	17	18	19	20	21	22	23
5	0.100944	0.128039	0.126742	0.121295	0.080221	0.080270	0.066905	0.092875	0.087347	0.086832	0.082980	0.066526
	24	25	26	27	28	29	30	31	32	33	34	35
5	0.075957	0.052554	0.034604	0.031599	0.006068	0.086611	0.249397	0.033492	0.082321	0.086976	0.037733	0.002279
	36	37	38	39	40	41	42	43				
5	0.005868	0.066870	0.061387	0.109601	0.026294	0.040125	0.055228	0.063532				

Tabela 3.18: Similaridade entre a aresta 5 (2,3) do grafo a ser reconhecido e as arestas do grafo modelo

Na Tabela 3.16, é possível observar que o menor valor equivale à coluna 0, significando que a aresta 1 do grafo alvo está relacionada com a aresta 0 do grafo modelo. Essa associação se mostra correta, como mostra a Figura 3.12 e a Tabela 3.13. Outras entradas da matriz de similaridade de arestas incluem a Tabela 3.17 e 3.18, que associam corretamente as arestas do grafo modelo e do grafo alvo.

Experimento 2

Neste teste, foram considerados o grafo modelo e o grafo resultante da segmentação da imagem cuja dimensionalidade foi reduzida pela aplicação do algoritmo PCA em combinação com o descritor de texturas LBP, como mostrado na Tabela 3.14. O vetor solução, apresentado na mesma tabela, representa a correspondência das regiões, estabelecida entre a figura da Tabela 3.12 e a figura da Tabela 3.14. Assim, temos que os vértices da primeira coluna representam a imagem a ser reconhecida e correspondem aos vértices da imagem modelo, presentes na segunda coluna da tabela.

Para cada teste, foram variados o fator de ponderação β da função de agregação, bem como a contribuição da textura no cálculo das distância probabilística utilizada na função de agrega-

ção. As medidas de distância também foram variadas. A Tabela 3.14 apresenta o resultado obtido para a matriz de similaridade entre vértices. Entre arestas, como a dimensão da matriz obtida foi muito alta (65×43), a mesma não será exibida em sua completude nesta dissertação, sendo apenas comentada.

Neste experimento, foram obtidas 4 correspondências incorretas para os vértices 2, 4, 15 e 24 do grafo que representa a imagem da Tabela 3.14. Para o vértice 2, o menor valor encontrado, que corresponde a possível correspondência entre o grafo alvo e o grafo modelo, ocorreu na coluna 1, enquanto a associação correta deveria apontar para o vértice 4 do modelo. Isso se deve à proximidade dos centros de massa dos vértices 2 e 1, fazendo com que a similaridade entre essas regiões seja mais alta. Para o vértice 4, entretanto, a similaridade entre os histogramas ponderou incorretamente a função de agregação, tendo por consequência associações incorretas para os dois vértices. O vértice 15 ocupa duas regiões na imagem modelo, e sua correspondência é possível tanto para o vértice 9 quanto para o vértice 11 do mesmo. E por último, o vértice 24 encontra-se em uma região de limiar, entre as regiões 3 e 14 do modelo, tornando equivocada sua classificação.

Para as arestas, os algumas associações são apresentadas nas Tabelas 3.20, 3.21 e 3.22.

	0	1	2	3	4	5	6	7	8	9	10	11
0	0.000327	0.008697	0.004153	0.089630	0.153831	0.087513	0.090230	0.131994	0.069634	0.152378	0.071330	0.073908
	12	13	14	15	16	17	18	19	20	21	22	23
0	0.090851	0.068930	0.071887	0.071172	0.085169	0.120726	0.044693	0.106095	0.133510	0.068745	0.090024	0.043609
	24	25	26	27	28	29	30	31	32	33	34	35
0	0.049174	0.055139	0.045580	0.071840	0.005598	0.129759	0.163149	0.071962	0.112100	0.112080	0.072011	0.000327
	36	37	38	39	40	41	42	43				
0	0.004689	0.043488	0.068142	0.116510	0.061275	0.042117	0.032477	0.028451				

Tabela 3.20: Similaridade entre a aresta 0 (0,1) do grafo a ser reconhecido e as arestas do grafo modelo

	0	1	2	3	4	5	6	7	8	9	10	11
2	0.001889	0.015712	0.019362	0.100668	0.166246	0.115828	0.100927	0.142208	0.080494	0.163864	0.083514	0.083990
	12	13	14	15	16	17	18	19	20	21	22	23
2	0.073266	0.080885	0.081874	0.079623	0.072123	0.106668	0.049836	0.112506	0.140135	0.076571	0.098804	0.049926
	24	25	26	27	28	29	30	31	32	33	34	35
2	0.054733	0.044828	0.038498	0.036078	0.000271	0.122051	0.157760	0.036794	0.100006	0.101872	0.038601	0.001889
	36	37	38	39	40	41	42	43				
2	0.000150	0.050271	0.081170	0.101371	0.060341	0.072107	0.054355	0.0692947				

Tabela 3.21: Similaridade entre a aresta 2 (0,5) do grafo a ser reconhecido e as arestas do grafo modelo

	0	1	2	3	4	5	6	7	8	9	10	11
4	0.004072	0.002768	0.005885	0.183224	0.249343	0.100487	0.183304	0.245562	0.223714	0.246913	0.225322	0.225461
	12	13	14	15	16	17	18	19	20	21	22	23
4	0.211834	0.222929	0.224835	0.222171	0.190763	0.191037	0.148308	0.184300	0.182103	0.181271	0.180797	0.148745
	24	25	26	27	28	29	30	31	32	33	34	35
4	0.153092	0.142679	0.038591	0.037028	0.014599	0.137286	0.395339	0.037865	0.188121	0.190426	0.039944	0.004072
	36	37	38	39	40	41	42	43				
4	0.014597	0.149231	0.099749	0.291463	0.044572	0.064912	0.054331	0.037769				

Tabela 3.22: Similaridade entre a aresta 4 (1,5) do grafo a ser reconhecido e as arestas do grafo modelo

Na Tabela 3.20, é possível observar que o menor valor equivale à coluna 0, significando que a aresta 0 do grafo alvo está relacionada com a aresta 0 do grafo modelo. Essa associação se mostra correta, como mostra a figura da Tabela 3.12 e a imagem da Tabela 3.14. Outras entradas da matriz de similaridade de arestas incluem a Tabela 3.21 e 3.22, que associam corretamente as arestas do grafo modelo e do grafo alvo.

Análise dos Resultados da Função de Agregação

Observou-se que a segmentação influencia diretamente no resultado das funções de agregação, assim como as características selecionadas. Outro fator determinante na qualidade da solução foi a delimitação correta do modelo, visto que grande parte das correspondências equivocadas ocorreram cujos centros de massa ficavam próximos demais de outras regiões. Isso sugere que, embora importante, o centro de massa pode ser melhor ponderado ao ser associado à distância entre os histogramas das regiões em avaliação. Além disso, analisando-se os resultados, concluiu-se que a influência dos vizinhos, embora necessite de um maior aprofundamento, melhorou os resultados obtidos, e o fator de ponderação β ideal se encontra no intervalo

[0.6,0.9]. Atribuindo uma influência maior aos vizinhos, obteve-se um resultado aquém em relação aos que consideraram apenas a similaridade vértice a vértice ou aresta a aresta. Isso se deve ao fato das duas imagens possuírem um número distinto de arestas, ou seja, vizinhos completamente distintos serão considerados na análise e sua provável correspondência será analisada, mesmo que ela não exista de fato. A medida de distância que obteve os melhores resultados foi a distância de Matusita, enquanto a distância de Kullback-Leibler obteve os piores resultados, seguindo o padrão encontrado na fase de segmentação. Assim, existe muito a ser pesquisado sobre a melhor escolha de parâmetros e sua influência na segmentação de imagens e sua análise.

4 *Conclusões e trabalhos futuros*

Este trabalho apresentou um método de segmentação de imagens monocromáticas e coloridas, utilizando o algoritmo *split-and-merge*, considerando características de cor e textura extraídas a partir da aplicação do descritor *Local Binary Pattern* para medir a similaridade entre as regiões. Diversos parâmetros foram analisados para aperfeiçoar o processo de segmentação. Observou-se que o tamanho mínimo da região utilizado na fase de divisão do algoritmo *split-and-merge* influenciou o resultado da segmentação e a escolha do limiar adequado mostrou-se um fator também muito importante para medir a similaridade entre as regiões, assim como o peso que indica a contribuição da textura. Este deve ser escolhido de forma que contenha todo o padrão da textura e não extrapole as características pertencentes à região em questão. A redução de dimensionalidade, ou seja, a diminuição do número de características selecionadas da imagem através da Análise de Componentes Principais, contribuiu significativamente para um melhor desempenho da segmentação em termos de memória utilizada, qualidade da solução, tempo e custo computacional. Observou-se também que as distâncias probabilísticas em sua forma discreta melhoraram o desempenho do método de segmentação em relação à distância euclidiana. Em compensação, distâncias probabilísticas em sua forma analítica retornaram uma imagem segmentada de forma bastante equivocada, não se adequando aos propósitos deste trabalho. Os experimentos demonstraram que a abordagem é bastante promissora, podendo ser aplicada a uma grande variedade de imagens. A funções de agregação, que devem converter os atributos em apenas um valor que indica a similaridade entre vértices e/ou arestas de dois grafos de atributos, acertaram, em média, 90% dos casamentos entre vértices e arestas. Entretanto, esse resultado depende da qualidade da segmentação e da construção do modelo.

A partir da investigação realizada, alguns trabalhos futuros podem ser delineados. Novos descritores devem ser pesquisados com o objetivo de caracterizar mais eficientemente a informação de textura, aumentando a precisão da segmentação. A análise multicritério da função de agregação também pode ser melhor estudada a fim de identificar novas formulações para o problema de comportar, em um único valor, a similaridade entre dois conjuntos de valores. Outro item a ser pesquisado são novos métodos para extrair e selecionar as características mais

relevantes de uma imagem, tendo em vista o bom resultado apresentado pelo PCA. Além disso, pode-se estudar os grafos obtidos pelo método em um ambiente de recuperação de imagens por conteúdo, ou mesmo em aplicações de geoprocessamento, adicionando informações georreferenciadas aos grafos, como latitude e longitude, área e perímetro das regiões. Através da utilização dos grafos para representar informações, pode-se utilizar diversos métodos existentes de comparação entre grafos. Esta vantagem permite a escolha de um método que seja mais eficiente e que atenda aos objetivos de uma aplicação específica, avaliando seu desempenho e técnica utilizada. Outros trabalhos futuros incluem adicionar outros algoritmos de segmentação ao ambiente, adicionar ferramentas para manipulação dos grafos como inserir, deletar e editar vértices e arestas e acrescentar ferramentas de desenho mais elaboradas para edição das imagens.

Referências Bibliográficas

- [1] GONZALEZ, R. C.; WOODS, R. E. *Digital Image Processing*. [S.l.]: Prentice Hall, 2008.
- [2] DEVIJVER, P. A.; KITTLER, J. *Pattern Recognition: A Statistical Approach*. London: Prentice/ Hall Int., 1982.
- [3] BASSEVILLE, M. Distance measures for signal processing and pattern recognition. *Signal Processing*, v. 18, n. 4, p. 349–369, 1989.
- [4] ZHOU, S. K.; CHELLAPPA, R. From sample similarity to ensemble similarity: Probabilistic distance measures in reproducing kernel hilbert space. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 28, n. 6, p. 917–929, 2006.
- [5] BUNKE, H.; MESSMER, B. T. Recent advances in graph matching. *International Journal of Pattern Recognition and Artificial Intelligence*, v. 11, n. 1, p. 169–203, 1997.
- [6] ISHITANI, Y. Flexible and robust model matching based on association graph for form image understanding. *Pattern Analysis and Applications*, v. 3, n. 2, p. 104–119, Junho 2000.
- [7] HONG, P.; HUANG, T. S. Spatial pattern discovery by learning a probabilistic parametric model from multiple attributed relational graphs. *International Workshop on Combinatorial Image Analysis*, v. 139, p. 113–135, Abril 2002.
- [8] GRACIANO, A. B. V.; CESAR-JR., R. M.; BLOCH, I. Graph-based object tracking using structural pattern recognition. In: *XX Brazilian Symposium on Computer Graphics and Image Processing*. Washington, DC, USA: IEEE Computer Society, 2007. p. 179–186.
- [9] VENTURA, V. de A. *Ambiente Gráfico para Representação de Imagens Através de Grafos*. 2007.
- [10] JOLLIFFE, I. *Principal Component Analysis*. New York: Springer, 1986. (Springer series in statistics).
- [11] CESAR, R.; BLOCH, I. First results on facial feature segmentation and recognition using graph homomorphisms. In: *VI Simpósio Ibero-Americano de Reconhecimento de Padrões SIARP*. [S.l.: s.n.], 2001. p. 95–99.
- [12] WONG, A. K. C.; YOU, M.; CHAN, S. C. An algorithm for graph optimal monomorphism. *IEEE Transactions on Systems, Man and Cybernetics*, v. 20, p. 628–636, 1990.
- [13] MAURO, C. de et al. Similarity learning for graph-based image representations. *Pattern Recognition Letters*, Elsevier Science Inc., New York, NY, USA, v. 24, n. 8, p. 1115–1122, 2003.

- [14] BIANCHINI, M. et al. Recursive neural networks for processing graphs with labelled edges: Theory and applications. *Neural Network*, Elsevier Science Ltd., Oxford, UK, UK, v. 18, n. 8, p. 1040–1050, 2005.
- [15] KAMP, Y.; HASLER, M. *Recursive Neural Networks for Associative Memory*. New York, NY, USA: John Wiley & Sons, Inc., 1990.
- [16] PETRAKIS, E. G.; FALOUTSOS, C.; LIN, K.-I. D. Imagemap: An image indexing method based on spatial similarity. *IEEE Transactions on Knowledge and Data Engineering*, IEEE Computer Society, Los Alamitos, CA, USA, v. 14, n. 5, p. 979–987, 2002.
- [17] OLIVEIRA, C. J. S. et al. Protótipo de um sistema de recuperação de imagens baseado na cor. In: *VIII Brazilian Symposium on Multimedia and Hypermedia Systems - SBMIDIA, Tools and Applications Workshop*. [S.l.: s.n.], 2002. p. 411–414.
- [18] P.S.SUHASINI; KRISHNA, K. S. R.; KRISHNA, I. M. Graph based segmentation in content based image retrieval. *Journal of Computer Science*, p. 699–705, 2008.
- [19] MADISETTI, V.; WILLIAMS, D. B. *The Digital Signal Processing Handbook*. [S.l.]: CRC PRESS, 1998.
- [20] FOSCHI, P. G. et al. Feature extraction for image mining. In: *Multimedia Information Systems*. [S.l.]: Arizona State University, 2002. p. 103–109.
- [21] VAJDA, P. et al. Graph-Based Approach for 3D Object Duplicate Detection. In: *10th International Workshop on Image Analysis for Multimedia Interactive Services*. [S.l.: s.n.], 2009.
- [22] TAKAHASHI, H.; NAKAJIMA, M. Region graph based text extraction from outdoor images. *International Conference on Information Technology and Applications*, IEEE Computer Society, Los Alamitos, CA, USA, v. 1, p. 680–685, 2005.
- [23] KORTING, T. S. et al. A new graph-based approach for urban image segmentation. In: *Proceedings*. [S.l.]: UNION Agency - Science Press, 2008.
- [24] BISHOP, C. M. *Neural Networks for Pattern Recognition*. 1. ed. [S.l.]: Oxford University Press, USA, 1996.
- [25] MATERKA, A.; STRZELECKI, M. *Texture Analysis Methods - A Review*. Lodz, Poland, 1998.
- [26] PAPATHOMAS, T.; KASHI, R.; GOREA, A. A human vision-based computational model for chromatic texture segregation. *IEEE Transactions on Systems, Man, and Cybernetics*, v. 27, n. 3, p. 428–440, Junho 1997.
- [27] POIRSON, A. B.; WANDELL, B. A.; W, B. A. Pattern-color separable pathways predict sensitivity to simple colored patterns. *Vision Research*, v. 36, p. 515–526, 1996.
- [28] DUBUISSON-JOLLY; GUPTA, A. Color and texture fusion: Application to aerial image segmentation and gis updating. *Image and Vision Computing*, v. 18, n. 10, p. 823–832, Julho 2000.

- [29] S.WANG; A.WANG. Segmentation of high-resolution satellite imagery based on feature combination. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences. ISPRS Congress Beijing*, v. 37, p. 1223–1229, 2008.
- [30] OJALA, T.; PIETIKÄINEN, M.; HARWOOD, D. A comparative study of texture measures with classification based on feature distribution. *Pattern Recognition*, v. 29, n. 1, p. 51–59, 1996.
- [31] OJALA, T.; PIETIKÄINEN, M. Unsupervised texture segmentation using feature distributions. *Pattern Recognition*, v. 32, n. 3, p. 477–486, 1999.
- [32] OJALA, T.; PIETIKÄINEN, M.; MÄENPÄÄ, T. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 24, n. 7, p. 971–987, 2002.
- [33] LEHMANN, T. M. et al. Content-based image retrieval in medical applications. *Methods Inf Med*, Department of Medical Informatics, Aachen University of Technology (RWTH), Aachen, Germany, v. 43, n. 4, p. 354–361, 2004.
- [34] LEE, K.-M.; MEER, P.; PARK, R.-H. Robust adaptive segmentation of range images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, IEEE Computer Society, Los Alamitos, CA, USA, v. 20, n. 2, p. 200–205, 1998.
- [35] GOLDBERG, D. E. *Genetic Algorithms in Search, Optimization and Machine Learning*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1989.
- [36] BRUMBY, S. P. et al. Investigation of image feature extraction by a genetic algorithm. In: . [S.l.]: SPIE, 1999. v. 3812, n. 1, p. 24–31.
- [37] SCHWARTZ, W. R.; PEDRINI, H. Método para classificação de imagens baseada em matrizes de coocorrência utilizando características de textura. In: *III Colóquio Brasileiro de Ciências Geodésicas*. Curitiba-PR, Brasil: [s.n.], 2003. p. 1–11.
- [38] ISHITANI, Y. Model matching based on association graph for form image understanding. *Third International Conference on Document Analysis and Recognition*, v. 1, p. 287, 1995.
- [39] OZER, B.; AKANSU, A. N.; WOLF, W. A graph based object description for information retrieval in digital image and video libraries. In: *IEEE Workshop on Content-Based Access of Image and Video Libraries*. Washington, DC, USA: IEEE Computer Society, 1999. p. 79.
- [40] POPE, A.; LOWE, D. Learning object recognition models from images. In: *Fourth International Conference on Computer Vision*. [S.l.: s.n.], 1993. p. 296–301.
- [41] NASRABADI, N.; LI, W.; CHOO, C. Object recognition by a hopfield neural network. *IEEE Transactions on Systems, Man and Cybernetics*, Third International Conference on Computer Vision, v. 21, p. 1523–1535, Dezembro 1991.
- [42] ZHANG, D.-Q.; CHANG, S.-F. Detecting image near-duplicate by stochastic attributed relational graph matching with learning. In: *12th annual ACM international conference on Multimedia*. New York, NY, USA: ACM, 2004. p. 877–884.

- [43] PETRAKIS, E. G. M.; FALOUTSOS, C. Similarity searching in medical image databases. *IEEE Transactions on Knowledge and Data Engineering*, IEEE Computer Society, v. 9, p. 435–447, Maio 1997.
- [44] BENGOETXEA, E. *Inexact Graph Matching Using Estimation of Distribution Algorithms*. Tese (Doutorado) — Ecole Nationale Supérieure des Télécommunications, Paris (France), Dezembro 2002.
- [45] ESHERA, M. A.; FU, K. S. An image understanding system using attributed symbolic representation and inexact graph matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, IEEE Computer Society, Washington, DC, USA, v. 8, n. 5, p. 604–618, 1986.
- [46] BENGOETXEA, E. et al. Inexact graph matching by means of estimation of distribution algorithms. *Pattern Recognition*, v. 35, p. 2867–2880, 2002.
- [47] RUSS, J. C. *The Image Processing Handbook*. 4. ed. [S.l.]: CRC Press, 2002.
- [48] SHAPIRO, L. G.; STOCKMAN, G. C. *Computer Vision*. [S.l.]: New Jersey, Prentice-Hall, 2001. 279-325 p.
- [49] RODRIGUES, F. A. *Localização e Reconhecimento de Placas de Sinalização Utilizando um Mecanismo de Atenção Visual e Redes Neurais Artificiais*. Dissertação (Mestrado) — Programa de Pós-graduação em Informática da Universidade Federal de Campina Grande, 2002.
- [50] PHAM, D. L.; XU, C.; PRINCE, J. L. A survey of current methods in medical image segmentation. *Annual Review of Biomedical Engineering*, v. 2, p. 315–337, 2000.
- [51] COSTA, L. da F.; JUNIOR, R. M. C. *Shape Analysis and Classification: Theory and Practice*. Boca Raton, FL, USA: CRC Press, Inc., 2000.
- [52] JAIN, A. K.; KARU, K. Learning texture discrimination masks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, IEEE Computer Society, Washington, DC, USA, v. 18, n. 2, p. 195–205, 1996.
- [53] ARIVAZHAGAN, S.; GANESAN, L. Texture classification using wavelet transform. *Pattern Recognition Letters*, Elsevier Science Inc., New York, NY, USA, v. 24, n. 9-10, p. 1513–1521, 2003.
- [54] AYALA, G.; DOMINGO, J. Spatial size distributions: Applications to shape and texture analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 23, n. 12, p. 1430–1442, 2001.
- [55] IDRISSE, M.; ACHEROY, M. Texture classification using Gabor filters. *Pattern Recognition Letters*, v. 23, n. 9, p. 1095–1102, 2002.
- [56] HUANG, Y. et al. Texture feature extraction for land-cover classification of remote sensing data in land consolidation district using semi-variogram analysis. *WSEAS Transactions on Computers*, World Scientific and Engineering Academy and Society (WSEAS), Stevens Point, Wisconsin, USA, v. 7, n. 7, p. 857–866, 2008.

- [57] SINGH, M.; SINGH, S. Spatial texture analysis: A comparative study. *International Conference on Pattern Recognition*, IEEE Computer Society, Los Alamitos, CA, USA, v. 1, p. 10676, 2002.
- [58] JULESZ, B. Textons, the elements of texture perception, and their interactions. *Nature*, v. 290, n. 5802, p. 91–97, Maio 1981.
- [59] BECK, J.; PRAZDNY, K.; ROSENFELD, A. A theory of textural segmentation. In: *Human and Machine Vision*. [S.l.]: Academic Press, New York, 1983. p. 1–38.
- [60] HARALICK, R. M.; SHANMUGAM, K.; DINSTEN, I. Textural features for image classification. *IEEE Transactions on Systems, Man and Cybernetics*, v. 3, n. 6, p. 610–621, 1973.
- [61] UNSER, M. Sum and difference histograms for texture classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, IEEE Computer Society, Washington, DC, USA, v. 8, n. 1, p. 118–125, 1986.
- [62] DEYOE, E. A.; ESSEN, D. C. V. Concurrent processing streams in monkey visual cortex. *Trends in Neurosciences*, v. 11, n. 5, p. 219–226, Maio 1988.
- [63] MOJSILOVIC, A. et al. Matching and retrieval based on the vocabulary and grammar of color patterns. *IEEE Transactions on Image Processing*, v. 9, n. 1, p. 38–54, 2000.
- [64] TAN, S.; KITTLER, J. Colour texture classification using features from colour histogram. In: *8th Scandinavian Conference on Image Analysis*. Tromso Norway, Scandinavian: [s.n.], 1993. p. 807–813.
- [65] BAYKUT, A. et al. Real-time defect inspection of textured surfaces. *Real-Time Imaging*, Academic Press Ltd., London, UK, UK, v. 6, n. 1, p. 17–27, 2000.
- [66] KUMAR, A.; PANG, G. K. H.; MEMBER, S. Defect detection in textured materials using Gabor filters. *IEEE Transactions on Industry Applications*, v. 38, p. 425–440, 2002.
- [67] BOUKOUVALAS, C. et al. Color grading of randomly textured ceramic tiles using color histograms. *IEEE Trans on Industrial Electronics*, v. 46, p. 219–226, 1999.
- [68] KAUPPINEN, H. *Development of a Color Machine Vision Method for Wood Surface Inspection*. Tese (Doutorado) — University of Oulu, 1999.
- [69] SCHWARTZ, W. R.; PEDRINI, H. Segmentação de imagens de terrenos baseada na associação de características de texturas com dependência espacial modelada por campo aleatório de markov. In: *XII Simpósio Brasileiro de Sensoriamento Remoto*. [S.l.: s.n.], 2005. p. 4311–4318.
- [70] OJALA, T. *Nonparametric Texture Analysis Using Spatial Operators, with Applications in Visual Inspection*. Tese (Doutorado) — University of Oulu, 1997. Department of Electrical Engineering, University of Oulu, Finland.
- [71] HE, D.; WANG, L. Texture unit, texture spectrum, and texture analysis. *IEEE Transactions on Geoscience and Remote Sensing*, v. 25, n. 4, p. 509–512, 1990.
- [72] PIETIKÄINEN, M.; OJALA, T.; XU, Z. Rotation-invariant texture classification using feature distributions. *Pattern Recognition*, v. 33, p. 43–52, 2000.

- [73] NADLER, M.; SMITH, E. P. *Pattern Recognition Engineering*. [S.l.]: John Wiley & Sons, 1993.
- [74] OOMMEN, B.; KASHYAP, R. A formal theory for optimal and information theoretic syntactic pattern recognition. *Pattern Recognition*, v. 31, p. 1159–1177, 1998.
- [75] PIETIKÄINEN, M. et al. View-based recognition of 3d-textured surfaces. In: *12th International Conference on Image Analysis and Processing*. Washington, DC, USA: IEEE Computer Society, 2003. p. 530.
- [76] SANTOS, D. *Seleção de Características: Abordagem via Redes Neurais Aplicada à Segmentação de Imagens*. Dissertação (Mestrado) — ICMC-USP, 2007.
- [77] WEBB, A. R. *Statistical Pattern Recognition*. 2. ed. [S.l.]: John Wiley & Sons, 2002.
- [78] MINGOTI, S. A. *Análise de Dados Através de Métodos de Estatística Multivariada*. [S.l.]: Editora UFMG, 2005.
- [79] CADIMA, J. F. Redução de dimensionalidade através duma análise em componentes principais: Um critério para o número de componentes principais a reter. *Revista de Estatística (INE)*, p. 37–49, 2001.
- [80] RICHARDS, J. A.; JIA, X. *Remote Sensing Digital Image Analysis An Introduction*. 4. ed. [S.l.]: Springer, 2006.
- [81] JOHNSON, R. A.; WICHERN, D. W. *Applied Multivariate Statistical Analysis*. 6. ed. [S.l.]: Prentice Hall, 2007.
- [82] MOIGNE, J. L. *Multi-Sensor Image Registration, Fusion and Dimension Reduction*. 2005. Disponível em: <<http://www.esto.nasa.gov/conferences>>.
- [83] BOTELHO, S. C. et al. C-nl pca: Extracting non-linear principal components of image datasets. *XII Simposio Brasileiro de Sensoriamento Remoto*, p. 3495–3502, Abril 2005.
- [84] THEODORIDIS, S.; KOUTROUMBAS, K. *Pattern Recognition*. 3. ed. Orlando, FL, USA: Academic Press, Inc., 2006.
- [85] CHA, S.-H. Taxonomy of nominal type histogram distance measures. In: *American Conference on Applied Mathematics*. Stevens Point, Wisconsin, USA: World Scientific and Engineering Academy and Society (WSEAS), 2008. p. 325–330.
- [86] GAUCH, H. G. *Multivariate Analysis in Community Ecology*. Cambridge, UK: Cambridge University Press, 1992.
- [87] CHENG, Y.-C.; CHEN, S.-Y. Image classification using color, texture and regions. *Image Vision Computing*, v. 21, n. 9, p. 759–776, 2003.
- [88] LIAPIS, S.; SIFAKIS, E.; TZIRITAS, G. Colour and texture segmentation using wavelet frame analysis, deterministic relaxation and fast marching algorithms. *Journal of Visual Communication and Image Representation*, v. 15, p. 1–26, 2004.
- [89] BARRETT, H. H.; MYERS, K. *Foundations of Image Science*. 3. ed. [S.l.]: John Wiley & Sons, 2004.

- [90] PETROU, M.; BOSDOGIANNI, P. *Image Processing: The Fundamentals*. New York, NY, USA: John Wiley & Sons, Inc., 2000. 335 p.
- [91] RAMOS, R. *Localização Industrial - Um Modelo Espacial para o Noroeste de Portugal*. Tese (Doutorado) — Universidade do Minho, 2000.
- [92] EASTMAN, J. *IDRISI for Windows: User's Guide. Version 2.0*. Worcester, MA, USA: Clark University, 1997.
- [93] WINTERFELDT, D.; EDWARDS, W. *Decision Analysis and Behavioural Research*. [S.l.]: Cambridge University Press, 1986.
- [94] MALCZEWSKI, J. *GIS and Multicriteria Decision Analysis*. [S.l.]: John Wiley & Sons, New York, 1999.
- [95] BOERES, M. C. *Heurísticas para reconhecimento de cenas por correspondência de grafos*. Tese (Doutorado) — Programa de Pós-Graduação de Engenharia da Universidade Federal do Rio de Janeiro (COPPE/UFRJ), 2002.

APÊNDICE A – Ambiente Gráfico para Representação de Imagens Através de Grafos

Em diversos trabalhos, grafos de atributos são utilizados para representar imagens [8, 23, 11, 46]. Entretanto, é difícil encontrar um programa que auxilie o pesquisador a criar uma representação para ser usada nessas aplicações. Nas abordagens que utilizam matrizes de similaridade, criar exemplos manualmente para algoritmos que analisam a similaridade entre imagens através de grafos, como o GRASP [95], é uma tarefa, por vezes, maçante e limitada.

Portanto, é importante que se tenha uma interface amigável que torne esta tarefa mais rápida, prática e com menor probabilidade de erro, facilitando a criação de exemplos para esses algoritmos e a posterior visualização dos resultados.

Assim, no trabalho realizado em [9] foi desenvolvido um ambiente, o *ImGraph*, que reúne as ferramentas necessárias para a criação desses exemplos em uma única interface gráfica, auxiliando o usuário a manipular imagens em um nível mais alto, sem precisar entender os processos de segmentação, tornando mais simples os passos para representação de imagens através de grafos de atributos e da criação de matrizes de similaridade entre vértices (resp. arestas) entre eles.

O ambiente foi desenvolvido utilizando-se C++ como linguagem de programação. O C++ é uma linguagem de programação de alto nível com facilidades para o uso em baixo nível, multiparadigma e de uso geral, que pode ser orientada a objeto. Outra característica muito importante é o fato de ela não estar sob o domínio (intelectual ou financeiro) de uma empresa, como acontece com Java (Sun), Visual Basic (Microsoft). Por fim, sua compatibilidade com “C” resulta em uma vasta base de códigos já desenvolvidos.

Para a construção da interface gráfica do ambiente, foi utilizado o GTK, um pacote de ferramentas multiplataforma utilizado para a criação de interfaces gráficas. Neste pacote estão presentes uma coleção de rotinas e sub-rotinas, originariamente desenvolvidas em C, que poderão ser utilizadas individualmente, ou em conjunto, para construir uma interface com o usuário.

Também foi utilizado o Glade, uma aplicação RAD (*Rapid Application Development*) que auxilia o desenvolvimento de interfaces gráficas que utilizam o GTK. A biblioteca “libglade” foi utilizada para montar a interface gráfica da aplicação em tempo de execução (*at runtime*) a partir de um arquivo descritivo desta interface (em formato XML), tornando possível modificar sua aparência sem a necessidade de compilar novamente o seu código fonte. Por ser um software livre, o Glade é distribuído sob a licença GPL (*General Public License*), e por sua praticidade e compatibilidade com as ferramentas utilizadas para a criação do ambiente tornou-se interessante para o desenvolvimento deste projeto.

Para a manipulação de grafos, utilizou-se o Graphviz, um pacote de ferramentas de código aberto (*open source*) que auxilia na visualização de grafos cuja estrutura esteja informada através de um arquivo texto. Este arquivo texto deverá ser criado obedecendo a “*DOT language*”, linguagem criada com o objetivo de ser uma maneira simples e intuitiva de descrever o conteúdo de grafos. Graphviz tem inúmeros recursos para confecção de diagramas, tais como: opções de cores, tipos de letras, estilos de linhas, atalhos de internet e diferentes formas de desenho definidas pelo programador. Na prática, grafos são geralmente gerados através de um arquivo externo, mas eles também poderão ser criados e editados manualmente.

Para manipulação e extração de características da imagem, foi utilizado o OpenCV (*Intel Open Source Computer Vision Library*), que consiste em uma biblioteca escrita em C e classes C++ com mais de 500 funções de geração e manipulação de imagens, que auxilia pesquisadores a desenvolverem aplicações de visão computacional, como o reconhecimento de face, de gestos, de formas ou objetos e atualmente, inclusive, o reconhecimento audiovisual para sistemas de segurança.

A seguir, serão descritas as funcionalidades do ambiente.

A.1 Funcionalidades

Foram identificadas as seguintes funcionalidades necessárias ao ambiente:

- **Abrir Imagem para Edição:** o usuário informa o arquivo que deseja abrir. A imagem será aberta em uma nova janela, para edição da imagem, a fim de selecionar regiões que irão representar vértices e definir a existência de arestas através de relações espaciais entre essas regiões.
- **Salvar Imagem:** responsável por salvar em arquivo a imagem modificada, resultante da edição.

- Salvar Imagem Como: responsável por salvar em arquivo a imagem modificada, resultante da edição, com a opção de escolher um novo nome ou o local a ser salva.
- Sair: responsável por finalizar e encerrar o programa.
- Criar Matriz de Similaridade Imagem-Imagem: responsável por criar as matrizes de similaridade entre vértices (resp. arestas) de dois grafos de atributos, tendo como base duas imagens: a imagem modelo e a imagem alvo.
- Criar Matriz de Similaridade Imagem-Texto: responsável por criar as matrizes de similaridade entre vértices (resp. arestas) de dois grafos de atributos, tendo como base a imagem alvo e arquivos de texto contendo dados relevantes sobre o modelo.
- Criar Matriz de Similaridade Texto-Texto: responsável por criar as matrizes de similaridade entre vértices (resp. arestas) de dois grafos de atributos, arquivos de texto contendo dados relevantes sobre o modelo e sobre a imagem alvo.
- Aplicar Segmentação: responsável por aplicar a segmentação automática na imagem escolhida, podendo exibir a imagem original, a imagem segmentada e o grafo que representa a imagem segmentada em janelas separadas se o usuário assim requisitar.
- Extrair Dados da Imagem: responsável por extrair o grafo de atributos da imagem, de acordo com a edição da imagem feita pelo usuário, podendo exibir o grafo extraído da imagem se o usuário assim requisitar.
- Exibir Solução: responsável por associar o grafo de atributos que representa a imagem modelo com o grafo de atributos que representa a imagem alvo, dado o vetor solução, que contém a correspondência entre a imagem modelo e a imagem a ser reconhecida.
- Abrir Imagem: responsável por exibir uma imagem gravada em arquivo em uma nova janela.
- Interpretar Dot: responsável por criar a visualização de um grafo a partir de um arquivo dot, podendo exibir a visualização do mesmo se o usuário assim requisitar.
- Criar Dot: responsável por criar o arquivo dot do grafo a partir de sua matriz de adjacências, podendo exibir a visualização do mesmo se o usuário assim requisitar.
- Ampliar: responsável por ampliar a imagem desejada, se esta não estiver aberta para edição.

- Reduzir: responsável por reduzir a imagem desejada, se esta não estiver aberta para edição.
- Restaurar: responsável por restaurar o tamanho da imagem desejada.
- Limpar: responsável por limpar a imagem editada, para que esta seja editada novamente.
- Desfazer: responsável por desfazer uma ação de edição.
- Refazer: responsável por refazer uma ação de edição.
- Definir Arestas: responsável por definir quais regiões possuem arestas que as conectam.
- Selecionar Região: responsável por selecionar uma região na imagem aberta para edição.
- Escolher Cor da seleção: responsável por selecionar a cor da seleção.
- Escolher Tamanho do Pincel de Seleção: responsável por definir o tamanho usado no pincel de seleção.
- Exibir Grafo: responsável por exibir o grafo gerado sobre a imagem aberta para edição.
- Exibir Seleções: responsável por exibir as seleções feitas pelo usuário sobre a imagem aberta para edição.

As próximas seções apresentarão os métodos utilizados para segmentação de imagens automática e manual para criação de imagens alvo e modelo, respectivamente, e as funções de agregação utilizadas para analisar os atributos vértice a vértice e aresta a aresta.

Algoritmo para Segmentação Automática da Imagem Alvo

O algoritmo de segmentação de imagens utilizado no ambiente foi o *split-and-merge*. Este algoritmo trabalha, primeiramente, dividindo as regiões em quatro partes, a começar pela imagem inteira, seguido das 4 partes resultantes dessa primeira divisão. Cada região será dividida novamente em outras 4 partes, e assim sucessivamente, até alcançar a região de menor unidade da imagem, o pixel. Esta fase é denominada *split*.

Depois da fase *split*, o algoritmo visita os vizinhos de conectividade-4 (ou seja, que se situam estritamente acima, abaixo, a esquerda e a direita) de cada região, verificando, de acordo com a tolerância fornecida pelo usuário, se é possível uni-las. A soma da diferença dos canais R, G e B das regiões analisadas deve ser menor do que a tolerância fornecida. E novamente, o algoritmo aplica esta operação sucessivamente, para todas as regiões, unindo-as ou não. Esta fase é denominada *merge* [1].

O algoritmo de segmentação foi alterado para que pudesse capturar informações da imagem segmentada, a fim de construir o grafo que a representa. Este processo é necessário para a posterior utilização dos atributos no processo de reconhecimento da imagem.

Assim, ao final da segmentação da imagem, foram adicionadas rotinas que capturam a cor de cada região (para a construção da matriz RGB), o centroide de cada região (que fornece também os atributos das arestas: comprimento e angulação) e os vizinhos de cada região (para a construção da matriz de adjacências).

Segmentação Manual da Imagem para Construção do Modelo

A imagem modelo, normalmente, é construída manualmente por necessitar de conhecimento prévio sobre o tipo de segmentação que se deseja considerar, como o atlas do cérebro humano, dividido de acordo com os conceitos estabelecidos na medicina.

Assim, a interface disponibiliza ferramentas de desenho que auxiliam o usuário na criação deste modelo. Atualmente, a ferramenta conta com ferramentas de seleção nas formas elípticas, retangulares e poligonais, implementadas através do OpenCV.

Função de Agregação

Segmentada a imagem e extraídos seus respectivos grafos de atributos, é necessário compará-los. Há vários critérios envolvidos no problema de comparar dois grafos. Para medir a similaridade, foi criada uma função de agregação. Uma função de agregação computa um único resultado para vários critérios dados como entrada.

Os critérios considerados na função de agregação desenvolvida para atributos dos vértices foram a média da cor da região e o centro de massa, e para atributos das arestas, foram considerados a angulação e o seu comprimento, medido através da distância euclidiana entre os dois centros de massa.

Todos os critérios foram normalizados, permitindo que valores de critérios não comparáveis entre si sejam normalizados para uma mesma escala, viabilizando a agregação entre eles. A maior parte dos processos de normalização utiliza o valor máximo e mínimo para a definição de uma escala. A forma mais simples é uma variação linear definida pela equação a seguir [92].

$$X_i = \frac{R_i - R_{min}}{R_{max} - R_{min}}$$

onde :

X_i : é o valor normalizado

R_i : é o valor a ser normalizado

R_{min} : é o valor mínimo para o critério

R_{max} : é o valor máximo para o critério

Assim, um conjunto de valores pode ser expresso (convertido) numa escala normalizada (por exemplo, entre zero e um), tornando-os comparáveis.

Depois que os critérios forem normalizados para uma escala de zero a um, é possível agregá-los de acordo com uma regra de decisão. Para analisar o quão semelhantes são os vértices (resp. arestas) do grafo modelo e do grafo alvo, foi considerado o seguinte cálculo:

$$D_i = 1 - |x_i^m - x_i^a|$$

onde:

D_i : é a diferença, ou seja, o valor que analisa quão distantes estão os valores normalizados dos vértices (resp. arestas) do grafo modelo e do grafo alvo.

x_i^m : é o valor normalizado do fator i do vértice (resp. arestas) do grafo modelo.

x_i^a : é o valor normalizado do fator i do vértice (resp. arestas) do grafo alvo.

Os valores analisados serão muito semelhantes se D_i estiver próximo de 1 e, respectivamente, mais discrepantes se D_i for próximo de 0.

Depois de se estabelecer a diferença entre dois valores normalizados, os resultados pertencentes ao mesmo critério, como cor, são novamente agregados, seguindo o seguinte cálculo:

$$C_i = \frac{\sum_{i=1}^k D_i}{k}$$

onde:

C_i : é o resultado da soma dos valores normalizados pertencentes ao mesmo critério, dividido pelo total de valores, ou seja, é o critério normalizado.

D_i : é a diferença dos valores normalizados

k : é o total da diferença de valores normalizados pertencentes ao mesmo critério.

Neste trabalho, os critérios foram agregados de acordo com a seguinte função:

$$S = \sum_{i=1}^n w_i c_i$$

onde:

S : valor final.

w_i : peso do critério i (com $i = 1, \dots, n$). Como os vértices (resp. arestas) possuem dois critérios a serem analisados, o peso fornecido pelo usuário será atribuído ao primeiro critério e seu complemento será atribuído ao segundo critério.

n : número de critérios.

c_i : critérios normalizados.

Os critérios considerados na análise dos vértices foram: a cor (em RGB) e o centro de cada região. Os critérios considerados na análise das arestas foram: a distância entre os centroides de cada região (comprimento das arestas) e a sua angulação (que foi considerada variando de 0 a 360 graus).

Esses critérios podem ser editados, de acordo com a preferência do usuário, podendo este considerar um ou mais fatores na função de agregação dos vértices (resp. arestas).

A próxima seção irá descrever a janela principal do ambiente ImGraph.

A.2 Protótipo Implementado

A Figura A.1 mostra a janela principal do ambiente, quando o mesmo é executado.



Figura A.1: Janela principal do ImGraph

A janela principal do ImGraph possui um menu e uma paleta de ferramentas. Tanto o menu quanto a paleta podem ser acessados através de um clique sobre cada um.

O menu do ImGraph possui as seguintes opções:

- Arquivo
 - Abrir para Edição

- Salvar
- Salvar Como
- Sair
- Editar
 - Aplicar Segmentação à Imagem
 - Criar Matrizes de Similaridade
 - * Imagem-Imagem
 - * Imagem-Texto
 - * Texto-Texto
 - Exibir Solução
 - Extrair Dados do Modelo
- Exibir
 - Modelo
 - Segmentada
 - Grafo
 - * Abrir Grafo
 - * Interpretar Dot
 - * Criar Grafo
- Ajuda
 - About

A barra de ferramentas conta com os seguintes botões, da esquerda para a direita, de acordo com a Figura A.1: Ampliar, Restaurar, Reduzir, Pincel, Cor, Limpar, Desfazer, Refazer, Exibir Grafo, Exibir Seleções, Selecionar Regiões Elípticas, Selecionar Regiões Retangulares, Selecionar Regiões Poligonais, Definir Arestas.