

Reconhecimento Semi-Automático e Vetorização de Regiões em Imagens de Sensoriamento Remoto

Este exemplar corresponde à redação final da
Dissertação devidamente corrigida e defendida
por Jefersson Alex dos Santos e aprovada pela
Banca Examinadora.

Campinas, 12 de março de 2009.



Ricardo da Silva Torres (Orientador)

Dissertação apresentada ao Instituto de Com-
putação, UNICAMP, como requisito parcial para
a obtenção do título de Mestre em Ciência da
Computação.

**FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DO IMECC DA UNICAMP
Bibliotecária: Maria Júlia Milani Rodrigues CRB8a / 2116**

Santos, Jefersson Alex dos
Sa59r Reconhecimento semi-automático e vetorização de regiões em imagens de sensoriamento remoto / Jefersson Alex dos Santos -- Campinas, [S.P. :s.n.], 2009.

Orientador : Ricardo da Silva Torres
Dissertação (mestrado) - Universidade Estadual de Campinas, Instituto de Computação.

1. Reconhecimento de padrões. 2. Programação genética (Computação). 3. Processamento de imagens. 4. Sensoriamento remoto. 5. Vetorização. I. Torres, Ricardo da Silva. II. Universidade Estadual de Campinas. Instituto de Computação. III. Título.

Título em inglês: Semi-automatic recognition and vectorization of regions in remote sensing images

Palavras-chave em inglês (Keywords): 1. Pattern recognition. 2. Genetic programming. 3. Image processing. 4. Remote sensing. 5. Vectorization.

Área de concentração: Recuperação de Informação / Processamento de Imagens

Titulação: Mestre em Ciência da Computação

Banca examinadora:

Prof. Dr. Ricardo da Silva Torres (IC-UNICAMP)
Prof. Dr. Hemerson Pistori (UCDB-MS)
Prof. Dr. Hélio Pedrini (IC-UNICAMP)

Data da defesa: 27/02/2009

Programa de pós-graduação: Mestrado em Ciência da Computação

TERMO DE APROVAÇÃO

Dissertação Defendida e Aprovada em 27 de fevereiro de 2009, pela Banca examinadora composta pelos Professores Doutores:

Hemerson Pistori

Prof. Dr. Hemerson Pistori

Centro de Ciências Exatas e da Terra / Universidade Católica Dom Bosco.

Hélio Pedrini

Prof. Dr. Hélio Pedrini

IC / UNICAMP.

Ricardo Torres

Prof. Dr. Ricardo da Silva Torres

IC / UNICAMP.

Reconhecimento Semi-Automático e Vetorização de Regiões em Imagens de Sensoriamento Remoto

Jefersson Alex dos Santos¹

12 de março de 2009

Banca Examinadora:

- Ricardo da Silva Torres (Orientador)
- Hemerson Pistori
Universidade Católica Dom Bosco
- Hélio Pedrini
Instituto de Computação - Unicamp
- Jansle Vieira Rocha (Suplente)
Faculdade de Engenharia Agrícola - Unicamp
- Neucimar Jerônimo Leite (Suplente)
Instituto de Computação - Unicamp

¹Suporte financeiro de: CAPES, 03/2007 à 08/2007 e Fapesp, 09/2007 à 03/2009.

Resumo

O uso de imagens de sensoriamento remoto (ISRs) como fonte de informação em aplicações voltadas para o agronegócio é bastante comum. Nessas aplicações, saber como é a ocupação espacial é fundamental. Entretanto, reconhecer e diferenciar regiões de culturas agrícolas em ISRs ainda não é uma tarefa trivial. Embora existam métodos automáticos propostos para isso, os usuários preferem muitas vezes fazer o reconhecimento manualmente. Isso acontece porque tais métodos normalmente são feitos para resolver problemas específicos, ou quando são de propósito geral, não produzem resultados satisfatórios fazendo com que, invariavelmente, o usuário tenha que revisar os resultados manualmente. A pesquisa realizada objetivou a especificação e implementação parcial de um sistema para o reconhecimento semi-automático e vetorização de regiões em imagens de sensoriamento remoto. Para isso, foi usada uma estratégia interativa, chamada realimentação de relevância, que se baseia no fato de o sistema de classificação poder aprender quais são as regiões de interesse utilizando indicações de relevância feitas pelo usuário do sistema ao longo de iterações. A idéia é utilizar *descritores de imagens* para codificar informações espectrais e de textura de partições das imagens e utilizar realimentação de relevância com Programação Genética (PG) para combinar as características dos descritores. PG é uma técnica de aprendizado de máquina baseada na teoria da evolução. As principais contribuições deste trabalho são: estudo comparativo de técnicas de vetorização de imagens; adaptação do modelo de recuperação de imagens por conteúdo proposto recentemente para realização de realimentação de relevância usando regiões de imagem; adaptação do modelo de realimentação de relevância para o reconhecimento de regiões em ISRs; implementação parcial de um sistema de reconhecimento semi-automático e vetorização de regiões em ISRs; proposta de metodologia de validação do sistema desenvolvido.

Abstract

The use of remote sensing images as a source of information in agrobusiness applications is very common. In these applications, it is fundamental to know how the space occupation is. However, the identification and recognition of crop regions in remote sensing images are not trivial tasks yet. Although there are automatic methods proposed to that, users prefer sometimes to identify regions manually. That happens because these methods are usually developed to solve specific problems, or, when they have a general purpose, they do not yield satisfying results. This work presents a semi-automatic method to vectorize regions from remote sensing images using relevance feedback based on genetic programming (GP). Relevance feedback is a technique used in content-based image retrieval (CBIR). Its objective is to aggregate user preferences to the search process. The proposed solution consists in using image descriptors to encode texture and spectral features from the images, applying relevance feedback based on GP to combine these features with information obtained from the users interactions and, finally, segment the image. Finally, segmented image (raster) is converted into a vector representation. The main contributions of this work are: comparative study of image vectorization techniques; extension of a recently proposed relevance feedback approach for dealing with image regions; extension of the relevance feedback model for region recognition in remote sensing images; partial implementation of the semi-automatic and vectorization system of remote sensing images regions; proposal a validation methodology.

Agradecimentos

Agradeço...

À minha mãe, Maria José, por todo o apoio, dedicação e sacrifício empregados para que eu tivesse condições de realizar meus objetivos.

À minha namorada, Flávia, pelo carinho, compreensão e por estar tão presente no meu dia-a-dia, sempre me ouvindo e compartilhando todos os desafios e conquistas desse período da minha vida.

Ao meu orientador, Prof. Ricardo, pela motivação, paciência, compreensão e toda a confiança que depositou em mim para que esse trabalho fosse desenvolvido da melhor maneira.

À Prof.^a Claudia, pelo apoio, ensinamentos e a oportunidade de trabalhar em seu grupo de pesquisa.

Aos professores da UEMS, em especial o Prof. Faccenda e o Prof. Fabrício, pelo conhecimento, motivação e por toda a confiança que depositaram no meu potencial.

Ao pesquisador da Embrapa CPAO, Éder, pelas dicas e ensinamentos sobre a arte de “geoprocessar”.

Ao pesquisador do CEPAGRI, Prof. Rubens, pelos auxílios e o conhecimento sobre sensoriamento remoto, tão importantes no desenvolvimento desse trabalho.

Aos companheiros do LIS, pelos auxílios técnicos e por tornar mais agradável o ambiente de trabalho.

A todos os meus parentes e amigos do Mato Grosso do Sul que, apesar de longe, sempre me fazem sentir em casa quando retorno.

Este projeto foi parcialmente financiado pela FAPESP (processo 2007/53607-9), CAPES, CNPq (projetos WebMAPS e BioCore) e o Instituto Virtual FAPESP Microsoft (projeto eFarms).

Lista de Abreviações

AG Algoritmos Genéticos

BIC *Border/Interior pixel Classification*

CBIR *Content-Based Image Retrieval* – recuperação de imagens por conteúdo

CEPAGRI Centro de Processamento Agrícola, Unicamp

GDAL Geospatial Data Abstraction Library

HSV *Hue, Saturation and Value* — cor, saturação e valor

IFT *Image Forest Transforming* – Transformada Imagem-Floresta

ISR Imagem de Sensoriamento Remoto

KNN *K-Nearest Neighbor* – K vizinhos mais próximos

LAP *Local Aggregation Pattern*

LSP *Local Similarity Pattern*

MaxVer *Maximum Likelihood Classification*

MSBR Modelo de Similaridade Baseado em Região

NDVI *Normalized Difference Vegetation Index*

OPF *Optimum Path Forest*

PG Programação Genética

PIB Produto Interno Bruto

RFSP *Region and Feature Saliency Pattern*

RGB *Red, Green and Blue*

SBSR Simpósio Brasileiro de Sensoriamento Remoto

SEMISH Seminário Integrado de Software e Hardware

SIBGRAPI *Brazilian Symposium on Computer Graphics and Image Processing*

SIG Sistema de Informação Geográfica

SVN *Support Vector Machines* – máquinas de vetores de suporte

Sumário

Resumo	vii
Abstract	ix
Agradecimentos	xi
1 Introdução	1
2 Conceitos Básicos e Trabalhos Correlatos	5
2.1 Sensoriamento Remoto	5
2.1.1 Classificação de ISRs	7
2.1.2 Métodos de Vetorização Automáticos e Manuais	8
2.2 Programação Genética	10
2.2.1 Indivíduos	12
2.2.2 População	13
2.2.3 Adequação de Indivíduos	13
2.2.4 Seleção de Indivíduos	13
2.2.5 Operadores Genéticos	14
2.3 Recuperação de Imagens por Conteúdo e Realimentação de Relevância . .	15
2.3.1 Recuperação de Imagens por Conteúdo	15
2.3.2 Técnicas de Realimentação de Relevância em CBIR	17
2.4 Realimentação de Relevância por Programação Genética	19
2.4.1 Arcabouço PG^+	19
2.4.2 Arcabouço PG^\pm	22
3 Método Semi-automático de Vetorização	23
3.1 Partição e Extração de Características	25
3.2 Reconhecimento das Regiões de Interesse	26
3.2.1 Arcabouço $GPLSP$	27
3.2.2 Arcabouço $GPSR$	30

3.3	Segmentação da Imagem	31
3.3.1	Algoritmo <i>Watershed</i>	31
3.3.2	Seleção de sementes	34
3.4	Vetorização das Regiões	35
4	Validação do Método de Realimentação de Relevância para Regiões	37
4.1	Projeto do Experimento	37
4.1.1	Descritores	37
4.1.2	Técnica de Referência	39
4.1.3	Banco de dados de Imagem	39
4.1.4	Detalhes de Implementação	40
4.1.5	Medidas de Avaliação	41
4.1.6	Simulação do Usuário	41
4.2	Resultados Experimentais	42
5	Validação do Método de Vetorização Semi-automática	45
5.1	Projeto do Experimento	45
5.1.1	Imagens de Sensoriamento Remoto	45
5.1.2	Medida de Avaliação	47
5.2	Aspectos de Implementação	48
5.2.1	Módulo de Partição	49
5.2.2	Módulos de Reconhecimento de Padrões e Realimentação de Relevância	51
5.2.3	Módulo de Segmentação	51
5.2.4	Módulo de Vetorização	52
5.2.5	Simulação do Usuário	52
5.3	Metodologia de Validação	53
5.4	Resultados Experimentais	56
5.4.1	Imagem PASTAGEM	56
5.4.2	Imagem CAFÉ	62
5.4.3	Comparação com o método MaxVer	64
5.5	Resultados da Vetorização	67
6	Conclusões e Trabalhos Futuros	75
6.1	Conclusões	75
6.2	Extensões	76
6.2.1	Estrutura do Particionamento	76
6.2.2	Descritores de Imagens	77
6.2.3	Padrões de Consulta	77

6.2.4	Realimentação de Relevância	78
6.2.5	Segmentação e Classificação dos Pixels	79
6.2.6	Comparação com Outros Métodos	79
6.2.7	Outras Aplicações	79

Bibliografia		80
---------------------	--	-----------

Lista de Tabelas

4.1	Parâmetros de configuração da PG.	40
5.1	Parâmetros do módulo de partição para as imagens PASTAGEM e CAFÉ.	51
5.2	Testes realizados para a imagem PASTAGEM.	55
5.3	Testes realizados para a imagem CAFÉ.	55

Lista de Figuras

2.1	Um recorte de uma ISR extraída de uma região de café no município de Monte Santo (MG) e sua decomposição nos canais de cor RGB.	6
2.2	Uma mesma imagem representada no formato (a) <i>raster</i> e (b) matricial. . .	9
2.3	Exemplo de vetorização manual no SPRING [3].	11
2.4	Exemplo de indivíduo.	12
2.5	Exemplo de combinação em PG.	15
2.6	Exemplo de mutação em PG.	16
2.7	Exemplo de indivíduo no arcabouço PG^+	21
3.1	Etapas do processo de vetorização	24
3.2	Arquitetura do sistema de vetorização proposto.	25
3.3	Exemplo de estimativa do tamanho das sub-imagens. (a) A ISR da região de interesse. (b) O polígono que representa a região de interesse. (c) A imagem dividida em 16 sub-imagens. (d) A mesma ISR dividida em 64 sub-imagens.	26
3.4	Exemplo de divisão de imagem em sub-imagens.	27
3.5	Exemplo do cálculo de similaridade utilizando o modelo por sub-imagens. .	28
3.6	Indivíduo PG no arcabouço GP_{LSP}	29
3.7	Exemplo de execução do algoritmo <i>watershed</i> . (a) Uma imagem gradiente com um corte. (b) O corte visto como um vale e seus mínimos regionais. (c) A “enchente” no tempo $t1$. (d) A enchente no tempo $t2$. (e) Resultado final da execução do algoritmo. (f) Imagem segmentada.	32
3.8	Exemplo de execução do algoritmo watershed com definição de marcadores. (a) Uma imagem gradiente com um corte e marcadores (em vermelho e azul). (b) O corte visto como um vale. (c) A “enchente” no tempo $t1$. (d) A enchente no tempo $t2$. (e) Resultado final da execução do algoritmo. (f) Imagem segmentada.	33
3.9	Exemplos de cortes na lista de sub-imagens ordenadas. (a) Cortes de 10% relevantes e 20% relevantes. (b) Cortes de 20% relevantes e 30% não relevantes.	34

3.10	Exemplo de segmentação considerando as 10% sub-imagens mais relevantes e as 20% menos relevantes como marcadores.	35
3.11	Exemplo de segmentação considerando as 20% sub-imagens mais relevantes e as 30% menos relevantes como marcadores.	36
4.1	Curvas de precisão-revoação mostrando a eficácia do método GP_{LSP}	42
4.2	Curvas de precisão-revoação mostrando a eficácia dos métodos GP_{LSP} e LAP na base do Corel.	43
5.1	Imagem PASTAGEM.	46
5.2	Máscara que indica as regiões correspondentes à pastagem na imagem PASTAGEM.	48
5.3	Imagem CAFÉ.	49
5.4	Máscara que indica as regiões correspondentes à café na imagem CAFÉ. . .	50
5.5	<i>Máscaras de relevância</i> para as imagens (a) PASTAGEM e (b) CAFÉ com percentual de relevância igual a 50%.	53
5.6	Curvas Kappa x Iterações para os parâmetros do Teste 1 da Tabela 5.2. . .	57
5.7	Curvas Kappa x Iterações para os parâmetros do Teste 2 da Tabela 5.2. . .	58
5.8	Curvas Kappa x Iterações para os parâmetros do Teste 3 da Tabela 5.2. . .	58
5.9	Curvas Kappa x Iterações para os parâmetros do Teste 4 da Tabela 5.2. . .	59
5.10	Curvas Kappa x Iterações para os parâmetros do Teste 5 da Tabela 5.2. . .	59
5.11	Curvas Kappa x Iterações para os parâmetros do Teste 6 da Tabela 5.2. . .	60
5.12	Curvas Kappa x Iterações para os parâmetros do Teste 7 da Tabela 5.2. . .	60
5.13	Curvas Kappa x Iterações para os parâmetros do Teste 8 da Tabela 5.2. . .	61
5.14	Curvas Kappa x Iterações comparando os resultados para o operador morfológico e de Sobel com a seleção de sementes por sub-imagem inteira na imagem PASTAGEM.	62
5.15	Curvas Kappa x Iterações comparando os resultados para o operador morfológico e de Sobel com seleção de sementes por ponto central na imagem PASTAGEM.	63
5.16	Curvas Kappa x Iterações comparando as melhores curvas para os métodos de seleção de marcadores na imagem PASTAGEM.	64
5.17	Curvas Kappa x Iterações para os parâmetros do Teste 1 da Tabela 5.3. . .	65
5.18	Curvas Kappa x Iterações para os parâmetros do Teste 2 da Tabela 5.3. . .	65
5.19	Curvas Kappa x Iterações para os parâmetros do Teste 3 da Tabela 5.3. . .	66
5.20	Curvas Kappa x Iterações para os parâmetros do Teste 4 da Tabela 5.3. . .	66
5.21	Curvas Kappa x Iterações comparando as melhores curvas para cada um dos testes realizados para a imagem CAFÉ.	67

5.22	Curvas Kappa x Iterações comparando o resultado do método proposto com o MaxVer para a imagem PASTAGEM.	68
5.23	Curvas Kappa x Iterações comparando o resultado do método proposto com o MaxVer para a imagem CAFE.	68
5.24	Resultado da vetorização da imagem PASTAGEM pelo método proposto. .	69
5.25	Resultado da vetorização sobrepondo a máscara de pastagem.	70
5.26	Resultado da vetorização da imagem CAFÉ pelo método proposto.	72
5.27	Resultado da vetorização sobrepondo a máscara de café.	73
6.1	Imagem de sensoriamento remoto indicando amostras de regiões com café com diversos padrões espectrais e de textura.	78

Capítulo 1

Introdução

A agricultura brasileira tem obtido resultados eficientes e competitivos. Na última década, o agronegócio aumentou sua contribuição no PIB, representando cerca de 25% [9]. Nesse cenário, há uma enorme demanda por sistemas de informação que forneçam suporte ao monitoramento e ao planejamento de atividades agrícolas no Brasil. Nessas aplicações geralmente é necessário armazenar, analisar e manipular dados geográficos. Os Sistemas de Informação Geográfica (SIGs) são sistemas automatizados que possuem essa finalidade [7]. Em SIGs, o mundo real é frequentemente modelado segundo duas visões complementares: o modelo de campos e o modelo de objetos [7]. O primeiro é formalizado como uma função matemática cujo domínio é uma (abstração da) região geográfica e cujo contra-domínio é o conjunto de valores que o campo pode tomar. No modelo de objetos, o mundo é representado por uma superfície ocupada por objetos identificáveis com geometria e características próprias. Por esse motivo, as aplicações que utilizam essa representação necessitam que as informações possuam formato vetorial. Uma representação vetorial é gerada a partir de descrições de formas geométricas (pontos, retas, polígonos, etc).

Uma das abordagens mais utilizadas para monitoramento de safras é baseada no uso de Imagens de Sensoriamento Remoto (ISRs). O sensoriamento remoto, segundo Câmara et al. [7], pode ser definido como o conjunto de processos e técnicas utilizados com o objetivo de medir e registrar as propriedades eletromagnéticas da superfície terrestre por meio da detecção do fluxo de energia radiante refletida ou emitida pelos alvos naturais, objetos ou fenômenos, por meio da utilização de sensores sem que haja contato direto entre eles. As imagens de sensoriamento remoto (ISR) são resultado desse processo e facilitam significativamente trabalhos de mapeamento e proporcionam a base para a criação de sistemas que fornecem suporte para o processo de tomada de decisão baseado nas mudanças de ocupação do solo. Nesse contexto, conhecer as áreas plantadas é uma questão fundamental na maioria das aplicações.

ISRs são matriciais ou *raster*, ou seja, são formadas por uma matriz bidimensional

de pixels (um ou mais valores que descrevem a cor da imagem em determinada posição). Desse modo, para que as informações vetoriais possam ser obtidas de ISRs é necessário realizar a vetorização. Wenyin & Dori [57] definem vetorização, também conhecida como conversão de matriz para vetor, como o processo de reconhecimento de objetos vetoriais em imagens matriciais.

Existem diversas aplicações nas quais o uso de dados vetoriais é necessário. Na agricultura, por exemplo, frequentemente o cálculo de área de uma determinada região é requisitado. Esse processo é facilmente realizado em uma representação vetorial, uma vez que a região de interesse é representada como um polígono. Em aplicações que envolvem tráfego urbano, dados em formato vetorial são bastante utilizados. Eles possibilitam, por exemplo, que as ruas de uma cidade sejam modeladas como estruturas para a análise de caminhos, entre outras facilidades.

Embora ISRs sejam uma fonte de informação útil para a identificação de áreas plantadas, a tarefa de obtenção das regiões não é fácil. Além de problemas típicos da área de reconhecimento de padrões, identificar áreas plantadas em ISRs inclui outros problemas, como distorções causadas pelo relevo. Além disso, ISRs, diferentemente de imagens comuns, não codificam apenas informação do espectro visível para os humanos, contendo normalmente outras faixas espectrais, como o infra-vermelho. Sendo assim, a tarefa de identificação normalmente requer a aplicação de estratégias de classificação que exploram aspectos das ISRs relativos aos padrões espectrais e de textura identificados. O processo de identificação das regiões é chamado de classificação e pode ser feito automática ou manualmente.

As abordagens manuais são baseadas no uso de editores de imagens digitais integrados a um SIG (por exemplo, “SPRING” [8]). Com o uso desses editores, o usuário pode definir ou desenhar polígonos que representam regiões de interesse usando a ISR como “pano de fundo”.

Muitas vezes os usuários preferem identificar manualmente as regiões de interesse uma vez que os resultados dos métodos automáticos nem sempre são satisfatórios. Os métodos bem sucedidos de classificação de ISRs normalmente são desenvolvidos para um alvo ou tipo de dado específico [58]. Os de propósito geral, por outro lado, são muito sensíveis a ruídos. Um exemplo disso são distorções que podem ser encontradas em áreas montanhosas. A resposta espectral e os padrões de textura de uma mesma cultura agrícola em terreno plano e em terreno inclinado são diferentes. Além disso, uma cultura agrícola pode ser plantada de diferentes formas e esse fator, aliado às diversas fases da vida de uma planta, tende a dificultar a diferenciação entre regiões. Por esses motivos, em situações práticas, os resultados obtidos precisam ser revisados.

A pesquisa realizada neste trabalho objetivou a especificação e implementação parcial de um sistema para o reconhecimento semi-automático e vetorização de regiões em ima-

gens de sensoriamento remoto. Para isso, foi usada uma estratégia interativa, chamada realimentação de relevância [61], que se baseia no fato de o sistema de classificação poder aprender quais são as regiões de interesse utilizando indicações de relevância feitas pelo usuário do sistema ao longo de iterações.

Neste trabalho, um método de realimentação de relevância proposto recentemente para busca de imagens [16, 43] é usado para a classificação de partições da ISR. Esse método adapta uma abordagem baseada em programação genética para aprender as preferências do usuário em uma sessão de consulta. Programação genética (PG) [23] é uma técnica de aprendizado de máquina usada em várias aplicações, como mineração de dados, processamento de sinais e regressão [2, 15, 60]. Essa técnica é baseada na teoria da evolução. A evolução é a mudança de características hereditárias de uma população da geração atual para a próxima. Nesse processo, os melhores indivíduos (soluções do problema-alvo) de uma população tendem a se sobressair, constituindo assim a chamada *seleção natural*. Dessa forma, uma população tende a evoluir ao longo das gerações se adequando às variações do ambiente [23]. O uso de PG é motivado nesse trabalho pelo sucesso anterior do uso dessa técnica em recuperação de informação textual [1, 15] e tarefas de busca de imagens por conteúdo [12].

Dessa forma, as principais contribuições deste trabalho de mestrado são:

- Estudo comparativo de técnicas de vetorização de imagens *raster*;
- Adaptação do modelo de recuperação de imagens por conteúdo proposto por Ferreira [17] para realização de realimentação de relevância usando regiões de imagem [43];
- Adaptação do modelo de realimentação de relevância para o reconhecimento de regiões em ISRs [44].
- Implementação parcial de um sistema de reconhecimento semi-automático e vetorização de regiões em ISRs;
- Proposta de metodologia de validação do sistema desenvolvido [44].

Esta dissertação de mestrado é parte do projeto e-Farms [33], um projeto multidisciplinar que combina pesquisa em Ciência da Computação e Ciências Agrícolas e é financiado pela *FAPESP-Microsoft Research Virtual Institute*. O objetivo é abordar na teoria e na prática problemas envolvendo gerenciamento de dados agrícolas provenientes de sensores e comunicação de dados em rede sem fio de baixo custo em zonas rurais do Brasil.

Alguns resultados obtidos foram publicados. Um deles nos anais do *XXI Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI)* e um outro no *XXXIV Seminário Integrado de Software e Hardware (SEMISH)*. Além desses, um artigo

foi aceito para publicação nos anais do XIV Simpósio Brasileiro de Sensoriamento Remoto (SBSR). O artigo publicado no SIBGRAPI descreve resultados preliminares da pesquisa do mestrado, especificamente o arcabouço GP_{LSP} (realimentação de relevância baseado em programação genética para informação local das imagens). No artigo do SEMISH, esse trabalho contribuiu com uma seção sobre aplicação de realimentação no contexto de imagens de sensoriamento remoto [54]. O artigo aceito no SBSR descreve como a técnica de realimentação de relevância baseado em programação genética foi utilizada para o reconhecimento de regiões de interesse em ISRs [44].

Este texto é organizado da seguinte forma: o Capítulo 2 apresenta os conceitos básicos e os trabalhos correlatos a esta dissertação; o Capítulo 3 descreve o método proposto; o Capítulo 4 descreve aspectos de validação do modelo de realimentação de relevância para regiões de imagem; o Capítulo 4 apresenta os aspectos de validação e os detalhes da implementação do protótipo do sistema de reconhecimento e vetorização semi-automático de regiões de ISR; o Capítulo 6, finalmente, apresenta as conclusões e possíveis extensões.

Capítulo 2

Conceitos Básicos e Trabalhos Correlatos

Este capítulo apresenta os conceitos básicos relacionados, bem como os trabalhos correlatos. A seção 2.1 trata dos conceitos referentes ao sensoriamento remoto com ênfase na classificação de imagens de sensoriamento remoto. Na seção 2.2, são apresentados conceitos de programação genética. Na seção 2.3, são apresentados conceitos e técnicas de recuperação de imagens por conteúdo e realimentação de relevância. Por fim, a seção 2.4 aborda técnicas de realimentação de relevância baseadas em programação genética, recentemente propostas.

2.1 Sensoriamento Remoto

O sensoriamento remoto, segundo Câmara et al. [7], pode ser definido como o conjunto de processos e técnicas utilizados com o objetivo de medir e registrar as propriedades eletromagnéticas da superfície terrestre por meio da detecção do fluxo de energia radiante refletida ou emitida pelos alvos naturais, objetos ou fenômenos, por meio da utilização de sensores sem que haja contato direto entre eles.

Para o entendimento do processo de detecção, por meio de sensores, da energia radiante refletida ou emitida pelos corpos, é fundamental o conhecimento do comportamento espectral do alvo, quer seja ele uma floresta, um campo, uma pastagem ou uma cultura agrícola, e a sua interação com a radiação solar [34]. Após o processo de detecção, os dados adquiridos pelos sensores são convertidos em formato digital. Os dados digitalizados são um conjunto de pixels, que, distribuídos em linhas e colunas, formam uma imagem.

Definição 1 *Uma imagem multibanda [10] \hat{I} é um par (D_I, \vec{I}) , onde:*

- D_I é um conjunto finito de pontos em \mathbb{Z}^n (domínio da imagem) e que n refere-se à dimensão da imagem, e
- $\vec{I} : D_I \rightarrow \mathcal{D}'$ é uma função que atribui a cada pixel p em D_I um conjunto de valores escalares $\{I_1(p), I_2(p), \dots, I_k(p)\}$, associados com alguma propriedade física. O valor de k refere-se ao número de bandas. Por exemplo, $\mathcal{D}' = \mathbb{R}^3$ quando uma cor, no sistema RGB, é atribuída a um pixel.

Um exemplo de ISR, no sistema RGB, é apresentado na Figura 2.1 onde foi detectado o fluxo de energia radiante de uma região do município de Monte Santo (sul do Estado de Minas Gerais), região tradicional no cultivo de café. Nessa imagem, o canal de cor vermelho (R) corresponde a valores captados na faixa espectral do infra-vermelho próximo, o canal verde (G) ao infra-vermelho médio e o canal azul (B) ao espectro do visível do verde.

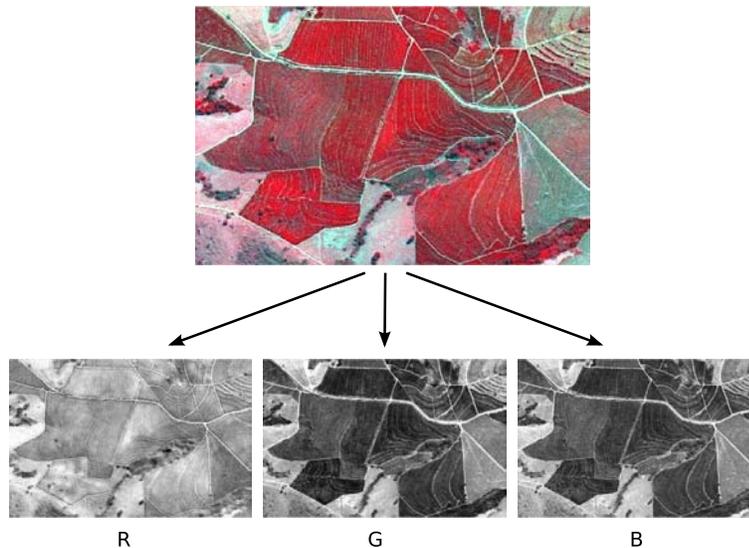


Figura 2.1: Um recorte de uma ISR extraída de uma região de café no município de Monte Santo (MG) e sua decomposição nos canais de cor RGB.

De acordo com Del Val Cura [6], as características mais relevantes são:

- ISRs são imagens multi-espectrais, isto é, de diferentes bandas do espectro de sinais. Podem ser classificadas em ativas, passivas e termiais, de acordo com suas características de detecção;

- Os diferentes valores multi-espectrais captados dependem fortemente da natureza física dos objetos da Terra observados pelos sensores, com alta probabilidade de introdução de ruído e variações segundo as condições climáticas no momento da coleta dos dados (luz solar, nuvens). Uma ISR normalmente precisa de transformações como filtragem e aumento de contrastes para a sua visualização.
- O caráter geográfico da informação implica a associação de coordenadas aos pixels. Como a superfície da Terra é esférica, a informação é captada e representada de forma planar o que introduz problemas de distorção. Esse aspecto gera a necessidade de correção geométrica, com introdução de possíveis erros.
- O conteúdo das ISRs é altamente complexo, diverso e não uniforme e as diferentes bandas das imagens não correspondem, em geral, ao espectro visível.

2.1.1 Classificação de ISRs

Desde que informações de satélites de imageamento se tornaram disponíveis para a comunidade civil na década de 1970, um grande esforço tem sido empregado para classificação de imagens com o objetivo de produzir mapas temáticos de alta qualidade e estabelecer inventários precisos referentes à utilização espacial da superfície da Terra [58]. Entretanto, as peculiaridades das ISRs associadas às dificuldades tradicionais de classificação de imagens tornam a classificação de ISRs uma tarefa ainda mais desafiadora.

Um estudo realizado em trabalhos publicados entre os anos de 1989 e 2003 analisou os resultados e implicações das pesquisas em classificação de ISRs [58]. De acordo com essa pesquisa, embora tenham sido propostas diversas abordagens nesse período, não houve melhora significativa nos resultados das classificações. Até então, a grande maioria dos métodos propostos realizavam classificação por pixels, utilizando, em geral, métodos estatísticos.

Talvez por isso mesmo, o algoritmo de classificação MaxVer (*Maximum Likelihood Classification*) [46] seja até hoje o mais popular. O MaxVer é considerado um algoritmo paramétrico que assume uma particular classe de distribuição estatística, normalmente a distribuição normal. A implementação do algoritmo MaxVer requer a computação da probabilidade de cada pixel pertencer às classes definidas. Cada pixel é então atribuído à classe de maior probabilidade calculada.

Outros trabalhos recentes vêm apresentando métodos estatísticos para a classificação de ISRs baseada em pixels. Em um desses trabalhos, Leite et al. [26] propuseram um classificador baseado em Modelos Ocultos de Markov para reconhecer plantações em suas diversas fases utilizando ISRs de resolução multi-temporal. As informações utilizadas

são matrizes de covariância e NDVI¹, além da assinatura espectral obtida pela análise temporal. Shiba et al. [32] propuseram uma abordagem baseada em aprendizagem por árvore de decisão utilizando apenas as características espectrais dos pixels.

Embora ainda existam trabalhos apresentando metodologias para a classificação baseada em informação de pixels, recentemente vem-se percebendo a importância de considerar informações baseadas em regiões das imagens de sensoriamento remoto e não somente em pixels individualmente. Gigandet et al. [19] propuseram um algoritmo para classificação de ISRs de alta resolução combinando segmentação não supervisionada com classificação supervisionada. Nesse método, as regiões foram classificadas utilizando distância de Mahalanobis e SVM (*Support Vector Machines* – máquinas de vetor de suporte). Lee et al. [24] propuseram uma metodologia de classificação baseada em regiões para imagens de alta resolução aplicando duas metodologias: MaxVer com as médias das regiões e MaxVer com Função Densidade de Probabilidade Gaussiana. Tanto no primeiro quanto no segundo trabalho, os resultados se mostraram melhores do que as abordagens baseadas em classificação de pixels. Yu et al. também propuseram uma metodologia para classificação de ISRs baseada em regiões [59]. O foco do trabalho, entretanto, era imagens de alta resolução obtidas de aviões. A segmentação das imagens foi feita por *evolução de redes fractais* e a classificação pelo método KNN (*K-Nearest Neighbor* – K vizinhos mais próximos) não paramétrico. Outro trabalho recente nessa área é o de Katartzis et al. [22]. Eles propuseram um método de classificação de regiões em imagens de sensoriamento remoto utilizando um Modelo Hierárquico de Markov. A segmentação é realizada pelo algoritmo de *watershed* utilizando mínimos regionais. Os experimentos foram feitos com imagens de alta resolução de aviões e imagens sintéticas (criadas para simulação).

2.1.2 Métodos de Vetorização Automáticos e Manuais

Quando uma imagem é representada digitalmente (escaneada, fotografada, etc) normalmente assume um formato matricial ou *raster*. Outra representação possível é a vetorial, que consiste de um conjunto de elementos independentes ou “objetos vetoriais” que possuem formato, posição e tamanho próprios [57].

A Figura 2.2 apresenta uma mesma imagem nos dois formatos de representação. A Figura 2.2 (a) demonstra uma representação matricial de 29 x 21 pixels para uma imagem. Na Figura 2.2 (b), a imagem é representada por formas geométricas e suas respectivas medidas. De acordo com a definição, imagens *raster* são compostas por pontos individuais (pixels). Quando aumentadas, as diferenças de cores entre os pixels ficam mais evidentes

¹NDVI (*Normalized Difference Vegetation Index*) é um indicador comumente utilizado em sensoriamento remoto para estimar a quantidade de vegetação em uma determinada região. É a razão entre a diferença dos valores espectrais de um pixel nas bandas correspondentes ao vermelho e ao infravermelho e à soma entre esses mesmos valores [27].

dando à imagem um aspecto “serrilhado”. Representações vetoriais mantêm a qualidade independentemente do seu tamanho. Isso acontece porque quando são aumentadas, os pontos que definem o objeto vetorial são recalculados pela função matemática que o define.

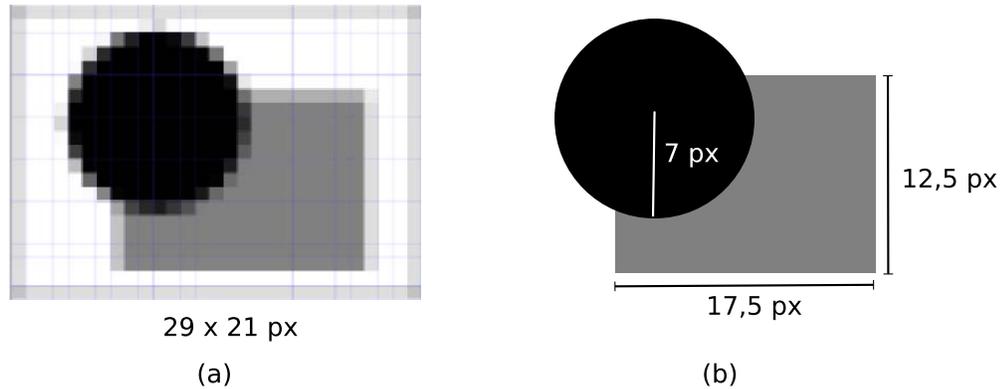


Figura 2.2: Uma mesma imagem representada no formato (a) *raster* e (b) matricial.

Definição 2 Vetorização [57] *é o processo de busca de objetos vetoriais em imagens matriciais com o objetivo de criar uma representação vetorial dessa imagem.*

No trabalho de Parker [38], a vetorização é feita automaticamente em imagens de contornos, ou seja, imagens que consistem em uma coleção de linhas que contrastam com o seu fundo. Ainda que o próprio trabalho justifique que frequentemente o usuário está interessado em alguma característica específica da imagem, algumas constatações são genéricas para qualquer problema de vetorização. Por exemplo, o processo de conversão de matriz para vetor pode ser dividido em duas etapas maiores: na primeira, as regiões de interesse devem ser identificadas e realçadas; o segundo passo consiste na identificação e na codificação dos vetores que compõem a imagem.

Uma revisão de diversos métodos de extração de vetores automaticamente em imagens de contorno é feita por Wenying & Dori [57]. Dentre os métodos, destacam-se: *Hough Transform-Based*, *Thinning Based*, *Contour-Based*, *Run Graph-Based*, *Mesh Pattern-Based* e *Sparse Pixel-Based*.

Lou et al. [31] propuseram uma abordagem para vetorização automática de ISRs baseada nas características dos dados matriciais. Para isso, o método foi dividido em três etapas: pré-processamento, processamento e pós-processamento. No pré-processamento, os pixels da imagem são classificados classe por classe. Depois eles são diferenciados como pixels de junção, nós ou pixels isolados. Na fase de processamento, os pixels do pré-processamento são utilizados para a geração dos polígonos. Esse processo é dividido

em quatro passos menores. O primeiro passo identifica o pixel de início do contorno. O segundo passo consiste em armazenar a localização dos pixels durante o traçado do contorno do polígono. Os contornos internos “holes” são também armazenados. Finalmente, os polígonos são criados com informação topológica baseando-se no traçado dos contornos externos e internos. O pós-processamento é uma otimização dos resultados da vetorização. Se a imagem original possui informação geográfica geométrica, os polígonos resultantes do processamento precisam ser revisados. Finalmente, o resultado é armazenado no banco.

Em geral, SIGs, como o SPRING [8], são equipados com ferramentas de extração de vetores de imagens raster pré-processadas, ou seja, imagens de contornos da região. A classificação das regiões, inclusive, é uma etapa normalmente contemplada com diversos algoritmos estatísticos nesses sistemas. Além disso, a vetorização manual também é possível por meio de editores que permitem demarcar polígonos e outras figuras geométricas nas ISRs. Um exemplo de edição vetorial da imagem é mostrado na Figura 2.3.

Uma outra ferramenta interessante é a GDAL (Geospatial Data Abstraction Library), uma biblioteca para tradução de formatos de dados geográficos [36]. Essa biblioteca possui uma ferramenta chamada “gdal_polygonize” que gera polígonos a partir de imagens raster. Esse aplicativo cria um polígono para todas as regiões com o mesmo valor de pixel em uma imagem.

Existem ainda várias outras ferramentas de vetorização criadas para outros fins, como vetorização de artes gráficas (logomarcas, desenhos artísticos, etc). Dentre esses, pode-se citar o Potrace [45]. O Potrace é uma ferramenta para vetorização de imagens, ou seja, transforma uma imagem matricial em um formato escalável (vetorial). A entrada é uma imagem (PBM, PGM, PPM ou BMP) e a saída é uma representação vetorial da imagem (PostScript, SVG ou bordas em PGM).

2.2 Programação Genética

Programação Genética (PG) é uma técnica de aprendizado de máquina que busca resolver problemas baseando-se nos princípios evolutivos da biologia.

A estrutura básica em PG é o *indivíduo*, que representa uma solução possível para um dado problema. Os indivíduos são programas que durante o processo evolutivo sofrem recombinações e perturbações sucessivas sendo, portanto, refinados. Ao final do processo, melhores soluções para o problema alvo são identificados. Os indivíduos são avaliados por meio de uma *função de adequação*. Assim, a programação genética pode ser entendida como uma busca no espaço de todas as possíveis soluções pelos indivíduos que melhor resolvem o problema. O Algoritmo 1 é o algoritmo evolutivo básico utilizado em PG.

De acordo com o Algoritmo 1, uma população inicial é criada (linha 1). Depois, sucessivos passos, limitados pelo número de gerações, repetem-se para evoluir esses indivíduos

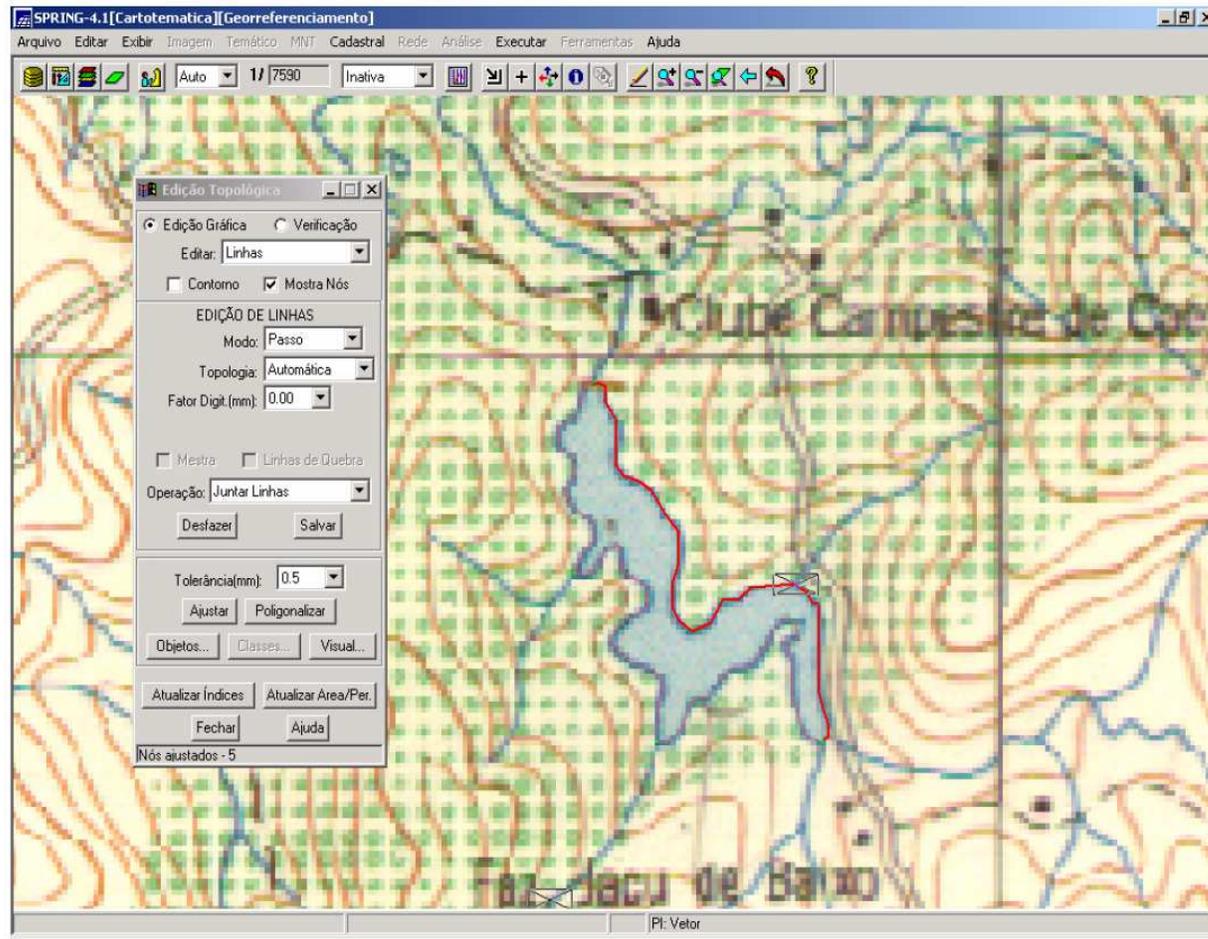


Figura 2.3: Exemplo de vetorização manual no SPRING [3].

(linha 2): o cálculo da *adequação* de cada indivíduo (linha 3); a *seleção* dos indivíduos (linha 4), baseada em sua adequação, para a geração de uma nova população, por meio de *operadores genéticos* (linha 5–7).

As próximas seções apresentam aspectos de PG em mais detalhes: a Seção 2.2.1 descreve a representação dos indivíduos; a Seção 2.2.2 apresenta o conceito de população e os diferentes métodos para a geração da população inicial de indivíduos; a Seção 2.2.3 descreve o processo de cálculo da adequação de um indivíduo; a Seção 2.2.4 apresenta diferentes métodos para a seleção de indivíduos; e a Seção 2.2.5 descreve os operadores genéticos mais comuns utilizados em PG.

Algoritmo 1 Algoritmo básico de evolução para programação genética.

- 1 Gere a **população inicial** de indivíduos
 - 2 **para** N gerações **faça**
 - 3 Calcule a **adequação** de cada indivíduo
 - 4 **Selecione** os indivíduos para operações genéticas
 - 5 Aplique **reprodução**
 - 6 Aplique **recombinação**
 - 7 Aplique **mutação**
 - 8 **fim para**
 - 9 **retorne** os melhores indivíduos
-

2.2.1 Indivíduos

Segundo Koza [23], em todo sistema de aprendizado pelo menos uma estrutura está sob adaptação. No caso de PG, as estruturas em adaptação são programas hierarquicamente estruturados, os *indivíduos*. O tamanho, a forma e o conteúdo desses programas podem ser dinamicamente alterados durante o processo.

A estrutura mais comumente adotada para representar um indivíduo em GP é a árvore. Nesse caso, o indivíduo é formado por dois tipos de nodos: as funções (nodos internos) e os terminais (folhas). Um exemplo pode ser observado na Figura 2.4.

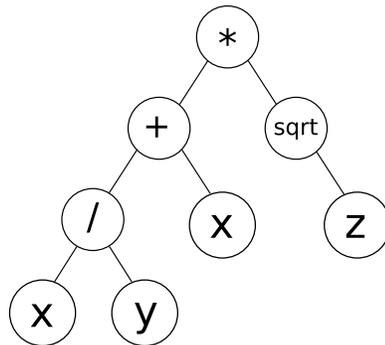


Figura 2.4: Exemplo de indivíduo.

Na Figura 2.4, os terminais são $\{x, y, z\}$, e seu conjunto de funções é $\{+, *, /, \sqrt{\cdot}\}$. Esse indivíduo representa a função

$$f(x, y, z) = \left(\frac{x}{y} + x\right) * \sqrt{z}.$$

2.2.2 População

Em programação genética, *população* é o conjunto de indivíduos que compõem uma geração. No início do processo evolutivo, é necessário gerar a população inicial. Dois tipos de inicialização são comuns: o crescimento (*grow*) e o cheio (*full*). Um aspecto desejável na criação da população inicial é a diversidade. Diversificar é importante para evitar problemas com ótimos locais, ou seja, aumenta a probabilidade de se encontrar uma boa solução no espaço de busca [23]. Além disso, é comum definir uma altura máxima para as árvores geradas.

O método do *crescimento* inicia a geração de um indivíduo escolhendo um terminal ou uma função para ser a raiz. O processo é repetido recursivamente para todos os ramos da raiz, se existirem. O crescimento é feito enquanto uma função for escolhida ou até que a altura máxima da árvore seja atingida.

O método *cheio* gera sempre uma *árvore cheia*. A estratégia é preencher todos os nodos da árvore com funções até que a mesma atinja a altura máxima. Quando a altura máxima é alcançada, os terminais são selecionados.

Para aumentar a diversidade da população inicial costuma-se utilizar a técnica “half-and-half”, uma forma híbrida de inicialização que une os dois métodos mencionados. Assim, 50% dos indivíduos são gerados pelo método *crescimento* e os outros 50% pelo método *cheio*.

2.2.3 Adequação de Indivíduos

A *adequação de um indivíduo* é uma medida quantitativa que determina a eficácia desse indivíduo para a resolução de um determinado problema [17]. O cálculo da adequação possibilita ao algoritmo de evolução escolher quais indivíduos serão selecionados para a próxima geração e para serem submetidos a operações genéticas, da mesma forma que permite selecionar os que serão excluídos da população.

O processo de adequação é normalmente calculado a partir de um conjunto de treinamento. O *conjunto de treinamento* é um conjunto de entradas de um problema para as quais as saídas são previamente conhecidas.

2.2.4 Seleção de Indivíduos

A cada iteração do processo evolutivo um conjunto de indivíduos deve ser selecionado para operações genéticas e servirem de base na próxima geração ou simplesmente serem excluídos da população. Os métodos de seleção [23] mais comuns são: o roleta, o baseado em torneio e o baseado em ordenação.

O *método da roleta* atribui a cada indivíduo uma probabilidade de ser selecionado. A escolha é feita então aleatoriamente de acordo com essas probabilidades. Dessa forma, é possível escolher os indivíduos mais bem adaptados, sem deixar de lado a diversidade dos menos adaptados.

A *seleção baseada em torneio* escolhe aleatoriamente um determinado número de indivíduos da população. O melhor entre os indivíduos é escolhido.

A *seleção baseada em ordenação* ordena os indivíduos da população pelos seus valores de adequação. De acordo com sua posição, cada indivíduo recebe uma probabilidade de ser selecionado.

2.2.5 Operadores Genéticos

Os *operadores genéticos* são os responsáveis pela geração de novos indivíduos de uma geração para a outra. Eles mantêm a diversidade da população e possibilitam a geração de melhores indivíduos durante o processo evolutivo. Os operadores genéticos mais comuns são: reprodução, combinação e mutação.

O operador de *reprodução* é a forma mais básica de um indivíduo sobreviver à seleção natural. Consiste em copiar o indivíduo selecionado e inserir a cópia na nova população.

O operador de *combinação* (*crossover*) introduz variabilidade na população produzindo novos indivíduos a partir de partes de outros dois, os pais. Dois indivíduos da população são selecionados para serem os “pais”. Partes de cada indivíduo são selecionadas e trocadas entre os “pais” para formar os indivíduos da próxima geração. Um exemplo de combinação pode ser observado na Figura 2.5.

Na Figura 2.5, dois indivíduos são selecionados (os “pais”) na geração n para a operação de combinação. Uma sub-árvore de cada indivíduo é selecionada. Depois é feita a troca das sub-árvores, gerando os dois indivíduos que irão fazer parte da geração $n + 1$. Por utilizarem um par de indivíduos, nesse operador o modo de geração é chamado sexuado. De forma similar, o operador de *reprodução* realiza uma operação assexuada.

O operador de *mutação* insere variabilidade na população criando perturbações na estrutura dos indivíduos. Uma sub-árvore é escolhida aleatoriamente em um dado indivíduo e trocada por uma nova sub-árvore gerada aleatoriamente. A Figura 2.6 é um exemplo do processo de mutação.

Na Figura 2.6, um indivíduo selecionado na geração n tem uma sub-árvore escolhida aleatoriamente. Então, uma outra sub-árvore é gerada e inserida nesse indivíduo no lugar da primeira gerando o indivíduo que fará parte da população da geração $n + 1$.

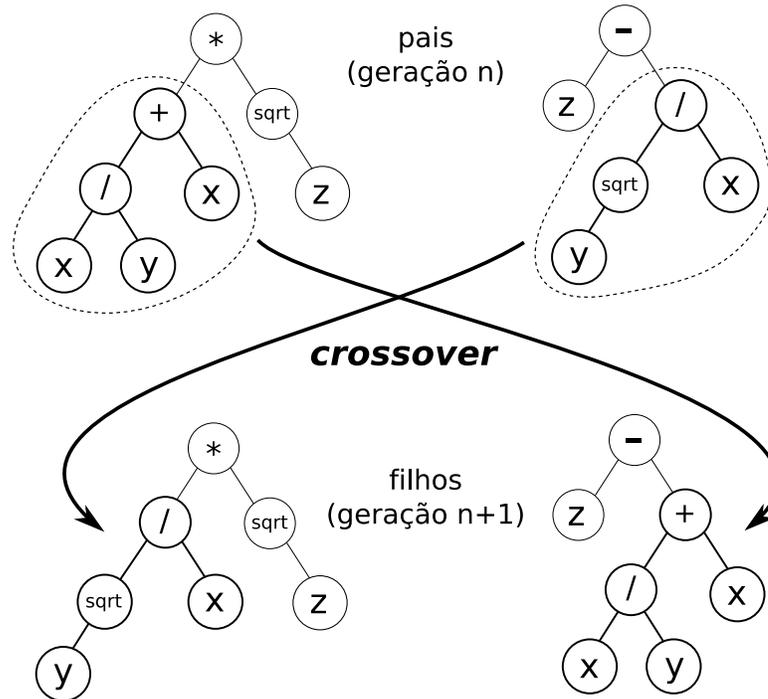


Figura 2.5: Exemplo de combinação em PG.

2.3 Recuperação de Imagens por Conteúdo e Realimentação de Relevância

Ao realizar o reconhecimento automático de uma região em uma imagem, independentemente do método utilizado, dificilmente se chegará a uma solução completamente satisfatória [28]. Como abordado na Seção 2.1.1, vários fatores dificultam a identificação de objetos em ISRs.

No processo de recuperação de imagens por conteúdo (CBIR) alguns problemas são semelhantes aos de classificação de ISR. A Seção 2.3.1 apresenta conceitos de recuperação de imagens por conteúdo e a Seção 2.3.2 apresenta trabalhos que envolvem a técnica de realimentação de relevância em sistema CBIR.

2.3.1 Recuperação de Imagens por Conteúdo

A recuperação de imagens baseada em conteúdo é centrada na noção de similaridade de imagens – dado um banco de dados com um grande número de imagens, o usuário deseja

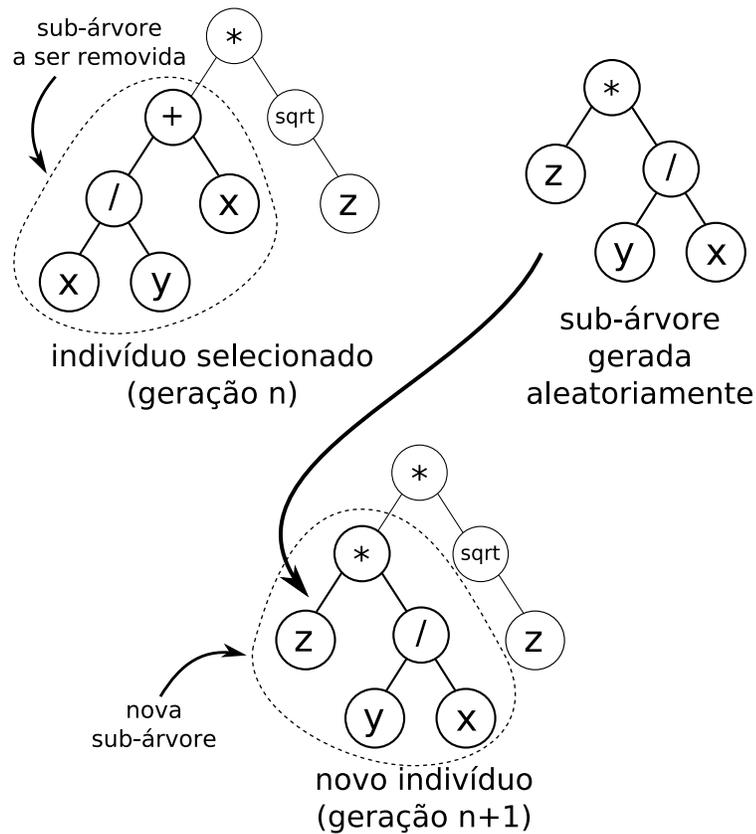


Figura 2.6: Exemplo de mutação em PG.

recuperar as imagens que são mais similares a um padrão de consulta (normalmente uma imagem de exemplo). Geralmente esses sistemas baseiam-se na extração de características visuais (cor, textura, forma e estrutura) e comparação das imagens por meio de descritores.

Definição 3 Um **v vetor de características** \vec{v}_f de uma imagem \hat{I} é um ponto no espaço \mathbb{R}^n : $\vec{v}_f = (v_1, v_2, \dots, v_n)$, onde n é a dimensão do vetor.

Exemplos de vetores de característica possíveis são um histograma de cor [52], uma curva fractal multi-escala [11], um conjunto de coeficientes de Fourier [39]. Estes descritores basicamente codificam propriedades das imagens, como cor, forma e textura. Note que diferentes tipos de vetores de características podem necessitar de funções de similaridade distintas.

Definição 4 Um *descriptor simples* [10] (ou simplesmente, *descriptor*) D é definido como uma tupla (ϵ_D, δ_D) , onde:

- $\epsilon_D : \hat{I} \rightarrow \mathbb{R}^n$ é uma função que extrai um vetor de características $\vec{v}_{\hat{I}}$ de uma imagem \hat{I} .
- $\delta_D : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$ é uma função de similaridade (por exemplo, baseada em uma medida de distância) que computa a similaridade entre duas imagens como uma função da distância entre seus vetores de características correspondentes.

Tanto na recuperação de imagens por conteúdo como no reconhecimento de regiões em ISRs podem-se utilizar as técnicas de realimentação de relevância [41]. No caso da CBIR, realimentação de relevância consiste em utilizar a interação humano-computador para refinar as consultas ao banco de imagens [10]. De modo geral, o processo de recuperação nessa abordagem consiste na indicação pelo usuário das imagens que são relevantes dentre aquelas retornadas em uma consulta. Feito isso, a consulta é reformulada e realizada novamente. Esse processo se repete até que o usuário esteja satisfeito. No caso do reconhecimento de regiões, esse processo pode ser aplicado de maneira semelhante. Porém, ao invés de várias imagens em uma consulta, o usuário poderia selecionar quais regiões são relevantes, podendo adicionar ou remover regiões buscadas em uma mesma imagem. O restante do processo é similar.

Diferentes abordagens são utilizadas para recuperação de imagens com realimentação de relevância. Em geral, a realimentação de relevância é utilizada em estratégias que permitem definir as funções de similaridade que combinam os descritores de imagem aplicados nas imagens de forma a encapsular as preferências do usuário. Isso significa que essa técnica pode ser útil na definição da função de similaridade $\delta_{\mathcal{D}}$ em um descriptor composto (Definição 5) de um sistema CBIR.

Definição 5 Um *descriptor composto* \hat{D} é definido como um par $(\mathcal{D}, \delta_{\mathcal{D}})$, onde:

- $\mathcal{D} = \{D_1, D_2, \dots, D_k\}$ é um conjunto de k descritores simples pré-definidos.
- $\delta_{\mathcal{D}}$ é um função de similaridade que combina os valores de similaridade obtidos de cada descriptor $D_i \in \mathcal{D}$, $i = 1, 2, \dots, k$.

2.3.2 Técnicas de Realimentação de Relevância em CBIR

Os trabalhos de Cox et al. [5] e Duan & Zhao [13] utilizaram a abordagem Bayesiana que estima uma distribuição de probabilidade sobre as imagens. A cada iteração, a distribuição de probabilidade é recalculada baseada na indicação do usuário do conjunto de resultados

relevantes. A regra de Bayes é utilizada para prever quais são as imagens mais relevantes para o usuário.

Na abordagem apresentada em [41], cada descritor utilizado recebe um peso. Este peso determina a relevância do descritor para uma determinada consulta. Os pesos são re-calculados a cada iteração com o usuário e a consulta é formulada novamente. Com isso, busca-se ressaltar as propriedades que o usuário deseja na consulta.

Diversas abordagens aplicam técnicas de inteligência artificial em suas implementações de realimentação de relevância. Nessas aplicações as consultas são ajustadas a cada iteração no intuito de encontrar soluções ótimas através de formulações heurísticas. A realimentação de relevância é implementada por Park & Jang com uma técnica baseada em redes neurais [42]. Muneesawang & Guan aplicam uma rede de função de base radial para implementar uma medida adaptativa baseado na indicação de imagens relevantes pelos usuários [35].

Máquina de Vetores de Suporte (SVM) é também uma técnica de aprendizado de máquina comumente usada em abordagens de realimentação de relevância. No trabalho proposto em [21], SVM é utilizada como método de aprendizado. Em [53], Tong et al. propuseram o uso de um método de SVM ativa de aprendizado para separar imagens relevantes de outras. Em cada interação, as imagens próximas do hiperplano de separação, as mais ambíguas, são mostradas para o usuário. No final desse processo, as mais distantes do hiperplano são mostradas.

Em [48] as similaridades de características regionais das imagens são utilizadas por Stejic et al. para recuperação de imagens. Eles propuseram um método baseado em algoritmos genéticos (AG) e um novo método *Local Similarity Pattern (LSP)*, para calcular a similaridade de imagens. AG e realimentação de relevância são utilizados para determinar as características que melhor descrevem cada região *LSP*. Em [50], Stejic et al. propuseram o *RFSP (Region and Feature Saliency Pattern)*. O *RFSP* é definido como uma estrutura similar ao *LSP*. Mas ao invés de utilizar AG para determinar a característica que melhor descreve cada região da imagem, no método *RFSP*, há um peso associado a cada região e o GA é aplicado para encontrar os melhores pesos para todas as características regionais. Uma outra técnica baseada em AG foi proposta em [49]: *Local Aggregation Pattern (LAP)*. No *LAP*, Stejic et al. utilizaram operadores de agregação matemática para combinar a similaridade das regiões. Nessa metodologia, a realimentação de relevância baseada em AG é utilizado para encontrar o melhor conjunto de operadores para a combinação das características das regiões.

Em [43], Santos et al. propuseram uma abordagem baseada em programação genética (PG) para combinar as características regionais das imagens, o *GP_{LSP}*. Nessa abordagem, os operadores utilizados para combinar as características são os indivíduos da PG, e, dessa forma, não são pré-definidos e sim obtidos a medida em que o usuário interage com o

sistema. O GP_{LSP} é apresentado em mais detalhes na Seção 3.2.1.

2.4 Realimentação de Relevância por Programação Genética

Utilizando realimentação de relevância baseada em programação genética, Ferreira et al. [16] propuseram dois arcabouços para recuperação de imagens por conteúdo. O primeiro arcabouço, chamado PG^+ , utiliza a informação provida pelas imagens indicadas pelo usuário como positivas no processo de aprendizado. O outro, PG^\pm , considera além das positivas também as imagens negativas para agregar as informações fornecidas durante as interações do usuário. A Seção 2.4.1 apresenta o arcabouço PG^+ enquanto o arcabouço PG^\pm é apresentado na Seção 2.4.2.

2.4.1 Arcabouço PG^+

O principal objetivo do algoritmo PG^+ é encontrar um conjunto de funções de combinação que melhor traduzam a percepção visual do usuário [17]. Então, procura-se redefinir as relações de similaridade entre as imagens, utilizando-se as funções de combinação encontradas, para que essas relações melhor se adequem ao conjunto de imagens que o usuário busca.

Seja L o número de imagens exibidas para o usuário em cada iteração. Seja o padrão de consulta $Q = \{q_1, q_2, \dots, q_M\}$, onde M é o número de elementos em Q . Q é formado pela imagem de consulta q_1 e todas as imagens rotuladas como relevante durante uma sessão de consulta [17].

O padrão de consulta utilizado no arcabouço PG^+ é composto não apenas pela imagem de consulta definida inicialmente pelo usuário, mas também por todas as imagens marcadas como relevante em todas as iterações. Desse modo, a abordagem é baseada em múltiplos pontos de consulta para o processo de recuperação. Segundo Ferreira [17], essa abordagem é mais robusta às imperfeições na definição do espaço de características, em relação à percepção do usuário. Com um único ponto de consulta dificilmente se conseguiria encapsular todas as preferências de consulta do usuário. A utilização de múltiplos pontos de consulta em conjunto com a redefinição da relação de similaridade proporcionada pelo aprendizado baseado em PG, possibilita uma melhor caracterização da percepção visual do usuário. O Algoritmo 2 apresenta o processo de realimentação utilizado pelo arcabouço GP^+ .

De acordo com o Algoritmo 2, no começo do processo de recuperação realizado pelo arcabouço PG^+ , o usuário define a imagem de consulta (linha 1). A partir dessa imagem, um conjunto inicial de imagens é selecionado para ser exibido para o usuário (linha 2).

Algoritmo 2 O processo de realimentação de relevância baseado em programação genética.

```

1 Definição da imagem de consulta  $q_1$ 
2 Exiba o conjunto inicial de imagens
3 enquanto o usuário não estiver satisfeito faça
4   Indicação do usuário
5   Atualize o padrão de consulta  $Q$ 
6   Utilize PG para encontrar os melhores indivíduos (funções de combinação de
   similaridades)
7   Ordene as imagens da base
8   Exiba as  $L$  imagens mais similares
9 fim enquanto

```

Utilizando o conjunto inicial, o processo de realimentação de relevância se inicia, com a indicação das imagens relevantes pelo usuário. São realizadas diversas iterações, que se repetem até que o usuário esteja satisfeito com o resultado (linha 3). Nessas iterações, são realizadas as seguintes etapas: indicação das imagens relevantes pelo usuário (linha 4); atualização do padrão de consulta (linha 5); aprendizado da percepção visual do usuário utilizando programação genética (linha 6); ordenação das imagens da base (linha 7); e exibição das imagens mais similares (linha 8).

A *seleção das imagens do conjunto inicial* exibido para o usuário é definida pela ordenação das imagens da base em relação a sua similaridade em relação à imagem de consulta. Esse processo é realizado em duas etapas. Inicialmente, cada descritor simples é utilizado para calcular a similaridade de cada imagem. Em seguida, os valores calculados são combinados pela sua média aritmética. Essa combinação utiliza todos os descritores empregados no processo de recuperação e atribui o mesmo grau de importância para cada um. Essa forma de combinação permite a definição do conjunto inicial de imagens retornadas para o usuário sem a necessidade de conhecimento prévio sobre o desempenho dos descritores. A partir da similaridade calculada em relação à imagem de consulta, as imagens da base são ordenadas e as L primeiras são exibidas. Assim, o usuário identifica nesse conjunto quais são as imagens relevantes. O próximo passo é inserir todas as imagens marcadas como relevantes no padrão de consulta Q .

O arcabouço PG^+ utiliza programação genética para combinar os descritores e codificar as necessidades do usuário. Um *indivíduo* na programação genética representa uma função cujo objetivo é combinar valores de similaridade definidos por diferentes descritores. Desse modo, cada nodo interno da árvore que representa o indivíduo é um operador aritmético enquanto as folhas são os valores de similaridade d_i de um descritor D_i entre uma imagem I_j e I_k . Um exemplo de indivíduo em PG^+ pode ser

visualizado na Figura 2.7. Esse indivíduo representa a função $f(d_{1I_j I_k}, d_{2I_j I_k}, d_{3I_j I_k}) = \left(\frac{d_{1I_j I_k} + d_{3I_j I_k}}{d_{2I_j I_k}}\right) - \sqrt{d_{2I_j I_k} * d_{3I_j I_k}}$.

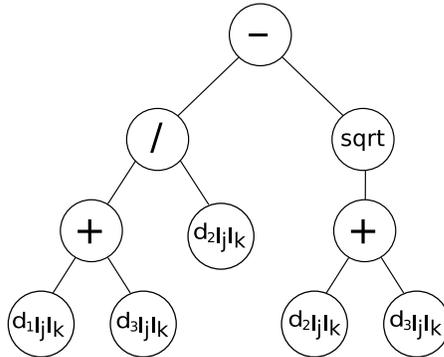


Figura 2.7: Exemplo de indivíduo no arcabouço PG^+ .

Durante o processo de aprendizado, os indivíduos que melhor codificam as preferências do usuário devem receber os maiores valores de adequação. A *adequação de um indivíduo* é calculada a partir de seu desempenho em um *conjunto de treinamento*. No arcabouço PG^+ o conjunto de treinamento é formado por um conjunto T de imagens e uma função r^+ . T é composto por imagens marcadas como relevante, por imagens exibidas para o usuário nas últimas iterações e por outras escolhidas aleatoriamente da base. A função r^+ indica a rotulação do usuário em relação a cada imagem $t \in T$ podendo assumir o valor 1 se t é relevante e 0 se não relevante ou não rotulado.

A adequação de um indivíduo é calculada com base na similaridade entre as imagens do padrão de consulta Q e todas as imagens do conjunto de treinamento T . O processo de cálculo da adequação de um indivíduo é realizada em três fases. Na primeira fase, são definidas listas ordenadas, cada uma considerando a similaridade entre todas as imagens do conjunto de treinamento e cada imagem do padrão de consulta. Na segunda fase, essas ordenações são avaliadas. Finalmente, na última fase, o valor final da adequação do indivíduo é calculado.

Após o cálculo da adequação dos indivíduos é possível realizar a *ordenação das imagens da base*. Como há a possibilidade de que vários indivíduos apresentem uma boa adequação, um critério foi definido para selecionar quais serão utilizados para ordenar a base: um esquema de *votação*. Primeiramente, um conjunto de indivíduos é selecionado para participar do processo de votação. Seja δ_{melhor} o melhor indivíduo obtido do processo de aprendizado na iteração corrente. O conjunto S de indivíduos selecionados para participar é definido como uma constante chamada *limitante para votação*:

$$S = \{\delta_i \mid \frac{F_{\delta_i}}{F_{\delta_{melhor}}} \geq \alpha\} \quad (2.1)$$

onde $\alpha \in [0, 1]$ é uma constante e F_{δ_i} o valor de adequação do indivíduo δ_i .

O primeiro passo do *processo de votação* é a ordenação das imagens da base utilizando os indivíduos selecionados. A ordenação é feita de acordo com a similaridade entre cada imagem da base e o padrão de consulta Q . Desse modo, para cada indivíduo do conjunto S vai existir uma lista das imagens da base ordenadas. Depois, cada imagem nas L primeiras posições em cada lista recebe um voto inversamente proporcional a sua posição. Por exemplo, a primeira imagem recebe um voto igual a 1; a segunda recebe 1/2; a terceira, 1/3 e assim por diante. Finalmente, as L imagens mais votadas são selecionadas para serem exibidas para o usuário.

2.4.2 Arcabouço PG^\pm

O arcabouço PG^\pm é uma extensão do arcabouço PG^+ visando incorporar as imagens não relevantes ao processo de aprendizado das preferências do usuário e ordenação da base [17]. Segundo Ferreira [17], utilizar a informação sobre imagens não relevantes pode trazer ganhos para o processo de recuperação.

O PG^\pm foi criado alterando-se dois componentes do PG^+ : a definição do conjunto de treinamento e o modo como os indivíduos selecionados para votação são utilizados para ordenar as imagens da base.

O novo *conjunto de treinamento* T contém, além de imagens relevantes e não-rotuladas, as imagens marcadas como não relevantes. Seja IRR o conjunto das imagens marcadas pelo usuário como não relevantes ao longo de todas as iterações. O conjunto de treinamento utilizado em PG^\pm é composto por: todas as imagens do padrão de consulta; imagens não rotuladas, escolhidas aleatoriamente da base; imagens não relevantes, escolhidas aleatoriamente do conjunto IRR .

O processo de *ordenação da base* é semelhante ao de PG^+ . A diferença é o modo como os indivíduos selecionados para votação são utilizados para ordenar as imagens da base. No caso PG^\pm , as imagens não relevantes interferem no *processo de votação* atuando com pontuação negativa. Essa estratégia privilegia indivíduos que posicionam imagens relevantes nas primeiras posições das ordenações e deixa nas últimas posições as imagens não relevantes.

Capítulo 3

Método Semi-automático de Vetorização

Este capítulo descreve um novo método semi-automático de vetorização de imagens de sensoriamento remoto utilizando realimentação de relevância baseada em programação genética.

Dentre os métodos existentes para classificação de ISRs, os mais comuns são os baseados em pixels, ou seja, aqueles em que o objetivo é encontrar a classe de cada pixel e não da região da qual faz parte. Nas abordagens de classificação baseadas em regiões, o processo normalmente consiste em segmentar a imagem e depois classificar as regiões segmentadas. No método proposto, entretanto, a classificação é realizada antes da segmentação.

O processo de extração de regiões de interesse em Imagens de Sensoriamento Remoto no “formato vetorial” pelo método proposto é composto por oito etapas, como pode ser observado na Figura 3.1.

Seja I uma ISR e p uma subimagem de I composta de $n \times n$ pixels (etapa 1). A imagem I é particionada em um conjunto de sub-imagens de mesmo tamanho (etapa 2). Depois, para cada sub-imagem p , são extraídos os vetores de características utilizando descritores de imagem (etapa 3). No próximo passo, o usuário seleciona uma sub-imagem que será o padrão de consulta (etapa 4). Utilizando a similaridade com relação ao padrão de consulta, as demais sub-imagens da imagem são classificadas e o resultado é mostrado ao usuário (etapa 5). Nesse ponto, se inicia o processo de realimentação de relevância (etapa 6) em que o usuário deve indicar as sub-imagens relevantes e não relevantes. O processo se repete enquanto o usuário achar necessário. Quando o usuário estiver satisfeito se inicia a próxima etapa do processo que consiste em utilizar as sub-imagens classificadas para extrair sementes para segmentar a imagem I (etapa 7). Com a imagem I segmentada, são extraídos polígonos referentes às regiões de interesse. Essa etapa é o que chamamos de vetorização (etapa 8). Baseado na Figura 3.1, o sistema foi modelado como pode ser

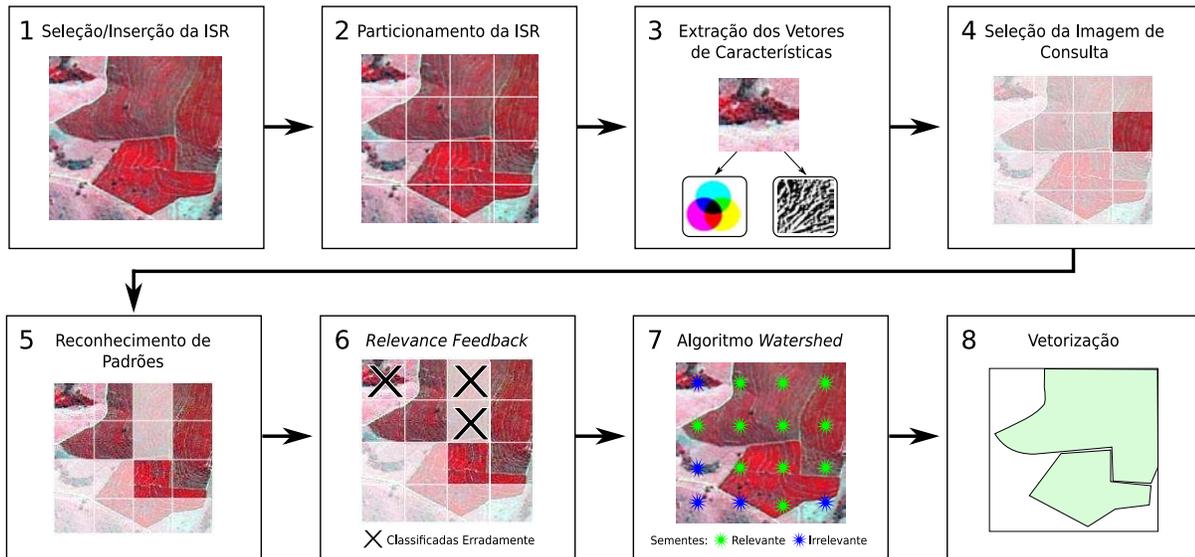


Figura 3.1: Etapas do processo de vetorização

visto na Figura 3.2.

Como mostra a Figura 3.2, a interface permite ao usuário a seleção/inserção da ISR, especificar uma consulta por meio de um padrão e visualizar os resultados da classificação durante o processo de realimentação de relevância. O módulo de processamento de consultas particiona a ISR e extrai o vetor de características das sub-imagens. Depois, esse módulo ordena as sub-imagens de acordo com a similaridade com relação ao padrão de consulta e retorna para o módulo de interface. O módulo realimentação de relevância recebe as indicações do usuário e reformula a classificação das sub-imagens. Quando o usuário estiver satisfeito com o resultado, as sub-imagens são utilizadas como fonte de marcadores pelo módulo de segmentação. Após segmentada, a ISR é vetorizada e armazenada em um repositório.

As etapas da Figura 3.1 podem ser agrupadas em quatro etapas principais que serão explicadas nas próximas seções: sub-imagem e extração de características da imagem (Seção 3.1), reconhecimento das regiões de interesse (Seção 3.2), segmentação da imagem (Seção 3.3) e vetorização das regiões encontradas (Seção 3.4).

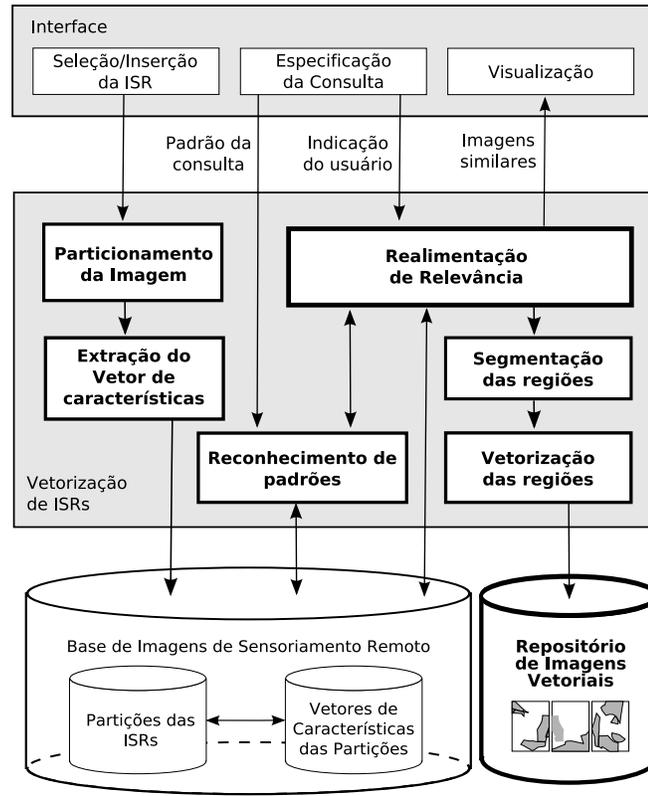


Figura 3.2: Arquitetura do sistema de vetorização proposto.

3.1 Partição e Extração de Características

O *processo de partição* da imagem consiste em criar uma grade de sub-imagens $n \times n$ pixels a partir de uma ISR I . Assim, uma ISR I é dividida em um conjunto de sub-imagens $P = \{p_1, p_2, \dots, p_{n_p}\}$ onde n_p é o número de sub-imagens.

Sejam a e b as dimensões (altura e largura, respectivamente) da ISR I , então o número de sub-imagens $n_p = (a \times b)/n^2$, onde n^2 é o número de pixels da sub-imagem ($n \times n$).

O valor de n é baseado no tamanho estimado que poderia ter uma região de interesse. Dessa forma, um valor ideal para n é o que faz com que várias sub-imagens estejam contidas dentro das regiões de interesse. Isso permite que as mesmas sejam utilizadas para reconhecer padrões de textura da região em que se encontra. O exemplo da Figura 3.3 simula uma situação que ilustra como o tamanho escolhido para a sub-imagem influencia no processo de reconhecimento.

Considere a ISR da Figura 3.3 (a). A Figura 3.3 (b) corresponde a um suposto polígono de interesse nessa imagem. As Figuras 3.3 (c) e (d) correspondem à imagem

dividida em 16 e 64 sub-imagens, respectivamente. Observe que se a imagem for dividida em 64 sub-imagens, a área da imagem considerada relevante é maior. Embora sub-imagens menores pareçam fornecer mais vantagem, o tamanho da sub-imagem não pode ser muito pequeno, pois isso atrapalharia a extração e codificação de informação de textura. É importante salientar que tamanhos de sub-imagens muito grandes também impedem a captura de informações de textura de forma adequada, uma vez que vários padrões podem ser encapsulados nessas sub-imagens.

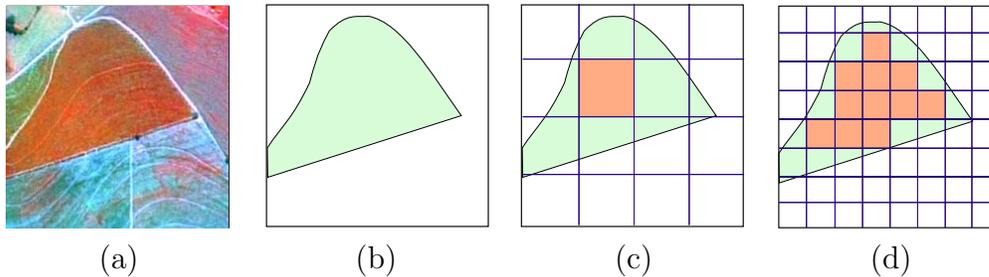


Figura 3.3: Exemplo de estimativa do tamanho das sub-imagens. (a) A ISR da região de interesse. (b) O polígono que representa a região de interesse. (c) A imagem dividida em 16 sub-imagens. (d) A mesma ISR dividida em 64 sub-imagens.

Após o particionamento, para cada sub-imagem, um conjunto de vetores de características $F_{r_i} = \{f_{1_{r_i}}, f_{2_{r_i}}, \dots, f_{n_D}\}$ é extraído usando um conjunto $\mathcal{D} = \{D_1, D_2, \dots, D_{n_D}\}$ contendo n_D descritores pré-definidos.

3.2 Reconhecimento das Regiões de Interesse

O processo de *identificação das sub-imagens de interesse* consiste em identificar as sub-imagens que compõem as regiões de interesse utilizando as características extraídas por meio dos descritores. Essa etapa considera as sub-imagens independentemente, tratando-as como uma base de imagens de um sistema de recuperação por conteúdo baseado em realimentação de relevância. Dessa forma, o usuário seleciona, inicialmente, uma sub-imagem de interesse como imagem de consulta. Uma busca por similaridade de características é realizada e o resultado é mostrado para o usuário que indica sub-imagens retornadas que sejam de interesse (sub-imagens relevantes). O processo é então refeito até que o usuário esteja satisfeito com o resultado.

A identificação das sub-imagens de interesse é realizada com o uso do arcabouço GP_{LSP} [43], uma adaptação do arcabouço GP^+ (Seção 2.4.1). O GP_{LSP} foi criado para realizar realimentação de relevância em sistemas de recuperação baseado em informações regionais das imagens. Apesar de apresentar bons resultados na recuperação de ima-

gens, esse arcabouço ainda precisou ser adaptado para trabalhar com ISRs, o GP_{SR} . Os arcabouços GP_{LSP} e GP_{SR} são apresentados nas Seções 3.2.1 e 3.2.2, respectivamente.

3.2.1 Arcabouço GP_{LSP}

GP_{LSP} [43] é um método de realimentação de relevância para recuperação por conteúdo que utiliza características locais das imagens. Esse método utiliza a mesma abordagem de programação genética (PG) utilizada pelos arcabouços propostos por Ferreira (Seção 2.4). Desse modo, GP_{LSP} utiliza PG para aprender as preferências do usuário e combinar a similaridade das regiões da imagem em uma seção de consulta. Mais detalhes são apresentados nas seções a seguir.

Modelo de Similaridade de Imagens Baseado em Regiões

Em geral, Modelos de Similaridade Baseados em Regiões (MSBRs) representam a similaridade entre imagens como a combinação da similaridade das regiões [50]. Há várias maneiras de modelar similaridade de imagens utilizando regiões. Algumas abordagens são baseadas em segmentação. Entretanto, dividir a imagem em sub-imagens de mesmo tamanho é mais fácil, uma vez que segmentação requer algoritmos e estruturas de dados mais complexas e difíceis de serem manipuladas.

Por esse motivo, o modelo utilizado pelo arcabouço GP_{LSP} é baseado nos métodos propostos por Stejic [48–50]. A especificação formal do modelo é apresentada a seguir.

Seja I um conjunto de imagens que representa a base de imagens. Cada imagem é dividida em um conjunto de sub-imagens $R = \{r_1, r_2, \dots, r_{n_R}\}$. A Figura 3.4 ilustra a divisão de uma imagem em 9 sub-imagens. Para cada sub-imagem, um conjunto de vetores de características $F_{r_i} = \{f_{1r_i}, f_{2r_i}, \dots, f_{n_D}\}$ é extraído por meio de um conjunto $\mathcal{D} = \{D_1, D_2, \dots, D_{n_D}\}$ de n_D descritores.

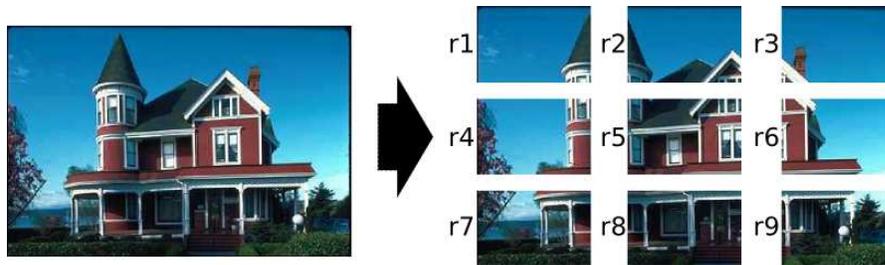


Figura 3.4: Exemplo de divisão de imagem em sub-imagens.

Um descritor D_i retorna o grau de similaridade de uma determinada característica

$d_i r_j I_a I_b$, utilizando a função de similaridade δ_{D_i} de um par de imagens I_a e I_b com relação à característica f_i de uma sub-imagem r_j . Dado um conjunto de valores correspondentes à similaridade de características, um descritor composto \mathcal{D} retorna o valor de similaridade $d_i I_a I_b$ das imagens I_a e I_b .

A Figura 3.5 mostra o modelo de similaridade baseado em regiões utilizado. Para cada sub-imagem de uma imagem, os vetores de características e similaridades são calculados utilizando os k descritores disponíveis. A função δ_D é utilizada para combinar a similaridade das sub-imagens. É possível observar que essa estrutura é um típico descritor composto como apresentado na Seção 2.3.

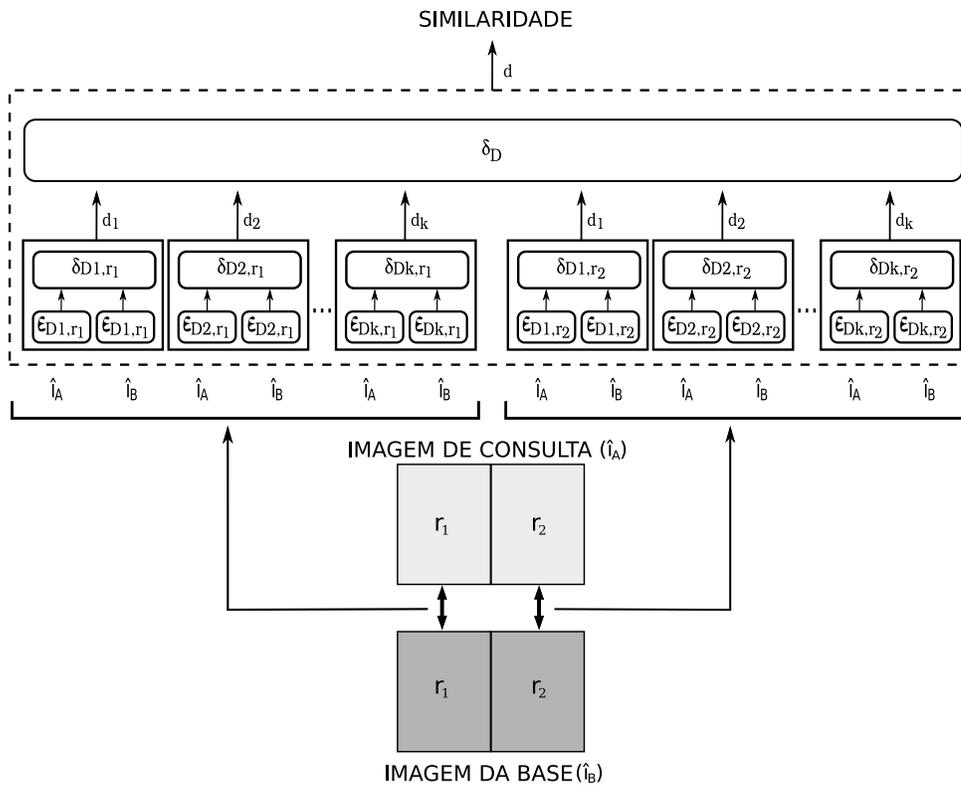


Figura 3.5: Exemplo do cálculo de similaridade utilizando o modelo por sub-imagens.

Especificação do Arcabouço

Nesse método, um descritor composto $\hat{D} = (\mathcal{D}, \delta_{\mathcal{D}})$ é empregado para ordenar as N imagens da base definidos como $DB = \{db_1, db_2, \dots, db_N\}$. O conjunto de K descritores simples \hat{D} é representado por $\mathcal{D} = \{D_1, D_2, \dots, D_{n_{\mathcal{D}}}\}$. As imagens da base são divididas

em um conjunto de sub-imagens $R = \{r_1, r_2, \dots, r_{n_R}\}$. A similaridade entre as sub-imagens de duas imagens I_{a,r_j} e I_{b,r_j} , computado por D_i , é representado por $d_i r_{j I_a I_b}$. Todas as similaridades $d_i r_{j I_a I_b}$ são normalizadas entre 0 and 1. Uma normalização Gaussiana [41] foi empregada para normalizar esses valores. Assim, a similaridade entre duas imagens I_a e I_b é obtido combinando as $n_D \times n_R$ similaridades das sub-imagens.

Seja L o número de imagens mostradas ao usuário em cada iteração. Seja Q o padrão de consulta $Q = \{q_1, q_2, \dots, q_M\}$, onde M é o número de elementos em Q , formado pela imagem de consulta q_1 e todas as imagens definidas como relevantes durante a seção de realimentação.

Como mencionado anteriormente, o objetivo do arcabouço GP_{LSP} é encontrar as funções de similaridade das sub-imagens que melhor codificam as preferências do usuário. Por ser uma estratégia baseada em PG (Seção 2.2), necessita a definição de uma série de componentes. Com exceção da *definição dos indivíduos* da programação genética, a *seleção do conjunto inicial* de imagens, *definição do conjunto de treinamento*, o *cálculo de adequação dos indivíduos* e o *processo de votação* são os mesmos do arcabouço PG^+ (Seção 2.4.1).

No arcabouço GP_{LSP} , cada indivíduo da PG representa uma função candidata δ_D , que é uma função de combinação de similaridades. O indivíduo é codificado como uma árvore. Os nodos internos contêm operadores aritméticos. As folhas da árvore possuem valores de similaridade $d_i r_{j I_a I_b}$, onde $1 \leq i \leq K$ e $1 \leq j \leq n_R$. A Figura 3.6 mostra um exemplo de indivíduo GP_{LSP} . O indivíduo da figura representa a função:

$$f(d_1 r_{1 I_a I_b}, d_2 r_{1 I_a I_b}, d_1 r_{2 I_a I_b}, d_2 r_{2 I_a I_b}) = \frac{d_1 r_{1 I_a I_b} * d_2 r_{1 I_a I_b}}{d_2 r_{2 I_a I_b}} + \sqrt{d_1 r_{2 I_a I_b}}$$

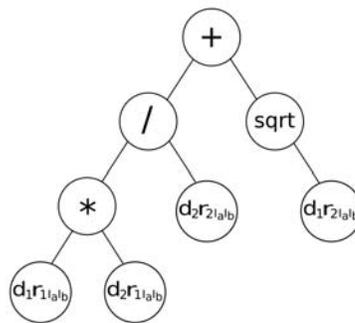


Figura 3.6: Indivíduo PG no arcabouço GP_{LSP} .

A Figura 3.6 considera o uso de três descritores distintos e o conjunto de operadores $\{+, /, *, sqrt\}$ como nós internos. A representação do indivíduo é ideal para o mecanismo de busca do arcabouço, uma vez que codifica diretamente os candidatos para a função δ_D .

3.2.2 Arcabouço GP_{SR}

O GP_{SR} é uma adaptação do arcabouço GP_{LSP} para a identificação de relevância entre sub-imagens de ISR. Dessa forma, o GP_{SR} não utiliza o modelo de similaridade de imagens baseado em regiões. No caso do arcabouço GP_{SR} , uma ISR é vista como uma base de imagens enquanto as sub-imagens podem ser vistas como imagens independentes.

Então, no GP_{SR} , seja \hat{I} uma ISR, um descritor composto $\hat{D} = (\mathcal{D}, \delta_{\mathcal{D}})$ é empregado para ordenar N sub-imagens definidas como $\hat{I} = \{r_1, r_2, \dots, r_N\}$. O conjunto de K descritores \hat{D} é representado por $\mathcal{D} = \{D_1, D_2, \dots, D_{n_{\mathcal{D}}}\}$ e a similaridade entre duas sub-imagens r_j e r_k , calculada por D_i , é representada por $d_{ir_j r_k}$.

A definição do indivíduo GP pelo arcabouço GP_{SR} é similar ao GP_{LSP} embora as sub-imagens, nesse caso, sejam as unidades a serem ordenadas posteriormente.

Algoritmo 3 O processo de reconhecimento de sub-imagens do arcabouço GP_{SR} .

- 1 *Indicação da sub-imagem de consulta q_1*
 - 2 *Mostre o conjunto inicial de sub-imagens*
 - 3 **Enquanto** o usuário não estiver satisfeito **faça**
 - 4 *Indicação das sub-imagens de interesse*
 - 5 *Atualize o padrão de consulta Q*
 - 6 *Utiliza PG para encontrar os melhores indivíduos (funções de similaridade)*
 - 7 *Ordene as sub-imagens*
 - 8 *Mostre as L sub-imagens mais similares*
 - 9 **fim enquanto**
-

O Algoritmo 3 inicia o processo de reconhecimento com a definição da sub-imagem de consulta pelo usuário (linha 1). A partir dessa sub-imagem, um conjunto inicial de sub-imagens é selecionado para ser exibido para o usuário (linha 2). Utilizando o conjunto inicial, o processo de realimentação de relevância se inicia, com a indicação das sub-imagens de interesse pelo usuário. São realizadas iterações até que o usuário esteja satisfeito com o resultado (linha 3). Nessas iterações, são realizadas as seguintes etapas: indicação das sub-imagens de interesse pelo usuário (linha 4); atualização do padrão de consulta (linha 5); atualização das funções de combinação de similaridade das sub-imagens utilizando programação genética (linha 6); ordenação das sub-imagens (linha 7); e exibição das sub-imagens mais similares (linha 8).

As etapas de *seleção do conjunto inicial* de imagens, definição do *conjunto de treinamento*, o *cálculo de adequação dos indivíduos* e *processo de votação* do arcabouço GP_{SR} , são similares às etapas do método PG^+ (Seção 2.4.1). A diferença é que, nas etapas em que as imagens são processadas no arcabouço PG^+ , o arcabouço GP_{SR} processa sub-imagens da ISR.

3.3 Segmentação da Imagem

A classificação das sub-imagens é utilizada como fonte de sementes (marcadores) para a segmentação da ISR por meio do algoritmo de *watershed*. A Seção 3.3.1 explica o algoritmo *watershed* enquanto a Seção 3.3.2 descreve o processo de extração de marcadores para o algoritmo de *watershed* baseado na classificação das sub-imagens.

3.3.1 Algoritmo *Watershed*

O processo de *segmentação da imagem* é baseado no algoritmo de “watershed” [29]. O algoritmo de *watershed* é um algoritmo de segmentação de processamento de imagens que divide uma imagem em áreas baseando-se na topologia da imagem. A largura do gradiente é interpretado como elevações. Durante sucessivas “inundações”, vales com bases adjacentes são construídos. Normalmente esse processo geraria uma super segmentação da imagem, especificamente para imagens ruidosas como, por exemplo, imagens médicas. A imagem deve ser pré-processada ou as regiões devem ser intercaladas nas bases com um critério de similaridade posterior [14, 29, 30].

Watershed Clássico

Um mínimo regional de uma imagem em níveis de cinza é uma área plana não adjacente a nenhuma outra área plana com menor altitude [56]. No algoritmo de *watershed clássico*, normalmente a transformação é aplicada em um *gradiente morfológico* da imagem que se deseja segmentar. Nessa transformação, cada mínimo regional é considerado um vale e ao final do processo corresponderá a um segmento. Como uma imagem gradiente possui muitos mínimos regionais (depende das texturas e ruídos da imagem original), a imagem resultante normalmente possui “super-segmentação”. As regiões criadas pelos mínimos regionais são chamadas *bacias primitivas* [56]. A Figura 3.7 ilustra o processo.

Suponha uma imagem gradiente como na Figura 3.7 (a). A Figura 3.7 (b) representa o “CORTE” da Figura 3.7 (a) como “vales” formados pelos diferentes níveis de cinza da imagem. As cores nos fundos dos vales representam os mínimos regionais. Quando o algoritmo é executado nessa imagem gradiente se inicia o processo de “inundação”. A Figura 3.7 (c) mostra o momento em que as águas de duas bacias (amarela e azul) se encontram, definindo o limite entre suas regiões. A Figura 3.7 (d) ilustra o momento em que as águas da região vermelha e cinza se encontram. O processo continua até que todas as regiões sejam cobertas por água. A Figura 3.7 (e) mostra o resultado final. A imagem resultante pode ser visualizada na Figura 3.7 (f).

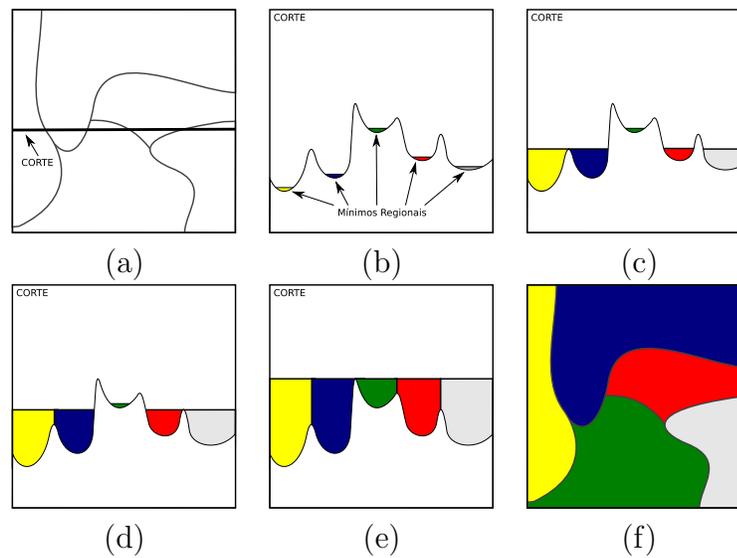


Figura 3.7: Exemplo de execução do algoritmo *watershed*. (a) Uma imagem gradiente com um corte. (b) O corte visto como um vale e seus mínimos regionais. (c) A “enchente” no tempo t_1 . (d) A enchente no tempo t_2 . (e) Resultado final da execução do algoritmo. (f) Imagem segmentada.

Watershed Baseado em Marcadores

O algoritmo de *watershed baseado em marcadores* é uma adaptação do *watershed clássico* que permite a redução dos segmentos na imagem resultante. A diferença básica é a inserção de *marcadores* (sementes). Nessa abordagem, cada vale do *watershed* será um marcador da imagem e não mais um mínimo regional. Os marcadores disputam as regiões e incorporam as *bacias primitivas*. A Figura 3.8 ilustra o processo.

Assim como na Figura 3.7 (a), suponha uma imagem gradiente agora com marcadores (Figura 3.8 (a)). A Figura 3.7 (b) representa o “CORTE” da Figura 3.8 (a) como “vales” formados pelos diferentes níveis de cinza da imagem e os marcadores (linhas em azul e vermelho). O processo de “inundação” se inicia e agora a água dos marcadores tende a inundar os outros vales da imagem. As Figuras 3.8 (c) e (d) mostram o momento em que as águas do marcador azul e do marcador vermelho “inundam” regiões desmarcadas, respectivamente. O processo continua até que todas as regiões sejam cobertas por água. A Figura 3.7 (e) mostra o resultado final, quando os dois marcadores se encontram. A imagem resultante pode ser visualizada na Figura 3.7 (f).

O algoritmo de *watershed baseado em marcadores* é uma abordagem interativa uma vez que os marcadores não são encontrados automaticamente no processo de segmentação ao contrário do *watershed clássico*. No método proposto neste trabalho, as sub-imagens identificadas pelo módulo de realimentação de relevância (Seção 3.2) são utilizadas como

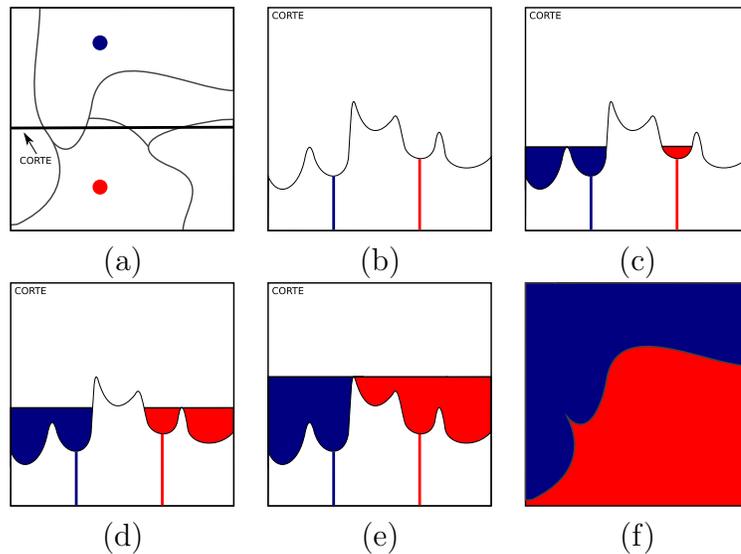


Figura 3.8: Exemplo de execução do algoritmo watershed com definição de marcadores. (a) Uma imagem gradiente com um corte e marcadores (em vermelho e azul). (b) O corte visto como um vale. (c) A “enchente” no tempo t_1 . (d) A enchente no tempo t_2 . (e) Resultado final da execução do algoritmo. (f) Imagem segmentada.

“marcadores” para o algoritmo de “watershed”. O processo de seleção de sementes é apresentado na Seção 3.3.2.

Watershed Baseado na Transformada Imagem-Floresta

A Transformada Imagem-Floresta (*Image Forest Transforming* – IFT) [14] é uma abordagem baseada em grafos para processamento de imagens. Dentre outras vantagens, a IFT permite que operadores de imagens sejam derivados de um mesmo algoritmo base. Além disso, o algoritmo da IFT pode ser executado em tempo linear em condições especiais [14].

Na IFT, a imagem I é vista como um grafo com pesos $G_I = (V, E, \omega)$, onde ω é uma função de dissimilaridade, como por exemplo: $\omega(u, v) = \max(I(u), I(v))$.

A abordagem de *watershed com marcadores* pode ser modelada como um problema de otimização em grafos por meio da IFT. Nesse caso, o algoritmo consiste na criação de uma floresta na qual existe um caminho de custo mínimo entre cada pixel e os marcadores. A *transformada watershed* pela IFT [14] assume uma função f_{max} para calcular o custo dos caminhos:

$$f_{max}(\langle v_1, v_2, \dots, v_n \rangle) = \max\{\omega(v_i, v_i + 1)\} = \max\{I(v_i)\}$$

Seja $\pi = \langle v_1, v_2, \dots, v_{n-1} \rangle$ um caminho pré-existente, quando uma aresta (v_{n-1}, v_n) é acrescentada a um π , o custo do novo caminho pode ser calculado em tempo constante a partir do custo de π .

O algoritmo gera uma floresta resultante P , formada por árvore de predecessores, em que cada conjunto de árvores cuja raiz tem o mesmo rótulo corresponde a uma região de interesse.

3.3.2 Seleção de sementes

A seleção de marcadores para a segmentação baseada em *watershed* no método proposto depende do corte que é feito na lista de sub-imagens gerada pelo módulo de reconhecimento de padrões. Como apresentado na Seção 3.2.1, o algoritmo GP_{LSP} gera uma lista ordenada das sub-imagens da ISR. Quanto mais próxima das primeiras posições, mais similar uma sub-imagem pode ser considerada da sub-imagem de consulta. Para utilizar a lista ordenada como fonte de semente para a segmentação, entretanto, é necessário saber em que ponto da lista se separam as sub-imagens relevantes (que contêm regiões de interesse) das não relevantes. Por esse motivo, é preciso realizar “cortes” na lista ordenada. Os “cortes” na lista de sub-imagens ordenadas são uma variável do método proposto. A Figura 3.9 ilustra exemplos de cortes em uma lista de sub-imagens ordenadas.

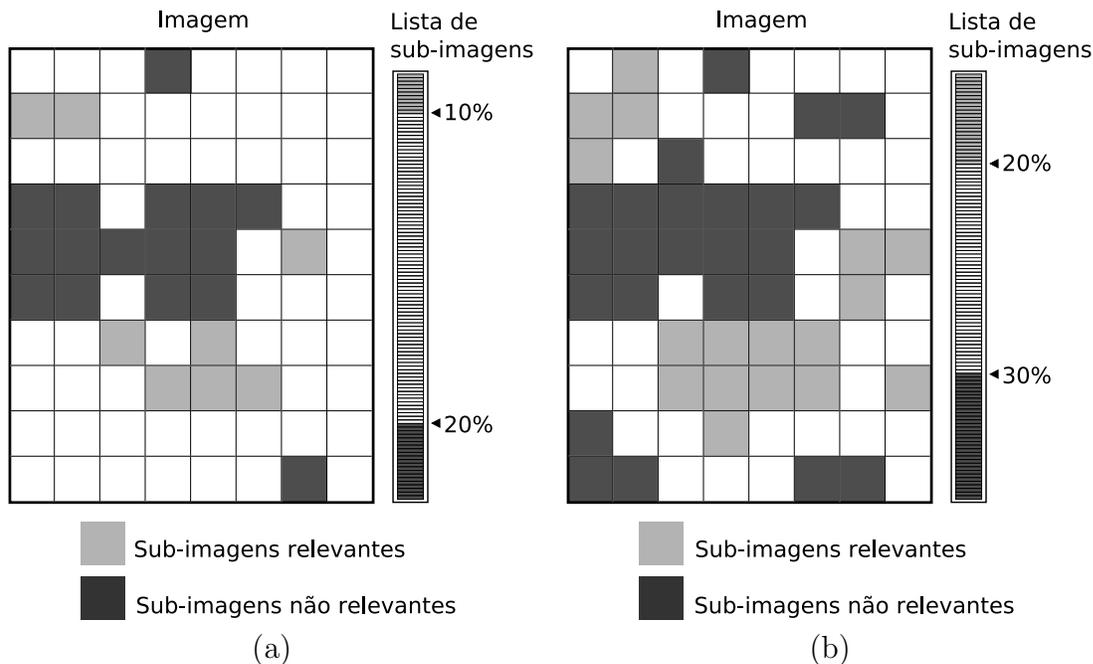


Figura 3.9: Exemplos de cortes na lista de sub-imagens ordenadas. (a) Cortes de 10% relevantes e 20% relevantes. (b) Cortes de 20% relevantes e 30% não relevantes.

Na Figura 3.9 (a), as 10% sub-imagens melhor posicionadas na lista ordenada foram selecionadas para serem marcadores de regiões de interesse e as 20% sub-imagens que

ocupam as últimas posições da lista foram selecionadas como marcadores de regiões não relevantes. Na Figura 3.9 (b), foram selecionadas para serem marcadores de regiões de interesse as 20% sub-imagens melhor posicionadas na lista ordenada e para marcadores de regiões não relevantes, as 30% sub-imagens que ocupam as últimas posições da lista. Observe que os índices de corte influenciam não apenas na quantidade de marcadores, mas também na distribuição deles. Esse fator faz com que diferentes níveis de corte gerem imagens segmentadas distintas, como pode ser observado nas Figuras 3.10 e 3.11.

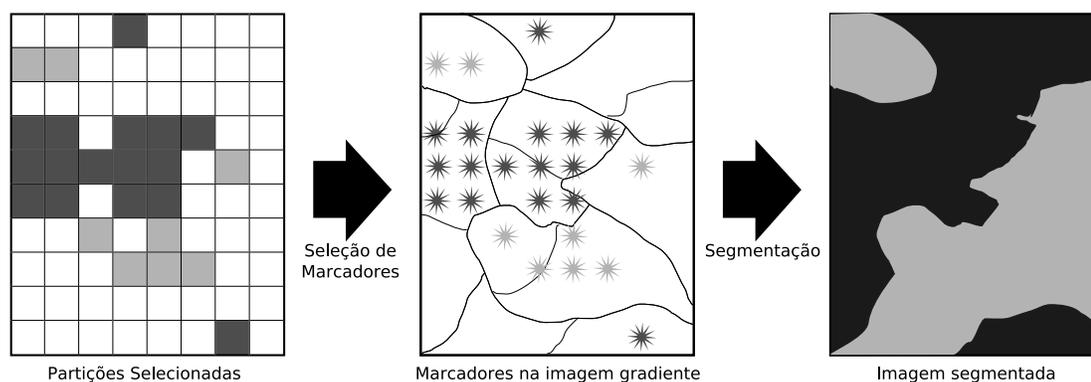


Figura 3.10: Exemplo de segmentação considerando as 10% sub-imagens mais relevantes e as 20% menos relevantes como marcadores.

A Figura 3.10 ilustra o processo de segmentação da imagem a partir das sub-imagens selecionadas no exemplo da Figura 3.9 (a). Das sub-imagens selecionadas são extraídos marcadores de relevância e não relevância. O *algoritmo watershed* é aplicado na imagem gradiente utilizando os marcadores selecionados, gerando a imagem segmentada.

Na Figura 3.11, o processo de segmentação da imagem é feito utilizando as sub-imagens selecionadas no exemplo da Figura 3.9 (b). É importante observar como os diferentes níveis de corte interferem no resultado final da segmentação.

3.4 Vetorização das Regiões

O processo de segmentação gera uma imagem binária nas mesmas dimensões da ISR original indicando as regiões da cultura agrícola de interesse e as demais. O *processo de vetorização* consiste em utilizar a imagem segmentada para extrair os polígonos das regiões de interesse. A ferramenta utilizada neste trabalho foi o Potrace [45]. Os parâmetros utilizados pelo Potrace para vetorização e detalhes de uso são descritos na Seção 5.2.4.

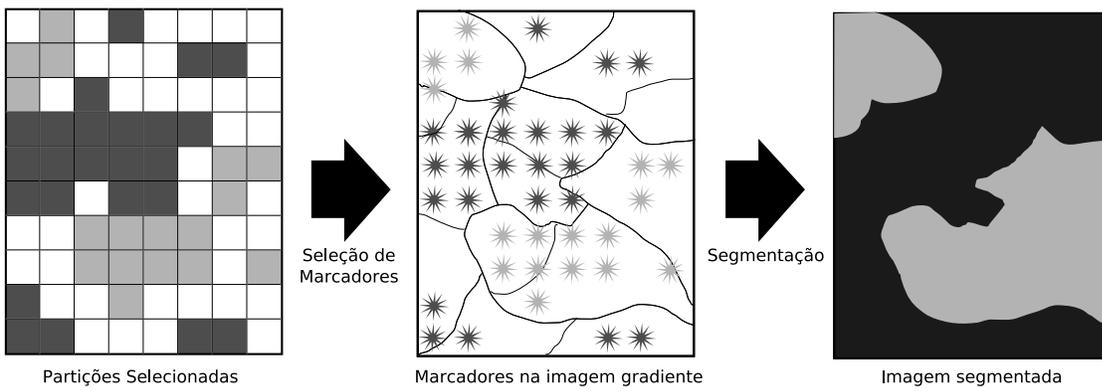


Figura 3.11: Exemplo de segmentação considerando as 20% sub-imagens mais relevantes e as 30% menos relevantes como marcadores.

Capítulo 4

Validação do Método de Realimentação de Relevância para Regiões

Para validar o GP_{LSP} (apresentado na Seção 3.2.1) foram realizados um conjunto de experimentos que são descritos neste capítulo. O objetivo principal dos experimentos realizados foi avaliar a utilização de programação genética na recuperação de imagens baseando-se nas informações regionais das imagens. A Seção 4.1 apresenta o projeto do experimento. Os resultados obtidos são apresentados na Seção 4.2.

4.1 Projeto do Experimento

Esta seção descreve o projeto dos experimentos para validar o arcabouço GP_{LSP} . A Seção 4.1.1 apresenta os descritores de cor e textura utilizados. A Seção 4.1.2 apresenta o método de comparação utilizado. A base de imagens utilizada nos experimentos é descrita na Seção 4.1.3. A Seção 4.1.4 apresenta os detalhes de implementação do arcabouço GP_{LSP} . Na Seção 4.1.5 é apresentada a metodologia de avaliação. Por fim, a simulação do usuário é explicada na Seção 4.1.6.

4.1.1 Descritores

Esta seção apresenta os descritores utilizados nos experimentos. Os parâmetros aplicados aos descritores de cor e textura aplicados foram obtidos de trabalhos anteriores e são detalhados em [17].

Descritores de Cor

Foram utilizados três descritores de cor, o *Histograma de Cor*, os *Momentos de Cor* e o *Border/Interior pixel Classification (BIC)*.

Histograma de Cor

É um dos descritores mais simples. Neste trabalho foi extraído o histograma [52] no espaço de cor *HSV*, quantizado em 16, 4, 4, para tonalidade (*Hue*), saturação (*Saturation*) e valor (*Value*), respectivamente, para todos os *pixels* das regiões das ISRs processadas. A extração gerou um vetor de características com 256 dimensões. A similaridade foi calculada utilizando-se a distância *L1*.

Momentos de Cor

O descritor Momentos de Cor [51] caracteriza a distribuição das cores de uma imagem como os três primeiros momentos de cada canal de cor. Esses momentos são a média, o desvio-padrão e a terceira raiz da obliquidade (*skewness*). Foi utilizado o espaço de cor *HSV* e o processo de extração originou um vetor de características com 9 dimensões. A função de similaridade utilizada é definida como uma variação da distância *L1*, chamada d_{mon} que atribui pesos pré-definidos para cada *bin* do vetor de características.

Border/Interior pixel Classification (BIC)

O descritor BIC [47] classifica cada *pixel* de uma imagem como interior ou borda, em um espaço de cor *RGB* quantizado em 4x4x4. Após a classificação, calcula dois histogramas, um considerando apenas os *pixels* de borda e outro considerando apenas os *pixels* interiores. Um *pixel* é classificado como interior se possuir a mesma cor dos *pixels* em sua vizinhança 4 (em cima, em baixo, à esquerda e à direita). Caso contrário, é classificado como borda. Com isso, uma imagem é descrita por dois histogramas, com 64 *bins* cada. Esses dois histogramas definem um vetor de características com 128 dimensões. A função de distância *dLog* [47] é utilizada para comparar dois vetores de características definidos pelo *BIC* em uma escala logarítmica.

Descritores de Textura

A informação de textura das sub-imagens das ISRs foram codificadas por meio de dois descritores de textura baseados em transformadas *Wavelet*: *Filtros de Gabor* e *Spline*.

Filtros de Gabor

O descritor Filtros de Gabor [25] utiliza a média e o desvio-padrão da distribuição da energia de uma imagem para representá-la. A representação utilizada neste trabalho considera 4 escalas e 4 diferentes rotações, resultando em um vetor de características com 32 dimensões. O cálculo da similaridade é realizado pela distância Euclidiana.

Spline

O descritor Spline [55] utiliza uma transformação *Wavelet Spline*. Essa transformação é estruturada em uma árvore para decompor três sub-bandas de uma imagem. Nesse trabalho foi empregada uma decomposição em três níveis, resultando em um vetor de características com 26 dimensões. Assim como no descritor de Filtros de Gabor, a distância Euclidiana foi utilizada para calcular a distância entre dois vetores de características.

4.1.2 Técnica de Referência

Comparamos o nosso método com a abordagem *LAP* proposta por Stejic et al. [49]. *LAP* [49] é um método para computar a similaridade da imagem baseado na similaridade das sub-imagens.

O método *LAP* computa a similaridade de duas imagens a partir da informação de suas sub-imagens. Esse processo é composto de duas etapas: na primeira, a similaridade das sub-imagens é computada; na segunda, os valores da similaridade das sub-imagens são combinados por meio de *operadores de agregação matemática*.

Para computar a similaridade entre duas imagens, as distâncias das sub-imagens são calculadas para cada vetor de característica usando uma função. Dessa forma, para cada sub-imagem, existem D_n valores de distância de sub-imagens, dependendo do número de descritores usados. Esses valores são combinados resultando em apenas um valor para cada sub-imagem. Depois de combinar os valores locais, eles são combinados em um valor final de distância de imagem.

Stejic et al. [49] definiram um operador de agregação matemática a como uma função de forma $a : [0, 1]^n \mapsto [0, 1]$. Eles usaram algoritmos genéricos para encontrar um bom conjunto de operadores para combinar os valores de similaridade. O conjunto completo é composto de 67 operadores de agregação.

4.1.3 Banco de dados de Imagem

O banco de dados de imagem usado nos experimentos foi um subconjunto da coleção heterogênea de 20000 imagens de Corel GALLERY Magic — Stock Photo Library 2. Essa é uma base bastante utilizada em testes de recuperação de imagens por conteúdo,

Tabela 4.1: Parâmetros de configuração da PG.

Parâmetro	Valor
Tamanho da população	60
Número de gerações	20
Altura máxima	5
Conjunto de funções	+, ×, / (<i>protegida</i>)
Conjunto de terminais	similaridade por descritores simples
População inicial	half and half
Profundidade inicial	2 – 5
Taxa de combinação	0,80
Taxa de mutação	0,20
Método de seleção	torneio (tamanho 2)
Tamanho do conjunto de treinamento	80
Limitante para Votação	0,999

tendo sido utilizada por diversos trabalhos anteriores [16, 17, 37]. O subconjunto usado é composto de 3906 imagens, distribuídas entre 85 classes. Essas classes têm tamanhos diferentes que variam entre 7 e 98.

4.1.4 Detalhes de Implementação

O sistema CBIR GP^+ [17] com os requisitos mínimos foi adaptado para validar nosso método. Os parâmetros de configuração usados na implementação do arcabouço são mostrados na Tabela 4.1.

Na Tabela 4.1, o “tamanho da população” corresponde ao número de indivíduos que compõe a população em uma geração. O parâmetro “Número de gerações” corresponde ao número de gerações do algoritmo evolutivo a cada execução. “Altura máxima” é o tamanho máximo que a árvore que representa um indivíduo pode ter durante o processo de evolução. O “Conjunto de funções” são os operadores matemáticos que um indivíduo pode ter nos nodos internos. O “Conjunto de terminais” são os possíveis valores para as folhas da árvore dos indivíduos. O parâmetro “População inicial” refere-se ao método utilizado para a criação da população inicial de indivíduos. “Profundidade inicial” indica um intervalo que representa as possíveis profundidades que um indivíduo pode ter. Os parâmetros “Taxa de combinação” e “Taxa de seleção” correspondem ao percentual de indivíduos selecionados para as operações de combinação e seleção, respectivamente. O “Método de seleção” indica o método utilizado para selecionar indivíduos para as operações genéticas. O “Tamanho do conjunto de treinamento” corresponde ao tamanho que o conjunto de treinamento possui. O parâmetro “Limitante para votação” é o valor que determina a

razão mínima entre o valor da adequação de um determinado indivíduo δ_i e a adequação do melhor indivíduo, para que δ_i seja selecionado para ordenar as imagens da base, ou seja, se a razão entre o valor de adequação de δ_i e o valor de adequação do melhor indivíduo for maior ou igual ao “Limitante para votação”, o indivíduo é selecionado para a votação.

O número máximo de gerações adotado foi 10. No entanto se um indivíduo tivesse valor de adequação normalizado (entre 0 e 1) igual a 1 antes da última geração, o algoritmo PG em execução era finalizado mais cedo.

Esses parâmetros foram determinados empiricamente depois da execução de vários experimentos. Como pode ser visto na Tabela 4.1, apenas os operadores de combinação e mutação foram usados no processo de busca. Devido ao pequeno tamanho da população, o uso de operador de reprodução fazia com que a diversidade da população caísse rapidamente. Por essa razão, esse operador não foi utilizado. A divisão protegida usada no conjunto da função retorna 1 se o valor do divisor for zero.

4.1.5 Medidas de Avaliação

Foram usadas curvas *precisão-revocação* para avaliar o desempenho dos métodos avaliados. Curva *precisão-revocação* é um critério de avaliação de atuação comumente usado em sistemas de recuperação de informações. A precisão $Pr(q)$ pode ser definida como o número de imagens relevantes recuperadas $R(q)$ sobre o número total de imagens recuperadas $N(q)$ para dada consulta q , ou seja, $Pr(q) = \frac{R(q)}{N(q)}$. Já a revocação $Re(q)$ é o número de imagens relevantes recuperadas $R(q)$ sobre o número total de imagens relevantes $M(q)$ apresentadas no banco de dados para dada consulta q , $Re(q) = \frac{R(q)}{M(q)}$. Espera-se que os valores de precisão e revocação sejam altos. Entretanto, essas medidas são dependentes. Para aumentar o número de imagens relevantes recuperadas (e a revocação), geralmente é necessário aumentar o número de imagens retornadas, e conseqüentemente, a precisão diminui. Por esse motivo, curvas *precisão-revocação* são utilizadas para mostrar essa relação e caracterizar a eficácia da técnica de recuperação.

4.1.6 Simulação do Usuário

O comportamento do usuário foi simulado nos testes. Em cada iteração, todas as imagens pertencentes à mesma classe da consulta são rotuladas como relevantes. Os experimentos consideraram 10 iterações para cada consulta. Em cada iteração, 20 imagens foram mostradas para usuário.

4.2 Resultados Experimentais

A Figura 4.1 mostra as curvas médias precisão-revocação do GP_{LSP} utilizando diferentes particionamentos de área com resolução de 3×3 , 4×4 , 5×5 , 6×6 e 7×7 sub-imagens. Os gráficos referem-se às curvas médias considerando 170 imagens de consulta diferentes. Observe que o $GP_{LSP(3 \times 3)}$ apresenta melhores resultados para valores de revocação maiores que 0,3. Isso pode ser atribuído ao fato dos experimentos terem sido executados e simulados utilizando uma base cujas imagens são distribuídas em classes de acordo com sua informação global.

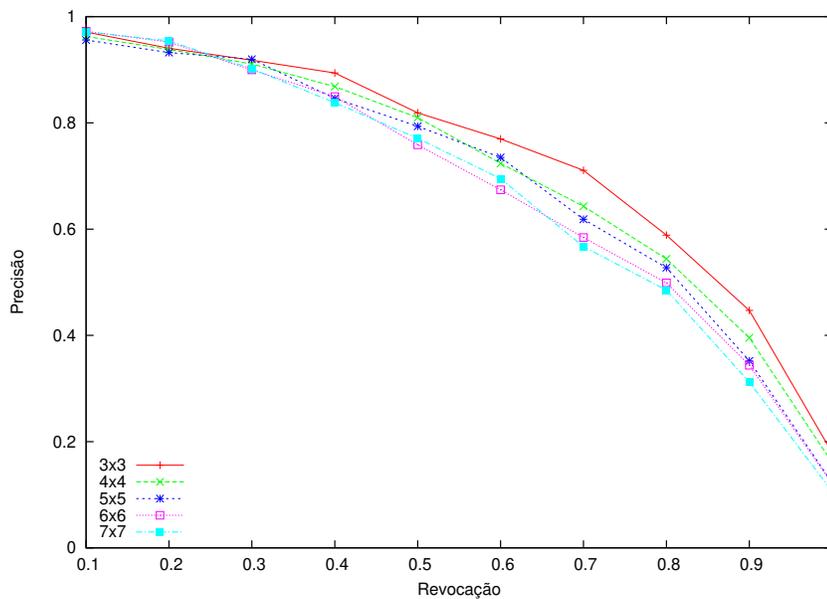


Figura 4.1: Curvas de precisão-revocação mostrando a eficácia do método GP_{LSP} .

A Figura 4.2 compara o melhor resultado do $GP_{LSP(3 \times 3)}$ com os resultados do método LAP . Como pode ser observado, o método proposto obteve melhor precisão para todos os valores de revocação.

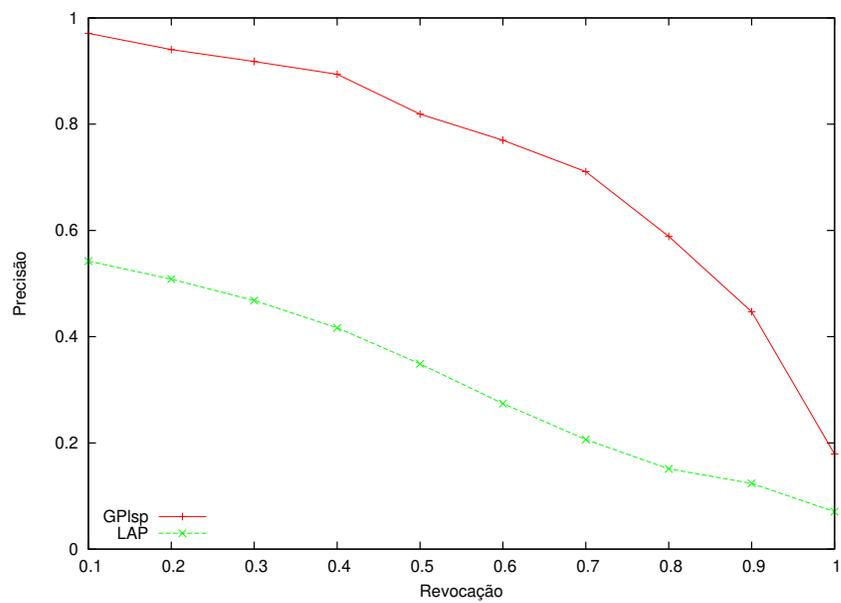


Figura 4.2: Curvas de precisão-revocação mostrando a eficácia dos métodos GP_{LSP} e LAP na base do Corel.

Capítulo 5

Validação do Método de Vetorização Semi-automática

5.1 Projeto do Experimento

Esta seção apresenta a caracterização dos experimentos realizados para validar o método de vetorização semi-automática proposto. Para codificar as características das regiões das imagens de sensoriamento remoto foram utilizados descritores de imagens que manipulam informação de cor e textura. Os descritores utilizados são os mesmos utilizados nos experimentos para validação do arcabouço GP_{LSP} (ver Seção 4.1.1). As imagens de sensoriamento utilizadas são apresentadas na Seção 5.1.1. A medida de avaliação utilizada é abordada na Seção 5.1.2.

5.1.1 Imagens de Sensoriamento Remoto

Para a realização dos testes, duas imagens de sensoriamento remoto foram utilizadas. Uma, em que as regiões de interesse dizem respeito a pastagens e uma outra em que as áreas de interesse são plantações de café. A primeira imagem será denominada PASTAGEM enquanto a segunda será chamada CAFÉ.

Diferentemente de imagens convencionais, as bandas ISRs normalmente não correspondem exatamente ao espectro visível pelo homem. Muitas delas codificam informação de ondas de infra-vermelho próximo, infra-vermelho médio, ultravioleta, etc. Assim, para utilizar descritores criados para imagens convencionais (que geralmente utilizam três canais de cor), é necessário selecionar algumas bandas fornecidas pela ISR e distribuí-las nos três canais de cor RGB disponíveis. Portanto, além das bandas espectrais que foram utilizadas, outras informações sobre as imagens são apresentadas a seguir.

Imagem PASTAGEM

A imagem PASTAGEM é um recorte de uma imagem captada pelo satélite sino-brasileiro CBERS do dia 20 de Agosto de 2005. O recorte corresponde ao vale do Córrego “Laranja Azeda”, no estado de Mato Grosso do Sul. Por ser uma região plana, há pouca interferência do relevo nos padrões espectrais dessa imagem, o que torna a classificação mais fácil. A resolução espacial da imagem é de 20 metros e a composição de bandas utilizada neste trabalho foi R-IR-G, ou seja, vermelho, infra-vermelho e verde nos canais R, G e B do esquema de cores RGB, respectivamente. As dimensões da imagem de pastagem são: 1310×1842 pixels. A imagem PASTAGEM pode ser visualizada na Figura 5.1.



Figura 5.1: Imagem PASTAGEM.

Da imagem PASTAGEM foram extraídas todas as regiões que correspondem à pastagens. Essa classificação deu origem à imagem mostrada na Figura 5.2, criado por pesquisadores da Embrapa Agropecuária Oeste e da Universidade Federal da Grande Dourados [40]. Para classificar essa imagem foi utilizado o software Spring [8]. Primeira-

mente a imagem foi segmentada com o uso do algoritmo “Crescimento de regiões” com similaridade igual a 6 e área igual a 6 [8]. Esse algoritmo é uma técnica de agrupamento de dados, na qual somente as regiões adjacentes espacialmente podem ser agrupadas. Inicialmente, esse processo de segmentação rotula cada “pixel” como uma região distinta. Calcula-se um critério de similaridade para cada par de região adjacente espacialmente. O critério de similaridade baseia-se em um teste de hipótese estatístico que testa a média de similaridade entre as regiões. A seguir, divide-se a imagem em um conjunto de sub-imagens e então realiza-se a união entre elas, segundo um limiar de agregação definido [8].

Depois da segmentação, as regiões encontradas foram classificadas utilizando o algoritmo de Bhattacharya com 90% de certeza.

Os resultados da classificação foram revisados pelos pesquisadores após algumas visitas ao local [40]. Algumas discordâncias foram constatadas e amostras foram coletadas. Utilizando tais amostras a imagem foi revisada e reclassificada gerando a máscara apresentada na Figura 5.2. Note que os pixels brancos correspondem às regiões de pastagem da Figura 5.1.

Imagem CAFÉ

A imagem CAFÉ é um recorte de 2400×2400 pixels de uma imagem maior, captada pelo satélite SPOT no dia 29 de Agosto de 2005 na região que compreende o município de Monte Santo, no estado de Minas Gerais, tradicional no cultivo de café. A imagem original é uma composição das bandas de infra-vermelho médio, infra-vermelho próximo e vermelho nos canais, R, G e B, respectivamente. Para melhorar a resolução espacial, a imagem foi transformada com o uso de uma imagem pancromática de resolução espacial de 2,5 metros. O recorte que corresponde à imagem CAFÉ pode ser visualizado na Figura 5.3.

A Figura 5.4 corresponde a uma máscara com todas as áreas plantadas com café da região. O processo de reconhecimento e demarcação das regiões de café foi feito de forma manual em todo o município de Monte Santo com a utilização da ISR do satélite SPOT pelos pesquisadores do Centro de Processamento Agrícola (CEPAGRI) da Unicamp.

5.1.2 Medida de Avaliação

Como mencionado anteriormente, o resultado da classificação do método proposto é relacionado com o número de interações. Assim, para avaliar o método, foi utilizado o índice kappa por interações. O kappa [4] é uma medida eficaz para comparar imagens classificadas. Para calculá-lo é necessária a criação de uma matriz de confusão. Uma matriz de confusão é uma matriz quadrada que expressa o número de unidades de amostra indicando o número de acertos e erros das classes especificando os falso-positivos, falso-negativos,



Figura 5.2: Máscara que indica as regiões correspondentes à pastagem na imagem PASTAGEM.

verdadeiro-positivos e verdadeiro-negativos. O índice kappa utiliza todos os elementos da matriz de erros, contemplando, assim, informações sobre todos os tipos de erros.

5.2 Aspectos de Implementação

O sistema proposto foi implementado na linguagem C com os requisitos mínimos para a validação, de acordo com a arquitetura proposta na Figura 3.2. O módulo de *extração* é composto pelos descritores detalhados na Seção 4.1.1. Os demais módulos são apresentados nas seções a seguir: o módulo de *sub-imagem*, na Seção 5.2.1; a Seção 5.2.2 apresenta os módulos de *reconhecimento* e *realimentação de relevância*; a Seção 5.2.3 descreve o módulo de *segmentação*; e, finalmente, módulo de *vetorização* é descrito na Seção 5.2.4. Além disso, a *interação do usuário com o sistema* foi simulada, o que necessitou a criação

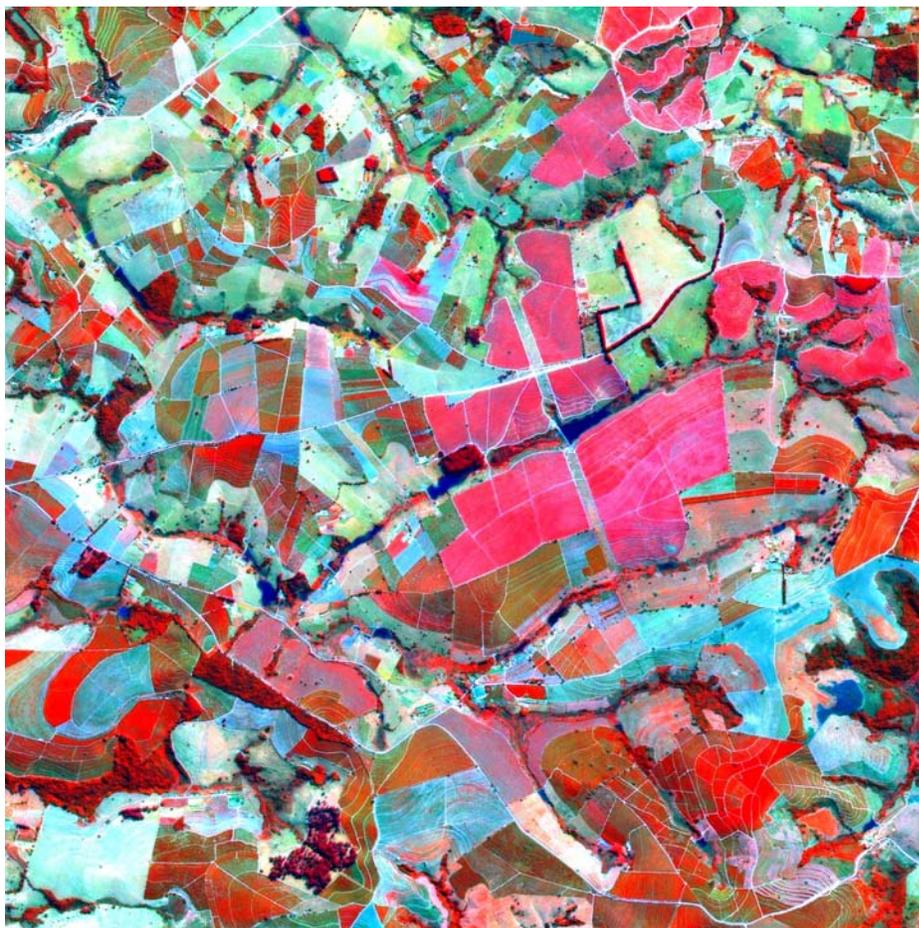


Figura 5.3: Imagem CAFÉ.

da metodologia apresentada na Seção 5.2.5.

5.2.1 Módulo de Partição

O *módulo de partição* é o responsável pelo particionamento da ISR. Além da ISR, um parâmetro de entrada desse módulo é o número de pixels n que as dimensões (altura e largura) da sub-imagem deve ter.

Como mencionado na Seção 3.1, o tamanho da sub-imagem deve ser menor que o tamanho das regiões de interesse, permitindo que as regiões de interesse possam conter várias sub-imagens em seu interior. Entretanto, o tamanho da sub-imagem tem que ser grande o suficiente para que os descritores consigam extrair características de textura relevantes. Considerando esses aspectos, foi definido o número de pixels que as sub-



Figura 5.4: Máscara que indica as regiões correspondentes à café na imagem CAFÉ.

imagens deveriam possuir para os experimentos com a imagem CAFÉ e PASTAGEM, baseando-se (empiricamente) no tamanho que uma propriedade da cultura de interesse comumente possui.

Para a imagem PASTAGEM estimaram-se 16 hectares (corresponde a $16.000 m^2$) como um valor adequado para as áreas de interesse. Embora o tamanho de uma área de pasto possua grandes extensões na criação de gado extensiva, no caso da região da imagem PASTAGEM (microbacia do córrego Laranja Azeda), a pecuária é uma atividade secundária e, portanto, as áreas destinadas à pastagem são pequenas se comparadas a outras regiões do mundo.

Conforme descrito na Seção 5.1.1, a imagem PASTAGEM possui resolução espacial de $20 m$. Então, cada pixel da imagem possui área igual a $400 m^2$. Dividindo a área estimada ($16.000 m^2$), pela área de um pixel ($400 m^2$), temos a quantidade de pixels que uma sub-imagem deve ter ($400 pixels$). Dessa forma, a imagem PASTAGEM foi dividida

Tabela 5.1: Parâmetros do módulo de partição para as imagens PASTAGEM e CAFÉ.

Parâmetro	PASTAGEM	CAFÉ
Área da sub-imagem/talhão (m^2)	16.000	5.625
Tamanho da sub-imagem (px)	20×20	30×30
Número de sub-imagens - n_p	5.980	6.400

em sub-imagens de 20×20 *pixels*, ou seja, $n = 20$ e o número de sub-imagens $n_p = 5980$.

O cálculo do tamanho da sub-imagem para a imagem CAFÉ foi feito de maneira similar. Assumiu-se o valor de $5625 m^2$ para um talhão, uma vez que o café é plantado em pequenas áreas na região da imagem CAFÉ, limitado, principalmente, pela declividade do terreno. Assim, como a resolução espacial da imagem é de $2,5 m$, a sub-imagem deve possuir 900 *pixels*. Assim, as sub-imagens nos experimentos com a imagem CAFÉ possuem 30×30 *pixels* e, portanto, $n = 30$ e o número de sub-imagens $n_p = 6400$.

A Tabela 5.1 resume os parâmetros utilizados para o cálculo do tamanho das sub-imagens nas imagens PASTAGEM e CAFÉ.

5.2.2 Módulos de Reconhecimento de Padrões e Realimentação de Relevância

Os módulos de *reconhecimento de padrões* e *realimentação de relevância* correspondem ao arcabouço GP_{SR} (Seção 3.2.2). O arcabouço GP_{SR} foi implementado utilizando-se a biblioteca *lilgp* [62], que, recentemente, vem sendo usada para modelagem e resolução de problemas com programação genética.

Por se tratar de um arcabouço que utiliza programação genética, vários parâmetros empregados no processo evolutivo precisam ser ajustados (por exemplo, o método de seleção dos indivíduos, a forma de inicialização da população, altura máxima das árvores que representam os indivíduos, dentre outros). Os parâmetros de configuração utilizados pelo módulo de *reconhecimento de padrões* foram baseados nos experimentos realizados por Ferreira em [17]. Estes parâmetros são os mesmos utilizados nos experimentos para validação do GP_{LSP} e podem ser vistos na Tabela 4.1.

5.2.3 Módulo de Segmentação

O módulo de *segmentação* é uma implementação do algoritmo IFT-Watershed [14]. A entrada para esse módulo é um conjunto de marcadores extraídos da classificação realizada pelo arcabouço GP_{SR} e uma imagem gradiente. Por esse motivo, antes de executar esse módulo é preciso extrair os marcadores e criar uma imagem gradiente em níveis de cinza a partir da ISR original.

Foram testadas duas metodologias: *seleção de marcador por sub-imagem inteira* e *seleção de marcador por ponto central*. No caso da *seleção de marcador por sub-imagem inteira*, todos os pixels de uma sub-imagem considerada relevante são utilizados como marcadores no algoritmo *watershed*. Na *seleção de marcador por ponto central* apenas o ponto central da sub-imagem é selecionado para ser marcador.

5.2.4 Módulo de Vetorização

O *módulo de vetorização* é o responsável pela extração dos polígonos referente às regiões da imagem binária gerada pelo módulo de segmentação.

No método proposto, a tarefa do *módulo de vetorização* é executada pela ferramenta Potrace [45]. Embora não seja uma ferramenta criada especificamente para a vetorização de ISRs, o Potrace [45] foi a ferramenta utilizada. Para utilização do Potrace na aplicação-alvo, verificou-se a necessidade de registrar a georreferência dos pontos em um arquivo de metadados. Esse arquivo deve conter as informações padrão de georreferência da ISR original (datum, sistema de projeção, etc) e a georreferência de um dos pontos dos polígonos da imagem vetorizada. Com essas informações é possível saber a georreferência de todos os pontos da representação vetorial.

Nas imagens vetorizadas foram utilizados os parâmetros padrão do Potrace [45], sem suavização de curvas.

5.2.5 Simulação do Usuário

Nos experimentos realizados, o comportamento do usuário é simulado. Para fazer isso, uma imagem das sub-imagens, que denominamos *máscara de relevância* (Definição 6), é feita, foi criada a partir da ISR. No caso da imagem PASTAGEM, a máscara utilizada pode ser vista na Figura 5.2. A máscara utilizada nos experimentos com a imagem CAFÉ pode ser observada na Figura 5.4.

Definição 6 *Máscara de relevância* é uma imagem binária que representa a relevância das sub-imagens.

Suponha a “máscara” da ISR particionada da mesma forma que a ISR. Na *máscara de relevância*, cada pixel representa uma sub-imagem da “máscara” da ISR. Uma sub-imagem é representada por um pixel com o valor ‘1’ se o número de pixels relevantes (valor ‘1’) na sub-imagem é maior que um índice percentual pré-definido. A Figura 5.5 apresenta as *máscaras de relevância* para as imagens PASTAGEM e CAFÉ com índice percentual igual a 50%. Note que um pixel da máscara de relevância possui valor ‘1’ se a quantidade de pixels relevantes na sub-imagem que ele representa for maior que 50%.

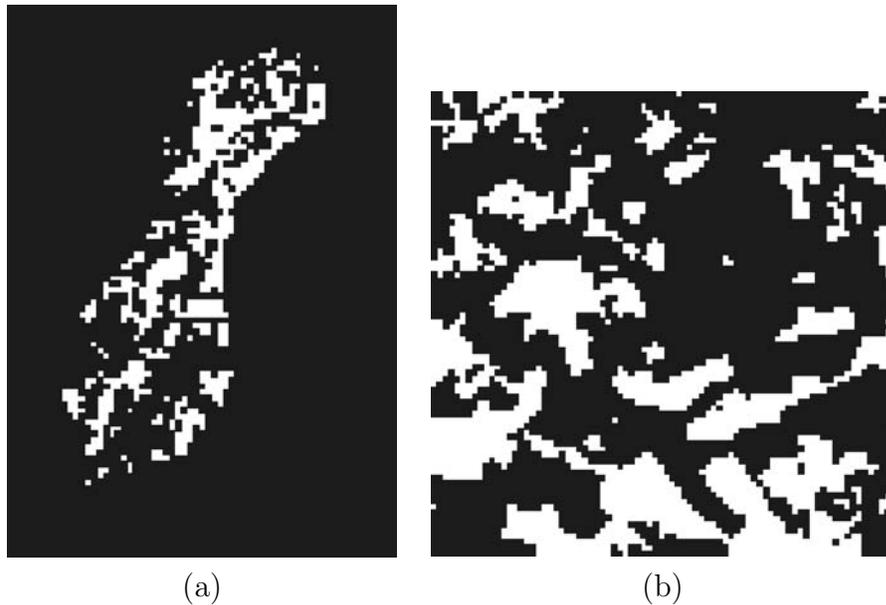


Figura 5.5: *Máscaras de relevância* para as imagens (a) PASTAGEM e (b) CAFÉ com percentual de relevância igual a 50%.

A realimentação de relevância é simulada utilizando a *máscara de relevância* ao invés das indicações do usuário. Para isso, nos momentos em que o algoritmo de realimentação mostra as imagens para que o usuário indique as sub-imagens de interesse, a indicação de relevância é feita verificando-se o valor das sub-imagens na *máscara de relevância*.

5.3 Metodologia de Validação

Os experimentos realizados visaram identificar a melhor combinação de parâmetros que afeta o método de vetorização semi-automática proposto. Os principais parâmetros são:

- **Valores de corte nas listas ordenadas de sub-imagens.** O arcabouço GP_{SR} gera uma lista das sub-imagens ordenadas pela similaridade com relação à sub-imagem de consulta. Desse modo, é necessário definir limites na lista para separar quais marcadores serão utilizados como pertencentes à classe de interesse. Os experimentos realizados visam analisar quais os melhores limiares. Desse modo, para cada experimento o “percentual de relevantes” e “percentual de não relevantes” são dois parâmetros que correspondem ao percentual de sub-imagens do topo da lista que serão indicadas como relevantes e o percentual de sub-imagens nas últimas posições da lista que serão indicadas como não relevantes, respectivamente.

- **Metodologia de seleção de marcadores.** Foram testadas as seguintes metodologias: *seleção de marcador por sub-imagem inteira* e *seleção de marcador por ponto central*. Por utilizar todos os pixels das sub-imagens relevantes, na *seleção de marcador por sub-imagem inteira*, a *máscara de relevância* deve ser criada com índice percentual igual a ‘1’. Na *seleção de marcador por ponto central*, entretanto, a *máscara de relevância* pode ser definida com índices percentuais menores que ‘1’, por não utilizar todos os pixels da sub-imagem relevante como marcador. A vantagem de usar índices percentuais menores é que mais regiões podem ser identificadas no processo de classificação. Entretanto, essa abordagem permite a inserção de erros na classificação, uma vez que a semente selecionada pode não pertencer a uma região relevante. O objetivo dos testes foi avaliar o impacto desses erros na classificação final.
- **Método de criação da imagem gradiente.** Foram testados dois métodos: “função morfológica simples” e o gradiente de Sobel com suavização por uma máscara gaussiana [20].

Os resultados apresentados nesta seção foram obtidos por meio de experimentos realizados nas imagens PASTAGEM e CAFÉ descritos em detalhe na Seção 5.1.1. Para cada ISR foram selecionadas 40 sub-imagens diferentes para padrão inicial de consulta. Foi utilizada uma máquina com dois processadores Xeon Quad Core e 4 GB de memória RAM.

Para a imagem PASTAGEM foram consideradas 8 configurações diferentes para combinar os parâmetros avaliados. A Tabela 5.2 apresenta os possíveis valores assumidos para cada parâmetro. O campo “Sel. de marcadores” representa o método de seleção utilizado para selecionar marcadores. O campo “Gradiente” representa a técnica utilizada para criar a imagem gradiente da ISR. O campo “MR” é o valor percentual de pixels para a sub-imagem ser considerada relevante na *máscara de relevância*. Os dois campos mais à direita da tabela são relacionados aos valores de corte da lista de sub-imagens ordenada.

Observe que nos testes com *seleção de marcadores por sub-imagem inteira*, os percentuais de sub-imagens relevantes testados possuem valores menores do que nos testes com *seleção de marcadores por ponto central*. Isso acontece por causa dos índices percentuais utilizados para a criação da *máscara de relevância*. Como o número de sub-imagens com 100% dos pixels pertencentes a uma região de interesse é uma quantidade pequena, a *máscara de relevância* com índice de relevância igual a ‘1’ apresentará um número menor de sub-imagens marcadas como relevante do que os criados com índices de relevância menores.

Para a imagem CAFÉ foram consideradas 4 configurações diferentes para combinar os parâmetros avaliados. A Tabela 5.3 apresenta os possíveis valores assumidos para cada

Tabela 5.2: Testes realizados para a imagem PASTAGEM.

Teste	Seleção de marcadores	Gradiente	MR	Percentual de relevantes	Percentual de não relevantes
1	sub-imagem inteira	morfológico	100%	1,0%; 2,5% e 5,0%	20,0%; 40,0%; e 60,0%
2	sub-imagem inteira	Sobel	100%	1,0%; 2,5% e 5,0%	20,0%; 40,0%; e 60,0%
3	ponto central	morfológico	90%	2,5%; 5,0% e 7,5%	5,0%; 7,5%; 10,0% e 20,0%
4	ponto central	morfológico	70%	2,5%; 5,0% e 7,5%	5,0%; 7,5%; 10,0% e 20,0%
5	ponto central	morfológico	50%	2,5%; 5,0% e 7,5%	5,0%; 7,5%; 10,0% e 20,0%
6	ponto central	Sobel	90%	2,5%; 5,0% e 7,5%	5,0%; 7,5%; 10,0% e 20,0%
7	ponto central	Sobel	70%	2,5%; 5,0% e 7,5%	5,0%; 7,5%; 10,0% e 20,0%
8	ponto central	Sobel	50%	2,5%; 5,0% e 7,5%	5,0%; 7,5%; 10,0% e 20,0%

parâmetro.

No caso da imagem CAFÉ, os resultados experimentais com *seleção de marcadores por ponto central* utilizando máscaras de relevância diferentes de 100% apresentaram resultados iniciais muito ruins e por isso, foram desconsiderados. Isso pode ser atribuído ao fato da imagem CAFÉ ser de alta-resolução, facilitando, dessa forma, a seleção de marcadores não relevantes como relevantes ou vice-versa.

Tabela 5.3: Testes realizados para a imagem CAFÉ.

Teste	Seleção de marcadores	Gradiente	MR	Percentual de relevantes	Percentual de não relevantes
1	sub-imagem inteira	morfológico	100%	1,0%; 2,5% e 5,0%	20,0%; 40,0%; e 60,0%
2	sub-imagem inteira	Sobel	100%	1,0%; 2,5% e 5,0%	20,0%; 40,0%; e 60,0%
3	ponto central	morfológico	100%	2,5%; 5,0% e 7,5%	20,0%; 40,0%; e 60,0%
4	ponto central	Sobel	100%	2,5%; 5,0% e 7,5%	20,0%; 40,0%; e 60,0%

5.4 Resultados Experimentais

Nesta seção são apresentados os resultados dos experimentos descritos na Seção 5.3. Nos gráficos das figuras que apresentam os resultados para os testes dos parâmetros das imagens PASTAGEM e CAFÉ (Figuras 5.6 a Figuras 5.13 e Figuras 5.17 a Figuras 5.21, respectivamente), cada curva representa os valores de kappa para cada iteração, onde “X/Y” representa os valores de corte testados, sendo X o índice de sub-imagens consideradas relevantes e Y o índice de sub-imagens não relevantes consideradas.

5.4.1 Imagem PASTAGEM

As Figuras 5.6 a 5.13 apresentam as curvas kappa-iterações para os parâmetros dos testes contidos na Tabela 5.2 (testes 1 a 8). Foram testadas todas as combinações de valores de corte para os percentuais de relevantes e não relevantes considerados na Tabela 5.2. Entretanto, para facilitar a visualização e a interpretação dos gráficos, as curvas com a maior parte dos valores kappa abaixo de 0,6 foram desconsideradas.

No gráfico da Figura 5.6, correspondente ao Teste 2 da Tabela 5.2, é possível observar que o melhor valor de corte é 2,5% para as mais relevantes da lista de sub-imagens com os parâmetros do Teste 1 da Tabela 5.2. No caso do percentual para a seleção das menos relevantes, não houve um limiar que pode ser considerado mais significativo. Entretanto, é possível observar que utilizar 20,0% das menos relevantes não é um bom corte. Uma razão seria porque o número de sub-imagens não relevantes deve fornecer um número muito maior de marcadores. Isso evitaria que os marcadores de relevância não invadam com facilidade as regiões que não são de interesse.

Os gráficos da Figura 5.7, referentes ao Teste 2 da Tabela 5.2, apresentam resultados similares aos da Figura 5.6 com relação aos valores de corte. Entranto, como é possível observar, as curvas ficaram próximas considerando diferentes valores de corte. Dessa forma, pode-se considerar que a aplicação do algoritmo de Sobel com suavização minimiza os impactos de um valor de corte inadequado.

Embora a configuração dos parâmetros do Teste 3 sejam diferentes dos Testes 1 e 2, os resultados apresentados foram bastante parecidos, como pode ser observado nos gráficos da Figura 5.8. Pode-se considerar, dessa forma, que o método de seleção de marcadores não influencia significativamente nos melhores resultados obtidos. Observe que para alguns valores de corte, os valores do kappa tiveram oscilações bruscas. Isso pode ter ocorrido devido ao fato de que tais valores de corte estão muito distantes dos valores ideais. Dessa forma, mesmo que o algoritmo de aprendizado consiga encontrar um número bom de sub-imagens relevantes, muitas sub-imagens não relevantes estão sendo utilizadas como marcadores de relevância. Isso inclui marcadores errados no processo de segmentação, gerando uma imagem com índice kappa baixo. Mesmo que o algoritmo

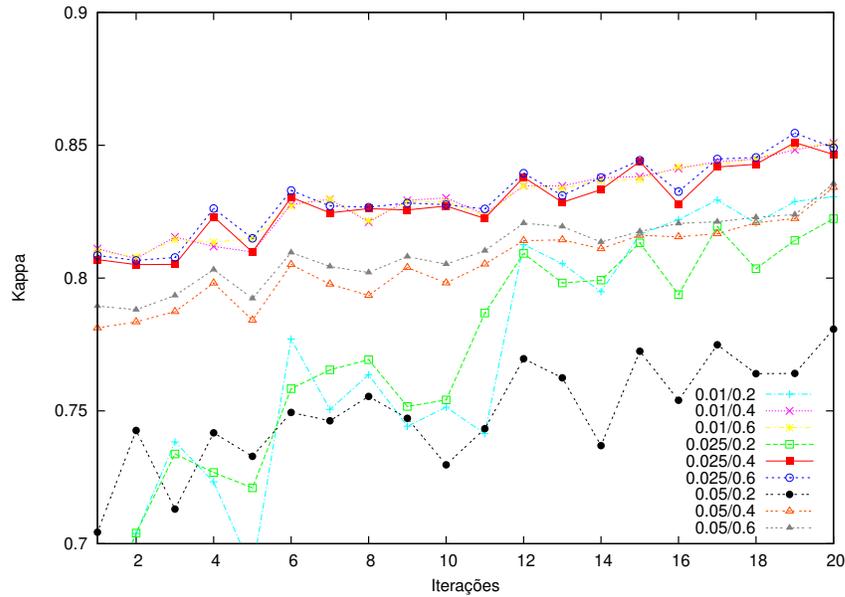


Figura 5.6: Curvas Kappa x Iterações para os parâmetros do Teste 1 da Tabela 5.2.

de aprendizado tenha encontrado um bom conjunto de indivíduos, como o *feedback* do usuário simulado foi ruim (devido ao kappa baixo), na próxima iteração o sistema tende a testar novos indivíduos, fazendo com que os resultados oscilem.

Nos gráficos das Figuras 5.9 e 5.10 (referente aos Testes 4 e 5 da Tabela 5.2, respectivamente) observa-se que os valores de corte mais adequados são parecidos com os Testes 1e 2. Comparando os resultados dos Testes 3, 4, e 5 (Figuras 5.8, 5.9 e 5.10, respectivamente) é possível observar que quanto menor o índice de relevância da máscara de relevância (ver Seção 5.2.3), menor é o impacto da realimentação de relevância no índice kappa. É possível concluir, então, que utilizar máscaras de relevância inadequada acrescenta tantos marcadores errados no processo de segmentação a ponto de anular ou diminuir o impacto das interações do usuário com o sistema.

Os gráficos das Figuras 5.11, 5.12 e 5.13, que correspondem aos testes com seleção de marcadores por ponto central utilizando o detector de contornos de Sobel com suavização gaussiana, apresentaram resultados similares aos testes de seleção por ponto central com gradiente morfológico (Figuras 5.8, 5.9 e 5.10). Isso nos permite concluir que não houve impacto nos resultados com relação à técnica de criação da imagem gradiente utilizada.

Nos Testes 1 e 2, que envolvem *seleção de marcadores por sub-imagem inteira*, os valores de corte 2,5% para relevantes e 0,6% para não relevantes se mostraram mais adequados, como pode ser observado nas curvas de kappa-iteração das Figuras 5.6 e 5.7.

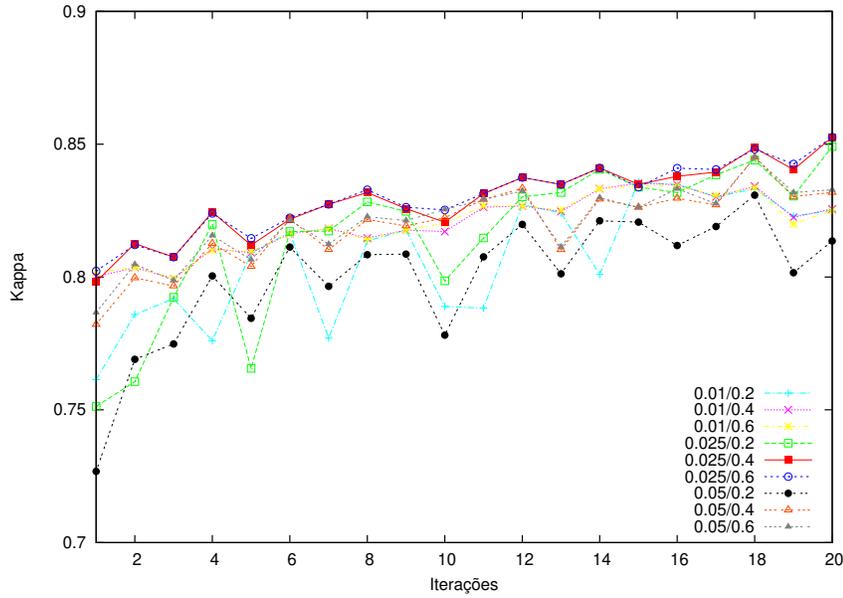


Figura 5.7: Curvas Kappa x Iterações para os parâmetros do Teste 2 da Tabela 5.2.

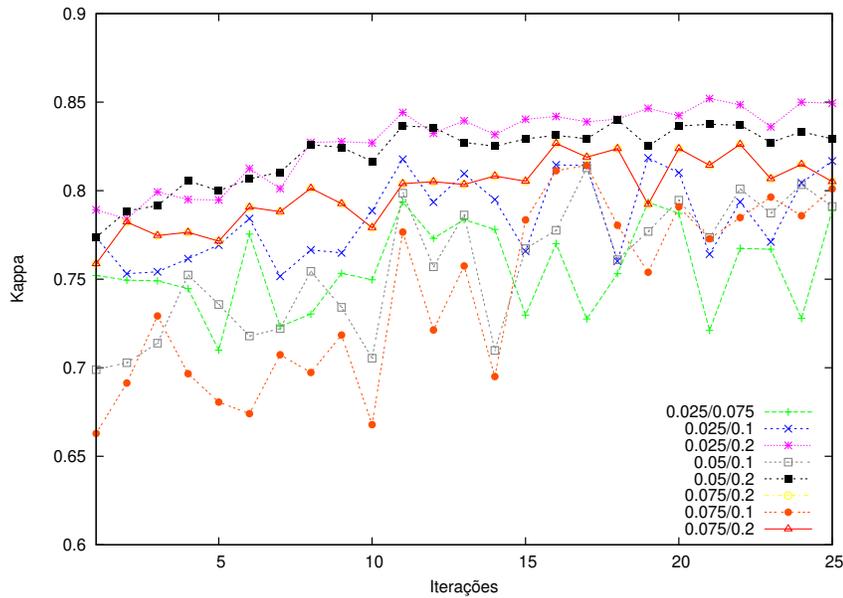


Figura 5.8: Curvas Kappa x Iterações para os parâmetros do Teste 3 da Tabela 5.2.

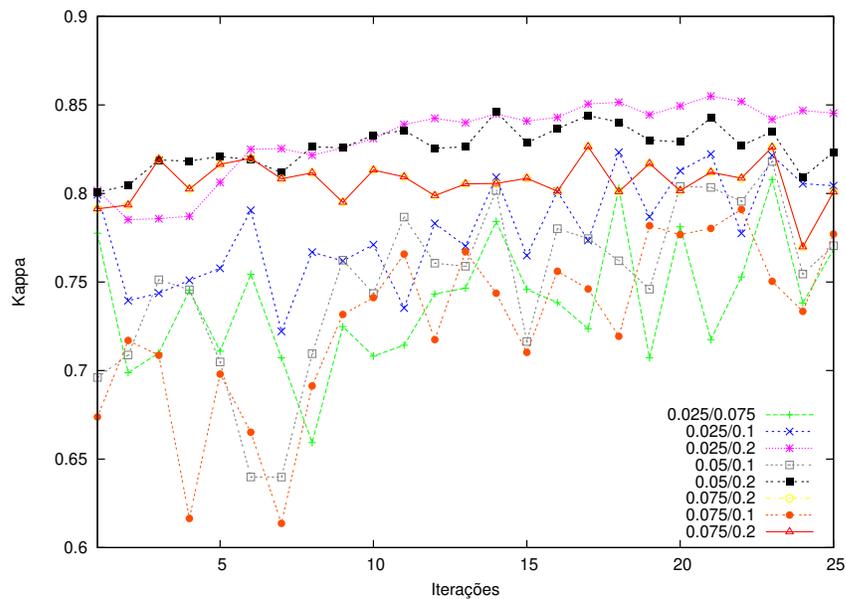


Figura 5.9: Curvas Kappa x Iterações para os parâmetros do Teste 4 da Tabela 5.2.

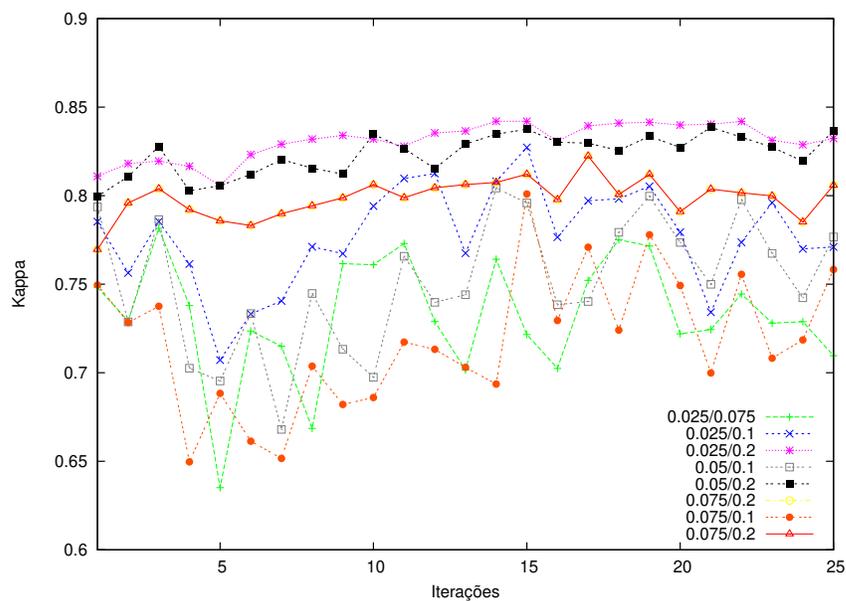


Figura 5.10: Curvas Kappa x Iterações para os parâmetros do Teste 5 da Tabela 5.2.

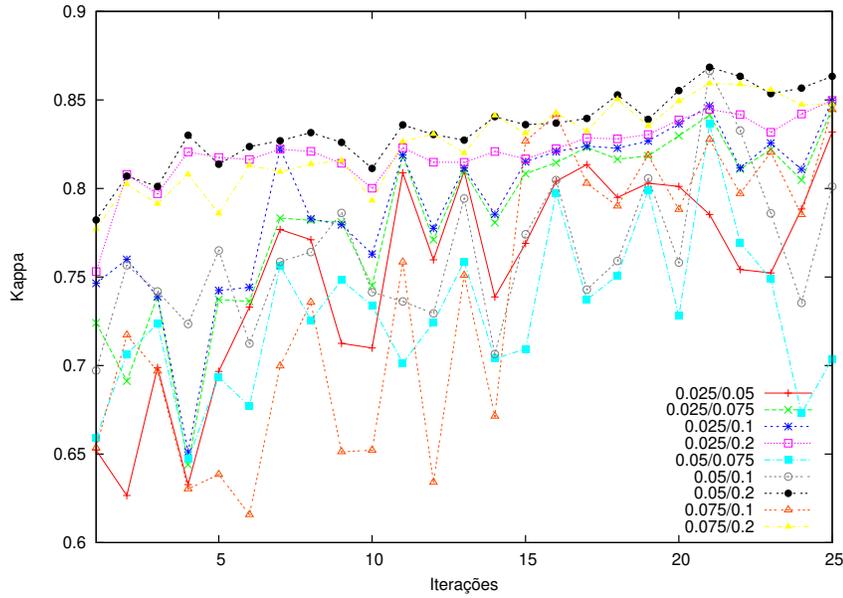


Figura 5.11: Curvas Kappa x Iterações para os parâmetros do Teste 6 da Tabela 5.2.

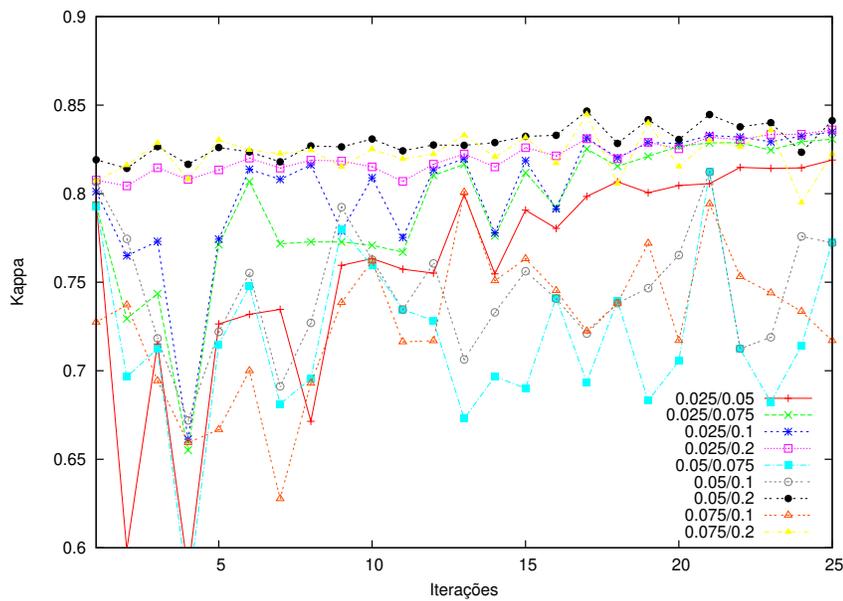


Figura 5.12: Curvas Kappa x Iterações para os parâmetros do Teste 7 da Tabela 5.2.

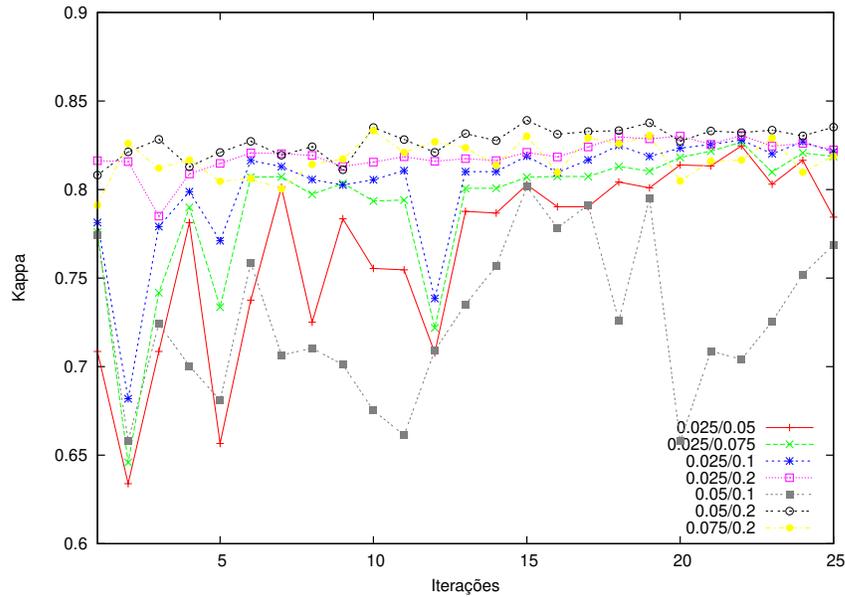


Figura 5.13: Curvas Kappa x Iterações para os parâmetros do Teste 8 da Tabela 5.2.

Como pode ser observado na Figura 5.14, comparando os melhores resultados dos dois testes, observa-se que não houve grandes diferenças com relação ao método de criação do gradiente aplicado.

Nos testes que envolvem *seleção de marcadores por ponto central* (Testes 3 a 8), aqueles com *máscaras de relevância* de 0,7 e 0,9 apresentaram melhores resultados. Os melhores valores de corte variam bastante, sendo diretamente relacionados aos índices das *máscaras de relevância*. De forma geral, os valores de corte 2,5% e 5% para relevantes e 20% para não relevantes apresentaram melhores resultados. A Figura 5.15 mostra as melhores curvas para os testes com gradiente morfológico e Sobel. Embora o gradiente por Sobel apresente resultados melhores nas primeiras iterações, no geral não houve diferenças significativas.

Entre os métodos de seleção de marcadores, os testes mostram que ambos podem chegar a bons resultados, como mostra a Figura 5.16.

Após todos os testes para a imagem PASTAGEM, pode-se concluir, de modo geral, que os resultados da classificação não são significativamente afetados pelos métodos de seleção e os métodos de criação de gradiente testados. Os valores de corte na lista de sub-imagens e a *máscara de sub-imagem*, entretanto, são parâmetros que influenciam diretamente na qualidade dos resultados. Com relação ao impacto das interações do usuário é possível observar que o valor do índice kappa aumenta significativamente na medida em que o

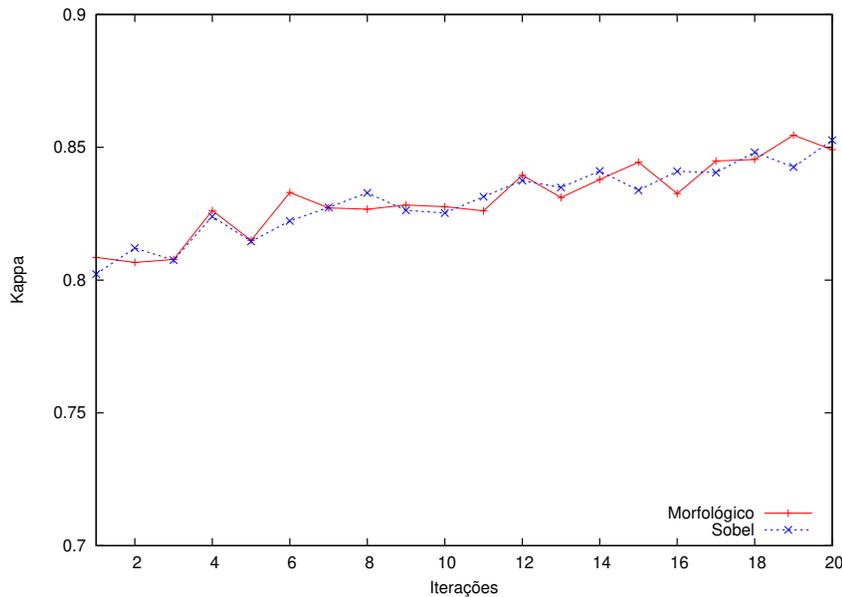


Figura 5.14: Curvas Kappa x Iterações comparando os resultados para o operador morfológico e de Sobel com a seleção de sementes por sub-imagem inteira na imagem PASTAGEM.

usuário interage com o sistema.

É importante salientar que os resultados foram obtidos a partir da média de 40 sub-imagens de consulta, selecionadas aleatoriamente. Para algumas sub-imagens de consulta, os resultados foram muito melhores. Isso significa que a sub-imagem de consulta tem um impacto grande no resultado da classificação e que, se escolhida adequadamente, pode melhorar significativamente os resultados.

5.4.2 Imagem CAFÉ

Analisando as curvas Kappa x Iterações de todos os testes realizados (Figuras 5.17 a 5.21), é possível observar que o melhor valor de corte é 2, 5% para as sub-imagens relevantes e 20% e 60% para as não relevantes. É importante lembrar que o valor de corte é diretamente relacionado à quantidade de sub-imagens relevantes que existem na imagem. Então, o melhor valor de corte deve ser aquele que chega mais próximo ao número de sub-imagens relevantes da *máscara de relevância*.

Nos gráficos das Figuras 5.17 e 5.18, que correspondem aos testes com seleção de marcadores por sub-imagem inteira (Testes 1 e 2, da Tabela 5.3), é possível observar que o índice kappa piora na medida em que são feitas mais iterações, além de oscilar

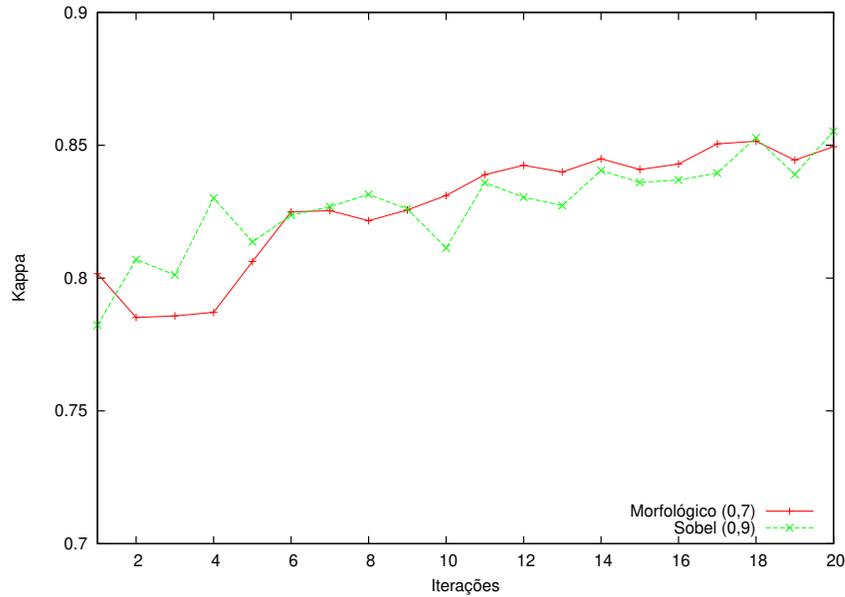


Figura 5.15: Curvas Kappa x Iterações comparando os resultados para o operador morfológico e de Sobel com seleção de sementes por ponto central na imagem PASTAGEM.

bastante. Uma hipótese provável é a de que, ao longo das iterações, os marcadores estejam extrapolando com facilidade as bordas de suas regiões e, conseqüentemente, rotulando regiões incorretas. Esse fator pode estar relacionado com o fato da imagem possuir alta-resolução, ou seja, muitos detalhes. Um marcador que indica relevância ganha uma região de não-relevância se não houver marcadores de não-relevância próximo à região em disputa. Isso fica mais evidente quando analisamos a máscara de relevância que corresponde à imagem de café (Figura 5.4). Observe os espaços entre os polígonos que representam regiões de interesse. Esses espaços dificilmente serão rotulados como “não-relevantes”, pois, para isso, seria necessário que houvesse um marcador de não relevância próximo à essa região. É importante salientar, entretanto, que o problema não é o fato da imagem possuir alta-resolução (esse fator na verdade é positivo, uma vez que imagens de alta-resolução fornecem mais informação para o sistema) e sim o fato de que o método de seleção de marcadores testado não ser adequado.

Nos gráficos das Figuras 5.19 e 5.21, é possível observar que, os valores do índice kappa também oscilem e decaiam ao longo das iterações. Observa-se que para a iteração 21 no caso do Teste 3 (Figura 5.19) e 18, no caso do Teste 4 (Figura 5.21) há um aumento do índice kappa.

A Figura 5.21 apresenta as melhores curvas para cada um dos testes realizados para

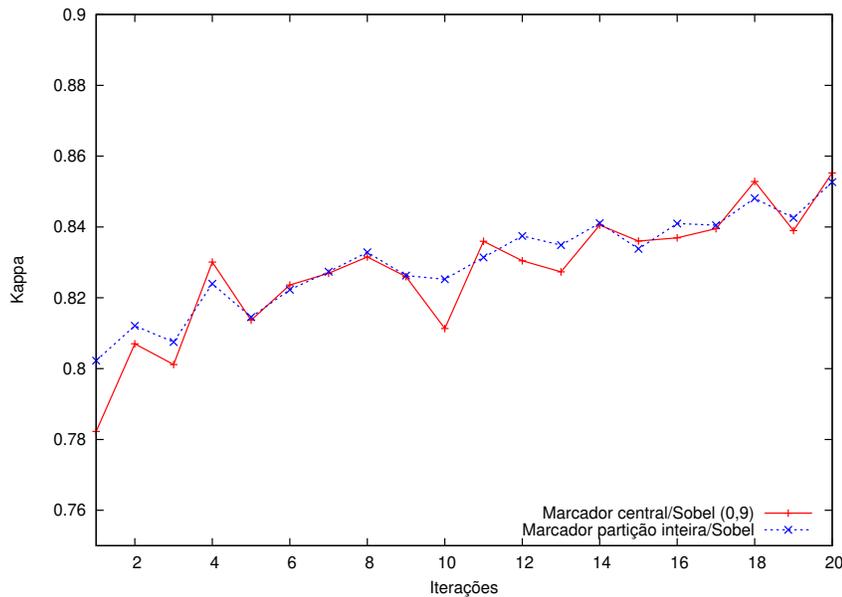


Figura 5.16: Curvas Kappa x Iterações comparando as melhores curvas para os métodos de seleção de marcadores na imagem PASTAGEM.

a imagem CAFÉ.

Na Figura 5.21 é possível observar que os testes com *seleção de marcadores por ponto central* apresentaram melhores resultados que *seleção de marcadores por sub-imagem inteira*. Isso pode ser atribuído ao fato de a imagem CAFÉ ser de alta resolução. As separações entre os talhões são bem representadas na máscara (Figura 5.4). Quando uma sub-imagem inteira é utilizada como marcador, as regiões não relevantes entre os talhões podem ser “inundadas”, extrapolando os limites da região de interesse.

Com relação ao método de criação da imagem gradiente, o *operador morfológico simples* apresentou resultados um pouco melhores, embora o *operador de Sobel* com suavização tenha apresentado um bom resultado com 18 iterações. Isso pode ser atrelado ao fato da suavização, em algumas partes da imagem, reduzir muito as bordas da região, fazendo com que alguns marcadores extrapolem os limites das regiões de interesse.

5.4.3 Comparação com o método MaxVer

O método proposto nesta dissertação de mestrado foi comparado com MaxVer (*Maximum Likelihood Classification*) [46]. MaxVer é o método de classificação supervisionado mais utilizado com dados de imagens de sensoriamento remoto. É considerado um algoritmo paramétrico e assume uma distribuição estatística de classe particular, em geral a distri-

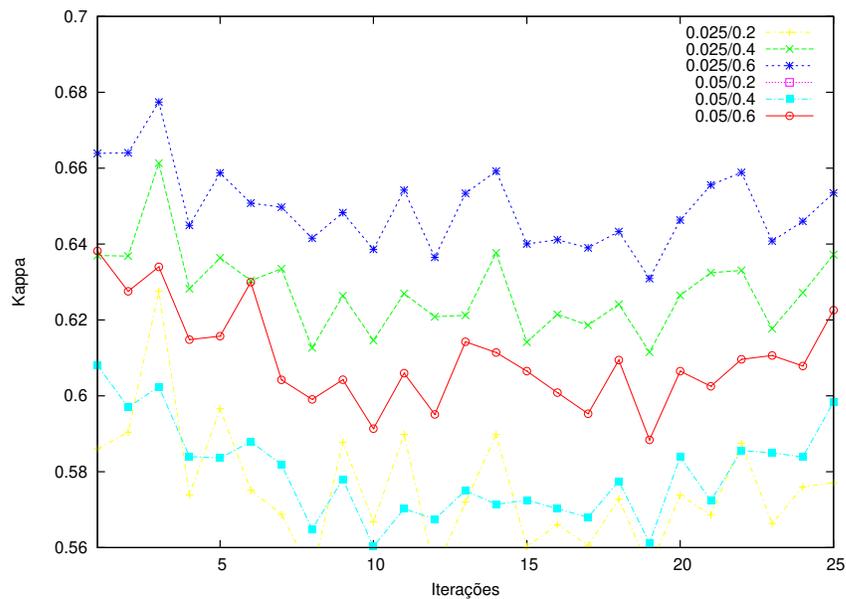


Figura 5.17: Curvas Kappa x Iterações para os parâmetros do Teste 1 da Tabela 5.3.

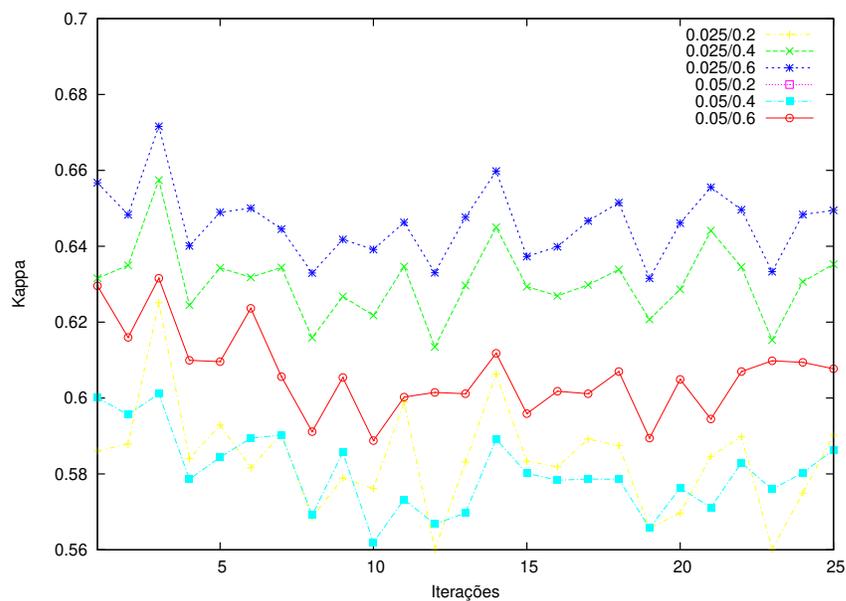


Figura 5.18: Curvas Kappa x Iterações para os parâmetros do Teste 2 da Tabela 5.3.

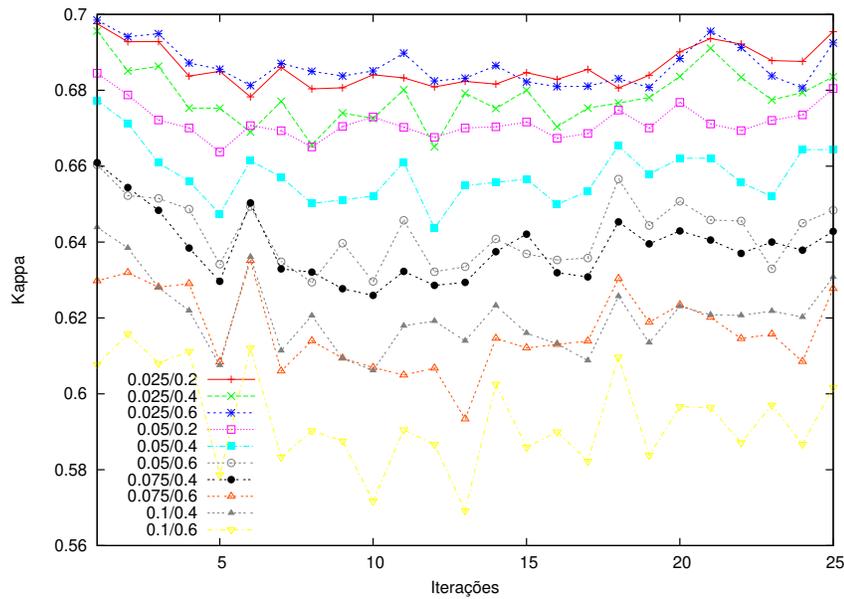


Figura 5.19: Curvas Kappa x Iterações para os parâmetros do Teste 3 da Tabela 5.3.

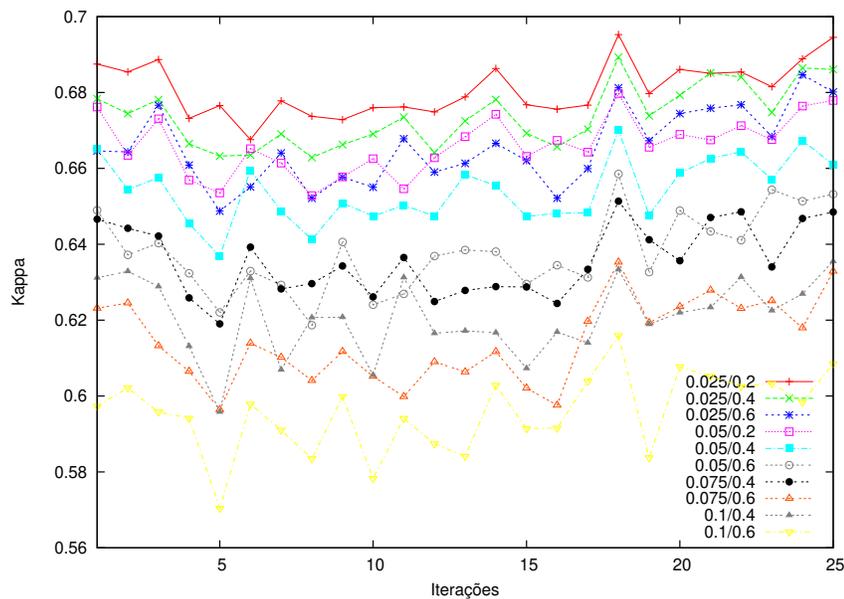


Figura 5.20: Curvas Kappa x Iterações para os parâmetros do Teste 4 da Tabela 5.3.

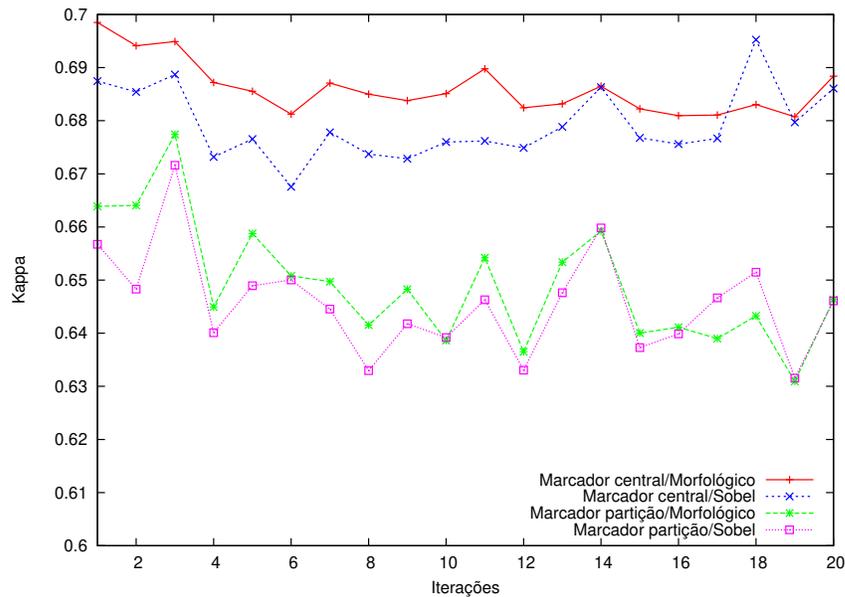


Figura 5.21: Curvas Kappa x Iterações comparando as melhores curvas para cada um dos testes realizados para a imagem CAFÉ.

buição normal, para usar no algoritmo de classificação.

A implementação do algoritmo MaxVer exige o cálculo da probabilidade de cada pixel pertencer a cada classe definida. Cada pixel é, então, classificado como pertencente à classe de maior probabilidade. A imagem PASTAGEM foi classificada pelo MaxVer com o limitante de probabilidade de 0,8 usando 20.580 pontos de amostra de pasto. A imagem CAFÉ foi classificada com limitante de probabilidade de 0,98, usando 43.630 pontos de amostra de café.

A Figura 5.22 ilustra a curva kapa-iterações do método proposto em comparação com o valor do kapa encontrado pelo MaxVer para a imagem PASTAGEM. Observe que o valor de kapa é sempre maior no método proposto que o calculado pelo MaxVer, aumentando ao longo das iterações.

A Figura 5.23 ilustra a curva kapa-iterações do método proposto em comparação com o valor do kapa encontrado pelo MaxVer para a imagem CAFÉ. Os valores de kapa para o método proposto foram melhores.

5.5 Resultados da Vetorização

Esta seção apresenta o resultado das imagens vetorizadas pelo método proposto.

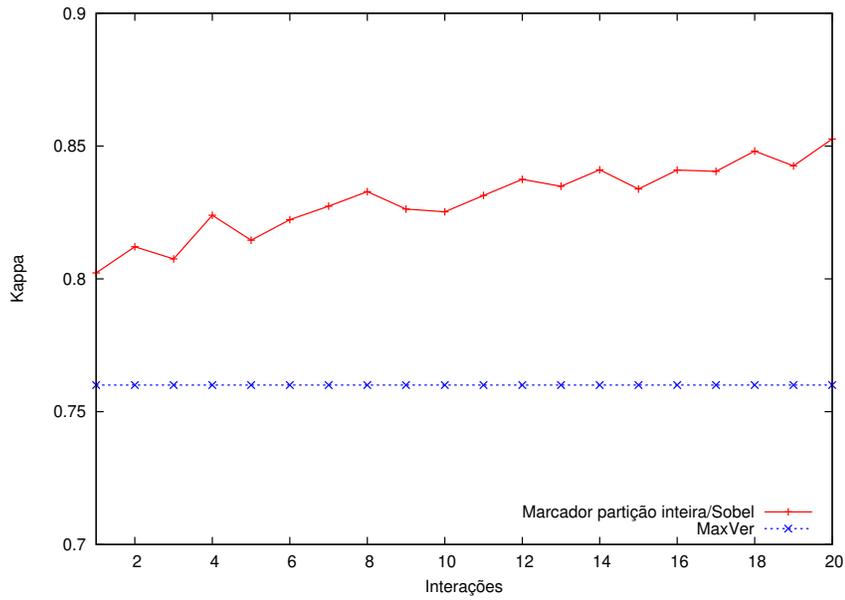


Figura 5.22: Curvas Kappa x Iterações comparando o resultado do método proposto com o MaxVer para a imagem PASTAGEM.

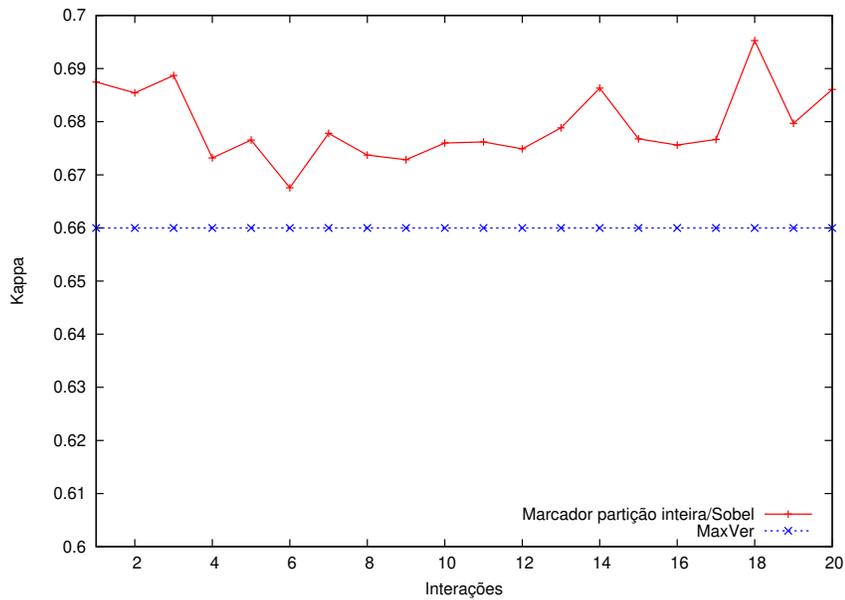


Figura 5.23: Curvas Kappa x Iterações comparando o resultado do método proposto com o MaxVer para a imagem CAFE.

A Figura 5.24 apresenta o resultado da vetorização da imagem PASTAGEM considerando as 5% melhores sub-imagens da lista ordenada como relevantes e as 20% piores como não relevantes. Foi aplicada a *seleção de marcador por sub-imagem inteira* com o gradiente a partir do operador de Sobel com duas suavizações gaussianas. Os polígonos em preto representam as regiões de pastagem encontradas pelo método proposto.



Figura 5.24: Resultado da vetorização da imagem PASTAGEM pelo método proposto.

A Figura 5.25 ilustra as áreas classificadas pelo método proposto (em vermelho) sobre as áreas da máscara de PASTAGEM (classificadas manualmente). Observe que grande parte das áreas relevantes (brancas) foram identificadas pelo método.

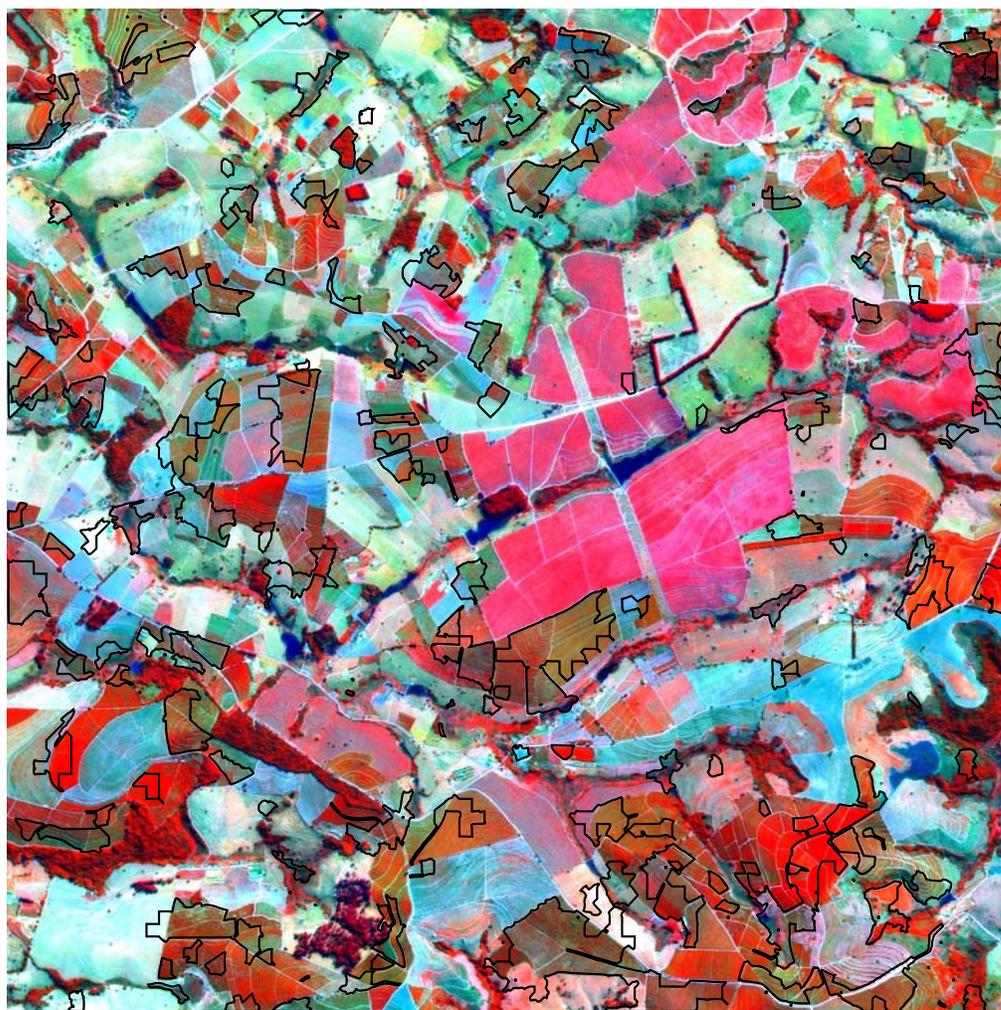


Figura 5.25: Resultado da vetorização sobrepondo a máscara de pastagem.

A Figura 5.26 apresenta o resultado da vetorização da imagem CAFÉ considerando as 2,5% melhores sub-imagens da lista ordenada como relevantes e as 60% piores como não relevantes. Foi aplicada a *seleção de marcador por ponto central* com o gradiente a partir do operador de Sobel com duas suavizações gaussianas. Os polígonos em preto representam as regiões de café encontradas pelo método proposto.

A Figura 5.27 ilustra as áreas classificadas pelo método proposto (em vermelho) sobre as áreas da máscara de CAFÉ (classificadas manualmente). Observe que, embora parte das áreas relevantes (em branco) não tenham sido identificadas pelo método, cada “bloco” de áreas relevantes possui uma área identificada. Isso indica que os marcadores utilizados para a segmentação não foram adequados uma vez que embora estejam cor-

retos, não alcançaram todas as regiões de relevância nas proximidades. Outra hipótese é a de que marcadores de não relevância muito próximos às áreas de relevância tenham sido selecionados, fazendo com que essas regiões relevantes fossem classificadas como não relevantes.



Sub-imagem de Consulta (ampliada)

Figura 5.26: Resultado da vetorização da imagem CAFÉ pelo método proposto.

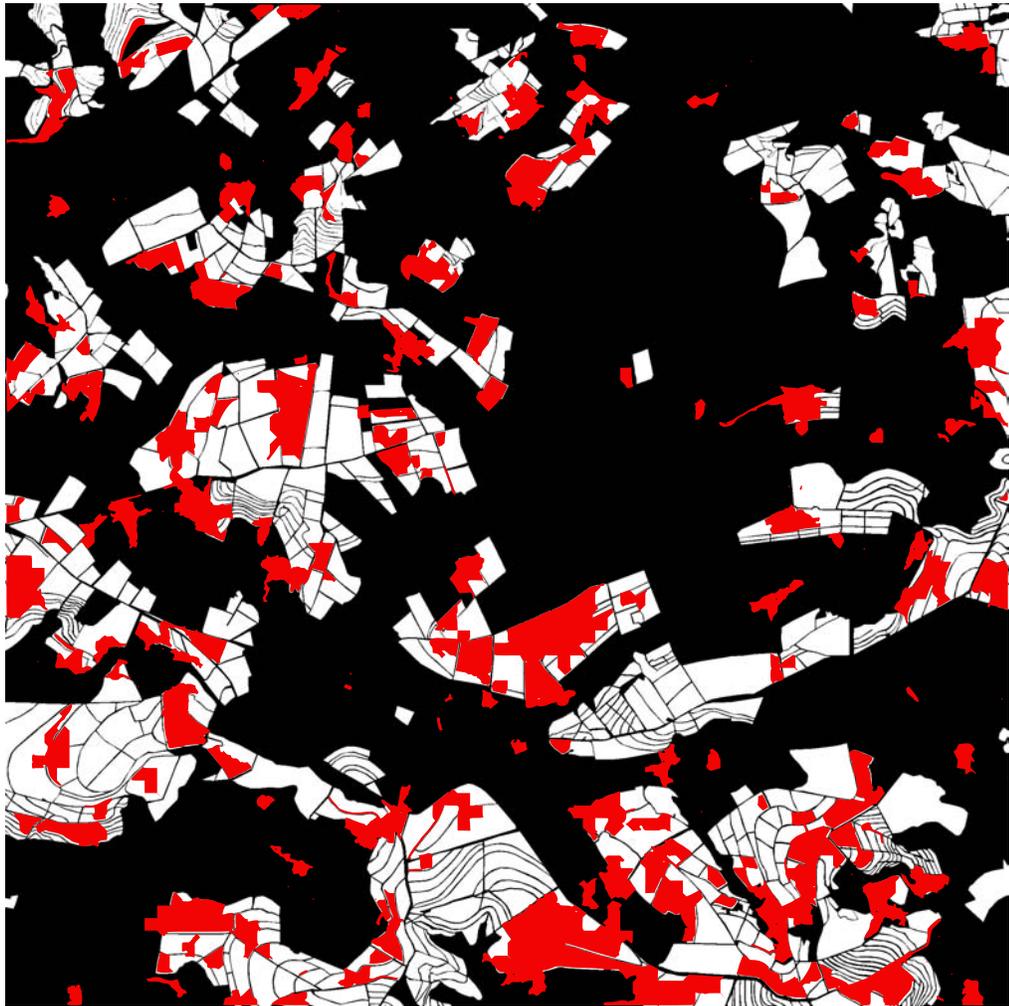


Figura 5.27: Resultado da vetorização sobrepondo a máscara de café.

Capítulo 6

Conclusões e Trabalhos Futuros

6.1 Conclusões

Imagens de Sensoriamento Remoto (ISRs) são uma importante fonte de informação para aplicações de uso agrícola. Entretanto, algumas aplicações necessitam que as ISRs sejam vetorizadas. Além disso, as características das ISRs tornam o problema de reconhecimento de regiões difícil. Desse modo, este trabalho apresentou um método de reconhecimento e vetorização semi-automático de ISRs.

Foi proposto um método para extração de regiões de interesse em Imagens de Sensoriamento Remoto no “formato vetorial” composto por quatro etapas principais: sub-imagem e extração de características da imagem, identificação das regiões de interesse, segmentação da imagem e vetorização das regiões encontradas. O módulo de identificação das regiões de interesse foi adaptado de um arcabouço criado para recuperação de imagens por conteúdo. O sistema utiliza descritores de imagens para codificar as características espectrais e de textura das ISRs, realimentação de relevância (para o usuário melhorar o padrão de consulta por meio de interações) e programação genética para combinar os descritores e agregar as contribuições obtidas das interações do usuário.

O arcabouço proposto para recuperação de imagens por conteúdo baseado em características locais, o GP_{LSP} , apresentou bons resultados, quando comparado ao método LAP .

O método proposto para vetorização semi-automática apresentou bons resultados com relação à imagem PASTAGEM, superando os resultados obtidos pelo algoritmo MaxVer. Observando as regiões identificadas é possível ver que houve um bom acerto das áreas de interesse. Nos testes com a imagem CAFÉ, os resultados mostraram diversos pontos que podem ser melhorados no método proposto, como melhorar a seleção de sementes, por exemplo. Ainda assim os resultados foram melhores do que a classificação realizada pelo MaxVer.

As principais contribuições são:

- Estudo comparativo de técnicas de vetorização de imagens *raster*;
- Adaptação do modelo de recuperação de imagens por conteúdo proposto por Ferreira para realização de realimentação de relevância usando regiões de imagem [43];
- Adaptação do modelo de realimentação de relevância para o reconhecimento de regiões em ISRs [44].
- Implementação parcial de um sistema de reconhecimento semi-automático e vetorização de regiões em ISRs;
- Proposta de metodologia de validação do sistema desenvolvido [44].

6.2 Extensões

Existem diversos tópicos que podem ser aprimorados para melhorar os resultados obtidos no processo de classificação das ISRs. As principais extensões deste trabalho podem ser agrupadas em cinco temas principais: estrutura do particionamento (Seção 6.2.1); descritores de imagens (Seção 6.2.2); padrões de consulta (Seção 6.2.3); realimentação de relevância (Seção 6.2.4) e; segmentação e classificação dos pixels (Seção 6.2.5). Além disso, o método proposto pode ser comparado a outros métodos (Seção 6.2.6).

6.2.1 Estrutura do Particionamento

No método proposto nesta dissertação, a imagem a ser classificada é dividida em sub-imagens regulares de mesmo tamanho. Essa abordagem visa preservar as características de textura das regiões de interesse e evitar que áreas de transição entre as regiões interfiram diretamente nessa fase do processo de segmentação, permitindo maior controle na fase de segmentação.

O tamanho que as sub-imagens possuem é definido empiricamente, baseado no tamanho estimado que um talhão da cultura agrícola que se deseja identificar na imagem normalmente possui. Um problema em aberto é encontrar uma forma automática de estimar o tamanho das sub-imagens. Uma alternativa seria inserir o tamanho da sub-imagem como uma variável no processo de classificação utilizando programação genética. Há ainda a possibilidade de se trabalhar com outras estruturas para representar as sub-imagens, como as *quadtrees* [18]. Uma estrutura como essa possibilitaria representar as sub-imagens em tamanhos diferentes, agrupando um número maior de pixels dentro das áreas de interesse e permitindo refinar a classificação das sub-imagens nas bordas das regiões de interesse.

6.2.2 Descritores de Imagens

A seleção dos descritores de imagem utilizados nesta pesquisa foi feita baseando-se nos trabalhos sobre recuperação de imagens por conteúdo. Verificar a relevância dos descritores para classificação de ISRs e a aplicação de outros descritores poderiam melhorar a qualidade e o tempo de execução do processo. Além disso, os descritores utilizados foram feitos para imagens de uso geral, que utilizam apenas informações do espectro visível codificadas no padrão RGB. ISRs, entretanto, podem disponibilizar informações em um número maior de bandas. Dessa forma, os descritores poderiam ser adaptados para trabalhar com uma quantidade maior de bandas, agregando mais informação ao processo de classificação.

Boa parte dos métodos de classificação existentes utilizam apenas informações de pixels isoladamente, calculando-se por exemplo, o índice NDVI e analisando a vizinhança dele para extrair informações de textura. Outros métodos, entretanto, utilizam outros índices e também informações sobre a região em que o pixel se encontra tais como perímetro, área, NDVI médio, etc [19, 22]. Há ainda índices característicos para determinadas culturas agrícolas. Assinaturas espectrais utilizando várias bandas ou série temporais também são utilizadas em alguns trabalhos para diferenciar culturas agrícolas. Uma possibilidade seria criar um descritor de culturas agrícolas baseando-se em características utilizadas comumente na classificação de ISRs.

6.2.3 Padrões de Consulta

O sistema atual utiliza apenas um padrão de entrada para consulta. Entretanto, uma mesma classe pode possuir diversos padrões de textura ou respostas espectrais que confundem alvos distintos. Um exemplo disso é o café. A Figura 6.1 ilustra uma ISR com algumas amostras de classes “café” e “não-café”. Nessa figura é possível visualizar a existência de diferentes padrões para a classe “café”, sendo que algumas delas inclusive são mais próximas (por resposta espectral ou textura) de amostras da classe “não-café” do que das outras amostras da classe “café”.

No caso do café, o modo como é plantado pode variar em uma mesma região. Ele pode ser plantado da forma convencional ou adensado (mais plantas em um mesmo espaço). Isso faz com que os pixels nas regiões de café adensado respondam de forma diferente aos talhões convencionais. A “idade” do talhão também faz com que os pixels contenham informação espectral diferente. E o maior problema é que, embora existam diferentes subclasses de café, o usuário normalmente deseja inclui-las em uma mesma classe: a classe das áreas plantadas com café.

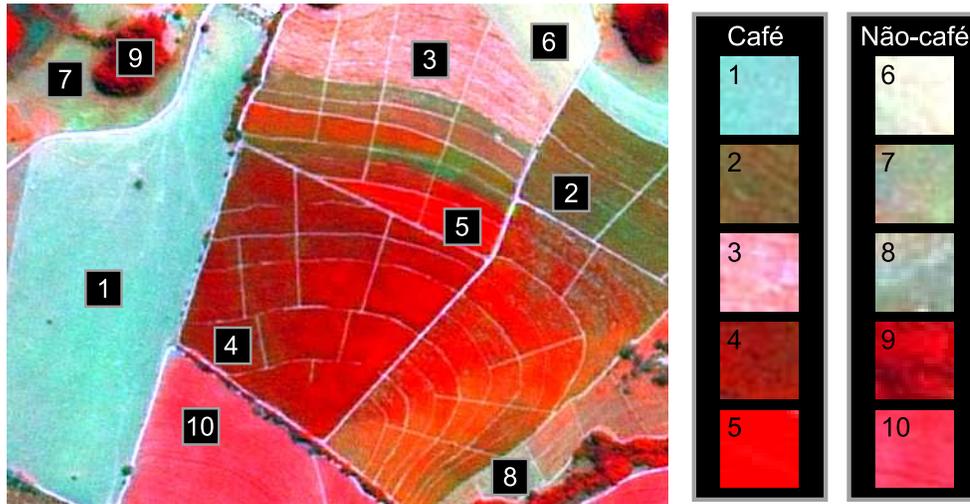


Figura 6.1: Imagem de sensoriamento remoto indicando amostras de regiões com café com diversos padrões espectrais e de textura.

6.2.4 Realimentação de Relevância

A etapa de realimentação de relevância é a responsável por agregar ao processo de classificação as informações fornecidas pelo usuário. Para isso, um arcabouço baseado em programação genética foi adaptado. Entretanto, não foram realizados experimentos para selecionar os melhores parâmetros para o módulo de PG [23]. Além disso, um arcabouço que considera tanto as imagens relevantes quanto irrelevantes no processo de aprendizado também poderia ser adaptado para classificar ISRs [16].

Ambos os arcabouços possuem etapas internas que poderiam ser alteradas. Uma mudança viável e que poderia melhorar o “feedback” do usuário é alterar o critério de seleção das sub-imagens que são mostradas a cada iteração para que sejam sempre sub-imagens vizinhas ao invés de mostrar imagens baseando-se apenas na relevância, o que permite mostrar ao usuário um conjunto de sub-imagens com distribuição espacial difícil de ser classificada.

Outra etapa que deve ser melhorada é com relação à resposta da realimentação de relevância. Ao invés de gerar uma lista ordenada das sub-imagens pela similaridade com a sub-imagem de consulta, para a classificação de ISRs, o ideal seria criar “respostas rotuladas” como: relevante, irrelevante e não rotulado. Isso facilitaria a classificação das sub-imagens, que posteriormente forneceria informação para o processo de segmentação e diminuiria a chance que as sub-imagens classificadas automaticamente de forma errada fossem utilizadas.

Os experimentos realizados até o momento foram feitos com usuários simulados, utilizando uma imagem já classificada para treinamento e teste. O ideal, no entanto, seria realizar testes com os especialistas da área para verificar a eficácia e eficiência na prática.

6.2.5 Segmentação e Classificação dos Pixels

No sistema atual, todas as sub-imagens são utilizadas como semente no processo de segmentação. Desse modo, se uma sub-imagem for classificada de forma errada, ela será utilizada como semente, e a região em que estiver inserida será classificada como relevante de forma errada. Uma alternativa para resolver esse problema seria utilizar apenas um percentual considerado relevante e um outro de sementes irrelevantes, fazendo com que as regiões conflitantes sejam disputadas pelas sementes rotuladas. Uma outra alternativa seria a utilização de um método de classificação por pixels, como o OPF (*Optimum Path Forest*) [37], para as regiões conflitantes. O OPF é um método de classificação supervisionada de padrões baseada em Floresta de Caminhos Ótimos, a qual modela o problema de reconhecimento de padrões como sendo um grafo, onde os nós são as amostras e os arcos definidos por uma relação de adjacência. Amostras mais relevantes (protótipos) são identificadas e um processo de competição entre elas é iniciado, as quais tentam oferecer caminhos de custo ótimo para as demais amostras da base de dados. O algoritmo OPF tenta minimizar esses custos para todas as amostras.

6.2.6 Comparação com Outros Métodos

O método proposto nesta dissertação foi comparado com o algoritmo de classificação MaxVer. Embora o algoritmo MaxVer seja o mais popular para a classificação de ISRs, ele é uma estratégia de classificação estatística baseada em pixels. Alguns métodos de classificação baseados em regiões [19, 22, 59] foram apresentados na Seção 2.1.1 e poderiam ser utilizados para comparação.

6.2.7 Outras Aplicações

O sistema atual foi criado para o reconhecimento de regiões em imagens de sensoriamento remoto visando alvos agrícolas e a solução de problemas de identificação de culturas em agricultura. Entretanto outras aplicações que utilizam ISRs carecem de soluções quando se trata de reconhecimento de padrões. Desse modo, o método proposto poderia ser testado, adaptado e aplicado na identificação de regiões urbanas, áreas desmatadas e até mesmo na análise de manchas de poluição em oceanos, por exemplo.

Referências Bibliográficas

- [1] H. M. Almeida. Uma abordagem de componentes combinados para a geração de funções de ordenação usando programação genética. Master's thesis, Departamento de Ciência da Computação, Universidade Federal de Minas Gerais, 2007.
- [2] B. Bhanu and Y. Lin. Object Detection in Multi-Modal Images Using Genetic Programming. *Applied Soft Computing*, 4(2):175–201, May 2004.
- [3] G. A. Carvalho, B. T. Leander, and A. C. M. Moura. Vetorização no spring. Technical report, Instituto de Geociências, Universidade Federal de Minas Gerais, Belo Horizonte, MG, Brasil, 2007.
- [4] R. G. Congalton and K. Green. *Assessing the Accuracy of Remotely Sensed Data: Principles and Practices*. Lewis Publishers, Washington, DC, 1977.
- [5] I. J. Cox, M. L. Miller, T. P. Minka, T. V. Papathomas, and P. N. Yianilos. The Bayesian Image Retrieval System, PicHunter: Theory, Implementation, and Psychophysical Experiments. *IEEE Transactions on Image Processing*, 9(1):20–37, January 2000.
- [6] L. M. Val Cura. *Um modelo para recuperação por conteúdo de imagens de sensoriamento remoto*. PhD thesis, Universidade Estadual de Campinas. Instituto de Matematica, Estatística e Computação Científica, Campinas, SP, 2002.
- [7] G. Câmara, M. A. Casanova, A. S. Hemerly, G. C. Magalhães, and C. M. B Medeiros. *Anatomia de Sistemas de Informação Geográfica*. Sagres Editora, Curitiba-PR, Brasil, 1997.
- [8] G. Câmara, R. Cartaxo, M. Souza, U. M. Freitas, J. Garrido, and F. M. Li. Spring: Integrating remote sensing and GIS by object oriented data modeling. *Computers and Graphics*, 20(3):395–403, 1996.
- [9] CNA Confederação da Agricultura e Pecuária do Brasil. Indicadores rurais, ano XI, Janeiro 2008.

- [10] R. da S. Torres and A. X. Falcão. Content-Based Image Retrieval: Theory and Applications. *Revista de Informática Teórica e Aplicada*, 13(2):161–185, 2006.
- [11] R. da S. Torres, A. X. Falcão, and L. da F. Costa. A Graph-based Approach for Multiscale Shape Analysis. *Pattern Recognition*, 37(6):1163–1174, June 2004.
- [12] R. da S. Torres, A. X. Falcão, M. A. Gonçalves, J. P. Papa, B. Zhang, W. Fan, and E. A. Fox. A genetic programming framework for content-based image retrieval. *Pattern Recognition*, 42(2):283–292, 2009.
- [13] L. Duan, W. Gao, W. Zeng, and D. Zhao. Adaptive relevance feedback based on Bayesian inference for image retrieval. *Signal Processing*, 85(2):395–399, February 2005.
- [14] A. X. Falcão, J. Stolfi, and R. A. Lotufo. The image foresting transform: Theory, algorithms, and applications. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(1):19–29, 2004.
- [15] W. Fan, M. D. Gordon, and P. Pathak. A generic ranking function discovery framework by genetic programming for information retrieval. *Information Processing & Management*, 40(4):587–602, July 2004.
- [16] C. Ferreira, R. da S. Torres, M. Gonçalves, and W. Fan. Image retrieval with relevance feedback based on genetic programming. In *23rd Brazilian Symposium on Database*, pages 120–134, Campinas, SP, Brasil, 2008.
- [17] C. D. Ferreira. Image retrieval with relevance feedback based on genetic programming. Master’s thesis, Instituto de Computação - Unicamp, July 2007.
- [18] R. A. Finkel and J. L. Bentley. Quad trees a data structure for retrieval on composite keys. *Acta Informatica*, 4(1):1–9, March 1974.
- [19] X. Gigandet, M.B. Cuadra, A. Pointet, L. Cammoun, R. Caloz, and J.-Ph. Thiran. Region-based satellite image classification: method and validation. *ICIP 2005. IEEE International Conference on Image Processing, 2005.*, 3:III–832–5, Sept. 2005.
- [20] Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1992.
- [21] P. Hong, Q. Tian, and T. S. Huang. Incorporate support vector machines to content-based image retrieval with relevant feedback. In *Proceedings of the 7th IEEE International Conference on Image Processing*, pages 750–753, 2000.

- [22] A. Katartzis, I. Vanhamel, and H. Sahli. A hierarchical markovian model for multiscale region-based classification of vector-valued images. *IEEE Transactions on Geoscience and Remote Sensing*, 43(3):548–558, March 2005.
- [23] J. R. Koza. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press, Cambridge, MA, USA, 1992.
- [24] J.Y. Lee and T. A. Warner. Image classification with a region based approach in high spatial resolution imagery. In *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, pages 181–187, Istanbul, Turkey, July 2004.
- [25] T. S. Lee. Image representation using 2D gabor wavelets. *IEEE Transactions Pattern Analysis Machine Intelligence*, 18(10):959–971, 1996.
- [26] P. B. C. Leite, R. Q. Feitosa, A. R. Formaggio, G. A. O. P. Da Costa, K. Pakzad, and I. D. Sanches. Crop type recognition based on hidden markov models of plant phenology. In *Brazilian Symposium on Computer Graphics and Image Processing, 21 (SIBGRAPI)*, Oct. 12–15, 2008.
- [27] T. M. Lillesand, R. W. Kiefer, and Jonathan W. Chipman. *Remote Sensing and Image Interpretation*. Wiley, 2007.
- [28] R. E. N. Loch. Alternativa de produção de mapas digitais em escala grande visando a análise temporal em SIG. In *Anais do VI Congresso Brasileiro de Cadastro Técnico Multifinalitário e II Encontro de CTM do Mercosul*. Florianópolis: UFSC, GT-Cadastro, Departamento de Engenharia Civil, Florianópolis, 2000.
- [29] R. A. Lotufo and A. X. Falcão. The ordered queue and the optimality of the watershed approaches. In *In Mathematical Morphology and its Applications to Image and Signal Processing*, pages 341–350. Kluwer Academic Publishers, 2000.
- [30] R. A. Lotufo, A. X. Falcão, and F. A. Zampirolli. IFT-watershed from gray-scale marker. In *SIBGRAPI '02: Proceedings of the 15th Brazilian Symposium on Computer Graphics and Image Processing*, pages 146–152, Washington, DC, USA, 2002.
- [31] X. Lou, W. Huang, A. Shi, and J. Teng. Raster to vector conversion of classified remote sensing image. *Geoscience and Remote Sensing Symposium, 2005. IGARSS '05. Proceedings. 2005 IEEE International*, 5:3656–3658, July 2005.
- [32] H. Y. Kim M. H. Shiba, R. L. Quintanilha. Classificação de imagens de sensoriamento remoto pela aprendizagem por árvore de decisão: uma avaliação de desempenho. In

- XII Simpósio Brasileiro de Sensoriamento Remoto*, pages 4320–4326, Goiânia, GO, Brasil, Abril 2005.
- [33] C. M. B Medeiros. About e-farms. <http://www.lis.ic.unicamp.br/projects/efarms/>.
- [34] M. A. Moreira. *Fundamentos do Sensoriamento Remoto e Metodologias de Aplicação*. UFV, Viçosa-MG, Brasil, third edition, 2005.
- [35] P. Muneesawang and L. Guan. An interactive approach for CBIR using a network of radial basis functions. *IEEE Transactions on Multimedia*, 6(5):703–716, November 2004.
- [36] OSGF. Gdal api tutorial. Technical report, Open Source Geospatial Foundation, 2008. Disponível em: <http://www.gdal.org/>.
- [37] J.P. Papa, A.A. Spadotto, A.X. Falcão, and J.C. Pereira. Optimum path forest classifier applied to laryngeal pathology detection. In *15th International Conference on Systems, Signals and Image Processing*, pages 249–252. IEEE, 2008.
- [38] J. R. Parker. Extracting vectors from raster images. Technical Report 87/259/07, University of Calgary, February 1987.
- [39] E. Persoon and K. Fu. Shape Discrimination Using Fourier Descriptors. *IEEE Transactions on Systems, Man, and Cybernetics*, 7(3):170–178, 1977.
- [40] A. F. N. Ribeiro. A utilização do geoprocessamento para caracterização do uso e ocupação do solo na microbacia do córrego laranja azeda. Trabalho de Conclusão de Curso, Universidade Federal da Grande Dourados, Dourados, MS, Brasil, 2006.
- [41] Y. Rui, T. S. Huang, M. Ortega, and S. Mehrotra. Relevance Feedback: A Power Tool for Interactive Content-Based Image Retrieval. *IEEE Transactions on Circuits and Systems for Video Technology*, 8(5):644–655, 1998.
- [42] K. Seo S. Park and D. Jang. Expert system based on artificial neural networks for content-based image retrieval. *Expert Systems with Applications*, 29(3):589–597, October 2005.
- [43] J. A. Santos, C. Ferreira, and R. da S. Torres. A genetic programming approach for relevance feedback in region-based image retrieval systems. In *Brazilian Symposium on Computer Graphics and Image Processing*, pages 155–162, Campo Grande, MS, Brasil, 2008.

- [44] J. A. Santos, R. A. C. Lampareli, and R. da S. Torres;. Using relevance feedback for classifying remote sensing images. In *XIV Brazilian Remote Sensing Symposium*, Natal, RN, Brazil, Abril 2009.
- [45] P. Selinger. Potrace: a polygon-based tracing algorithm. available at <http://potrace.sourceforge.net/potrace.pdf>, September 2003.
- [46] R. Showengerdt. *Techniques for Image Processing and Classification in Remote Sensing*. Academic Press, New York, 1983.
- [47] R.O. Stehling, M.A. Nascimento, and A.X. Falcão. A Compact and Efficient Image Retrieval Approach Based on Border/Interior Pixel Classification. In *Proceedings of the 11th ACM International Conference on Information and Knowledge Management*, pages 102–109, McLean, Virginia, USA, November 2002.
- [48] Z. Stejic, Y. Takama, and K. Hirota. Genetic algorithms for a family of image similarity models incorporated in the relevance feedback mechanism. *Applications on Software Computing*, 2(4):306–327, 2003.
- [49] Z. Stejic, Y. Takama, and K. Hirota. Mathematical aggregation operators in image retrieval: effect on retrieval performance and role in relevance feedback. *Signal Processing*, 85(2):297–324, 2005.
- [50] Z. Stejic, Y. Takama, and K. Hirota. Relevance feedback-based image retrieval interface incorporating region and feature saliency patterns as visualizable image similarity criteria. *IEEE Transactions on Industrial Electronics*, 50(5):839–852, Oct. 2003.
- [51] M. A. Stricker and M. Orengo. Similarity of Color Images. In *Storage and Retrieval for Image and Video Databases (SPIE)*, pages 381–392, 1995.
- [52] M. Swain and D. Ballard. Color Indexing. *International Journal of Computer Vision*, 7(1):11–32, 1991.
- [53] S. Tong and E. Y. Chang. Support vector machine active learning for image retrieval. In *Proceedings of 9th ACM international conference on Multimedia*, pages 107–118, New York, NY, USA, 2001.
- [54] R. S. Torres, J. A. M. Zegarra, J. A. Santos, C. D. Ferreira, O. A. B. Penatti, F. A. Andaló, and J. G. Almeida Júnior. Recuperação de imagens: Desafios e novos rumos. In *XXXV Seminário Integrado de Software e Hardware (SEMISH)*, pages 223–237, Belém, PA, Brasil, 2008.

- [55] M. Unser, A. Aldroubi, and M. Eden. A family of polynomial spline wavelet transforms. *Signal Processing*, 30(2):141–162, 1993.
- [56] L. Vincent and P. Soille. Watersheds in digital spaces: an efficient algorithm based on immersion simulations. *Transactions on Pattern Analysis and Machine Intelligence*, 13(6):583–598, 1991.
- [57] L. Wenyin and D. Dori. From raster to vectors: Extracting visual information from line drawings. *Pattern Anal. Appl*, 2(1):10–21, 1999.
- [58] G.G. Wilkinson. Results and implications of a study of fifteen years of satellite image classification experiments. *IEEE Transactions on Geoscience and Remote Sensing*, 43(3):433–440, March 2005.
- [59] Q. Yu, P. Gong, N. Clinton, G. Biging, M. Kelly, and D. Schirokauer. Object-based detailed vegetation classification with airborne high spatial resolution remote sensing imagery. *Photogrammetric Engineering Remote Sensing*, 72(7):799–811, 2006.
- [60] B. Zhang, M. A. Gonçalves, W. Fan, Y. Chen, E. A. Fox, P. Calado, and M. Cristo. Combining structural and citation-based evidence for text classification. In *Proceedings of the 13th ACM Conference on Information and Knowledge Management*, pages 162–163, 2004.
- [61] X. S. Zhou and T. S. Huang. Relevance feedback in image retrieval: A comprehensive review. *Multimedia Systems*, 8(6):536–544, 2003.
- [62] D. Zonker and B. Punch. lil-gp 1.01 User’s Manual. Technical report, College of Engineering, Michigan State University, Michigan, USA, 1996.