

Segmentação interativa de imagens
via transformação *watershed*

Bruno Klava

DISSERTAÇÃO APRESENTADA
AO
INSTITUTO DE MATEMÁTICA E ESTATÍSTICA
DA
UNIVERSIDADE DE SÃO PAULO
PARA
OBTENÇÃO DO TÍTULO
DE
MESTRE EM CIÊNCIAS

Programa: Ciência da Computação
Orientador: Prof^a. Dr^a. Nina S. T. Hirata

Durante o desenvolvimento deste trabalho o autor recebeu auxílio financeiro do CNPq

São Paulo, julho de 2009

Segmentação interativa de imagens via transformação *watershed*

Este exemplar corresponde à redação
final da dissertação devidamente corrigida
e defendida por Bruno Klava
e aprovada pela Comissão Julgadora.

Banca Examinadora:

- Prof^a. Dr^a. Nina Sumiko Tomita Hirata (orientador) - IME-USP.
- Prof. Dr. Roberto Marcondes Cesar Junior - IME-USP.
- Prof. Dr. Roberto de Alencar Lotufo - UNICAMP.

Agradecimentos

Este trabalho não seria possível sem as oportunidades que tive. E nenhuma dessas oportunidades seriam possíveis se não fosse pelo esforço e o apoio dos meus pais. Por isso fica aqui o mais profundo agradecimento aos meus pais, Rita e Nelson, por seu amor, por me ensinarem o valor da educação e por me proverem com tudo o que esteve aos seus alcances.

Aos meus amigos, pelo apoio incondicional, companheirismo, conselhos e todos os momentos de alegria.

À professora Nina pelo apoio e orientação desde a graduação. Pelas suas idéias ao longo do desenvolvimento deste trabalho, pela sua confiança no meu potencial, pela sua compreensão e paciência. Fica aqui o meu agradecimento e a minha admiração pelo seu exemplo de competência.

Aos demais membros das bancas do exame de qualificação e da banca de mestrado, professores Roberto Marcondes Cesar Jr., José Coelho de Pina Jr. e Roberto de Alencar Lotufo, pelas valiosas sugestões que contribuíram para a qualidade desta dissertação e com idéias para trabalhos futuros.

Aos demais professores e funcionários do Instituto de Matemática e Estatística da Universidade de São Paulo, pelo ensino de qualidade e pela excelente formação que proporcionam aos alunos.

A todos que, de uma forma ou outra, contribuíram para a realização deste trabalho.

Finalmente, ao CNPq, pelo auxílio financeiro concedido, permitindo minha dedicação ao programa de mestrado e ao desenvolvimento deste trabalho de pesquisa.

Resumo

A transformação *watershed* é uma abordagem bastante utilizada em segmentação de imagens. O *watershed* a partir de marcadores e o *watershed* hierárquico são abordagens derivadas da transformação *watershed* e adequadas para segmentação interativa de imagens: na primeira, o usuário pode editar os marcadores controlando a partição resultante; na outra, o usuário pode selecionar uma partição num conjunto hierárquico de partições da imagem. O propósito deste trabalho é estudar a relação entre esses dois paradigmas. Em particular, apresentamos as condições necessárias e formas de se alternar entre os dois paradigmas num processo de segmentação interativa (implementado na ferramenta disponível em <http://watershed.sourceforge.net/>), de forma que o usuário possa aproveitar os pontos positivos de cada uma dessas abordagens para obter a partição desejada.

Palavras-chave: segmentação, interatividade, *watershed*.

Abstract

The watershed transform is a well-known approach for image segmentation. Watershed from markers and hierarchical watershed are approaches derived from the watershed transform and are suitable for interactive image segmentation: in the former, the user can edit markers controlling the resultant partition; in the latter, the user can select an image partition from a nested set of partitions. The purpose of this work is to study the relation between these two approaches. More specifically, we show the necessary conditions and how to transition from one approach to the other in an interactive segmentation process (implemented in the tool available at <http://watershed.sourceforge.net/>), in a way that the user can exploit the strengths of both approaches to obtain the desired partition.

Keywords: segmentation, interactivity, watershed.

Sumário

Lista de Abreviaturas	v
Lista de Figuras	vii
Lista de Algoritmos	xi
1 Introdução	1
1.1 Objetivos e contribuições	5
1.2 Organização do texto	5
2 Preliminares	7
2.1 Grafos	7
2.2 Imagens digitais	8
2.2.1 Imagens digitais como grafos	8
2.2.1.1 Componentes conexos numa imagem	9
2.3 Morfologia matemática	10
2.3.1 Gradiente morfológico	12

2.4	Segmentação	12
3	Transformação <i>watershed</i>	15
3.1	<i>Watershed</i> clássico	15
3.2	<i>Watershed</i> a partir de marcadores	16
3.2.1	<i>Watershed</i> através da <i>Image foresting transform</i>	18
3.2.2	Marcadores nas bacias primitivas	20
3.3	<i>Watershed</i> hierárquico	21
3.3.1	Construção da hierarquia de partições	21
3.3.2	Representação da hierarquia de partições	22
3.3.2.1	Localização de partições na hierarquia	24
3.3.2.2	Tipos de inundação	28
4	Alternando entre os paradigmas do <i>watershed</i>	33
4.1	Hierarquia para marcadores	33
4.1.1	Criação de marcadores	34
4.1.2	MSS no RAG <i>versus</i> MSS em G_I	35
4.2	Marcadores para hierarquia	42
5	Interatividade no <i>watershed</i>	45
5.1	Possibilidades de interação	45
5.2	Características adicionais	48
5.3	Comparação das abordagens	49
5.3.1	<i>Watershed</i> a partir de marcadores	49
5.3.2	<i>Watershed</i> hierárquico	50
5.3.3	Abordagem integrada	51

	iii
6 Conclusões	59
6.1 Contribuições do trabalho	59
6.2 Trabalhos futuros	60
A Publicações	61
B Ferramenta	63
B.1 Descrição dos pacotes	63
B.1.1 br.usp.ime.klava.segmentit.exceptions	64
B.1.2 br.usp.ime.klava.segmentit.filters	65
B.1.3 br.usp.ime.klava.segmentit.formats	66
B.1.4 br.usp.ime.klava.segmentit.gui	66
B.1.5 br.usp.ime.klava.segmentit.gui.actions	70
B.1.6 br.usp.ime.klava.segmentit.structures	70
B.1.7 br.usp.ime.klava.segmentit.util	70
B.1.8 br.usp.ime.klava.segmentit.watershed	73
B.2 Interface gráfica	74
B.2.1 Ferramentas	74
B.2.2 Camadas	76
B.2.3 Menus	76
B.2.4 Outras janelas	79
Referências Bibliográficas	81

Lista de Abreviaturas

AGM	Árvore geradora mínima
ALC	Árvore dos lagos críticos
RAG	Grafo de adjacência de regiões (<i>region adjacency graph</i>)
MSS	Conjunto mínimo de marcadores (<i>minimal seed set</i>)
NRRR	Região receptiva não redundante (<i>non redundant receptive region</i>)
RR	Região receptiva de marcadores
RRR	Região receptiva redundante
TZ	Zona de empate (<i>tie-zone</i>)

Lista de Figuras

1.1	Facilitando o processo de segmentação.	2
1.2	Diferentes partições manuais.	2
1.3	Imagens com bordas não aparentes.	4
2.1	Imagem digital.	9
2.2	Representação matricial da Fig. 2.1.	9
2.3	Diferenças entre 4-conectividade e 8-conectividade.	10
2.4	Imagem e seu gradiente morfológico.	12
2.5	Segmentação por limiarização.	13
2.6	Efeitos da escolha de diferentes limiares.	14
3.1	Partição super-segmentada da Fig. 2.4.	16
3.2	Inundação por marcadores.	17
3.3	Partições obtidas através da inundação por marcadores.	17
3.4	Construção de uma hierarquia de partições.	23
3.5	Partições representadas como fusões de bacias primitivas.	24

3.6	Partições obtidas utilizando somente a árvore geradora mínima.	25
3.7	Construção da ALC a partir da AGM.	26
3.8	Localização de partições através de cortes na ALC.	27
3.9	Partições geradas pela inundação uniforme.	28
3.10	Partições da Fig. 3.9(a) geradas por inundações síncronas.	30
4.1	NRRRs de uma partição do RAG.	35
4.2	RRs da região azul da Fig. 4.1(b).	36
4.3	RRs da região vermelha da Fig. 4.1(b).	37
4.4	Marcadores calculados no RAG aplicados em G_I	39
4.5	NRRRs de cada região de interesse, computadas em G_I	40
5.1	Segmentando através da hierarquia e de marcadores.	47
5.2	Hierarquia restrita a uma região.	48
5.3	Informação de cor reduzindo o esforço de interação.	49
5.4	Limitação de partições devido à estrutura da ALC.	51
5.5	Diagrama da abordagem integrada.	52
5.6	Segmentando as teclas de uma calculadora.	54
5.7	Segmentando a imagem da Fig. 2.4(a).	55
5.8	Segmentando imagem com bordas não aparentes.	57
B.1	Notações utilizadas nos diagramas UML.	64
B.2	Diagrama de classes do pacote <i>br.usp.ime.klava.segmentit.exceptions</i>	64
B.3	Diagrama de classes do pacote <i>br.usp.ime.klava.segmentit.filters</i>	65
B.4	Diagrama de classes do pacote <i>br.usp.ime.klava.segmentit.formats</i>	67
B.5	Diagrama de classes do pacote <i>br.usp.ime.klava.segmentit.gui</i>	68

B.6	Diagrama de classes do pacote <i>br.usp.ime.klava.segmentit.gui.actions</i>	69
B.7	Diagrama de classes do pacote <i>br.usp.ime.klava.segmentit.structures</i>	71
B.8	Diagrama de classes do pacote <i>br.usp.ime.klava.segmentit.util</i>	72
B.9	Diagrama de classes do pacote <i>br.usp.ime.klava.segmentit.watershed</i>	73
B.10	Interface principal.	74
B.11	Menus do <i>SegmentIt</i>	77
B.12	Interface da abordagem hierárquica.	79
B.13	Interface do filtro gradiente com pesos.	80

Crédito das imagens:

- Fig. 1.1(a) - [27]
http://commons.wikimedia.org/wiki/Image:Cerebral_Angiogram_Lateral.jpg
- Fig. 1.1(b) - [27]
http://commons.wikimedia.org/wiki/Image:Bluescreen_weather.JPG
- Fig. 1.2 - [24]
<http://www.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/BSDS300/html/dataset/images/gray/277095.html>
- Fig. 1.3(a), Fig. 5.8(a) - [24]
<http://www.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/BSDS300/html/dataset/images/gray/24063.html>
- Fig. 1.3(b) - [24]
<http://www.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/BSDS300/html/dataset/images/gray/118035.html>
- Fig. 2.4(a), Fig. 3.3(a), Fig. 3.9(a) - [24]
<http://www.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/BSDS300/html/dataset/images/gray/42049.html>
- Fig. 3.8(a) - [24]
<http://www.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/BSDS300/html/dataset/images/gray/14037.html>
- Fig. 5.1(a) - [7]
http://www.mmorph.com/handson/handson/images/img_hofig7s17_001.jpg
- Fig. 5.2(a), Fig. 5.3(a), Fig. 5.3(c) - [12]
<http://maps.google.com/?ie=UTF8&t=k&ll=-15.623037,1.054688&spn=127.248296,307.96875&z=2>
- Fig. 5.6(a) [7]
Versão reduzida de http://www.mmorph.com/handson/handson/images/img_hofig7s13_001.jpg

Lista de Algoritmos

1	Rotulação de componentes conexos	11
2	Limiarização	13
3	Watershed através da IFT	19
4	Construção da ALC a partir da AGM	26

A segmentação é uma etapa importante em praticamente todos os problemas que envolvem análise de imagens digitais [11]. O processo de segmentação tem por objetivo particionar o domínio espacial da imagem, de forma a demarcar as regiões de interesse, que correspondem aos objetos alvos da análise em questão.

A segmentação é um processamento difícil, uma vez que imagens variam muito e também porque muitas vezes não é fácil descrever formalmente o que desejamos segmentar. Por estes motivos, geralmente são utilizadas estratégias para facilitar o processo de segmentação. Por exemplo, em alguns tipos de exames médicos é comum a utilização de alguma substância de contraste, para ressaltar o tecido que será analisado em relação aos demais, como pode ser observado na Fig. 1.1(a). Na indústria televisiva e cinematográfica, a técnica de *chroma key* é amplamente utilizada para segmentar pessoas e elementos cenográficos, de cenas gravadas em estúdio, para fundi-los com cenas gravadas em ambientes externos ou ambientes virtuais, como pode ser observado na Fig. 1.1(b).

Como nem sempre é possível facilitar o processo de segmentação, geralmente são desenvolvidos sistemas para segmentar imagens restritas a um domínio específico. Em muitos casos é necessário um pós-processamento da segmentação, obtida através de técnicas automáticas, para que se obtenha um resultado satisfatório. Além disso, a escolha da abordagem para resolver um determinado problema de segmentação não é uma tarefa trivial, uma vez que pode ser complexo especificar formalmente os parâmetros para obter uma partição desejada e também pelo fato de que o conceito de uma boa partição depende do propósito do seu uso. Por exemplo, na Fig. 1.2 são exibidas partições de uma mesma imagem feitas manualmente por diferentes usuários, o que mostra que o conceito de uma boa segmentação varia para cada indivíduo.

Em muitas situações, o resultado da segmentação deve ser avaliado por um especialista, com o intuito de garantir a precisão ou a qualidade da partição da imagem analisada. Nestes

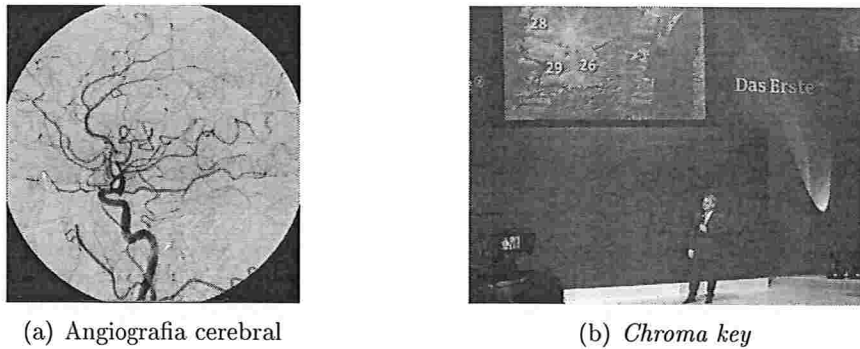


Figura 1.1: Facilitando o processo de segmentação: (a) Utilização de substância de contraste de iodo para destacar vasos sanguíneos em uma radiografia cerebral. (b) Utilização da técnica de *chroma key* em programa meteorológico.

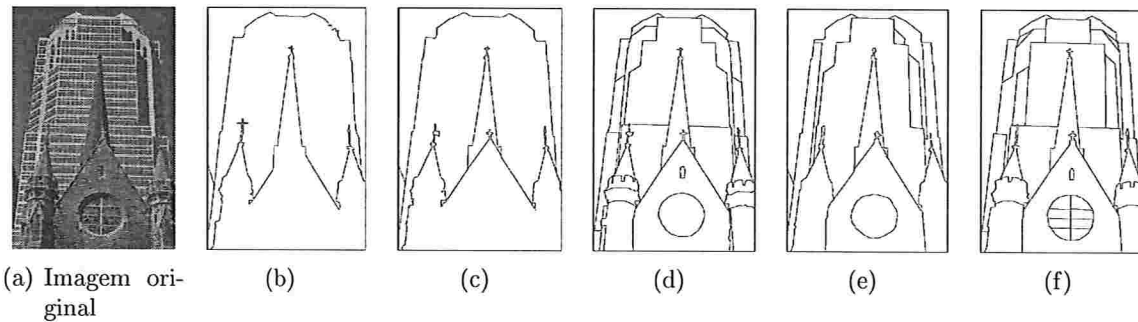


Figura 1.2: Diferentes partições manuais.

casos, é comum que o processo de segmentação não seja automatizado, ou seja, o processo é feito manualmente pelo especialista. Dentre estes casos, podemos citar:

- localização de estruturas orgânicas em imagens médicas, em que a precisão dos resultados é decisiva no diagnóstico;
- separação entre objetos e fundo em edição de fotos e vídeos, em que as bordas dos objetos devem ser suaves, para que o resultado final da composição fique o mais natural possível;
- geração de *ground-truth*, que corresponde à segmentação ideal, sendo utilizada geralmente para avaliar algoritmos de segmentação ou para desenvolver sistemas de segmentação baseados em aprendizado via exemplos. Em geral é produzida via edição manual, por especialistas do domínio da imagem quando este é bem definido e

restrito (por exemplo, [25] no caso de imagens médicas) ou como em [24] para imagens genéricas.

Em outras situações, não há a necessidade de desenvolvimento de um sistema robusto para a segmentação automática, mas há o interesse em se obter uma segmentação de forma mais rápida, com possibilidade de intervenção humana para tal:

- quando o objeto de estudo não é a segmentação em si, mas a análise posterior dos objetos contidos na imagem;
- criação de imagens segmentadas para inserção em documentos e apresentações;
- anotação de vídeos, para possibilitar busca em bases de dados multimídia, através da rotulação de alguns objetos da cena e posterior associação de metadados a cada rótulo, por exemplo, através do uso de descritores do MPEG-7¹.

Sistemas interativos para segmentação são apropriados para lidar com essas questões, já que o usuário pode direcionar o processo de segmentação, tendo controle sobre o nível de detalhes para cada região de interesse na partição.

A transformação *watershed* [5] é uma ferramenta bem conhecida para segmentação na área de morfologia matemática, possuindo duas abordagens adequadas para segmentação interativa:

- a abordagem do *watershed a partir de marcadores* reduz o problema de segmentação ao problema de achar um conjunto de marcadores para as regiões de interesse. Manipulando o conjunto de marcadores interativamente, o usuário consegue direcionar o processo para obter a partição desejada;
- o *watershed hierárquico* [3] gera um conjunto de partições aninhadas, com partições da imagem em diversos níveis de resolução. A hierarquia é construída com base numa partição super-segmentada formada por micro-regiões (resolução mais alta), que são fundidas de forma a gerar os níveis intermediários da hierarquia, até que toda a imagem esteja contida numa única região (partição com menor resolução na hierarquia). O usuário pode navegar através da hierarquia para selecionar uma partição com o nível de detalhes desejado para cada região de interesse.

¹O MPEG-7 é uma especificação de metadados para conteúdos multimídia, criado para padronizar a representação das informações sobre mídia digital, permitindo a globalização e interoperabilidade destes dados. Os descritores do MPEG-7 permitem a descrição semântica e estrutural de conteúdos multimídia.

Apesar das duas abordagens serem apropriadas para interação, ambas têm deficiências. Utilizando a primeira abordagem, pode ser necessário um esforço de interação considerável no caso de haver muitas regiões de interesse, já que é necessário desenhar um marcador para cada uma delas. No outro caso, existem partições que não podem ser localizadas na hierarquia pelo modo como esta é construída; além do mais, as bordas das partições contidas na hierarquia são necessariamente um subconjunto das bordas das micro-regiões. Isto é uma limitação em particular nos casos de bordas que não aparecem numa imagem, mas que sabidamente existem, como por exemplo nas imagens da Fig. 1.3.

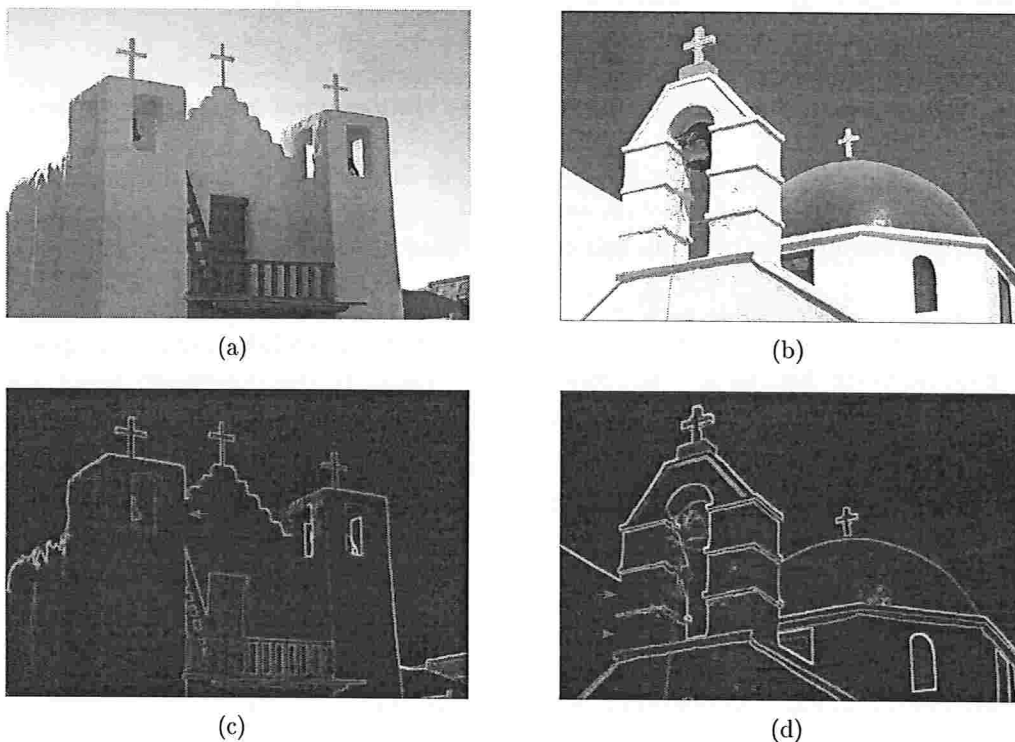


Figura 1.3: Imagens com bordas não aparentes: (a-b) imagens originais. (c-d) Gradientes das imagens, que realçam suas bordas (bordas não aparentes indicadas pelas setas vermelhas).

Uma abordagem possível para contornar as limitações de cada uma das abordagens descritas é explorar o uso de marcadores e hierarquias conjuntamente. A limitação dos trabalhos existentes que utilizam tal abordagem, como [13], [18] e [22], é que são formulados de forma a operar no nível de precisão de micro-regiões, o que implica que as limitações do *watershed* hierárquico permanecem.

Nossa proposta é integrar o *watershed* hierárquico e o *watershed* a partir de marcadores,

independentemente do nível de precisão desejado (*pixels* ou micro-regiões).

1.1 Objetivos e contribuições

O propósito deste trabalho é integrar o paradigma de marcadores e o *watershed* hierárquico, de modo a permitir que os usuários aproveitem os pontos mais fortes de cada um deles e obtenham os resultados de segmentação desejados com menor esforço de interação. Em particular, será investigado como mapear uma partição correspondente a um conjunto de marcadores a uma hierarquia de partições e, no sentido contrário, como mapear uma partição de uma dada hierarquia a um conjunto de marcadores que recupera a mesma partição. Se tais mapeamentos são possíveis, então é possível alternar entre os dois paradigmas.

Uma dificuldade nesse processo é devida ao fato que o *watershed* hierárquico opera somente no nível de precisão das micro-regiões, limitação que não existe na abordagem de marcadores, que pode ser aplicada tanto no nível de precisão dos *pixels* como no nível de precisão das micro-regiões (a abordagem de marcadores é utilizada no nível de precisão das micro-regiões geralmente por questões de eficiência computacional). Novos resultados a respeito dessa questão são apresentados e, baseado neles, diversas possibilidades de interação são propostas. As possibilidades propostas foram implementadas e testadas na ferramenta interativa para segmentação de imagens denominada *SegmentIt*, disponível em <http://watershed.sourceforge.net/>.

Como objetivo secundário, faremos uma avaliação da abordagem integrada proposta, de forma a validar os benefícios e possibilidades que tal abordagem proporciona aos usuários de uma ferramenta interativa para segmentação. Apresentamos vários exemplos que ilustram esse uso integrado.

1.2 Organização do texto

No Capítulo 2 são revisados conceitos das áreas correlatas, necessários para o desenvolvimento deste trabalho. No Capítulo 3, são apresentados detalhes da transformação *watershed* e de suas abordagens adequadas à interação. No Capítulo 4 são discutidas as limitações e formas de se alternar entre os paradigmas do *watershed*. No Capítulo 5 são apresentadas as possibilidades de interação baseadas nos resultados apresentados no Capítulo 4, juntamente com exemplos que ilustram seu uso e uma análise comparando os paradigmas e a abordagem proposta. Finalmente, no Capítulo 6 são apresentadas as conclusões e as futuras investigações no contexto deste trabalho.

Nos apêndices é apresentada uma lista de publicações realizadas no contexto deste trabalho e detalhes referentes à ferramenta implementada, que disponibiliza todas as pos-

sibilidades de interação apresentadas no Capítulo 5.

Observação: Este texto é melhor visualizado em cores, pois vários detalhes importantes são ilustrados nas figuras através do uso de cores.

Neste capítulo são apresentados conceitos teóricos e terminologias das áreas correlatas ao trabalho, bem como notações utilizadas ao longo deste texto.

2.1 Grafos

Um *grafo* $G = (V, E)$ é uma estrutura formada por um conjunto V de vértices e um conjunto E de arestas, tal que $E \subseteq V \times V$.

Um *grafo com pesos (nas arestas)* é um grafo $G = (V, E, \omega)$, onde ω é uma função que associa um peso (não negativo) a cada aresta de E .

Dois vértices $u, v \in V$ são ditos *adjacentes* ou *vizinhos* no grafo $G = (V, E)$ se $(u, v) \in E$.

$N(u)$ denota a vizinhança de u : $N(u) = \{v \in V : (u, v) \in E\}$.

Um *caminho* $\pi(u, v)$ entre dois vértice u e v é uma seqüência $\langle u = v_1, v_2, \dots, v_n = v \rangle$ de vértices, tal que $(v_i, v_{i+1}) \in E$, $1 \leq i < n$, e $v_i \neq v_j$, $1 \leq i < j \leq n$.

O comprimento de um caminho é igual ao número de arestas que o compõem.

A concatenação de dois caminhos $\pi(u, v) = \langle u = v_1, v_2, \dots, v_n = v \rangle$ e $\pi(v, w) = \langle v = w_1, w_2, \dots, w_m = w \rangle$ é denotada por $\pi(u, v) \cdot \pi(v, w)$ e corresponde à seqüência $\langle u = v_1, v_2, \dots, v_n = v = w_1, w_2, \dots, w_m = w \rangle$.

Um caminho de u a v com custo mínimo de acordo com uma função de custo f_C é representado por $\pi^*(u, v)$.

$\pi^*(M, v)$ denota um caminho de custo mínimo entre um conjunto M , $M \subseteq V$, ao vértice v , isto é, $f_C(\pi^*(M, v)) = f_C(\pi^*(m_i, v))$, para algum $m_i \in M$, e $f_C(\pi^*(m_i, v)) \leq f_C(\pi^*(m_j, v))$, $\forall m_j \in M$.

Um *componente conexo* em um grafo $G = (V, E)$ é um conjunto de vértices $C \subseteq V$ tal que, se $v \in C$ e $\exists \pi(v, w)$, então $w \in C$, $\forall w \in V$.

Um grafo $G' = (V', E')$ é um *subgrafo* de $G = (V, E)$, denotado por $G' \subseteq G$, se $V' \subseteq V$, $E' \subseteq V' \times V'$ e $E' \subseteq E$.

Um grafo $G' = (V', E')$ é um *grafo gerador* de $G = (V, E)$ se $G' \subseteq G$ e $V' = V$.

Um grafo é dito *acíclico* se existe, no máximo, um único caminho entre quaisquer de seus vértices.

Uma *floresta* de G é um subgrafo acíclico de G . Cada componente conexo de uma floresta é denominada *árvore*.

Uma árvore T é uma *árvore geradora mínima* (AGM) de um grafo com pesos $G = (V, E, \omega)$ se T é um subgrafo gerador de G e a soma dos pesos de suas arestas é mínima, considerando todas as possíveis árvores geradoras de G .

2.2 Imagens digitais

Uma imagem I pode ser definida como uma função bidimensional, $I = f(x, y)$, na qual x e y são coordenadas espaciais e o valor assumido por f no ponto (x, y) é chamado de intensidade da imagem no ponto (x, y) . Quando as coordenadas espaciais e a intensidade assumem valores finitos e discretos, chamamos a imagem de *imagem digital*.

Cada elemento da imagem digital é chamado de *pixel* (contração de *picture element*), e associa um ponto (x, y) do domínio espacial a uma intensidade. Denotaremos a intensidade do *pixel* p na imagem I por $I(p)$.

Por exemplo, a Fig. 2.1 representa uma imagem digital de 9 *pixels* de largura por 11 *pixels* de altura. Uma forma equivalente de representar a imagem é através de uma matriz, cujos elementos são dados pelas intensidades dos *pixels* nas posições correspondentes, como na Fig. 2.2.

2.2.1 Imagens digitais como grafos

Uma imagem digital I pode ser representada como um grafo G_I , no qual cada *pixel* de I corresponde a um vértice, e as arestas ligam os *pixels* vizinhos, segundo alguma relação de vizinhança. As relações de vizinhança mais utilizadas são a *4-conectividade* e a *8-conectividade*:

4-conectividade: Os *pixels* vizinhos por 4-conectividade de um *pixel* p são os *pixels* posicionados ao redor de p na direção vertical ou horizontal, indicados em cinza na

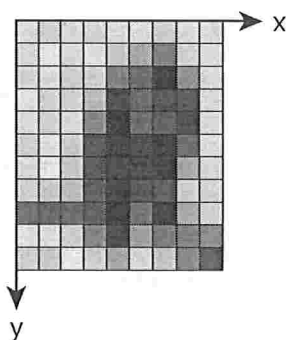


Figura 2.1: Imagem digital.

$$f = \begin{bmatrix} 197 & 189 & 199 & 195 & 206 & 208 & 189 & 203 & 201 \\ 196 & 208 & 200 & 211 & 184 & 151 & 110 & 198 & 202 \\ 194 & 203 & 193 & 201 & 111 & 87 & 39 & 129 & 205 \\ 207 & 196 & 205 & 178 & 56 & 89 & 62 & 68 & 208 \\ 201 & 195 & 207 & 124 & 39 & 94 & 79 & 57 & 209 \\ 189 & 208 & 196 & 69 & 42 & 59 & 39 & 89 & 209 \\ 199 & 219 & 198 & 67 & 56 & 53 & 42 & 158 & 208 \\ 195 & 196 & 187 & 78 & 51 & 71 & 85 & 209 & 207 \\ 86 & 79 & 87 & 71 & 39 & 100 & 45 & 159 & 202 \\ 207 & 208 & 196 & 114 & 50 & 141 & 103 & 109 & 121 \\ 187 & 185 & 218 & 194 & 145 & 208 & 184 & 115 & 69 \end{bmatrix}$$

Figura 2.2: Representação matricial da Fig. 2.1, na qual o valor de cada entrada corresponde ao nível de cinza do *pixel* de posição correspondente na imagem.

Fig. 2.3(a). Define um grafo 4-conectado, como na Fig. 2.3(c).

8-conectividade: Os *pixels* vizinhos por 8-conectividade de um *pixel* p são os *pixels* posicionados ao redor de p , tanto nas direções vertical e horizontal quanto nas diagonais, indicados em cinza na Fig. 2.3(b). Define um grafo 8-conectado, como na Fig. 2.3(d).

2.2.1.1 Componentes conexos numa imagem

No Algoritmo 1 (baseado numa busca em largura) apresentamos como rotular os componentes conexos de uma imagem colorida, considerando como vizinhos os *pixels* conectados segundo a relação de adjacência dada e que têm a mesma cor. Os *pixels* do fundo, cujas

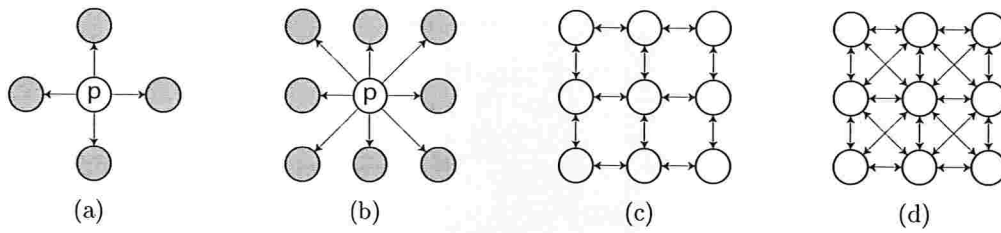


Figura 2.3: Diferenças entre 4-conectividade e 8-conectividade.

cores são iguais a *corFundo*, são desconsiderados e recebem 0 como rótulo. Como saída do algoritmo, obtemos a imagem L com cada componente conexo rotulado com um número inteiro diferente.

2.3 Morfologia matemática

Na área de morfologia matemática, baseando-se na teoria de reticulados¹ e topologia, é estudado, entre outros assuntos, a decomposição de operadores de imagens em função de dois operadores elementares: a erosão e a dilatação.

Seja I uma imagem. A *erosão* de I , definida pela Eq. 2.1, é formada pelo valor mínimo dos níveis de cinza da imagem na vizinhança de x , para todo *pixel* x de I . A *dilatação* de I , definida pela Eq. 2.2, é formada pelo valor máximo dos níveis de cinza da imagem na vizinhança de x , para todo *pixel* x de I . Para os dois operadores, a vizinhança é definida por B , chamado de elemento estruturante².

$$[\varepsilon_B(I)](x) = \min\{I(z) : z \in B \text{ transladado de } x\} \quad (2.1)$$

$$[\delta_B(I)](x) = \max\{I(z) : z \in B \text{ transladado de } x\} \quad (2.2)$$

¹Neste trabalho, é considerado o reticulado das imagens em níveis de cinza.

²As definições de erosão, dilatação e gradiente morfológico aqui apresentadas são válidas para elementos estruturantes planos e simétricos.

Algoritmo 1: Rotulação de componentes conexos

```

1  $L = \text{rotulacao}(I, \text{corFundo})$ 
2  $I$ : imagem de entrada (colorida)
3  $\text{corFundo}$ : cor dos pixels do fundo da imagem
4  $L$ : imagem de saída rotulada
5  $\text{fila}$ : fila simples (política First In First Out)
6  $\text{rotulo}$ : rótulo do componente conexo sendo rotulado
7  $\text{corAtual}$ : cor do componente conexo sendo rotulado
8 1. Inicialização
9 para cada  $p \in I$  faça
10   se  $I(p) = \text{corFundo}$  então
11      $L(p) \leftarrow 0$ 
12   senão
13      $L(p) \leftarrow -1$ 
14  $\text{rotulo} \leftarrow 0$ 
15 2. Rotulação
16 para cada  $p \in I$  faça
17   se  $L(p) = -1$  então
18      $\text{corAtual} \leftarrow I(p)$ 
19      $\text{rotulo} \leftarrow \text{rotulo} + 1$ 
20      $L(p) \leftarrow \text{rotulo}$ 
21      $\text{fila.insere}(p)$ 
22     enquanto não  $\text{fila.vazia}()$  faça
23        $q \leftarrow \text{fila.remove}()$ 
24       para cada  $v \in N(q)$  faça
25         se  $L(v) = -1$  e  $I(v) = \text{corAtual}$  então
26            $L(v) \leftarrow \text{rotulo}$ 
27            $\text{fila.insere}(v)$ 

```

2.3.1 Gradiente morfológico

Um dos operadores derivados a partir da dilatação e da erosão é o *gradiente morfológico*, definido pela Eq. 2.3.

$$[\nabla_B(I)](x) = [\delta_B(I)](x) - [\varepsilon_B(I)](x) \quad (2.3)$$

O gradiente morfológico possibilita achar as regiões da imagem em que há maior variação dos níveis de cinza entre *pixels* adjacentes. Como uma borda (limite entre dois objetos) é uma região onde os níveis de cinza variam mais, o gradiente responde mais forte (resultando em *pixels* com nível de cinza mais claros) nessas regiões do que no restante da imagem. Na Fig. 2.4, observamos o realce das bordas no gradiente morfológico da imagem original, utilizando o quadrado 3x3 como elemento estruturante.

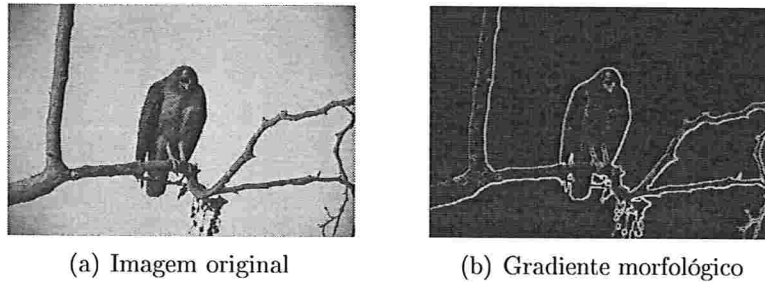


Figura 2.4: Imagem e seu gradiente morfológico.

2.4 Segmentação

Os algoritmos existentes para segmentação de imagens baseiam-se geralmente na descontinuidade ou similaridade dos níveis de cinza dos *pixels*. Dentre as diversas abordagens, podemos citar as baseadas em limiarização, crescimento de regiões, redes neurais, métodos estatísticos, transformada de Fourier, transformada de *wavelet* e na transformação *watershed*.

Para exemplificar as dificuldades existentes em aplicar técnicas automáticas de segmentação, como citado no Capítulo 1, iremos mostrar uma das técnicas mais intuitivas para a segmentação, chamada de limiarização, ou *thresholding*, especificada no Algoritmo 2. A ideia por trás desta técnica é que os objetos de interesse têm níveis de cinza diferentes em relação ao fundo da imagem, logo, se for possível escolher um limiar adequado, separamos facilmente os objetos do fundo. A escolha do limiar pode ser feita de forma automática,

por exemplo, através da análise do histograma da imagem, que corresponde a um gráfico com a frequência dos níveis de cinza nos *pixels* da imagem.

Algoritmo 2: Limiarização	
1	$L = \text{thresholding}(I, \lambda)$
2	I : imagem de entrada
3	L : imagem binária de saída
4	λ : limiar de separação
5	para cada $p \in I$ faça
6	se $I(p) \leq \lambda$ então
7	$L(p) \leftarrow 0$
8	senão
9	$L(p) \leftarrow 255$

Na Fig. 2.5, temos um exemplo simples de segmentação por limiarização, abordagem comum em sistemas para reconhecimento de caracteres.

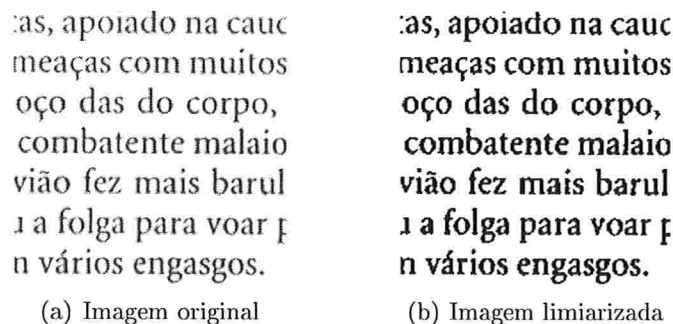


Figura 2.5: Segmentação por limiarização.

Em certos casos, é muito difícil, ou até mesmo impossível, encontrar um limiar adequado. Por exemplo, para segmentar a imagem da Fig. 2.4(a) a partir da limiarização de seu gradiente morfológico invertido (figura Fig. 2.6(a)), se escolhermos um limiar muito baixo, nem todas as bordas são mantidas, como na Fig. 2.6(b). Aumentando o limiar de forma a manter mais bordas, acabamos mantendo também detalhes da textura dos objetos, como na figura Fig. 2.6(c), o que pode prejudicar a análise posterior. Além de não conseguirmos especificar o nível de detalhes resultante da segmentação, as bordas obtidas são grosseiras, sendo necessário um pós-processamento para deixá-las com uma espessura adequada (1 *pixel*).

Uma alternativa à limiarização é a segmentação através da localização e filtragem de

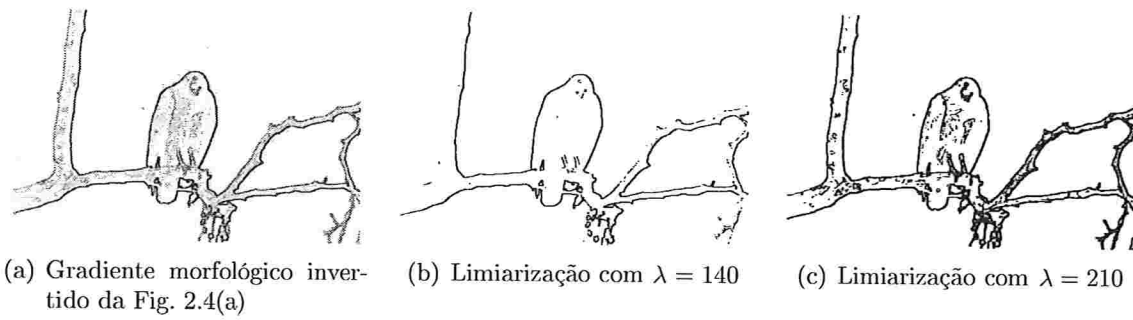


Figura 2.6: Efeitos da escolha de diferentes limiares.

máximos e mínimos regionais por operadores morfológicos como reconstrução por abertura, conforme descrito em [8].

No Capítulo 3, iremos mostrar detalhes da transformação *watershed*, ferramenta clássica da morfologia matemática para segmentação de imagens.

Transformação *watershed*

O estudo de *watersheds* (linhas de partição de águas) é um clássico na área de topografia. Seu uso em problemas de segmentação foi introduzido por S. Beucher e C. Lantuejoul [4] e o problema de super-segmentação de tal abordagem foi solucionado posteriormente por F. Meyer [19]. Em [23, 22] é revisada a transformação *watershed* e seu uso na área de morfologia matemática para segmentação de imagens digitais.

Neste capítulo descrevemos a transformação *watershed* e seus paradigmas, enfatizando os detalhes relevantes para as transições entre o *watershed* a partir de marcadores e o *watershed* hierárquico, como estruturas de dados e algoritmos utilizados.

3.1 *Watershed* clássico

A definição de *watershed* é dada por uma formulação física bem intuitiva. Considere uma superfície topográfica e gotas de água caindo sobre tal superfície. Uma bacia de retenção é dada pela região tal que, quando gotas de água caem sobre tal região, estas percorrem o relevo da superfície, através de fluxos descendentes, até chegar a um mesmo mínimo da superfície. As linhas que separam diferentes bacias de retenção são chamadas de *watersheds*, linhas sobre as quais, quando gotas de água caem sobre elas, não é possível afirmar para qual mínimo irão escoar.

Uma formulação alternativa e que é facilmente expressa na forma de algoritmo para a transformação *watershed* é baseada na simulação de inundação: considerando a imagem de entrada em níveis de cinza como uma superfície topográfica, o objetivo é produzir linhas de divisão de águas nesta superfície. Para tal, um furo é feito em cada mínimo regional¹ m_k da superfície, que é, então, submersa a uma taxa constante, de modo que a água entre

¹Um *mínimo regional* em uma imagem em níveis de cinza é uma zona plana não adjacente a nenhuma

pelos mínimos regionais. Quando frentes de água, vindas de diferentes mínimos regionais, estão prestes a se encontrar, uma barreira é construída para evitar tal encontro. Em algum momento, o processo chega a um estado tal que somente os topos das barreiras estão visíveis acima do nível da água, correspondendo às linhas de *watershed*. Dessa forma, a cada mínimo regional m_k é associada uma *bacia de retenção* CB_k .

No contexto do *watershed clássico*, em que furos são feitos em cada mínimo regional da superfície considerada, cada bacia de retenção é chamada de *bacia primitiva*. Em [26] é descrito um algoritmo eficiente para o *watershed clássico*.

A transformação *watershed* geralmente é aplicada no gradiente morfológico da imagem que se deseja analisar, pois este filtro morfológico responde mais fortemente nas bordas dos objetos. Dessa forma, quando a transformação *watershed* é aplicada, induzimos o aparecimento de linhas de divisão de águas sobre as bordas dos objetos.

Um problema do *watershed clássico* é a super-segmentação da imagem conforme observamos na figura Fig. 3.1, uma vez que o gradiente morfológico costuma apresentar muitos mínimos regionais, principalmente devido a ruído e texturas na imagem original.

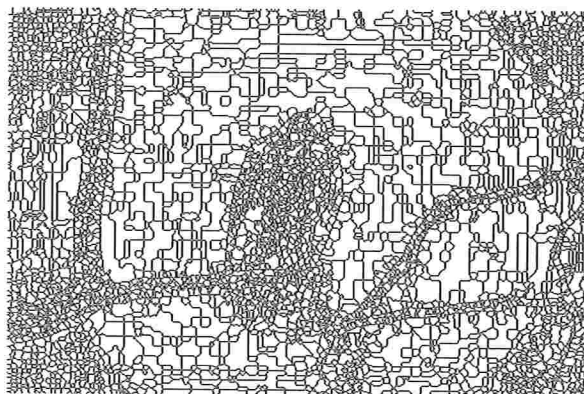


Figura 3.1: Partição super-segmentada da Fig. 2.4.

3.2 *Watershed* a partir de marcadores

Uma abordagem alternativa para contornar o problema de super-segmentação é a utilização de marcadores: um furo é feito na posição de cada marcador, e o processo de inundação segue da mesma forma como no *watershed clássico*. Assim, obtemos uma bacia de retenção para cada marcador. O problema da segmentação reduz-se, então, a localizar um conjunto

outra zona plana com menor altitude (nível de cinza). Uma *zona plana* é um componente conexo maximal cujos *pixels* têm todos o mesmo nível de cinza.

de marcadores adequados para os objetos da imagem a ser analisada. Esta abordagem é conhecida como *watershed a partir de marcadores*. A localização, formato e tamanho exatos dos marcadores não é importante, o que torna essa abordagem bastante robusta.

No processo de inundação por marcadores, ilustrado na Fig. 3.2, a água vai entrando pelos furos correspondentes aos diferentes marcadores, submergindo inicialmente as bacias primitivas nas quais estão localizados os marcadores. Ao longo do processo de inundação, quando o nível da água ultrapassa a altura mínima das bordas com bacias primitivas vizinhas, estas são submersas. A inundação prossegue até que todo o relevo correspondente à imagem esteja submerso.

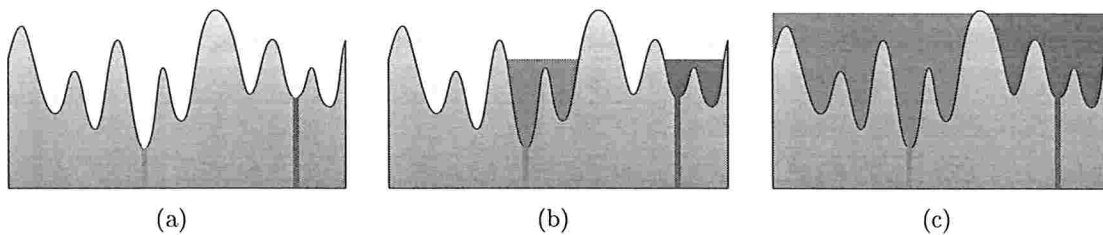


Figura 3.2: Inundação por marcadores.

Conforme observamos na Fig. 3.3, para cada marcador obtemos uma região de interesse na partição correspondente.

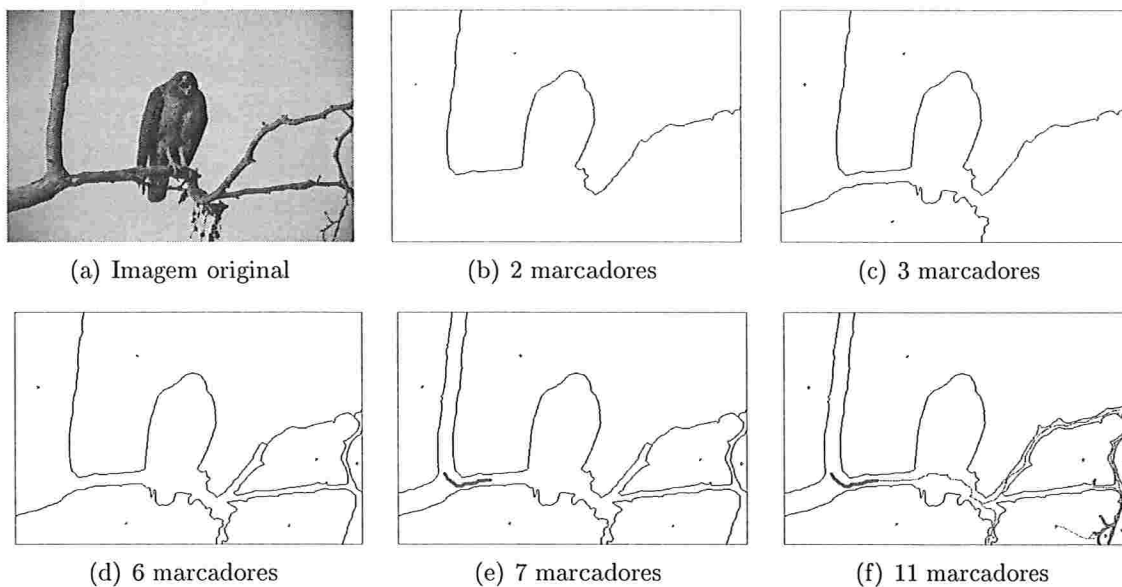


Figura 3.3: Partições obtidas através da inundação por marcadores.

3.2.1 Watershed através da *Image foresting transform*

A abordagem de *watershed* a partir de marcadores pode ser reformulada como um problema de otimização em grafos através da *image foresting transform* (IFT) [9], correspondendo à criação de uma floresta na qual existe um caminho de custo mínimo entre cada *pixel* e o conjunto de marcadores (o algoritmo da IFT é uma generalização do algoritmo de Dijkstra). A imagem I é vista como um grafo com pesos $G_I = (V, E, \omega)$, onde ω é uma função de dissimilaridade.

$$\omega(u, v) = \max\{I(u), I(v)\} \quad (3.1)$$

Tipicamente a imagem de entrada I é dada pelo gradiente da imagem de interesse.

A *transformação watershed pela IFT* (IFT-WS) assume que a função f_{max} é utilizada para calcular os custos dos caminhos:

$$\begin{aligned} f_{max}(\langle v_1, v_2, \dots, v_n \rangle) &= \max_{1 \leq i < n} \{\omega(v_i, v_{i+1})\} \\ &= \max_{1 \leq i \leq n} \{I(v_i)\} \end{aligned} \quad (3.2)$$

Note que, ao acrescentar uma aresta (v_{n-1}, v_n) a um caminho $\pi = \langle v_1, v_2, \dots, v_{n-1} \rangle$ pré-existente, o custo do novo caminho pode ser calculado em tempo constante a partir do custo de π :

$$\begin{aligned} f_{max}(\langle v_1, v_2, \dots, v_{n-1} \rangle \cdot (v_{n-1}, v_n)) &= f_{max}(\pi \cdot (v_{n-1}, v_n)) \\ &= \max\{f_{max}(\pi), \omega(v_{n-1}, v_n)\} \end{aligned} \quad (3.3)$$

A floresta resultante P , formada por árvores de predecessores, define uma partição da imagem I , na qual cada conjunto de árvores cujas raízes têm o mesmo rótulo corresponde a uma região de interesse da partição. Denotamos essa partição por $IFT-WS(G_I, M)$, onde I é a imagem de entrada e M é o conjunto de marcadores.

No Algoritmo 3, é apresentada uma versão modificada do algoritmo para *IFT-WS* exibida em [9], com a seguinte alteração:

- marcação de *pixels* como pertencentes à linha de *watershed* (linhas 33 – 34), como no algoritmo descrito em [7].

Um problema existente na utilização da função de custo f_{max} é conhecido como “pro-

Algoritmo 3: Watershed através da IFT

```

1  $L = IFT-WS(G_I, M)$ 
2  $G_I = (V, E, \omega)$ : grafo derivado a partir dos pixels da imagem  $I$ 
3  $M$ : conjuntos de marcadores (previamente rotulados por  $\lambda$ )
4  $L$ : rotulação (partição) dos pixels de  $I$ 
5  $P$ : árvores de predecessores (floresta de caminhos de custo mínimos)
6  $C$ : custo mínimo dos caminhos de algum marcador até cada pixel
7  $WS$ : conjunto dos pixels localizados na linha de watershed
8  $Q$ : fila de prioridades (política First In First Out)

9 1. Inicialização
10  $WS \leftarrow \emptyset$ 
11 para cada  $p \in V$  faça
12    $P(p) \leftarrow nil$ 
13   se  $p \in M$  então
14      $L(p) \leftarrow \lambda(p)$ 
15      $C(p) \leftarrow I(p)$ 
16      $Q.insere(p, C(p))$ 
17   senão
18      $L(p) \leftarrow 0$ 
19      $C(p) \leftarrow +\infty$ 

20 2. Propagação
21 enquanto não  $Q.vazia()$  faça
22    $s \leftarrow Q.remove()$ 
23   para cada  $t \in N(s)$  faça
24     se  $C(t) > C(s)$  então
25        $C' \leftarrow f_{max}(\pi^*(M, s) \cdot \langle s, t \rangle)$ 
26       se  $C' < C(t)$  então
27         se  $C(t) \neq +\infty$  então
28            $Q.remove(t, C(t))$ 
29            $P(t) \leftarrow s$ 
30            $C(t) \leftarrow C'$ 
31            $L(t) \leftarrow L(s)$ 
32            $Q.insere(t, C(t))$ 
33   se  $L(t) \neq L(s)$  e  $t \notin WS$  e  $s \notin WS$  então
34      $WS \leftarrow WS \cup \{s\}$ 

```

blema do planalto”, caracterizado pela existência de caminhos com mesmo custo a partir de diferentes marcadores até zonas planas que não são mínimos regionais (planaltos). Neste caso, a rotulação dos *pixels* do planalto depende de detalhes da implementação, como por exemplo, ordem na qual os *pixels* marcadores são percorridos, ordem na qual os vizinhos de um dado *pixel* são percorridos e política de acesso aos elementos da fila. Uma solução para tal problema é a utilização de uma função de custo lexicográfico com dois componentes, $f_{lex} = (f_{max}, f_d)$, no qual f_d corresponde à distância à borda do planalto, fazendo com que diferentes frentes de água inundem o planalto com uma mesma velocidade:

$$f_d(\langle v_1, \dots, v_n \rangle) = \max_{k \in [0, n-1]} \{k, f_{max}(\langle v_1, \dots, v_n \rangle) = f_{max}(\langle v_1, \dots, v_{n-k} \rangle)\} \quad (3.4)$$

Note que, ao acrescentar uma aresta (v_{n-1}, v_n) a um caminho $\pi = \langle v_1, v_2, \dots, v_{n-1} \rangle$ pré-existente, o custo f_{lex} do novo caminho pode ser calculado em tempo constante a partir do custo de π :

$$f_{lex}(\pi \cdot (v_{n-1}, v_n)) = \begin{cases} (\omega(v_{n-1}, v_n), 0) & \text{se } \omega(v_{n-1}, v_n) > f_{max}(\pi) \\ (f_{max}(\pi), f_d(\pi) + 1) & \text{c.c.} \end{cases}$$

Como $IFT-WS(G_I, M)$ pode ter diferentes soluções ótimas, a *tie-zone watershed* [2] visa determinar uma solução ótima única associada ao grafo G_I de uma imagem e ao conjunto M de marcadores: um *pixel* de I é associado a uma bacia de retenção se em todas as soluções possíveis ele pertence a tal bacia de retenção. Caso contrário, é dito que o *pixel* pertence à zona de empate, denotada por $TZ-WS(G_I, M)$.

3.2.2 Marcadores nas bacias primitivas

A abordagem de marcadores também pode ser utilizada em um grafo derivado da partição do *watershed* clássico: o *grafo de adjacência de regiões*, denotado por RAG^2 .

No RAG , para cada bacia primitiva (correspondente a um mínimo regional da imagem) criamos um vértice. Para cada par de bacias primitivas vizinhas, ligamos os vértices correspondentes através de uma aresta que tem como peso o nível de cinza mínimo na borda entre as duas bacias, valor que corresponde ao nível de água em que as bacias primitivas em questão iriam se fundir, ou seja:

$$\omega(CB_i, CB_j) = \min_{\substack{p_i \in CB_i \\ p_j \in CB_j \\ p_j \in N(p_i)}} \{\omega(p_i, p_j)\}.$$

² RAG : do inglês *region adjacency graph*

No RAG há menos vértices que em G_I e, por esse motivo, esse grafo é utilizado por questões de eficiência computacional. Podemos utilizar o RAG também em situações em que não há necessidade da precisão no nível dos *pixels* para a partição de interesse.

O Algoritmo 3 pode ser aplicado utilizando o RAG como grafo de entrada com apenas duas modificações:

- não faz sentido definir o conjunto WS , pois neste caso os vértices são bacias primitivas e não *pixels*, logo, as linhas 33 – 34 devem ser excluídas do algoritmo;
- o conjunto de marcadores M deve ser pré-processado de forma a não haver mais de uma rotulação possível para uma bacia primitiva. O rótulo de cada marcador $m \in M$ deve ser mapeado para a bacia primitiva a qual pertence, logo, se houverem marcadores com rótulos diferentes dentro de uma mesma bacia primitiva, então tal bacia poderia assumir diferentes rótulos, dependendo da ordem de processamento dos marcadores. Para que uma única rotulação seja possível, podemos excluir a intersecção de múltiplos marcadores com cada bacia primitiva se processarmos os marcadores de forma *offline*. No contexto de uma aplicação interativa, em que o algoritmo será aplicado a cada edição de marcadores, podemos processar os marcadores de forma *online*: ao se criar um marcador, as intersecções de marcadores pré-existentes com as bacias primitivas pelas quais o novo marcador passa devem ser excluídas, priorizando então o rótulo do novo marcador.

3.3 Watershed hierárquico

Outra abordagem utilizada para contornar o problema de super-segmentação obtido com o *watershed* clássico é o *watershed hierárquico*, através do qual é construído um conjunto hierárquico de partições da imagem, no qual partições mais finas ficam aninhadas sob partições mais grossas.

3.3.1 Construção da hierarquia de partições

A construção da hierarquia de partições de uma imagem, ilustrada na Fig. 3.4, pode ser feita da seguinte maneira:

- O processo de inundação se inicia como no *watershed* clássico, com a água entrando pelos diferentes mínimos regionais, a partir dos quais surgem as bacias primitivas, que constituem a partição mais fina da hierarquia, conforme a Fig. 3.4(a). Para cada bacia primitiva é criado um vértice que será uma folha na árvore que representa a hierarquia das partições.

- No momento em que frentes de água vindas de diferentes bacias se encontram, fundimos as bacias em questão numa nova bacia de retenção. Para representar este evento, é criado um novo vértice na hierarquia, que é colocado em um nível acima dos vértices correspondentes às bacias que se fundiram. Note que podemos fundir bacias primitivas e/ou bacias de retenção criadas por fusões anteriores. Na Fig. 3.4(b) pode ser observada a fusão das bacias primitivas rotuladas por B e C numa nova bacia com rótulo D.
- O processo de fusão de bacias adjacentes continua até que todas as bacias sejam fundidas em uma única região. Nesse ponto, a bacia de retenção que engloba todos os *pixels* da imagem define a partição mais grossa da hierarquia. O vértice correspondente a essa bacia é a raiz da hierarquia das partições (Fig. 3.4(c)).

3.3.2 Representação da hierarquia de partições

Nesta seção iremos detalhar a representação de partições de imagens através de hierarquias e as estruturas de dados envolvidas na construção descrita na Seção 3.3.1.

A partição mais fina da hierarquia é formada pelas bacias primitivas, correspondendo ao resultado obtido pelo *watershed* clássico, partição definida pelo RAG.

A partir do RAG podemos obter as demais partições da hierarquia, fundindo bacias primitivas vizinhas que são ligadas por arestas com peso menor ou igual a um determinado limiar. Aumentando tal limiar, que representa o nível de água num cenário de simulação de imersão, fundimos mais bacias, obtendo partições mais grossas da imagem. O processo é ilustrado na Fig. 3.5, na qual cada componente da partição, indicado em azul, é formado por um subconjunto dos vértices do RAG. Cada região da partição é um componente conexo resultante da supressão das arestas do RAG com peso maior que o limiar especificado.

Analisando o processo de fusão na Fig. 3.5, percebe-se que somente as arestas com menores pesos entre as arestas que ligam diferentes componentes conexos são importantes no processo de inundação, ou seja, a AGM do RAG contém toda a informação do RAG relevante ao processo de inundação. De fato, realizando a inundação da AGM obtemos o mesmo resultado obtido inundando todo o RAG, conforme pode ser observado comparando-se a Fig. 3.5 com a Fig. 3.6. Note que, inundando a AGM, cada componente da partição selecionada pelo limiar corresponde a uma AGM do componente selecionado pelo mesmo limiar quando utilizado no processo de inundação levando em consideração todas as arestas do RAG. Uma partição é representada, então, por uma floresta geradora.

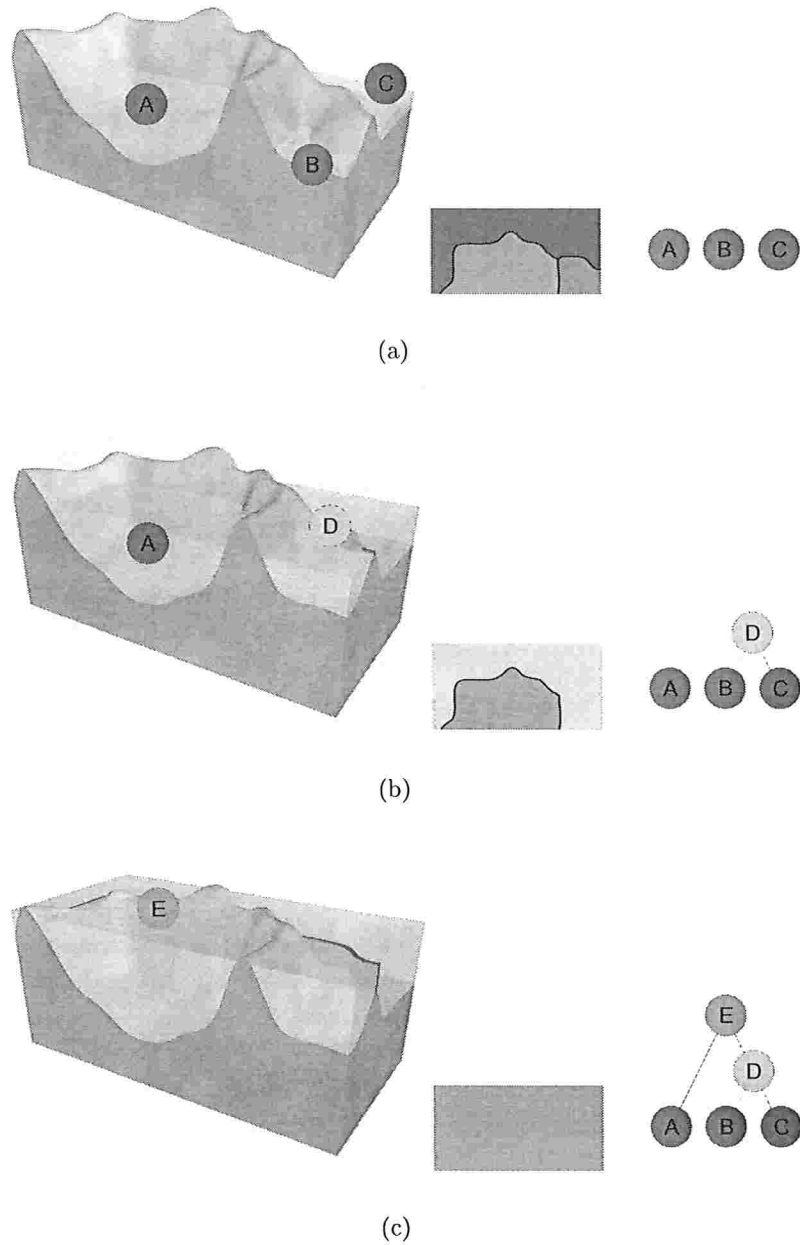


Figura 3.4: Construção de uma hierarquia de partições: (a) Bacias primitivas definindo a partição mais fina da hierarquia. (b) Partição intermediária. (c) Partição mais grossa, na qual todas as bacias de retenção estão fundidas em uma única região.

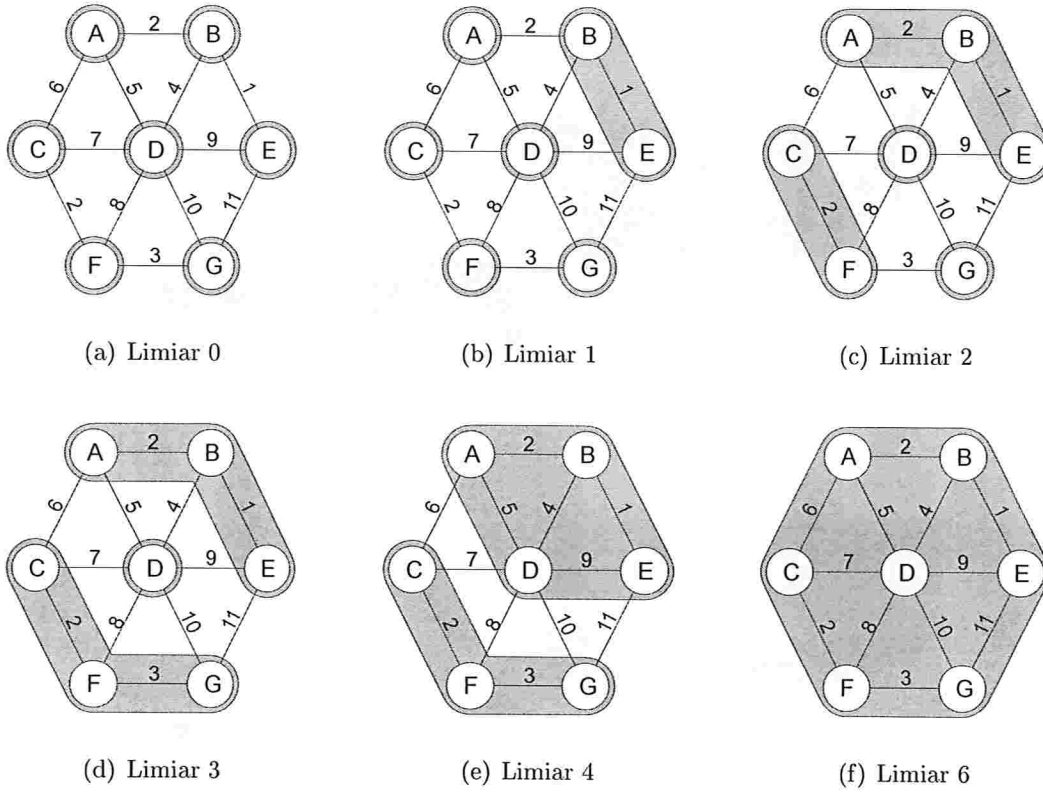


Figura 3.5: Partições representadas como fusões de bacias primitivas.

3.3.2.1 Localização de partições na hierarquia

Com a estrutura hierárquica construída de forma automática, o problema de segmentação passa a ser o de localizar a partição desejada na hierarquia, o que pode ser feito através das seguintes operações, como proposto em [2S]:

- escolha do número de regiões³, através da seleção de um limiar adequado para fusão de bacias primitivas, conforme o processo descrito na Seção 3.3.2;
- re-segmentação de uma região, através da supressão de uma aresta de peso máximo na árvore correspondente a região em questão;

³Nem sempre é possível encontrar uma partição com um determinado número de regiões somente pela escolha de um limiar, uma vez que os pesos das arestas da AGM podem não ser únicos. Se for de interesse obter diferentes partições com 1 a N regiões (sendo N o número de bacias primitivas), os pesos das arestas da AGM podem ser ligeiramente modificados, de forma que não existam duas arestas com o mesmo peso na AGM.

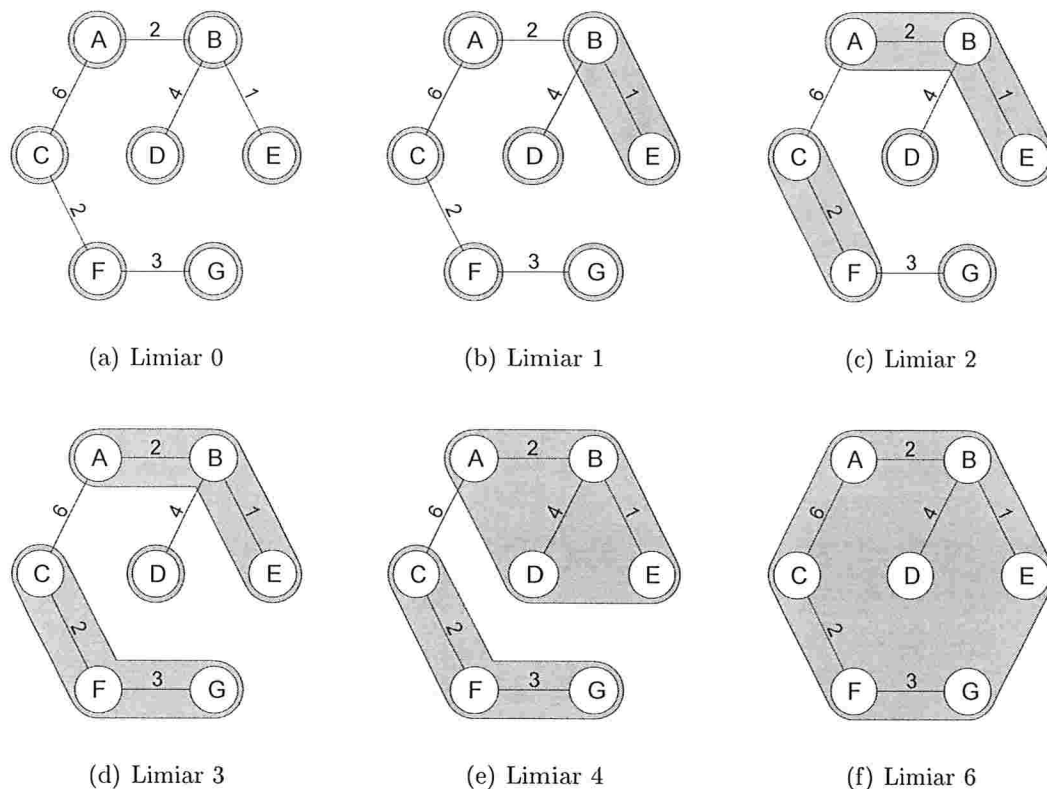


Figura 3.6: Partições obtidas utilizando somente a árvore geradora mínima.

- fusão de regiões, através da inclusão de uma aresta de peso mínimo que liga a árvore correspondente à região selecionada com alguma outra árvore da floresta que representa a partição.

Para realizar as operações descritas na lista acima sem a necessidade de fazer buscas nas arestas da AGM pela ordem de seus pesos, devemos armazenar a hierarquia de uma forma que seja possível acessar as arestas de acordo com tal ordem⁴. Uma estrutura eficiente para tal, descrita em [20], é a *árvore dos lagos críticos*, denotada por ALC. A ALC é justamente a estrutura hierárquica descrita na Seção 3.3.1, e pode ser obtida a partir da AGM através do Algoritmo 4. Para ilustrar a execução do algoritmo, exibimos na Fig. 3.7 a transformação de uma AGM em uma ALC. Note que o caminho de qualquer vértice até a raiz da ALC percorre uma seqüência de arestas cujos pesos estão em ordem não-decrescente.

⁴Note que os pesos das arestas do RAG podem não ser únicos. Não é utilizado nenhum critério para desempatar arestas com mesmo peso, ou seja, a AGM e a ALC obtidas são geradas a partir de ordenações arbitrárias entre arestas de mesmo peso.

Algoritmo 4: Construção da ALC a partir da AGM

- 1 $arestasOrdenadas \leftarrow$ arestas da AGM ordenadas por peso
- 2 **para cada** aresta a de $arestasOrdenadas$ **faça**
- 3 Sejam $b1$ e $b2$ os vértices ligados por a e p o peso da aresta a
- 4 Crie um novo vértice $lagoCritico$
- 5 Ligue os vizinhos de $b1$ e $b2$ a $lagoCritico$
- 6 Crie uma aresta orientada de $b1$ a $lagoCritico$ com peso p
- 7 Crie uma aresta orientada de $b2$ a $lagoCritico$ com peso p
- 8 Suprima a do conjunto de arestas

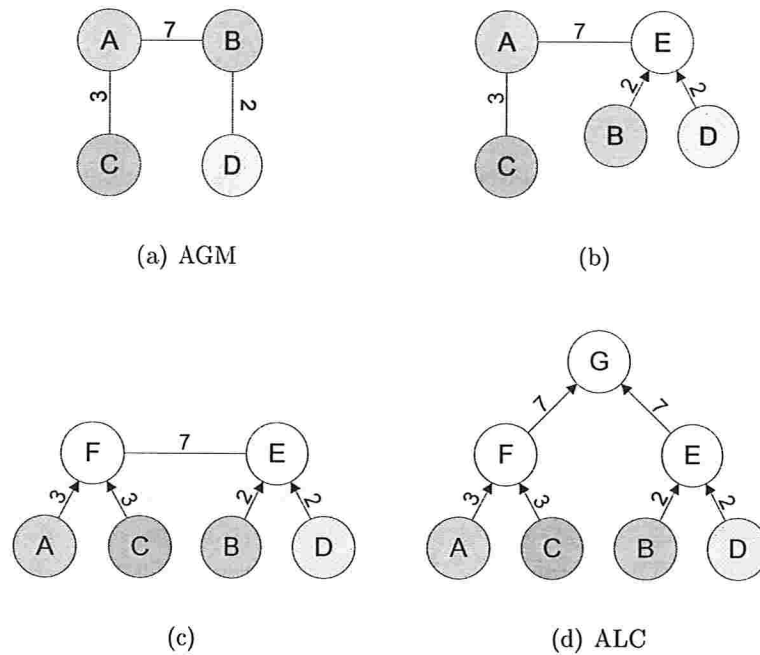


Figura 3.7: Construção da ALC a partir da AGM.

Com as arestas ordenadas por peso, as operações de supressão de arestas na floresta geradora que representa a partição podem ser implementadas como cortes na estrutura hierárquica da ALC (conforme ilustração utilizando uma hierarquia simplificada na Fig. 3.8):

- a seleção do limiar para fusão de bacias da partição mais fina determina um corte em um mesmo nível através de toda a ALC, como ilustrado na Fig. 3.8(c);
- a re-segmentação de uma região determina um corte em um nível abaixo do nível atual da região selecionada, como ilustrado na Fig. 3.8(d);
- a fusão de regiões determina um corte em um nível acima do nível atual da região selecionada, como ilustrado na Fig. 3.8(e).

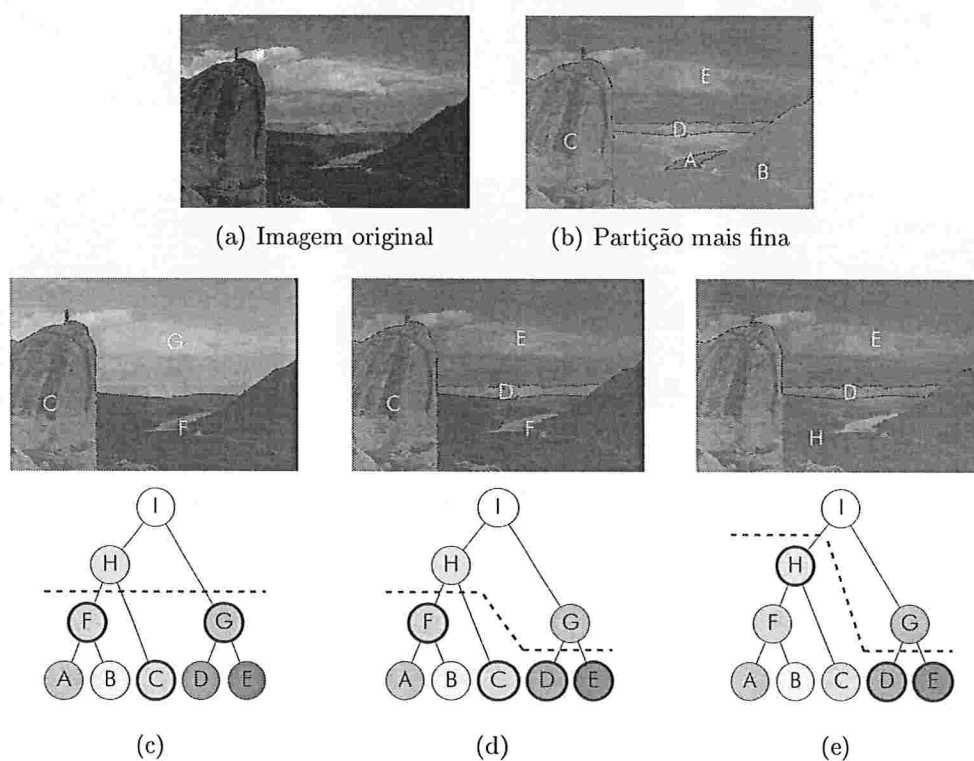


Figura 3.8: Localização de partições através de cortes na ALC: (c) 3 regiões (F, C e G) selecionadas por um único corte. (d) Região G dividida em D e E. (e) Regiões F e C fundidas em H.

3.3.2.2 Tipos de inundação

Inundação uniforme

O processo de inundação descrito até este ponto é chamado de inundação uniforme, pois o nível de água se mantém uniforme para todas as bacias de retenção. Na Fig. 3.9 exibimos algumas partições geradas por essa forma de inundação. Percebemos que não há uma relação muito forte entre esse processo de inundação e as possíveis regiões de interesse: as regiões das partições obtidas pela inundação uniforme dificilmente correspondem a regiões de interesse.

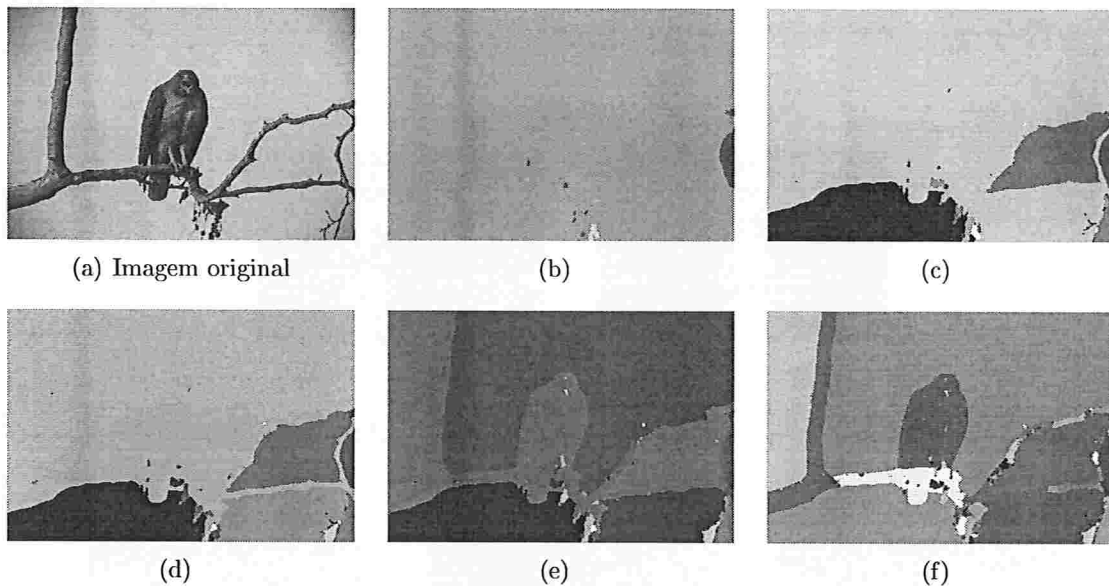


Figura 3.9: Partições geradas pela inundação uniforme.

Inundação síncrona

A seguir descrevemos o processo de inundação síncrona:

- Em cada mínimo regional da superfície é criada uma fonte de água, cuja vazão é controlada de forma a manter uma mesma métrica (profundidade, área ou volume) em todas as bacias de retenção.

- Quando o nível de água em uma bacia atinge o valor de sua dinâmica⁵, a vazão da fonte de água correspondente é interrompida totalmente.
- Conforme o nível de água vai aumentando, as bacias com menores métricas vão sendo fundidas com as bacias de maiores métricas, até que todas as bacias sejam fundidas em uma única (raiz da hierarquia).

Através desse processo de inundação síncrona, as partições nos níveis superiores da hierarquia são formadas por bacias com métricas maiores em relação às bacias das partições nos níveis inferiores. Utilizando diferentes métricas das bacias, privilegiamos diferentes características das regiões na construção da hierarquia de partições:

- Ao utilizarmos a *profundidade* das bacias, privilegiamos regiões de acordo com seus contrastes;
- Ao utilizarmos a *área* das bacias, privilegiamos regiões de acordo com seus tamanhos;
- Ao utilizarmos o *volume* das bacias, privilegiamos regiões de acordo com seus contrastes e tamanhos.

Utilizando tais critérios no processo de inundação síncrona, obtemos hierarquias com partições geralmente mais interessantes que as da hierarquia obtida pelo processo de inundação uniforme, conforme podemos observar na Fig. 3.10.

Um fato interessante que relaciona os diferentes tipos de inundação é que todos geram AGMs que são diferentes somente pelos pesos de suas arestas, como demonstrado em [21], já que a fusão de bacias vizinhas depende somente de suas dinâmicas, valores que independem do tipo de inundação. Por isso, podemos simular o processo de inundação síncrona utilizando uma AGM obtida anteriormente pelo processo de inundação uniforme, como descrito a seguir:

- inicialmente consideramos a floresta geradora formada apenas pelos vértices da AGM original;
- percorremos as arestas da AGM original em ordem de seus pesos e, para cada aresta a :
 - considere v_1 e v_2 os vértices ligados por a na AGM original e T_1 e T_2 as árvores da floresta geradora que contém os vértices v_1 e v_2 , respectivamente;

⁵A dinâmica de uma bacia é o valor máximo que o nível de água pode alcançar nessa bacia de forma que ela não escoe para uma bacia com mínimo regional de menor altitude.

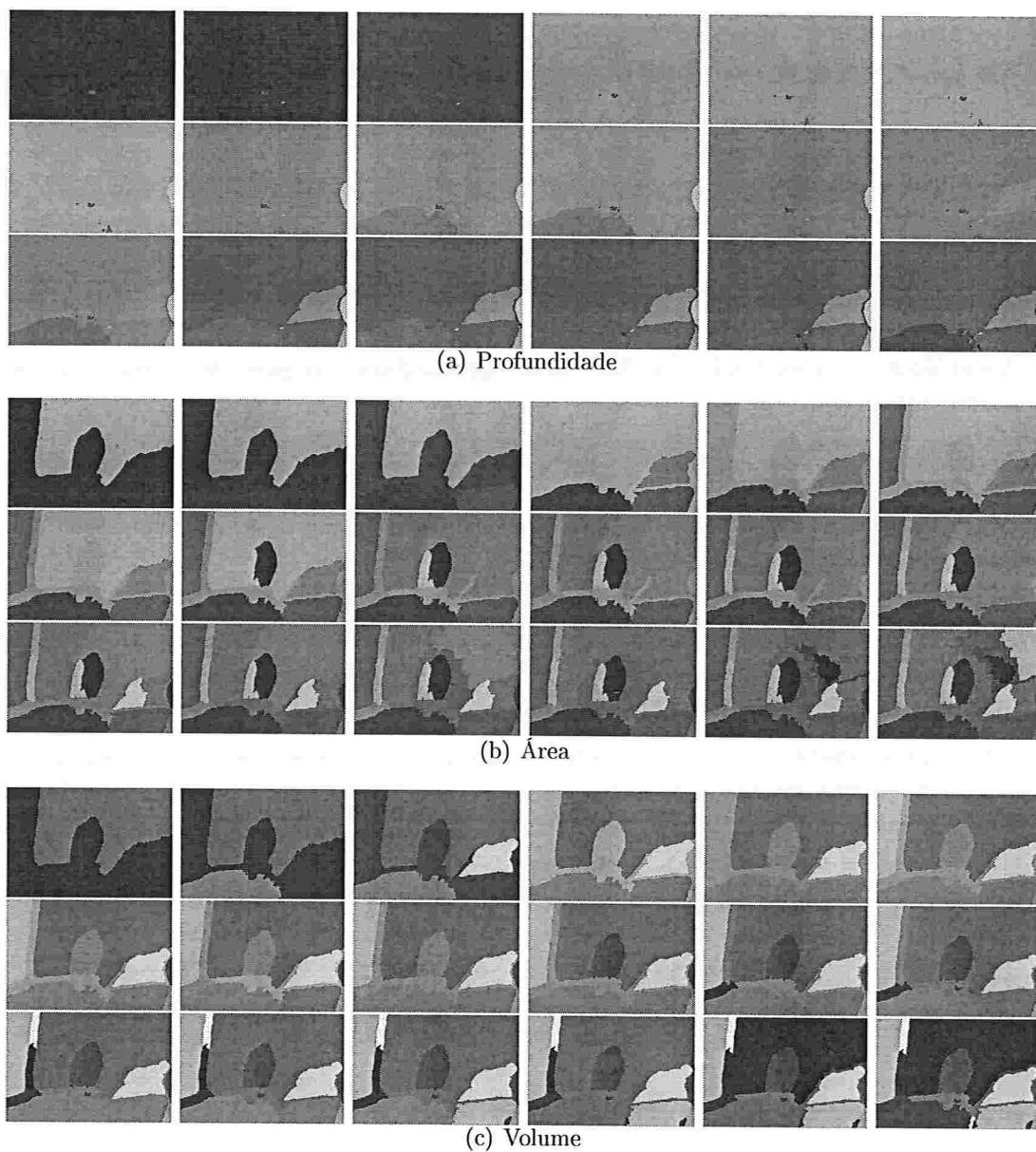


Figura 3.10: Partições da Fig. 3.9(a) geradas por inundações síncronas.

- para cada T_i , $i \in \{1, 2\}$, calculamos o valor m_i de sua métrica (profundidade, área ou volume) utilizando o conjunto P_i , formado pelos *pixels* da imagem I pertencentes às bacias primitivas cujos vértices correspondentes pertencem a T_i , e o nível de água n , definido pelo peso de a na AGM original;
 - definimos o novo peso de a como sendo o valor mínimo entre m_1 e m_2 ;
 - acrescentamos a na floresta geradora, ligando os vértices v_1 e v_2 .
- ao final do processo, obtemos uma AGM com a mesma estrutura da AGM original, mas com os pesos compatíveis com a métrica utilizada.

Para o conjunto de *pixels* P_i e o nível de água n indicados acima, definimos o conjunto P'_i por

$$P'_i = \{p \in P_i : I(p) \leq n\}.$$

Para P'_i e n calculamos o valor m_i da métrica desejada da seguinte forma:

Profundidade

$$m_i = n - \min_{p \in P'_i} \{I(p)\}$$

Área

$$m_i = |P'_i|$$

Volume

$$m_i = \sum_{p \in P'_i} n - I(p)$$

Alternando entre os paradigmas do *watershed*

As abordagens de *watershed* hierárquico e *watershed* a partir de marcadores podem ser utilizadas de forma integrada somente se for possível alternar entre ambas. Mais especificamente, dada uma partição P selecionada no conjunto hierárquico de partições de uma imagem I , é necessário achar um conjunto de marcadores M tal que $IFT-WS(G, M) = P$, onde G é um grafo derivado a partir de I com a precisão desejada (grafo dos *pixels* G_I ou um RAG de I). Reciprocamente, dado um conjunto de marcadores M e a partição correspondente $P = IFT-WS(G, M)$, desejamos obter um conjunto hierárquico de partições que inclua P .

Neste capítulo serão descritas maneiras possíveis para se alternar entre as abordagens do *watershed* consideradas. Além disso, serão caracterizadas as diferenças que podem surgir ao se utilizar diferentes granularidades em cada uma das abordagens.

4.1 Hierarquia para marcadores

Qualquer partição de uma imagem I pode ser obtida através da inundação por marcadores (considerando G_I como grafo de entrada): se para cada região da partição considerarmos o marcador formado por todos os *pixels* de tal região, obtemos um conjunto trivial de marcadores que gera exatamente a partição desejada. Em especial, se tal partição foi obtida através do *watershed* hierárquico, o conjunto trivial de marcadores recupera esta partição. Porém, desejamos obter um conjunto de marcadores que seja facilmente editável com o intuito de modificar a partição em questão, o que é impraticável para o conjunto descrito. Para tal iremos nos basear no *conjunto mínimo de marcadores* (MSS¹), que é o problema inverso da segmentação por *watershed*: dada uma partição obtida pelo *watershed*,

¹MSS: do inglês *minimal seed set*

consiste em encontrar um conjunto mínimo de marcadores que pode ser utilizado para obter a mesma partição. O MSS foi proposto inicialmente em [17], sendo calculado no RAG, e estendido em [1] para ser calculado diretamente no grafo G_I dos *pixels* da imagem.

Considere o grafo $G = (V, E, w)$ e a rotulação L_P de seus vértices derivada de uma partição $P = \{P_1, \dots, P_n\}$: cada região de interesse P_i é formada por um subconjunto conexo de V e é associada a um rótulo distinto $\lambda_i \in \Lambda$. Os *nós fronteira* $F(\lambda)$ de uma região de interesse rotulada por λ são dados por:

$$F(\lambda) = \{p \in V : \exists q \in N(p), L_P(q) \neq L_P(p) = \lambda\}$$

A *força* $S(p)$ de um nó fronteira p é dada por:

$$S(p) = \min\{\omega(p, q) : q \in N(p), L_P(q) \neq L_P(p)\}$$

A *região receptiva* (RR) $RR(p)$ associada ao nó fronteira p é dada por:

$$RR(p) = \{q \in V : L_P(q) = L_P(p), f_{max}(\pi^*(p, q)) < S(p)\} \quad (4.1)$$

Uma região receptiva é *redundante* (RRR) se contém estritamente alguma outra região receptiva:

$$RR(p) \text{ é redundante} \Leftrightarrow \exists q, RR(q) \subset RR(p) \quad (4.2)$$

e, neste caso, a força da RR contida é estritamente menor que a força da RR que a contém:

$$RR(q) \subset RR(p) \Rightarrow S(q) < S(p) \quad (4.3)$$

Para ilustrar o processo descrito, apresentamos na Fig. 4.1 uma partição em 2 regiões, para a qual desejamos obter as regiões receptiva não redundantes (NRRRs²). Nas Fig. 4.2 e Fig. 4.3 observa-se o fato de que uma RRR contém uma outra RR.

4.1.1 Criação de marcadores

Para recuperar a partição desejada, basta encontrar as NRRRs de cada região e, para cada uma delas, criar um marcador (contido totalmente na região) que passe por todas as NRRRs.

²NRRR: do inglês *non redundant receptive region*

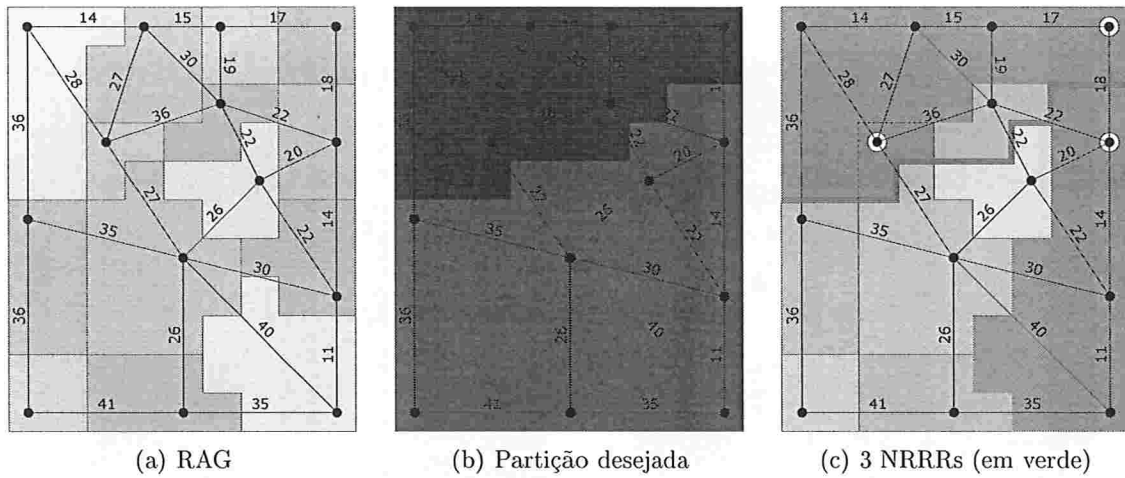


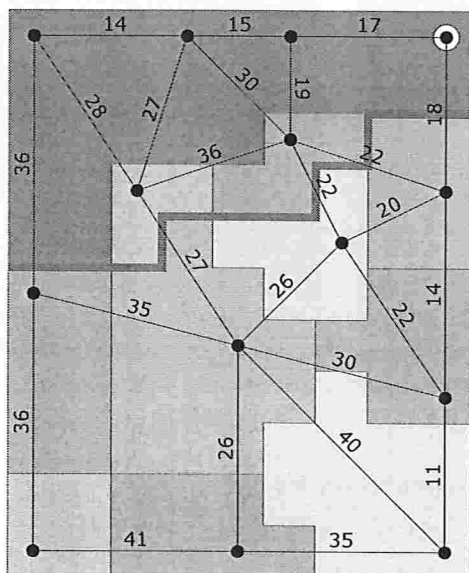
Figura 4.1: NRRRs de uma partição do RAG.

Caso uma região tenha apenas uma NRRR, é trivial desenhar um marcador que recupere tal região: basta marcar pelo menos um *pixel* em tal NRRR. Se a NRRR foi calculada no RAG, podemos marcar, por exemplo, o mínimo regional de alguma bacia primitiva de tal NRRR, ou então, para obter um marcador fácil de ser visualizado, marcar todos os *pixels* de alguma bacia primitiva da NRRR em questão. Alguns critérios para criação de marcadores mínimos são analisados em [6].

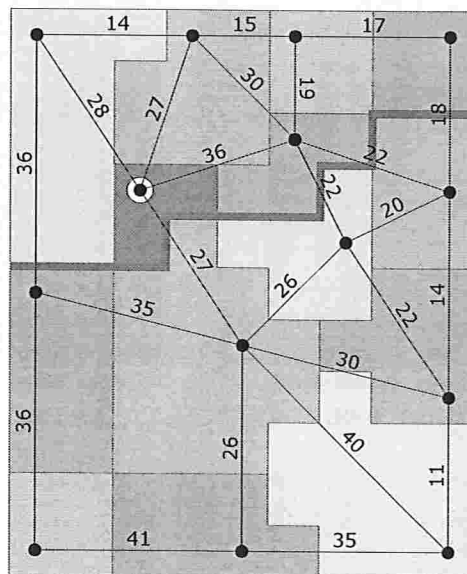
Caso uma região tenha mais de uma NRRR, desejamos criar um marcador conexo que passe por todas as NRRRs da região. Neste caso, podemos criar um pedaço do marcador em algum vértice de cada NRRR, como descrito acima, e conectá-los por algum caminho contido totalmente na região. Se as NRRRs foram calculadas no RAG podemos utilizar o caminho da AGM que liga os vértices marcados: para cada par de vértices consecutivos nesse caminho, ligamos seus mínimos regionais aos *pixels* que determinaram o valor da aresta que liga tais vértices através de caminhos não-decrescentes (dentro de uma bacia primitiva sempre existe um caminho não decrescente do mínimo regional a qualquer de seus *pixels*; por exemplo, todos os caminhos contidos nas árvores de predecessores resultantes da IFT quando os mínimos regionais são utilizados como marcadores têm essa propriedade).

4.1.2 MSS no RAG *versus* MSS em G_I

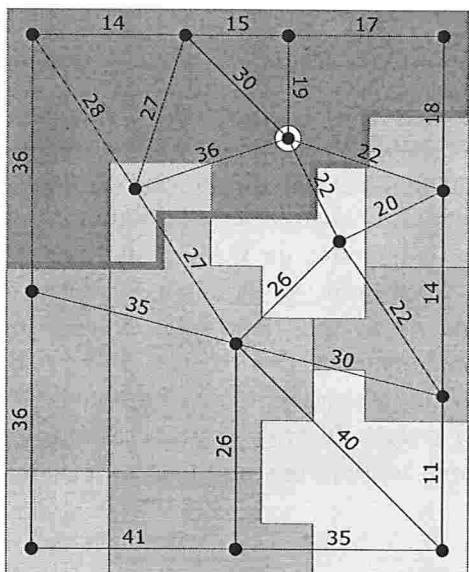
Dada uma partição P selecionada em uma hierarquia, o MSS pode ser calculado diretamente no RAG, já que a hierarquia é construída baseada neste grafo. Seja M um conjunto de marcadores construído nas NRRRs do RAG, como descrito na Seção 4.1.1. Só é possível garantir que M recupere exatamente a partição P se for aplicado de volta no RAG, onde as



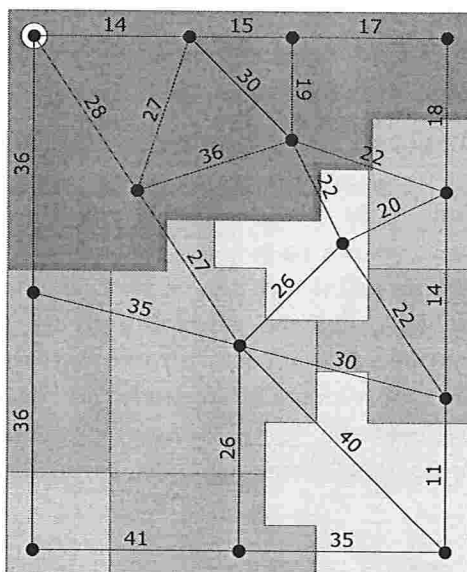
(a) NRRR associada ao nó fronteira de força 18



(b) NRRR associada ao nó fronteira de força 27

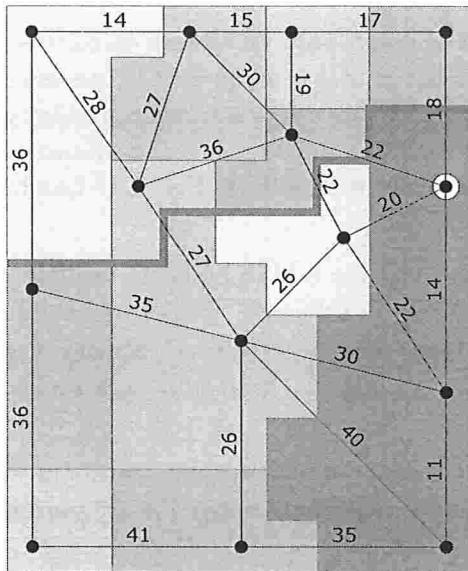


(c) RRR associada ao nó fronteira de força 22

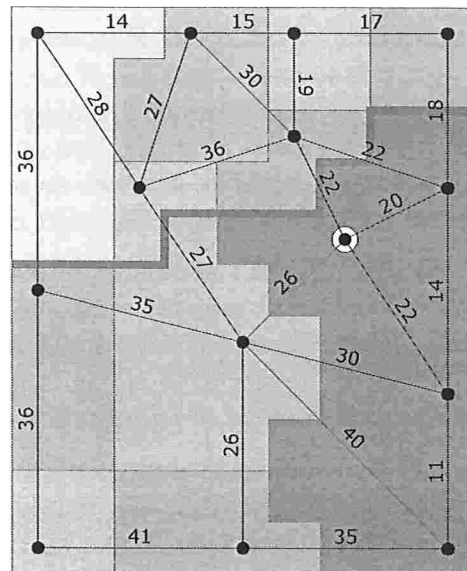


(d) RRR associada ao nó fronteira de força 36

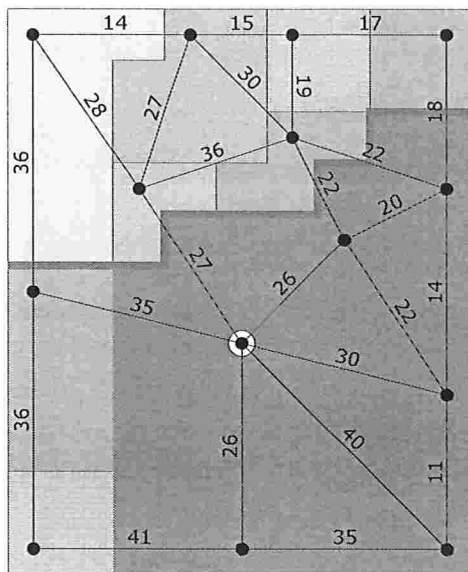
Figura 4.2: RRs da região azul da Fig. 4.1(b).



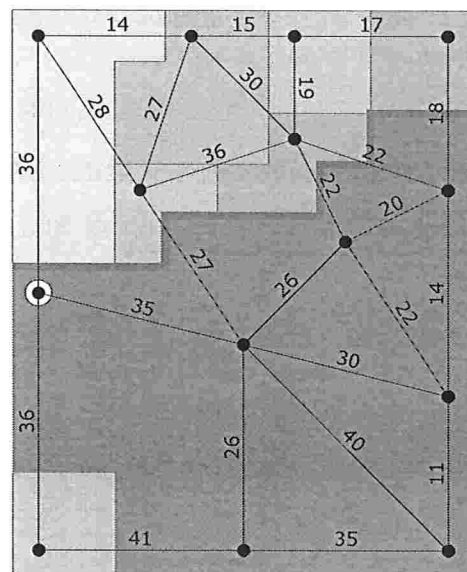
(a) NRRR associada ao nó fronteira de força 18



(b) RRR associada ao nó fronteira de força 22



(c) RRR associada ao nó fronteira de força 27



(d) RRR associada ao nó fronteira de força 36

Figura 4.3: RRs da região vermelha da Fig. 4.1(b).

bacias primitivas são consideradas unidades atômicas. Se M for aplicado em G_I , a partição resultante $P' = IFT-WS(G_I, M)$ pode diferir de P , como pode ser observado na Fig. 4.4.

Por outro lado, se o MSS de P é calculado em G_I , o conjunto resultante de marcadores M' recupera exatamente a partição P , i.e., $IFT-WS(G_I, M') = P$, já que M' garante que $TZ-WS(G_I, M') = \emptyset$. Entretanto, os marcadores obtidos desta forma tendem a se localizar nas bordas das regiões de interesse, como pode ser observado na Fig. 4.5. Esta figura exhibe as NRRRs de cada região de interesse da mesma imagem utilizada na Fig. 4.4. Lembre-se que M' deve incluir pelo menos um *pixel* de cada NRRR.

Em contrapartida, os marcadores M obtidos ao computar o MSS no RAG tendem a ser compactos e localizados mais ao centro de cada região de interesse. Experimentalmente, observamos que as diferenças entre a partição desejada P e a partição P' obtida a partir de tais marcadores localizam-se próximas às bordas das regiões. Esta evidência nos leva a presumir que as diferenças se encontram na TZ.

A seguir, é provado que uma partição P , selecionada na hierarquia, também é um solução ótima de $IFT-WS(G_I, M)$. Isto implica que as diferenças entre P e P' pertencem necessariamente a $TZ-WS(G_I, M)$.

Proposição 1. *Seja $P = \{P_1, P_2, \dots, P_n\}$ uma partição da imagem I obtida pelo watershed hierárquico. Considere um conjunto de marcadores $M = \{M_1, M_2, \dots, M_n\}$, calculados no RAG de I , onde M_i é o marcador associado a P_i , i.e., para $i = 1, \dots, n$:*

$$M_i \subseteq P_i \text{ e } M_i \cap R \neq \emptyset, \forall R \in NRRR(P_i) \quad (4.4)$$

onde $NRRR(P_i)$ denota o conjunto das NRRRs de P_i , calculadas no RAG de I .

P é uma solução ótima de $IFT-WS(G_I, M)$.

Demonstração. Seja $P' = \{P'_1, P'_2, \dots, P'_n\}$ uma solução ótima arbitrária de $IFT-WS(G_I, M)$. Se $\exists p$ tal que $\lambda_\alpha = L_P(p) \neq L_{P'}(p) = \lambda_\beta$, então é suficiente mostrar que $f_{max}(\pi^*(M_\alpha, p)) = f_{max}(\pi^*(M_\beta, p))$, onde M_α e M_β são os marcadores rotulados por λ_α e λ_β , respectivamente.

Sem perda de generalidade, suponha que $\lambda_\alpha = 1$ and $\lambda_\beta = 2$. Pela otimalidade de P' , segue que

$$f_{max}(\pi^*(M_2, p)) \leq f_{max}(\pi^*(M_1, p)). \quad (4.5)$$

O MSS é calculado de forma a não haver TZ, ou seja,

$$f_{max}(\pi^*(M_i, CB)) < f_{max}(\pi^*(M_j, CB)), i \neq j, \forall CB \in P_i \quad (4.6)$$

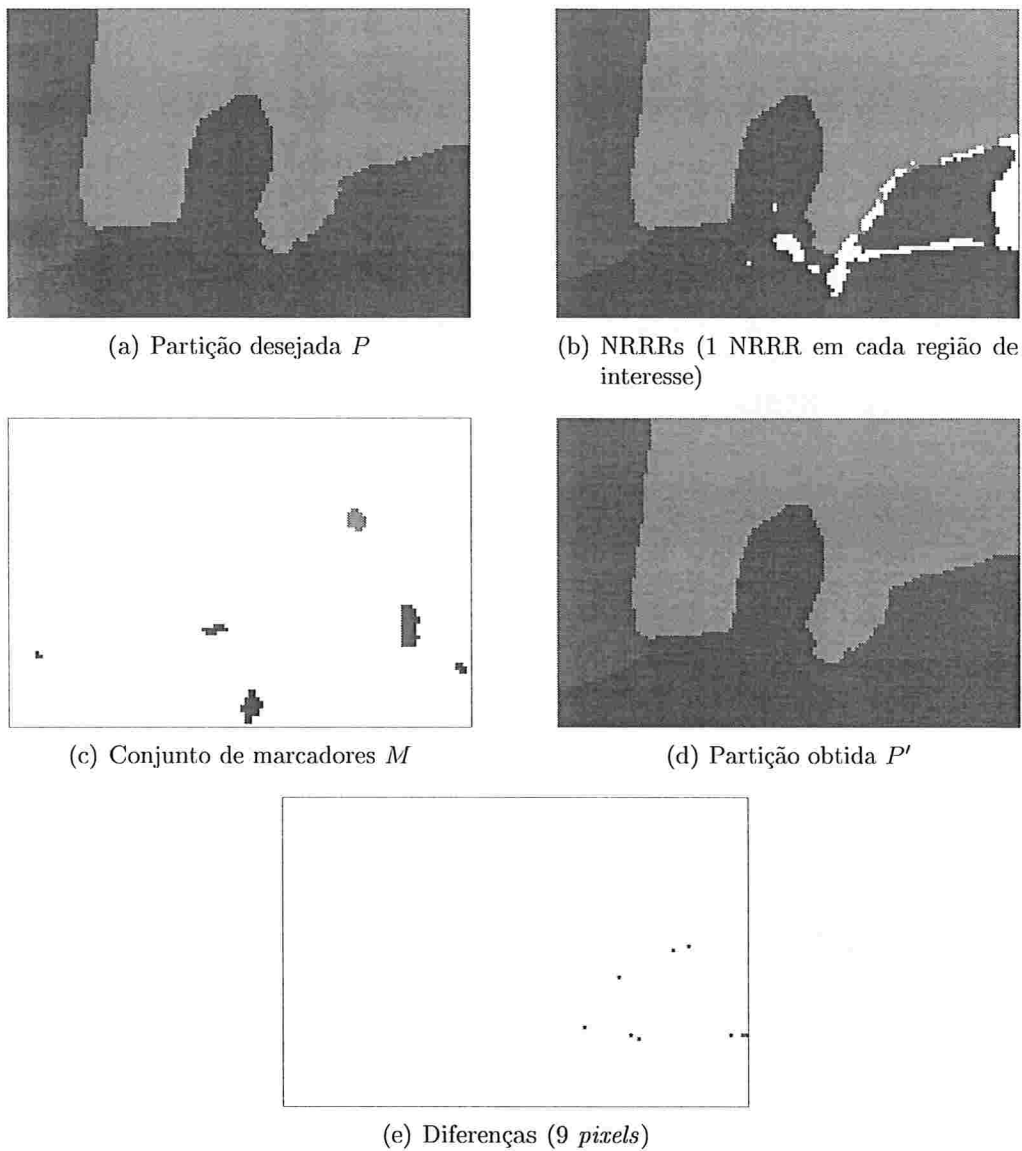


Figura 4.4: Marcadores calculados no RAG aplicados em G_I : (a) Partição desejada P , selecionada num conjunto hierárquico de partições. (b) NRRRs de cada região de interesse de P . (c) Marcadores selecionados nas NRRRs de P . (d) Partição obtida P' . (e) *Pixels* com rotulação diferente da desejada.

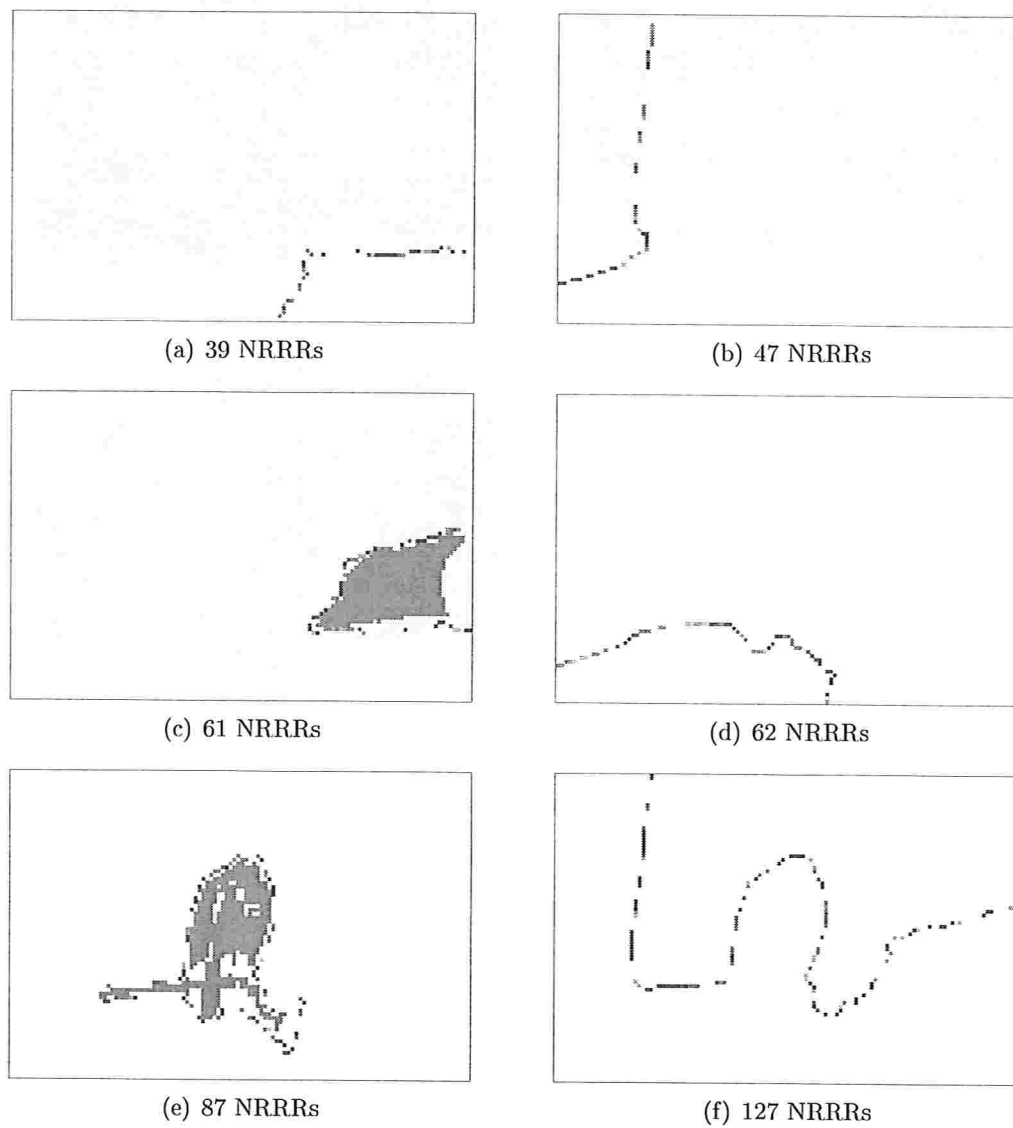


Figura 4.5: NRRRs de cada região de interesse, computadas em G_I .

Em especial,

$$f_{max}(\pi^*(M_1, V)) < f_{max}(\pi^*(M_2, V)) \quad (4.7)$$

onde V é a bacia primitiva a qual p pertence.

Para qualquer caminho $\langle CB_1, CB_2, \dots, CB_n \rangle$ no RAG de I existe um caminho $\langle r_1, \dots, r_2, \dots, r_n \rangle$ de mesmo custo em G_I , onde r_i é um *pixel* do mínimo regional da bacia primitiva CB_i (tal caminho é descrito na Seção 4.1.1).

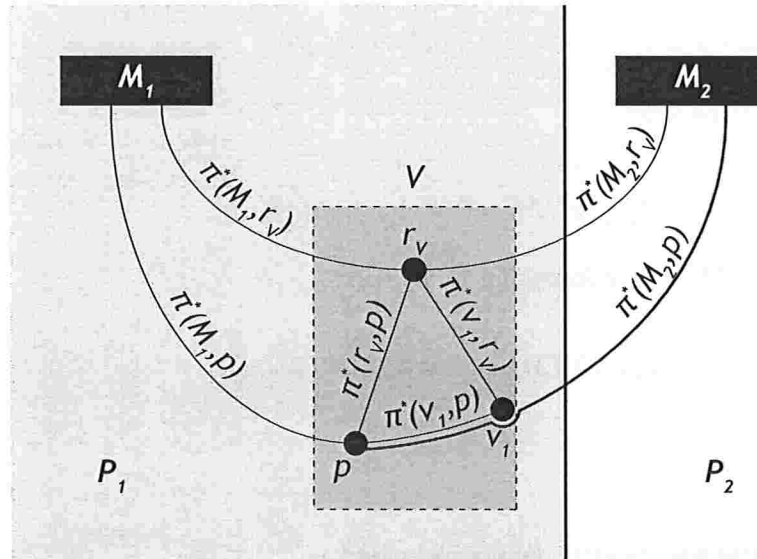
Desse fato, segue que:

$$f_{max}(\pi^*(M_i, V)) = f_{max}(\pi^*(M_i, r_V)) \quad (4.8)$$

De Eq. 4.8 e Eq. 4.7, temos que:

$$f_{max}(\pi^*(M_1, r_V)) < f_{max}(\pi^*(M_2, r_V)) \quad (4.9)$$

Seja v_1 o primeiro *pixel* de V em $\pi^*(M_2, p)$.



Pela escolha de v_1 :

$$\begin{aligned}
f_{max}(\pi^*(M_2, p)) &= f_{max}(\pi^*(M_2, v_1) \cdot \pi^*(v_1, p)) \\
&= \max\{f_{max}(\pi^*(M_2, v_1)), f_{max}(\pi^*(v_1, p))\} \\
&= \max\{f_{max}(\pi^*(M_2, v_1)), f_{max}(\pi^*(v_1, r_V)), f_{max}(\pi^*(r_V, p))\} \\
&= \max\{f_{max}(\pi^*(M_2, v_1)), f_{max}(\pi^*(v_1, r_V)), I(p)\} \\
&= \max\{f_{max}(\pi^*(M_2, r_V)), I(p)\}
\end{aligned} \tag{4.10}$$

Como $\pi^*(M_1, p)$ é ótimo, temos que:

$$\begin{aligned}
f_{max}(\pi^*(M_1, p)) &\leq f_{max}(\pi^*(M_1, r_V) \cdot \pi^*(r_V, p)) \\
&= \max\{f_{max}(\pi^*(M_1, r_V)), f_{max}(\pi^*(r_V, p))\} \\
&= \max\{f_{max}(\pi^*(M_1, r_V)), I(p)\}
\end{aligned} \tag{4.11}$$

Substituindo Eq. 4.9 e Eq. 4.10 em Eq. 4.11:

$$\begin{aligned}
f_{max}(\pi^*(M_1, p)) &\leq \max\{f_{max}(\pi^*(M_1, r_V)), I(p)\} \\
&\leq \max\{f_{max}(\pi^*(M_2, r_V)), I(p)\} \\
&= f_{max}(\pi^*(M_2, p))
\end{aligned} \tag{4.12}$$

Logo,

$$f_{max}(\pi^*(M_1, p)) \leq f_{max}(\pi^*(M_2, p)) \tag{4.13}$$

De Eq. 4.13 e Eq. 4.5, concluímos que

$$f_{max}(\pi^*(M_1, p)) = f_{max}(\pi^*(M_2, p)). \tag{4.14}$$

□

4.2 Marcadores para hierarquia

Dada uma partição P obtida a partir de um conjunto de marcadores M , independentemente da granularidade utilizada, desejamos construir uma AGM do RAG cuja hierarquia

correspondente inclua a partição P . Para tal, dois pontos devem ser levados em consideração.

Primeiro, as bacias primitivas são consideradas unidades atômicas na hierarquia, ou seja, nenhuma partição da hierarquia contém bordas além das bordas das bacias primitivas. Se o RAG for utilizado na abordagem de marcadores, então esta condição é automaticamente satisfeita. Porém, se a granularidade de *pixels* foi utilizada na abordagem de marcadores, então a partição P pode conter bordas que atravessam bacias primitivas. Neste caso a partição P não pode ser representada na hierarquia. Uma abordagem possível é associar um mesmo rótulo para todos os *pixels* dentro de cada bacia primitiva, por exemplo, o rótulo mais freqüente entre eles. A partição P' obtida dessa forma pode ser representada em um conjunto hierárquico de partições e, dentre todas as partições cujas bordas são subconjuntos das bordas das bacias primitivas, é a partição com menor número de *pixels* com rotulação diferente de P .

Além disso, também é necessário que cada conjunto de bacias primitivas com mesmo rótulo (que corresponde a uma região de interesse) seja um conjunto conexo considerando as arestas da árvore geradora do RAG a partir da qual a hierarquia é construída. Se tal condição não for satisfeita, a partição desejada não pode ser obtida na hierarquia. Para garantir que essa condição seja satisfeita, ao invés de considerar uma AGM arbitrária do RAG, derivamos uma árvore geradora (não necessariamente uma AGM) do RAG da seguinte maneira:

1. considere uma AGM para cada região de interesse (conjunto de bacias primitivas com mesmo rótulo), obtendo uma floresta geradora do RAG;
2. complete a floresta geradora, com arestas de peso mínimo entre as arestas do RAG ainda não utilizadas, de forma a obter uma árvore geradora do RAG.

Na construção da ALC, ao invés de utilizar todas as arestas da árvore geradora ordenadas pela ordem de seus pesos, as arestas da floresta geradora obtida no primeiro passo devem ser utilizadas antes das arestas adicionadas no segundo passo.

Interatividade no *watershed*

Baseado nas formas de se alternar entre as abordagens do *watershed* apresentadas no capítulo anterior, são propostas algumas possibilidades de interação para obter partições de imagens digitais. As possibilidades descritas foram implementadas em uma ferramenta interativa para segmentação (no Apêndice B encontra-se uma descrição dos pacotes que compõe o código do *SegmentIt* e das funcionalidades disponíveis através da sua interface gráfica).

Também será apresentada uma análise de como a abordagem de marcadores em conjunto com a abordagem hierárquica pode beneficiar os usuários, reduzindo o esforço de interação necessário ao se considerar uma abordagem integrada.

5.1 Possibilidades de interação

1. **Construção de hierarquias de partições** utilizando inundação uniforme, obtendo partições como as da Fig. 3.9, ou inundações síncronas, utilizando a profundidade, a área ou o volume como critérios de fusão de bacias de retenção, obtendo partições como as da Fig. 3.10.

Ao alternar entre os diferentes critérios de fusão de bacias de retenção, é possível reconstruir a ALC a partir da AGM já existente do RAG, bastando recalcular os pesos das arestas da AGM e construindo a nova ALC através do Algoritmo 4. Dessa forma, o usuário pode alternar rapidamente entre as diferentes hierarquias possíveis, podendo verificar qual critério é mais adequado para diferenciar suas regiões de interesse.

Para navegar na hierarquia de partições de uma imagem, o usuário pode utilizar um controle deslizante para selecionar um limiar para o peso das arestas a serem suprimidas. Além disso, pode utilizar operações locais na partição selecionada na

hierarquia, como ilustrado na Fig. 3.8, de forma a afetar somente as regiões desejadas. As operações ficam disponíveis em um menu de contexto, de forma que o usuário seleciona diretamente na imagem a região a ser alterada pela operação. As mesmas operações locais (fusão e re-segmentação de regiões) também são acessíveis utilizando o *mouse wheel*, que possibilita o acesso às operações com menor esforço de interação e uma navegação mais rápida na hierarquia de partições.

Pelo fato da utilização do limiar e das operações serem feitas diretamente na ALC, que estrutura as arestas da AGM do RAG pela ordem de seus pesos, a navegação pela hierarquia de partições é feita em tempo real para o usuário, uma vez que a ALC já tenha sido construída.

2. **Geração automática de marcadores para partições selecionadas na hierarquia**, como descrito na Seção 4.1. Após alternar da abordagem hierárquica para os marcadores, o usuário pode refinar a partição com precisão no nível dos *pixels*, como foi feito, por exemplo, para separar as células na Fig. 5.1(g).
3. **Edição de marcadores utilizando a ferramenta *pincel/borracha***. A função que rotula os marcadores é derivada da rotulação de componentes conexos, apresentada no Algoritmo 1. Dessa forma, o usuário não precisa mudar a cor do pincel para marcar cada região de interesse. Se for necessário que marcadores para diferentes regiões de interesse sejam adjacentes, o usuário pode utilizar o pincel com cores diferentes para distinguir tais marcadores. O pincel/borracha é disponível como uma ferramenta única, de forma que o usuário pode editar (desenha e apagar) marcadores utilizando os diferentes botões do *mouse*.
4. **Fusão de regiões adjacentes através de um *pincel de fusão***. Isto é feito de forma mais flexível que a operação local de fusão na hierarquia, já que pode fundir regiões utilizando arestas que não estão na AGM do RAG. Para tal, uma partição com as regiões selecionadas fundidas é derivada (associando o mesmo rótulo a elas) e esta partição é utilizada para calcular os marcadores ou para derivar uma hierarquia (dependendo da abordagem sendo utilizada).
Este pincel foi utilizado para obter a partição da Fig. 5.1(c), já que esta não pode ser localizada nas hierarquias originais (obtidas por inundações uniforme e síncronas).
5. **Uso da hierarquia restrita a uma região**. Isto permite que o usuário refine uma região, obtida através de marcadores do conjunto automático de marcadores gerados por um aplicação anterior do *watershed* hierárquico, ou marcadores criados/editados manualmente. Por exemplo, para obter as bacias primitivas da terra de uma imagem de satélite, o usuário pode primeiro desenhar marcadores para separar terra de água e, então, aplicar a hierarquia na região da terra e selecionar a partição mais fina,

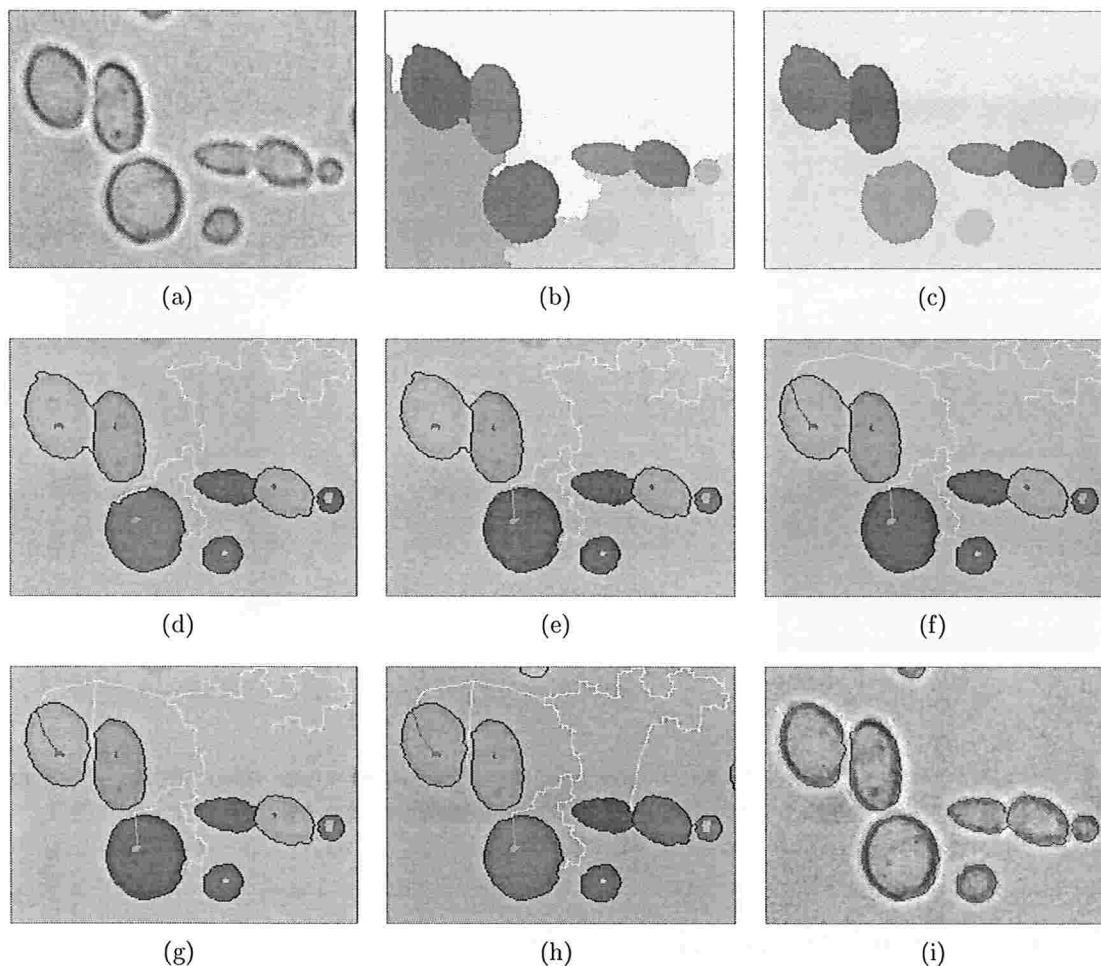


Figura 5.1: Segmentando através da hierarquia e de marcadores: (a) Imagem original. (b) Partição selecionada na hierarquia. (c) Partição após fusão de regiões. (d) Marcadores gerados automaticamente. (e) Marcadores editados para recuperar corretamente a maior célula na parte inferior da imagem. (f) Marcadores editados para segmentar corretamente a célula mais à esquerda na imagem, fundindo o *blob* na parte superior da célula ao fundo. (g) Marcadores editados para separar as duas células mais à esquerda na imagem. (h) Marcadores editados para separar um pequeno detalhe entre as células rotuladas em azul e mais dois marcadores criados para segmentar pedaços de células visíveis na borda da imagem. (i) Resultado final.

como na Fig. 5.2. Esta é uma maneira mais direta de se obter tal partição, ao invés de primeiro obter a partição resultante do *watershed* clássico para toda a imagem e fundir todas as bacias primitivas localizadas na água.

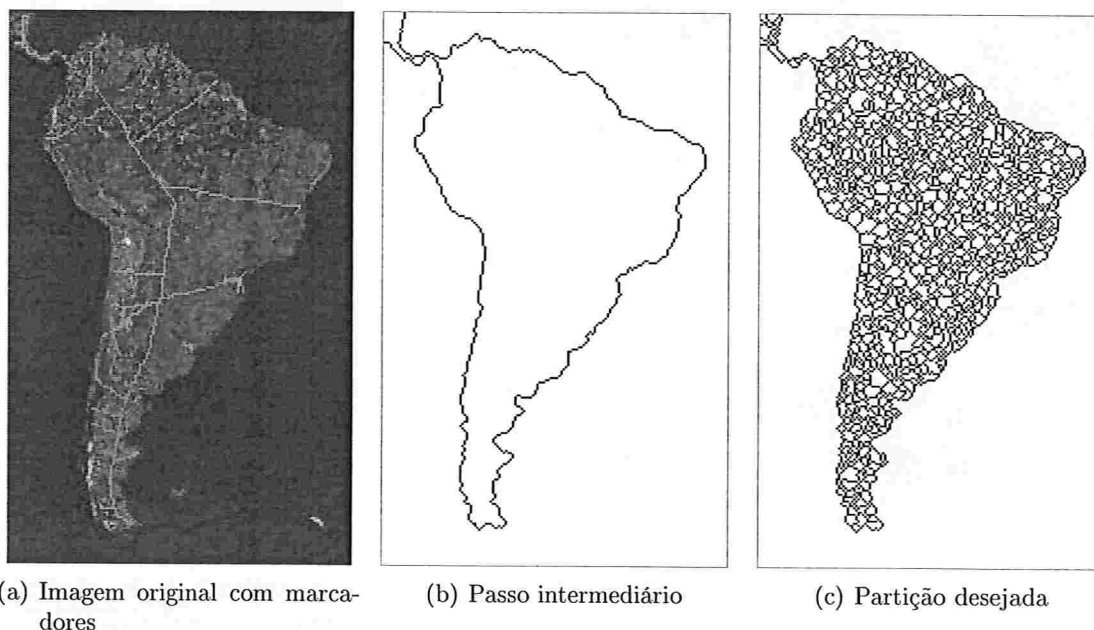


Figura 5.2: Hierarquia restrita a uma região: (a) Imagem original com marcadores. (b) Partição separando terra de água. (c) Bacias primitivas localizadas em terra.

5.2 Características adicionais

A ferramenta implementada também inclui características adicionais como:

- opção *Segmentar automaticamente*, que executa o algoritmo a cada edição de marcadores (pode ser desabilitada para melhorar a performance ao desenhar marcadores para segmentar imagens grandes, por exemplo);
- salvar a imagem de marcadores para refinamentos posteriores;
- controle sobre a imagem de saída (o usuário pode escolher que imagens serão sobrepostas: imagem original, imagem do gradiente, bacias de retenção, linhas de *watershed*);

- suporte para segmentação de imagens coloridas através do gradiente com pesos, como descrito em [10]. O usuário pode usar os pesos calculados automaticamente ou escolher os pesos para cada um dos gradientes considerados (o gradiente com pesos consiste em uma média ponderada de gradientes definidos em cada uma das bandas da imagem: matiz, saturação e brilho). A informação de cor pode ser um ponto chave na redução do esforço de interação necessário para obter uma partição desejada, como pode ser observado na Fig. 5.3.

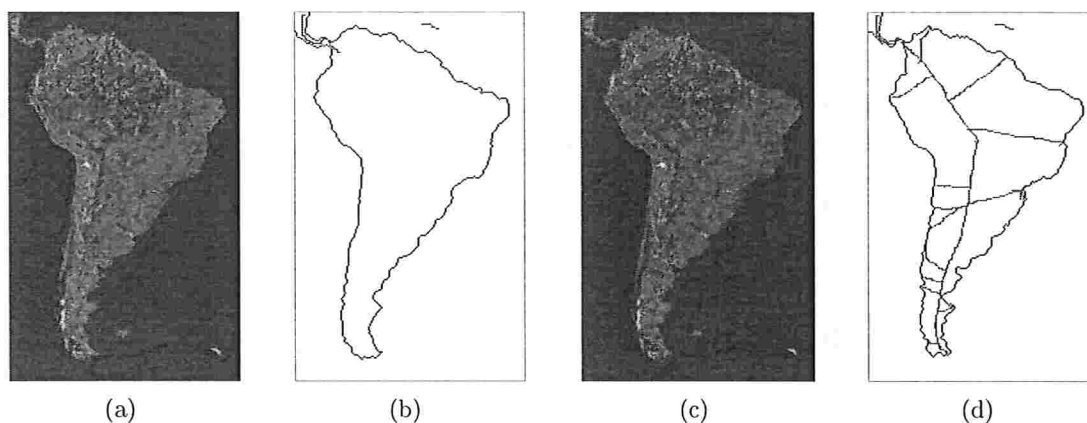


Figura 5.3: Informação de cor reduzindo o esforço de interação (marcadores em vermelho, linhas de *watershed* em preto): (a)-(b) Marcadores simples são suficiente para separar terra de água quando se utiliza o gradiente com pesos. (c)-(d) Marcadores complexos são necessários quando é utilizado o gradiente morfológico da imagem obtida sem a informação de cor.

5.3 Comparação das abordagens

Nessa seção destacaremos as vantagens e desvantagens na utilização de cada uma das abordagens.

5.3.1 *Watershed* a partir de marcadores

Pontos fortes:

- É fácil segmentar uma região específica da imagem (basta fazer marcadores internos e externos para tal região, estendendo tais marcadores até obter a partição desejada);

- funciona nas duas granularidades (trabalhando na granularidade de bacias primitivas há uma maior eficiência na execução, devido à redução no número de vértices do grafo sendo utilizado; trabalhando na granularidade dos *pixels* é possível controlar com precisão os detalhes de cada região de interesse da partição);
- permite “forçar” bordas fracas ou até mesmo inexistentes na imagem (se houver algum efeito – forte iluminação, por exemplo – que dificulte ou impossibilite a visualização de alguma borda de interesse na imagem, basta criar marcadores nos dois lados da borda desejada para que essa seja recuperada ou imposta pelo algoritmo do *watershed*).

Pontos fracos:

- um marcador deve ser feito para cada região de interesse (o que exige um considerável esforço de interação caso haja muitas regiões na partição desejada);
- maior esforço de interação para criar marcadores para regiões de estruturas mais complexas.

5.3.2 *Watershed* hierárquico

Pontos fortes:

- Fácil segmentar diversos objetos de acordo com suas métricas, como contraste e/ou tamanho;
- definida somente no nível de precisão das bacias primitivas (o que implica em eficiência).

Pontos fracos:

- definida somente no nível de precisão das bacias primitivas (implicando em limitações às partições que podem ser encontradas na hierarquia, uma vez que as bordas das partições sempre serão um subconjunto das bordas das bacias primitivas);
- as partições são limitadas devido à estrutura da ALC (para fundir duas regiões, é necessário selecionar o menor ancestral comum dos vértices correspondentes, o que pode fundir outras regiões, conforme ilustrado na Fig. 5.4);
- pode ser difícil segmentar uma região específica (não é possível especificar diretamente as regiões de interesse: o usuário deve ir navegando na hierarquia – não é possível escolher onde será feita a divisão de uma região – até que apareçam todas as bordas desejadas, e nesse processo podem aparecer outras bordas indesejadas).

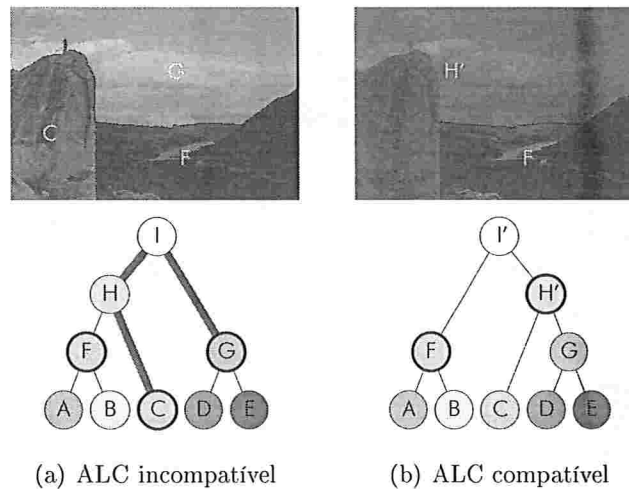


Figura 5.4: Limitação de partições devido à estrutura da ALC: (a) Se desejamos fundir as regiões C e G da Fig. 3.8(c), o menor ancestral comum delas (que corresponde à raiz da hierarquia, caminhos destacados em vermelho) deve ser selecionado, o que implica que a região F também deve ser fundida com C e G. (b) ALC modificada, como descrito no item 4 da Seção 5.1, para ser possível localizar a partição desejada na hierarquia.

5.3.3 Abordagem integrada

Analisando as vantagens e desvantagens de cada uma das abordagens consideradas, vemos que a maioria das limitações de cada abordagem corresponde a pontos fortes da outra. Logo, ao se utilizar as duas abordagens concomitantemente há um ganho de versatilidade para resolver problemas de segmentação.

No diagrama da Fig. 5.5 são relacionadas as transições possíveis entre as abordagens interativas e representações de partições.

Neste contexto, o principal ponto a ser considerado é a escolha da granularidade: na abordagem de marcadores podemos utilizar o RAG como entrada para o algoritmo do *watershed* (há limitação no conjunto de marcadores que pode ser utilizado, mas numa abordagem interativa essa limitação pode ser contornada como descrito na Seção 3.2.2, representada pela aresta azul na Fig. 5.5), ou o grafo dos *pixels* G_I , quando é necessária maior precisão na partição desejada; na abordagem hierárquica, somente o nível de precisão do RAG é considerado. Logo, se o nível de precisão dos *pixels* é utilizado, ao alternar para a abordagem hierárquica, primeiro devemos converter a partição de G_I para uma partição do RAG, através da unificação dos rótulos dos *pixels* internos a cada bacia primitiva (como descrito na Seção 4.2), o que pode implicar em perda de informação (representada por uma aresta vermelha na Fig. 5.5).

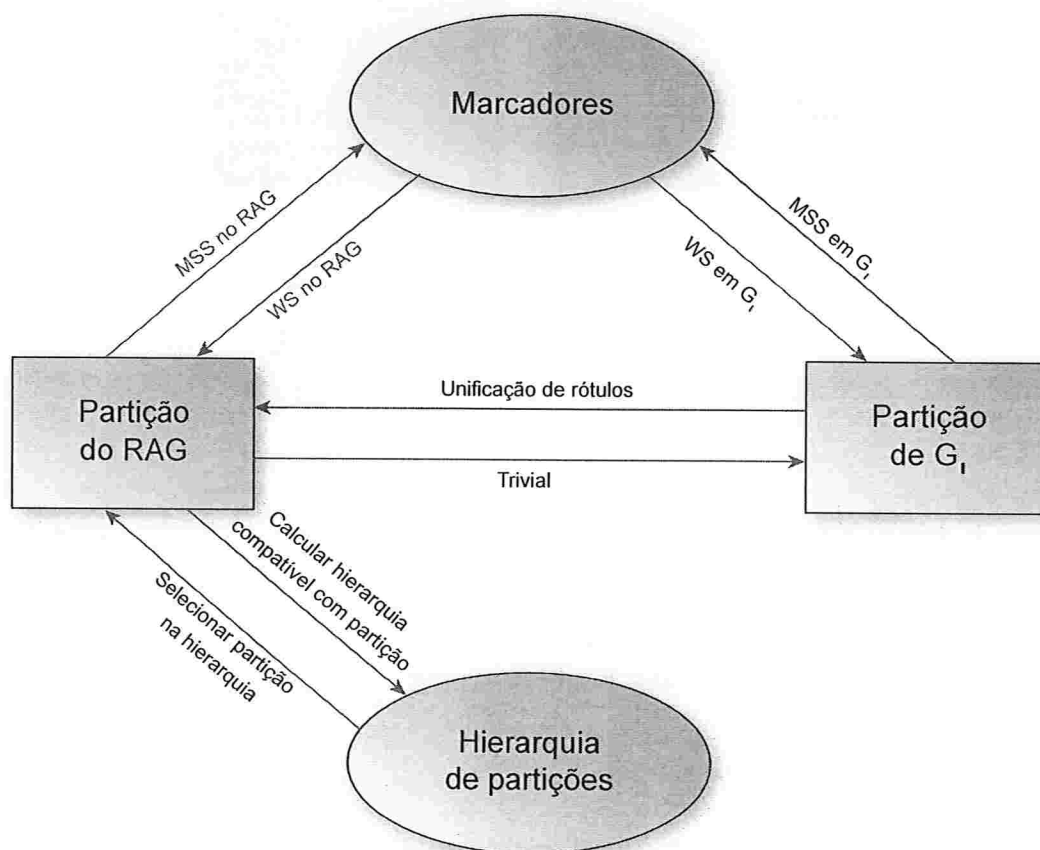


Figura 5.5: Diagrama da abordagem integrada: arestas em verde indicam transições que são sempre possíveis. Arestas em azul indicam transições possíveis com limitação. Arestas em vermelho indicam transições com perda de informação ou que dificultam interatividade.

Na transição para a abordagem de marcadores, o cômputo do MSS pode ser feito tanto sobre G_I quanto sobre o RAG: calculando o MSS em G_I , os marcadores tendem a se localizar nas bordas das regiões de interesse, o que os torna inadequados para edição (representada por uma aresta vermelha na Fig. 5.5); já se o RAG é considerado ao calcular o MSS, os marcadores obtidos tendem a ser mais compactos e localizados mais ao centro de cada região de interesse. Por esses motivos, os marcadores calculados no RAG são mais adequados para posteriores edições manuais pelos usuários. Assim, escolhemos computar o MSS no RAG, independentemente do nível de precisão do grafo sendo utilizado na abordagem de marcadores. Com essa escolha, a partição obtida ao se alternar da abordagem hierárquica para a abordagem de marcadores pode diferir ligeiramente da partição anterior

a tal transição, como mostrado na Seção 4.1.

Como refinamentos feitos por marcadores no grafo dos *pixels* da imagem podem ser desfeitos ao alternar para a abordagem hierárquica ou para o nível de precisão das bacias primitivas, o processo de segmentação que integra as duas abordagens do *watershed* deve seguir uma abordagem *coarse to fine*: inicialmente o usuário pode utilizar as duas abordagens para obter uma partição mais próxima possível da partição desejada, utilizando somente o RAG (que, por conter menos vértices que G_I , implica na redução do tempo de execução do *IFT-WS*); então, utilizando somente os marcadores (i.e., editando os marcadores), o usuário pode refinar a partição com precisão no nível dos *pixels*. Desta forma são feitas somente transições sem perda de informação (sem passar pelas arestas vermelhas da Fig. 5.5).

Através dos exemplos a seguir, mostraremos como a utilização da abordagem integrada proposta, considerando as possibilidades de interação descritas anteriormente e implementadas no *SegmentIt*, pode facilitar a obtenção rápida de partições de imagens digitais.

Exemplo 1:

Para separar as teclas da calculadora da Fig. 5.6(a), começamos pela abordagem hierárquica e, utilizando somente o limiar e operações locais de divisão de regiões, obtemos a partição da Fig. 5.6(b). Utilizando o pincel de fusão arbitrária, podemos fundir as regiões que foram separadas devido à estrutura da ALC, obtendo a partição da Fig. 5.6(c).

Para corrigir o pedaço do fundo que ficou conectado à tecla de igual, podemos alternar para a abordagem de marcadores no nível de precisão dos *pixels*, Fig. 5.6(d) (detalhe na Fig. 5.6(e)), e estender o marcador do fundo, obtendo a partição da Fig. 5.6(f) (detalhe na Fig. 5.6(g)). Esta é uma forma mais direta de fazer tal correção do que continuar na abordagem hierárquica, dividindo a região que contém a tecla de igual até que a borda desejada apareça (Fig. 5.6(h)), para então fundir o pedaço desejado com o fundo e unir os pedaços nos quais a tecla foi dividida.

Exemplo 2:

Para separar os elementos da imagem do pássaro apresentada na parte inicial desta dissertação (Fig. 2.4(a)), podemos utilizar a abordagem hierárquica e, através de operações locais de divisão de regiões, obter a partição da Fig. 5.7(a). Utilizando o pincel de fusão arbitrária, podemos fundir os três pedaços do galho, obtendo a partição da Fig. 5.7(b).

Para completar o pedaço do galho que falta, uma possibilidade seria continuar na abordagem hierárquica e ir dividindo as regiões até que as bordas desejadas apareçam, o que resultaria na partição da Fig. 5.7(c)). Isto implicaria na necessidade de utilizar o pincel de fusão arbitrária para fundir todas as regiões criadas e não desejadas. Porém, podemos obter o mesmo resultado, de forma mais rápida, se alternarmos para a abordagem

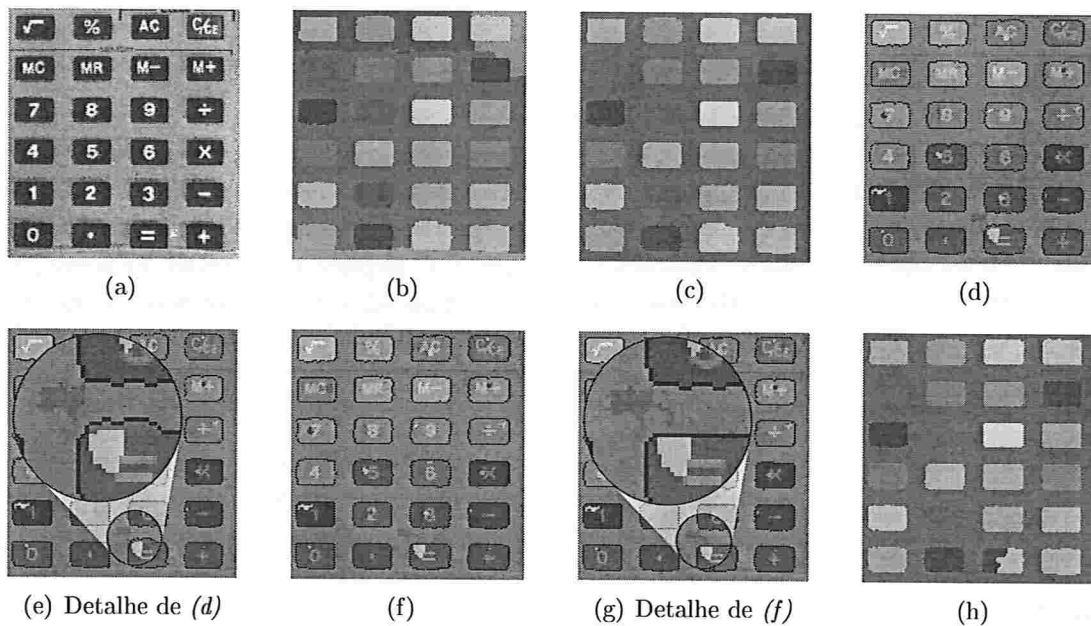


Figura 5.6: Segmentando as teclas de uma calculadora: (a) Imagem original. (b) Partição obtida através de seleção de limiar e operações locais de divisão de regiões na abordagem hierárquica. (c) Partição obtida usando o pincel de fusão, fundindo regiões do fundo que foram separadas devido à estrutura da ALC. (d) Marcadores gerados automaticamente. (e) Detalhe da tecla que não foi segmentada como desejado. (f) Marcador estendido de forma a segmentar corretamente a tecla de igual. (g) Detalhe da partição e marcador após correção. (h) Partição obtida somente através de operações locais de divisão de regiões na partição de (c), a partir da qual podemos fundir regiões de forma a obter o mesmo resultado de (f).

de marcadores (Fig. 5.7(d)) e estendermos o marcador gerado para o galho, conforme a Fig. 5.7(e).

Para segmentar as pequenas regiões do fundo visíveis no meio do galho, uma forma interessante seria utilizar a abordagem hierárquica com o critério de profundidade. Para tal, basta selecionar a ferramenta de hierarquia, clicar na região do galho (as operações hierárquicas estarão restritas a essa região), selecionar o critério desejado e então utilizar o limiar da hierarquia para segmentar as regiões que têm maior contraste com suas regiões adjacentes (Fig. 5.7(f)). Dessa forma, além de segmentar as regiões desejadas, também segmentamos pequenos pedaços na parte inferior da imagem, que podem ser facilmente fundidos de volta ao galho com o pincel de fusão arbitrária, obtendo então a partição da Fig. 5.7(g) (estão exibidas todas as regiões da partição, não somente às quais as operações hierárquicas estão restritas).

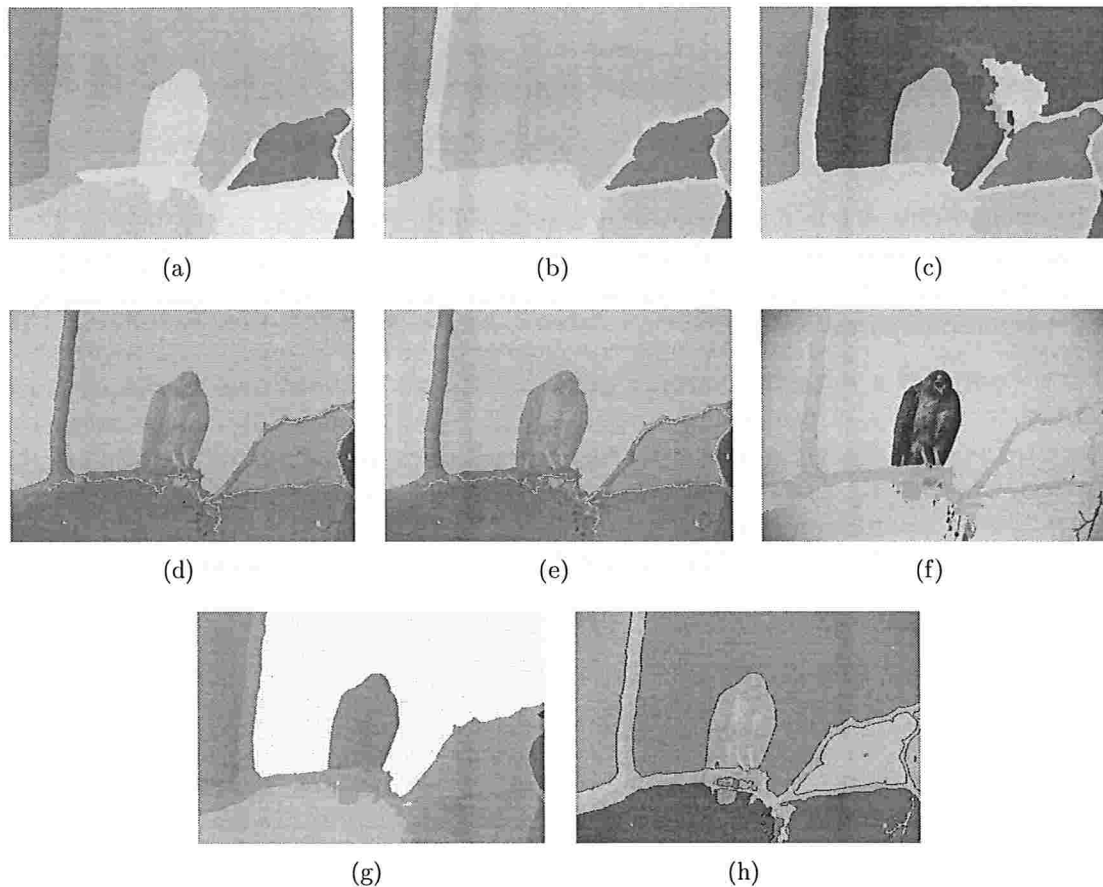


Figura 5.7: Segmentando a imagem da Fig. 2.4(a): (a) Partição obtida somente através de operações locais de divisão de regiões. (b) Pedacos do galho unificados na mesma região através do uso do pincel de fusão. (c) Muitas regiões são separadas para se obter o pedaço do galho que falta em (b) se utilizarmos somente a abordagem hierárquica. (d) Marcadores obtidos automaticamente a partir de (b). (e) Completando o pedaço que faltava do galho com a simples extensão de um único marcador. (f) Pedacos do fundo que ficaram na mesma região que o galho são separados facilmente na abordagem hierárquica utilizando o critério de profundidade (contraste). (g) Pedacos na parte inferior do galho que foram separados no passo anterior são unidos com o pincel de fusão. (h) Partição final, obtida através do refinamento de marcadores no nível de precisão do *pixels*.

Alguns detalhes são difíceis de se obter utilizando a hierarquia, como por exemplo segmentar o pedaço fino do galho no canto inferior direito da imagem e o pedaço da cauda do pássaro que está atrás do galho mas tem pouco contraste com este. Para refinar a partição de forma a incluir tais detalhes, a forma mais fácil é alternar para a abordagem de marcadores e para o nível de precisão dos *pixels*, para então refinar os marcadores manualmente, obtendo a partição desejada, apresentada na Fig. 5.7(h).

Exemplo 3:

A principal dificuldade para segmentar a imagem da Fig. 5.8(a) são os efeitos de iluminação, que não permite a visualização de algumas bordas. De fato, na Fig. 5.8(b) são exibidas bacias primitivas que deveriam ser divididas para que todas as bordas desejadas sejam possíveis na partição.

Para segmentar a imagem, podemos primeiro utilizar a abordagem hierárquica para selecionar uma partição próxima à desejada (Fig. 5.8(c)) com muito pouco esforço de interação. Depois, alternando para a abordagem de marcadores e para o nível de precisão dos *pixels*, Fig. 5.8(d), é possível editar os marcadores criados automaticamente para se obter as bordas não aparentes na imagem original e refinar detalhes em outras partes da imagem, conforme a Fig. 5.8(e). Na Fig. 5.8(f) é exibida a partição obtida.

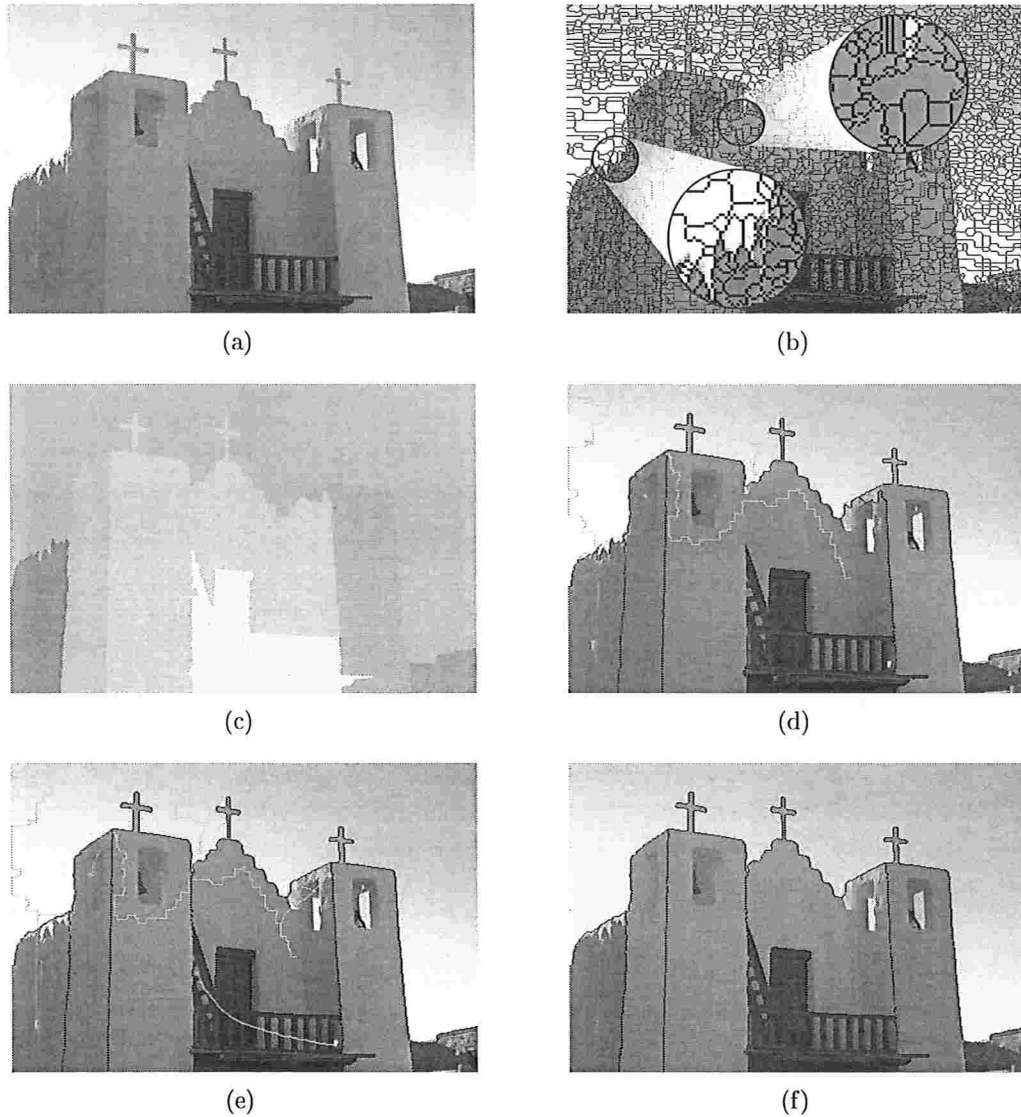


Figura 5.8: Segmentando imagem com bordas não aparentes devido a efeitos de iluminação: (a) Imagem original. (b) Detalhes de bacias primitivas que desejamos subdividir para obter as bordas das regiões de interesse. (c) Partição obtida rapidamente na abordagem hierárquica. (d) Marcadores gerados automaticamente. (e) Marcadores editados no nível de precisão dos *pixels* de forma a obter as bordas desejadas. (f) Resultado final.

Neste trabalho, apresentamos uma abordagem que permite a utilização alternada do *watershed* a partir de marcadores com o *watershed* hierárquico. Dessa forma, o usuário pode usufruir das vantagens de cada paradigma, obtendo os resultados desejados com menor esforço de interação do que quando se utiliza apenas um deles isoladamente.

Podem existir diferenças entre as partições ao se alternar entre os paradigmas considerados, devido à diferença de granularidade que é utilizada em cada um (mais especificamente, há diferenças se a granularidade dos *pixels* é utilizada no paradigma dos marcadores). Porém, tal diferença é bem caracterizada e em geral desprezível ao se considerar os benefícios obtidos com a abordagem proposta.

Com a possibilidade de se alternar entre os paradigmas, é possível desenvolver métodos interativos versáteis para segmentação baseada na transformação *watershed*. Diversas possibilidades de interação conhecidas, bem como as novas propostas, foram implementadas numa ferramenta de segmentação interativa, distribuída como um projeto de software livre.

6.1 Contribuições do trabalho

- Integração do *watershed* a partir de marcadores com o *watershed* hierárquico;
- implementação de uma ferramenta para segmentação integrando diversas possibilidades de interação descritas na literatura da área;
- novo resultado a respeito das diferentes granularidades (*pixels* e bacias primitivas) no problema do MSS;
- proposta do uso de hierarquias limitadas a uma região, facilitando a segmentação de diferentes regiões de uma mesma imagem que têm diferentes características;

- criação da operação de fusão arbitrária, que possibilita superar a limitação das partições que podem ser localizadas na hierarquia devido à estrutura da ALC.

6.2 Trabalhos futuros

- Estudos de casos para avaliar a redução do esforço de interação devido à abordagem integrada e melhorar a usabilidade da ferramenta;
- prover meios de criar marcadores nos dois lados de bordas de interesse (para suavizar bordas ou facilitar a segmentação de imagens com texturas, por exemplo);
- estender o suporte à segmentação de imagens coloridas (atualmente limitado);
- considerando detalhes de implementação do algoritmo *IFT-WS*, investigar a possibilidade de reduzir o MSS calculado no nível de precisão dos *pixels* (como marcadores calculados no nível das bacias primitivas são mais simples e recuperam praticamente a mesma partição, espera-se que marcadores mais simples também sejam possíveis de ser calculados automaticamente no nível dos *pixels*);
- investigar como construir hierarquias com maior representatividade do conteúdo das imagens, por exemplo, aproveitando a interação do usuário ao corrigir a estrutura da ALC (com o pincel de fusão arbitrária) como entrada para técnicas de aprendizado de máquina. Pretende-se, dessa forma, reduzir a lacuna semântica criada ao utilizar somente informações de baixo nível (nível de cinza dos *pixels*) para a construção da hierarquia.



Publicações

A seguir listamos as publicações produzidas no contexto deste trabalho:

- **Watershed segmentation: Switching back and forth between markers and hierarchies** [14] - resumo expandido publicado nos anais do *8th International Symposium on Mathematical Morphology - ISMM 2007* e apresentado na sessão de pôsteres do referido simpósio;
- **Watershed Transform based Interactive Image Segmentation Tool** [15] - resumo expandido publicado nos anais do *XXIst Brazilian Symposium on Computer Graphics and Image Processing - SIBGRAPI 2008* e apresentado na sessão de pôsteres do referido simpósio;
- **Interactive image segmentation with integrated use of the markers and the hierarchical watershed approaches** [16] - artigo completo publicado nos anais da *International Conference on Computer Vision Theory and Applications - VISAPP 2009*.



Neste apêndice são descritos os pacotes da ferramenta implementada tendo como base as funcionalidades descritas nos capítulos anteriores, apresentando alguns detalhes de implementação relevantes. Também são apresentadas as telas da interface gráfica do *SegmentIt*, listando todas as funcionalidades disponíveis aos usuários.

B.1 Descrição dos pacotes

Observações:

- Para simplificar os diagramas UML¹ apresentados, os construtores foram excluídos e são exibidos somente os atributos e métodos públicos;
- não pertencem aos pacotes do *SegmentIt* as classes/interfaces do Java (*java.** e *javax.**), e destas estão exibidos somente os métodos sobrescritos/implementados nas classes filhas;
- no pacote *br.usp.ime.klava.segmentit.gui* não estão exibidas as interfaces do Java para tratamento de eventos (do pacote *java.awt.event*: *ActionListener*, *MouseListener*, *MouseMotionListener*, *MouseWheelListener*, *WindowListener*; do pacote *javax.swing.event*: *ChangeListener*, *InternalFrameListener* e *MenuListener*);
- na Fig. B.1 são apresentadas as notações utilizadas nos diagramas UML;
- a documentação detalhada da API² desenvolvida encontra-se disponível em <http://watershed.sourceforge.net/javadoc/>;

¹UML: do inglês *unified modeling language*

²API: do inglês *application programming interface*

- a descrição feita neste capítulo corresponde à versão 1.0.2 do *SegmentIt*.

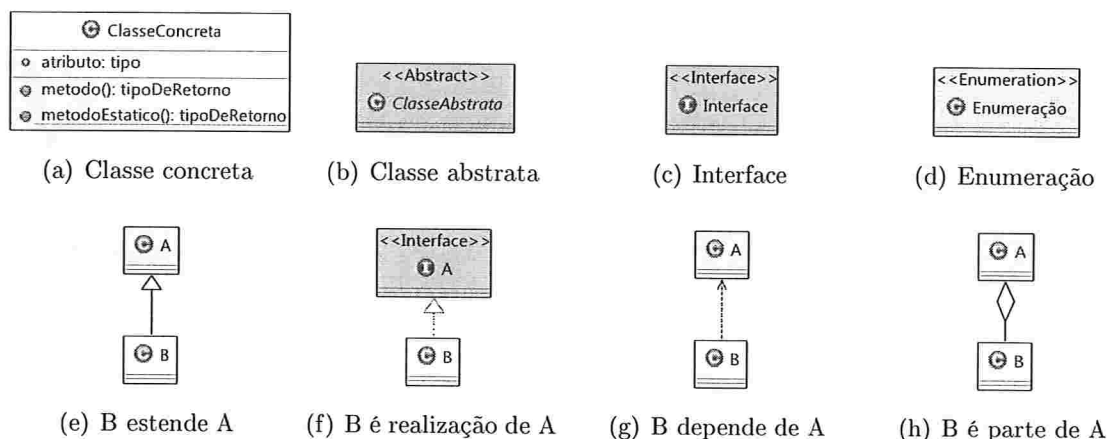


Figura B.1: Notações utilizadas nos diagramas UML.

B.1.1 `br.usp.ime.klava.segmentit.exceptions`

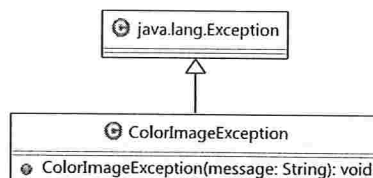


Figura B.2: Diagrama de classes do pacote `br.usp.ime.klava.segmentit.exceptions`.

Neste pacote encontram-se as exceções do sistema.

Atualmente existe apenas uma exceção específica do sistema – `ColorImageException` – lançada pelos métodos que recebem uma imagem como argumento e necessitam que tal imagem seja uma imagem em níveis de cinza.

Ao tratar esta exceção, é possível gerar uma versão em níveis de cinza da imagem de entrada (através das classes de filtros apropriadas, por exemplo `br.usp.ime.klava.segmentit.filters.WeightedGradient`, filtro padrão para imagens coloridas no sistema).

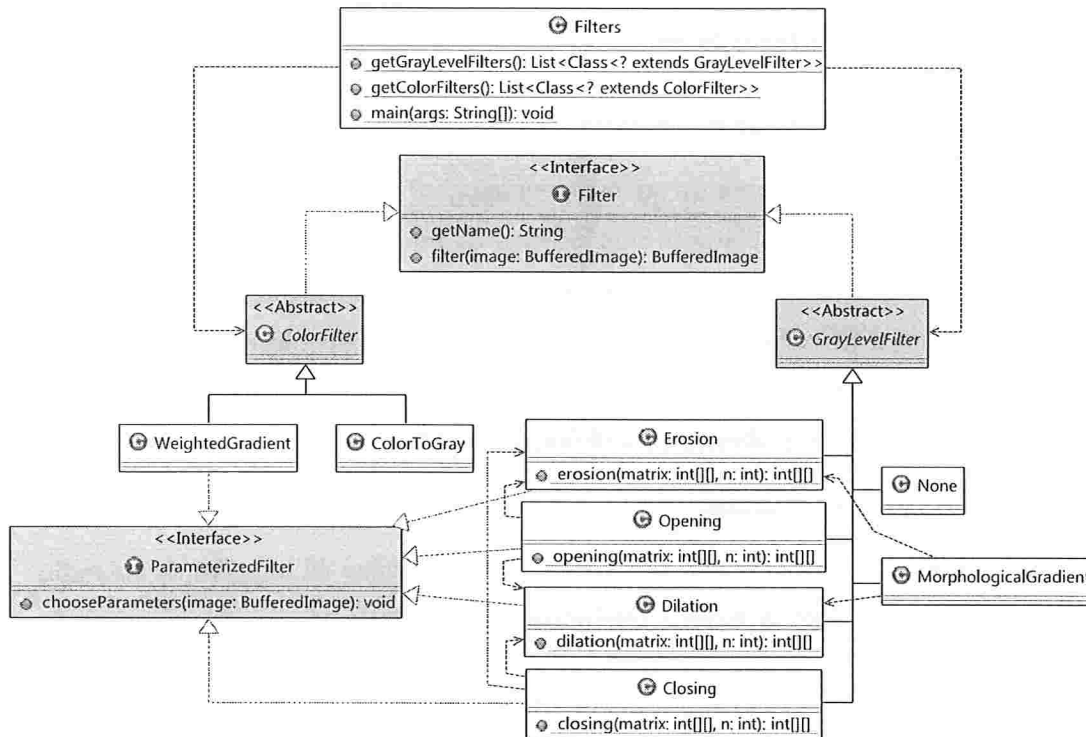


Figura B.3: Diagrama de classes do pacote *br.usp.ime.klava.segmentit.filters*.

B.1.2 *br.usp.ime.klava.segmentit.filters*

Neste pacote encontram-se os filtros utilizados para gerar as imagens de entrada para os algoritmos da transformação *watershed*.

As classes de filtros devem estender as classes abstratas de acordo com o tipo de imagem que recebem como entrada: *ColorFilter* para imagens coloridas e *GrayLevelFilter* para imagens em níveis de cinza. Filtros que possuem parâmetros devem implementar a interface *ParameterizedFilter* (elementos estruturantes para filtros morfológicos são especificados através do método *select* da classe *br.usp.ime.klava.segmentit.util.StructuringElement*).

A classe *Filters* é responsável por listar os filtros disponíveis no sistema, analisando o conteúdo do pacote e devolvendo os filtros de acordo com o tipo de entrada suportado. Novas classes de filtros devem ser criadas neste pacote, bastando implementar uma das classes bases, *ColorFilter* ou *GrayLevelFilter*, e opcionalmente implementando a interface *ParameterizedFilter*.

Atualmente os filtros disponíveis são:

- Filtros para imagens em níveis de cinza
 - *None*, devolve a própria imagem de entrada;
 - *Dilation*, devolve a dilatação morfológica da imagem de entrada;
 - *Erosion*, devolve a erosão morfológica da imagem de entrada;
 - *MorphologicalGradient*, devolve o gradiente morfológico da imagem de entrada;
 - *Closing*, devolve o fechamento morfológico da imagem de entrada;
 - *Opening*, devolve a abertura morfológica da imagem de entrada.
- Filtros para imagens coloridas
 - *ColorToGray*, devolve uma versão em níveis de cinza da imagem de entrada;
 - *WeightedGradient*, devolve o gradiente com pesos [10] da imagem de entrada.

Em futuras versões da ferramenta, o pacote de filtros deverá ser expandido para possibilitar a composição de filtros e melhorar o suporte a imagens coloridas.

B.1.3 `br.usp.ime.klava.segmentit.formats`

Nesse pacote encontram-se as classes responsáveis pelo tratamento dos formatos suportados para leitura e escrita de arquivos de imagens, utilizados por classes da camada de interface gráfica (pacotes *br.usp.ime.klava.segmentit.gui* e *br.usp.ime.klava.segmentit.gui.actions*).

B.1.4 `br.usp.ime.klava.segmentit.gui`

Neste pacote encontram-se as classes da interface gráfica da ferramenta.

A classe *SegmentationContainer* é a classe principal da GUI³ da ferramenta, através da qual é possível fazer a segmentação de diversas imagens ao mesmo tempo, cada uma em um *SegmentationDocument*. Um *SegmentationDocument* é composto por camadas, cujas visibilidades podem ser controladas de forma a facilitar a criação de marcadores e visualização dos resultados sobrepostos à imagem original ou filtrada. A classe *SegmentViaHierarchy* é responsável por disponibilizar as funcionalidades de segmentação baseadas no *watershed* hierárquico.

³GUI: do inglês *graphical user interface*

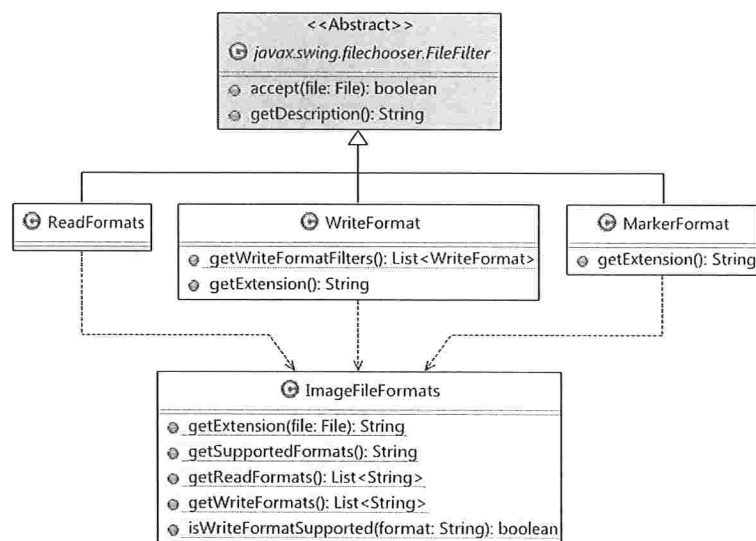


Figura B.4: Diagrama de classes do pacote *br.usp.ime.klava.segmentit.formats*.

A classe *ZoomableImageIcon* estende a classe para exibição de imagens do Java – *javax.swing.ImageIcon* – provendo métodos para controlar a opacidade e o nível de ampliação/redução para visualização de imagens.

A classe *MarkersEditor* possibilita a criação, edição e exclusão de marcadores. Caso a abordagem de marcadores esteja sendo utilizada no nível de precisão das bacias primitivas, é possível utilizar métodos – *erasePreviousMarkers* e *eraseNonUniqueLabeledBasins* – para alterar o conjunto de marcadores de forma que a rotulação das bacias primitivas seja única (se existir mais de um marcador em uma bacia primitiva, esta pode receber o rótulo de qualquer um desses; se este for o caso, o rótulo do último marcador editado é selecionado, e a intersecção de outros marcadores com a bacia em questão é apagada, de forma que a imagem de marcadores seja compatível com tal rotulação).

A classe *StatusBar* possibilita a exibição de ajuda sobre a utilização da ferramenta – *Tool* – atualmente selecionada na barra de ferramentas.

A classe *FileChooser* provê janelas que utilizam os formatos suportados (gerenciados pelas classes do pacote *br.usp.ime.klava.segmentit.formats*) em operações de leitura e escrita de arquivos de imagens.

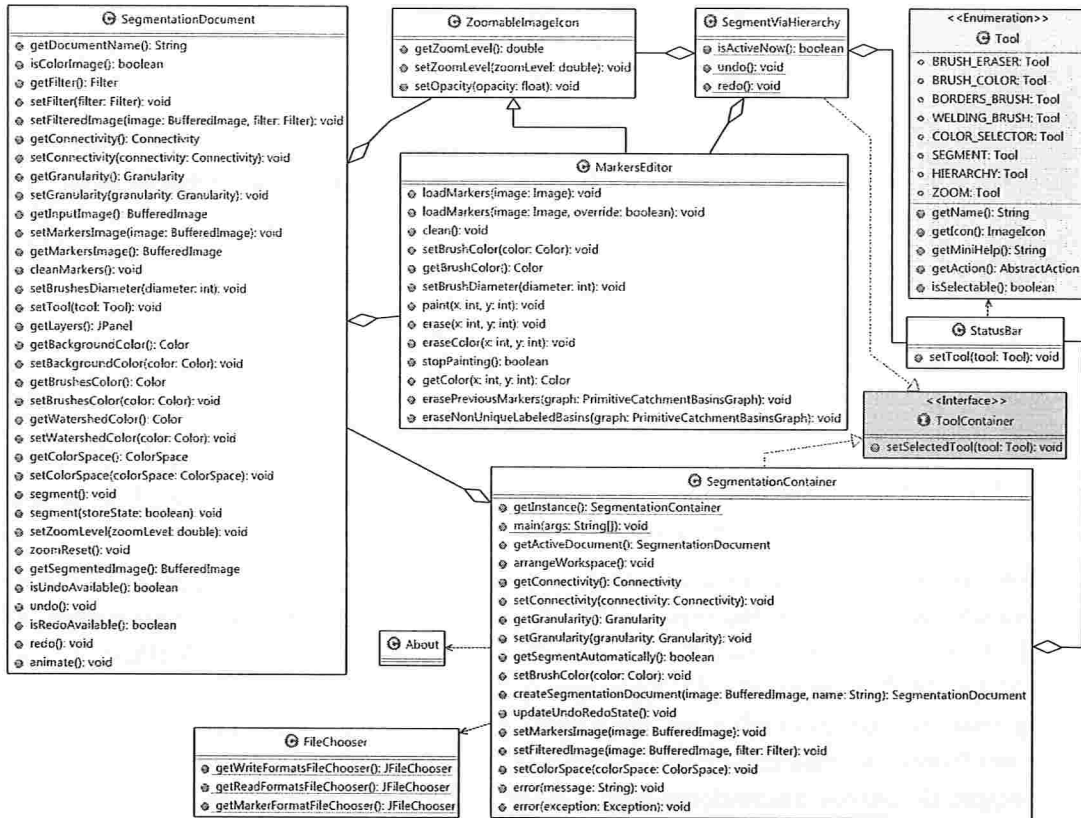


Figura B.5: Diagrama de classes do pacote *br.usp.ime.klava.segmentit.gui*.

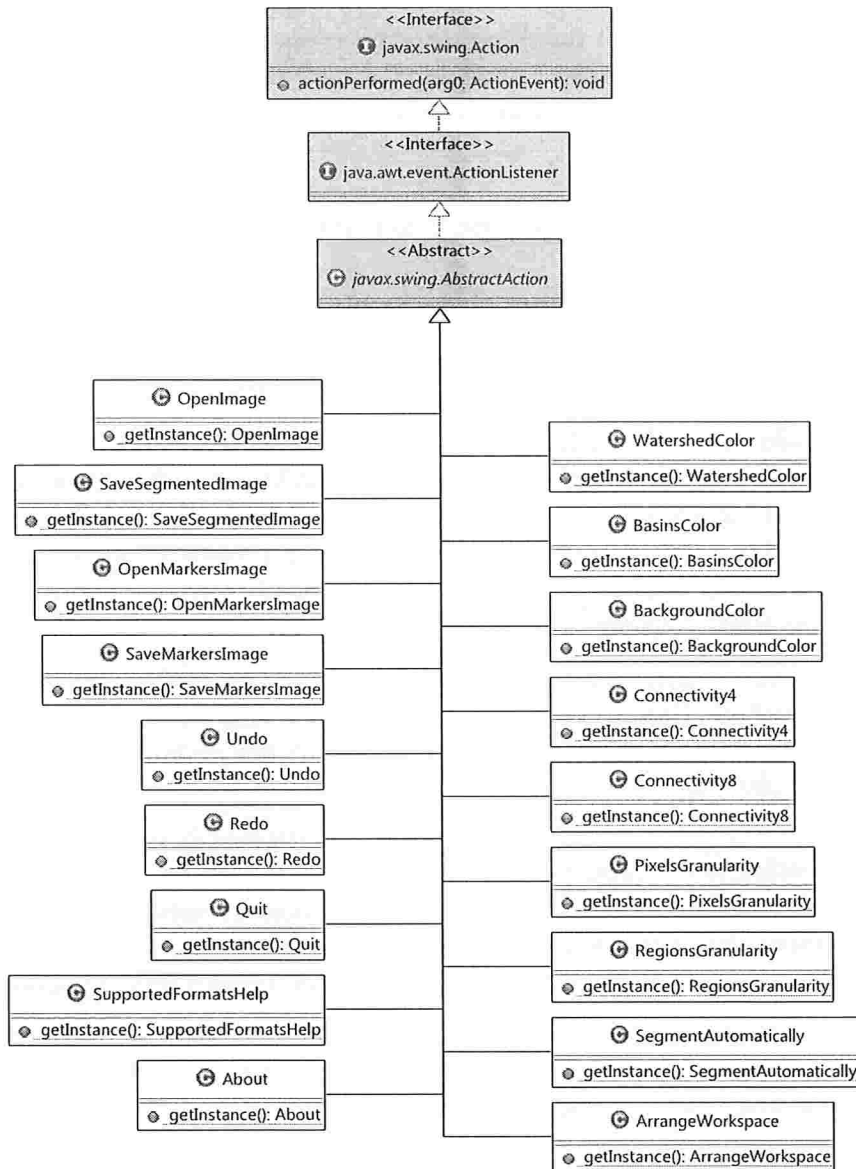


Figura B.6: Diagrama de classes do pacote *br.usp.ime.klava.segmentit.gui.actions*.

B.1.5 br.usp.ime.klava.segmentit.gui.actions

Neste pacote encontram-se ações vinculadas a opções de menu das classes *SegmentationContainer* e *SegmentViaHierarchy*. Todas as ações implementam o padrão de projeto *Singleton* para que seja possível habilitar/desabilitar suas execuções em qualquer parte do código. Uma vez que uma ação é vinculada a uma opção de menu, este fica habilitado/desabilitado de acordo com o estado da ação.

B.1.6 br.usp.ime.klava.segmentit.structures

Neste pacote encontram-se as estruturas de dados utilizadas no *SegmentIt*.

A classe *Pixel* é a unidade básica de informação, tendo atributos e métodos para manter o seu custo e predecessor em execuções do algoritmo IFT-WS.

De *CatchmentBasin* são derivadas duas classes: *CriticalLake*, que representa um nó interno da estrutura hierárquica, ligando duas *CatchmentBasins*; e *PrimitiveCatchmentBasin*, unidade atômica quando operando nesse nível de precisão na abordagem de marcadores, além de ser o tipo das folhas da estrutura hierárquica. Em cada *PrimitiveCatchmentBasin* é mantido um histograma dos níveis de cinza dos *pixels* contidos nesta, permitindo o cálculo de métricas para construção de hierarquias (*BasinsHierarchy*) de forma rápida.

Na classe que representa grafos de bacias primitivas (*PrimitiveCatchmentBasinsGraph*) são utilizadas estruturas *UnionFind* (com heurística de compressão de caminho e união por *rank*) para calcular árvores geradoras mínimas ou árvores geradoras com os requisitos apresentados na Seção 4.2.

A classe *RegionOfInterest* representa uma região de interesse de uma partição, e contém um método – *paintMarker* – para calcular um marcador que passe por todas as NRRRs de tal região. Este método calcula caminhos entre os nós de cada NRRR que são mais distantes das bordas. Em futuras versões, este método pode ser modificado para tentar minimizar o tamanho dos marcadores gerados ao invés de como é feito atualmente.

B.1.7 br.usp.ime.klava.segmentit.util

Neste pacote encontram-se classes auxiliares utilizadas nos demais pacotes.

A classe *ImageUtils* provê métodos, por exemplo, para gerar a versão de uma imagem em outra escala e controlar o brilho de imagens em níveis de cinza.

A classe *MatrixImageUtils* provê métodos para conversão entre imagens e matrizes: criação de matrizes de *Pixels* a partir de imagens, criação de matrizes com níveis de cinza

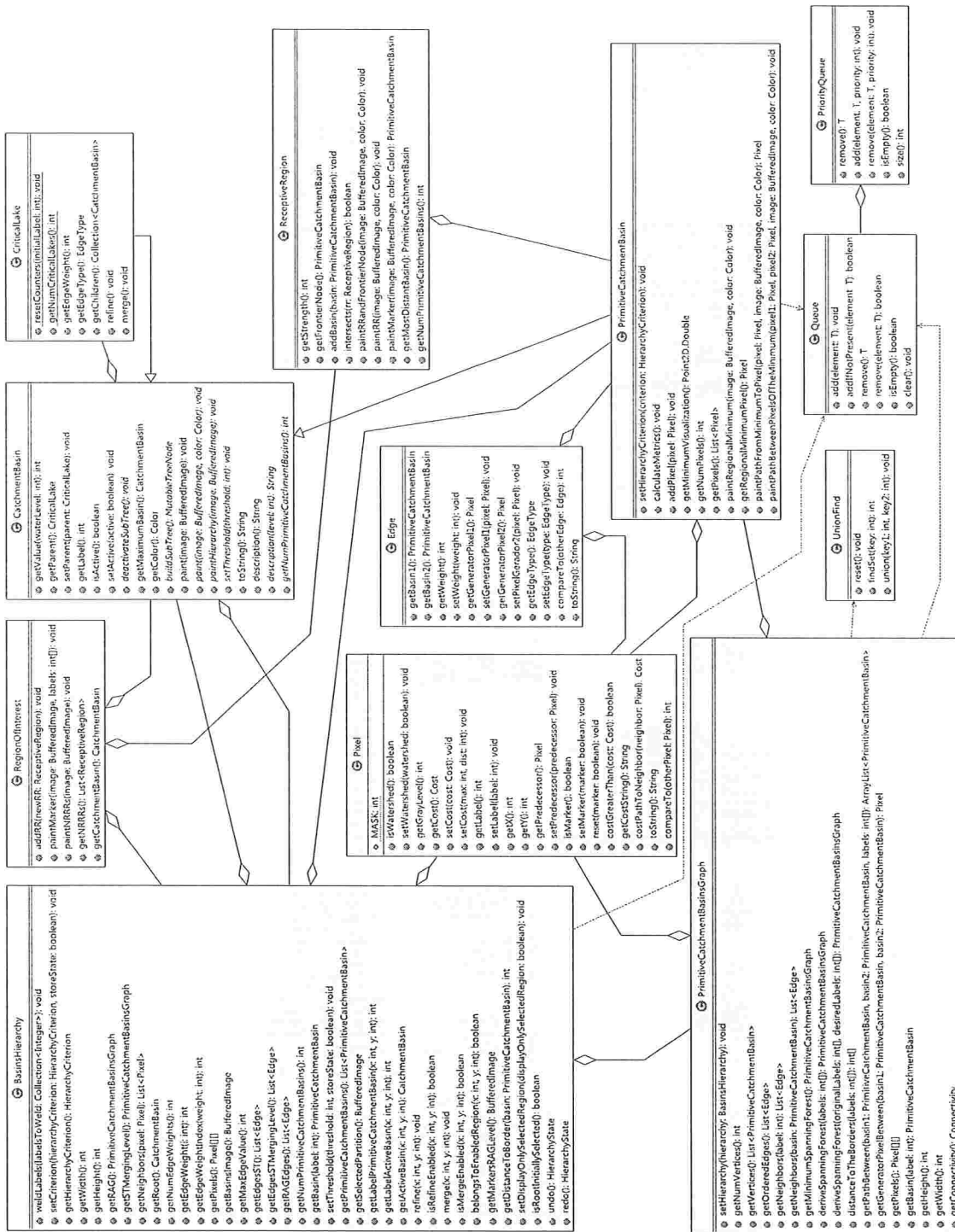


Figura B.7: Diagrama de classes do pacote *br.usp.ime.klava.segmentit.structures*.

de uma imagem, geração de imagens que representam uma matriz de *pixels* rotulados, entre outros.

A classe *Messages* centraliza todas as informações textuais exibidas na interface gráfica do *SegmentIt*, facilitando a internacionalização dessa (a interface gráfica da ferramenta exibe informações textuais em inglês ou português, dependendo da configuração de idiomas da JVM⁴ sobre a qual está sendo executada).

A enumeração *ColorSpace* provê métodos para gerar uma cor única para cada número num dado intervalo, o que é utilizado, por exemplo, para associar uma cor única a cada bacia de retenção (bacia primitiva ou lago crítico) de uma ALC ou cada região de uma partição. Existem dois *ColorSpaces* disponíveis: *DEFAULT*, que utiliza a ordenação padrão de cores do Java, e *HUES*, que gera cores que variam uniformemente ao longo da dimensão das matizes (brilho e saturação sempre máximos).

A classe *UndoRedo* possibilita o armazenamento de estados do sistema, assim como métodos para gerenciá-los e possibilitar desfazer/refazer ações. A classe *HierarchyState* representa o estado do sistema durante a segmentação hierárquica, armazenando o critério utilizado para a construção de uma dada hierarquia, bem como o limiar utilizado e uma rotulação das bacias primitivas resultante de operações locais e fusão arbitrárias.

B.1.8 br.usp.ime.klava.segmentit.watershed

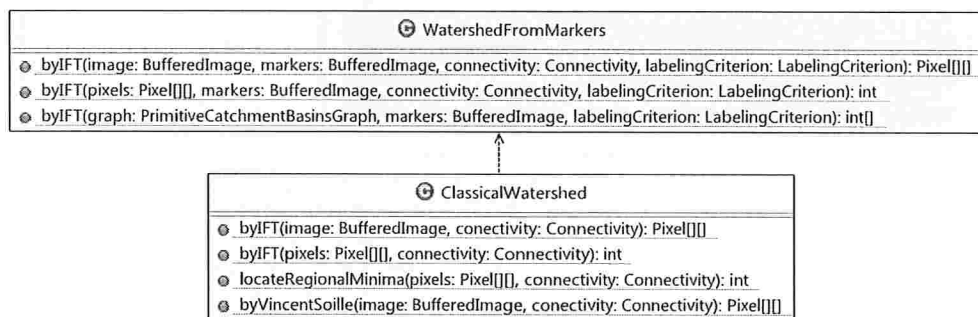


Figura B.9: Diagrama de classes do pacote *br.usp.ime.klava.segmentit.watershed*.

Neste pacote encontram-se implementações para o *watershed* clássico e para o *watershed* a partir de marcadores (tanto para a granularidade de *pixels* quanto para a granularidade de bacias primitivas).

Na classe *ClassicalWatershed* encontram-se métodos para o *watershed* clássico pela IFT, já que o algoritmo de Vincent e Soille [26] não associa todos os *pixels* de uma imagem a

⁴Java Virtual Machine

uma bacia primitiva, limitação que não existe no Algoritmo 3, que sempre associa um *pixel* a uma bacia primitiva, mesmo que este pertença à linha de *watershed*. Para calcular os mínimos regionais para serem utilizados como marcadores na IFT, é utilizada um versão simplificada do algoritmo de Vincent e Soille – método *locateRegionalMinima*.

B.2 Interface gráfica

Na interface principal do *SegmentIt*, exibida na Fig. B.10, o usuário pode manipular as imagens que deseja segmentar através da utilização da caixa de ferramentas, painel de camadas e barra de menus.

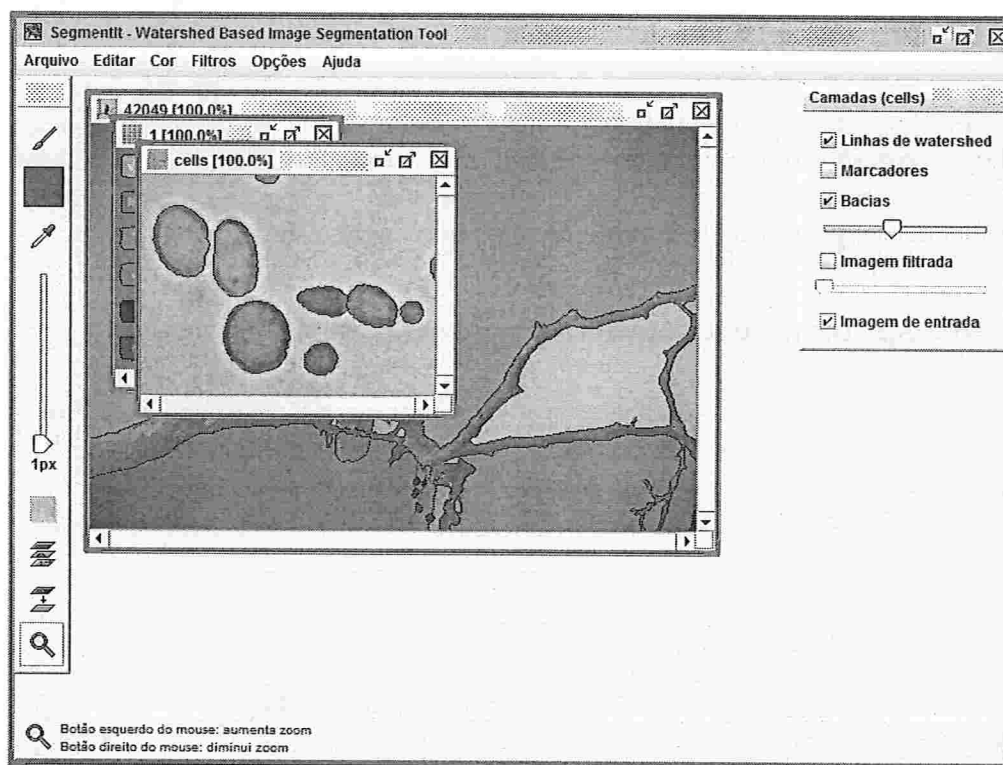


Figura B.10: Interface principal.

B.2.1 Ferramentas

A seguir, são descritas todas as ferramentas disponíveis na caixa de ferramentas do *SegmentIt*.

- **Pincel / borracha:** Permite editar (criar e apagar) marcadores; com o botão esquerdo do *mouse* é possível criar marcadores (utilizando a cor selecionada atualmente), com o botão central do *mouse* (ou ALT + botão direito do *mouse*) é possível apagar marcadores, com o botão direito do *mouse* é possível apagar marcadores somente da cor selecionada atualmente; um clique duplo sobre a ferramenta permite limpar a camada de marcadores;
- **Cor dos pincéis:** Exibe a cor atual utilizada pelo *Pincel / borracha* e pelo *Pincel de fusão de regiões*, permitindo selecionar outras cores em tabelas e seletores HSB⁵ e RGB⁶;
- **Seletor de cores:** permite selecionar uma cor utilizada na camada de marcadores com o botão esquerdo do *mouse* (seleciona o *Pincel / borracha* automaticamente após uma cor ser selecionada);
- **Diâmetro dos pincéis:** exibe e permite alterar o diâmetro utilizado pelo *Pincel / borracha* e pelo *Pincel de fusão de regiões*;
- **Segmentar:** executa o algoritmo de *watershed* utilizando os marcadores desenhados até o momento (disponível somente se a opção *Segmentar automaticamente*, Seção B.2.3, não estiver selecionada no menu *Opções*);
- **Hierarquia:** permite alternar para a abordagem de segmentação hierárquica (Seção B.2.4), utilizando a imagem inteira ou a região sobre a qual o usuário clica com o *mouse*, caso já exista uma partição;
- **Pincel de fusão de regiões** (disponível também na janela de segmentação hierárquica, Seção B.2.4): permite desenhar marcadores e fundir as regiões que intersectam com o marcador desenhado, computando automaticamente marcadores que recuperam a partição que funde as regiões selecionadas e preserva as demais regiões separadas (esta operação é realizada sobre o RAG, portanto pode haver diferenças se a precisão de *pixels* estiver sendo utilizada);
- **Zoom** (disponível também na janela de segmentação hierárquica, Seção B.2.4): permite alterar a ampliação de visualização (a ferramenta de *Zoom* não precisa ser selecionada para ser utilizada, estando associada à *mouse wheel* e disponível independentemente da ferramenta selecionada); através do botão esquerdo do *mouse* permite aumentar a ampliação (multiplica por 2 o nível atual), enquanto que com o botão direito do *mouse* permite diminuir a ampliação (divide por 2 o nível atual); um clique duplo sobre a ferramenta restabelece o nível de ampliação para 100%.

⁵HSB: do inglês *hue, saturation and brightness*

⁶RGB: do inglês *red, green and blue*

Na barra de *status* (parte inferior da tela do *SegmentIt*) é exibida uma breve ajuda sobre a utilização da ferramenta selecionada (ferramentas selecionáveis: *Pincel / borracha*, *Seletor de cores*, *Hierarquia*, *Pincel de fusão de regiões* e *Zoom*).

B.2.2 Camadas

A seguir, são descritas as camadas que constituem o painel de camadas do *SegmentIt*.

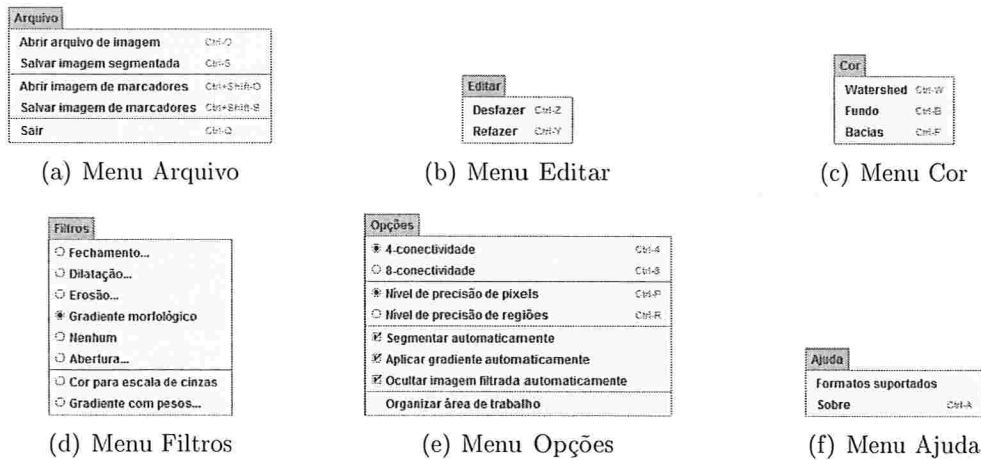
- **Linhas de *watershed*:** exibe as linhas de *watershed* obtidas pelo Algoritmo 3 (disponível apenas se a precisão de *pixels* estiver sendo utilizada);
- **Marcadores:** exibe os marcadores utilizados como entrada para o algoritmo de *watershed*;
- **Bacias:** associa uma cor a cada região de interesse da partição atual, sendo possível controlar sua opacidade através de um controle deslizante;
- **Imagem filtrada:** exibe a imagem filtrada utilizada, sendo possível controlar seu contraste através de um controle deslizante;
- **Imagem de entrada:** exibe a imagem de entrada (alvo da segmentação).

A caixa de seleção ao lado de cada camada permite controlar a sua visibilidade.

B.2.3 Menus

A seguir, são descritas todas as opções disponíveis nos menus do *SegmentIt*, apresentados na Fig. B.11.

- **Arquivo**
 - **Abrir arquivo de imagem:** abre uma imagem para ser segmentada no *SegmentIt*, criando um *SegmentationDocument* (veja Seção B.1.4) para contê-la;
 - **Salvar imagem segmentada:** salva a imagem segmentada do *SegmentationDocument* em foco, sobrepondo as imagens das camadas visíveis (Seção B.2.2);
 - **Abrir imagem de marcadores:** abre uma imagem para ser utilizada como marcadores para o *SegmentationDocument* em foco, sobrescrevendo marcadores pré-existentes;
 - **Salvar imagem de marcadores:** salva a imagem de marcadores do *SegmentationDocument* em foco em um arquivo;

Figura B.11: Menus do *SegmentIt*.

- **Sair**: finaliza a execução do *SegmentIt*.
- **Editar** (disponível também na janela de segmentação hierárquica, Seção B.2.4)
 - **Desfazer**: desfaz a última ação⁷ realizada;
 - **Refazer**: refaz a última ação desfeita.
- **Cor**
 - **Watershed**: modifica a cor das linhas de *watershed*;
 - **Fundo**: modifica a cor do fundo (visível somente quando as imagens de entrada e filtrada estão ocultas);
 - **Bacias**: modifica o espaço de cores utilizado para pintar as bacias de retenção (regiões de interesse) da partição atual. As opções disponíveis são: padrão (escolhe cores uniformemente distribuídas utilizando o endereçamento padrão de cores utilizado pelo Java) e matizes (escolhe cores uniformemente distribuídas no domínio das matizes, utilizando saturação e brilho máximos).
- **Filtros**⁸
 - **Fechamento**: utiliza o fechamento morfológico da imagem de entrada como entrada para o algoritmo de *watershed*;

⁷As ações que são possíveis ser desfeitas/refeitas são: edição de marcadores e operações hierárquicas. É armazenado um histórico de 10 ações.

⁸Atualmente somente o último filtro selecionado é aplicado sobre a imagem de entrada, ou seja, não há composição de filtros, limitação que será resolvida em futuras versões da ferramenta.

- **Dilatação:** utiliza a dilatação morfológica da imagem de entrada como entrada para o algoritmo de *watershed*;
- **Erosão:** utiliza a erosão morfológica da imagem de entrada como entrada para o algoritmo de *watershed*;
- **Gradiente morfológico** (padrão para imagens em níveis de cinza): utiliza o gradiente morfológico da imagem de entrada como entrada para o algoritmo de *watershed*;
- **Nenhum:** utiliza a própria imagem de entrada como entrada para o algoritmo de *watershed*;
- **Abertura:** utiliza a abertura morfológica da imagem de entrada como entrada para o algoritmo de *watershed*;
- **Cor para escala de cinzas:** utiliza uma versão em níveis de cinza da imagem de entrada como entrada para os algoritmos do *watershed*;
- **Gradiente com pesos** (padrão para imagens coloridas): utiliza o gradiente com pesos (Seção 5.2) da imagem de entrada como entrada para o algoritmo de *watershed*.

- **Opções**

- **4-conectividade:** utiliza a 4-conectividade como relação de adjacência para criação do grafo que representa a imagem sendo segmentada;
- **8-conectividade:** utiliza a 8-conectividade como relação de adjacência para criação do grafo que representa a imagem sendo segmentada;
- **Nível de precisão de *pixels*:** utiliza da granularidade de *pixels* na abordagem de marcadores;
- **Nível de precisão de regiões:** utiliza a granularidade de bacias primitivas na abordagem de marcadores (por padrão é a opção utilizada para cada *SegmentationDocument* criado, induzindo à utilização da abordagem *coarse to fine*);
- **Segmentar automaticamente:** executa o algoritmo de *watershed* a cada edição de marcadores (pode ser desabilitada para melhorar a performance durante a segmentação de imagens grandes; neste caso, a ferramenta Segmentar pode ser utilizada para executar o algoritmo de *watershed*);
- **Aplicar gradiente automaticamente:** aplica o gradiente morfológico automaticamente ao abrir uma imagem (o gradiente com pesos é aplicado caso a imagem de entrada seja colorida);
- **Ocultar imagem filtrada automaticamente:** oculta automaticamente a camada da imagem filtrada ao abrir uma imagem;

- **Organizar área de trabalho:** reposiciona a caixa de ferramentas e o painel de camadas, além de exibir as imagens abertas em cascata.
- **Ajuda**
 - **Formatos suportados:** exibe informação sobre os formatos de imagens suportados;
 - **Sobre:** exibe janela de informações sobre o *SegmentIt*.

B.2.4 Outras janelas

Na janela de **segmentação hierárquica** (exibida na Fig. B.12 e acessível a partir da ferramenta *Hierarquia*, Seção B.2.1), é possível navegar no conjunto hierárquico de partições. É possível criar hierarquias utilizando os seguintes critérios: nível de fusão, profundidade, área e volume (seleção padrão). A navegação é feita através de um controle deslizante, utilizado para selecionar um limiar que afeta todas as regiões da hierarquia, e das operações locais (divisão e fusão de regiões), acessíveis através do menu de contexto ou da *mouse wheel*. As ferramentas disponíveis são o *Pincel de fusão de regiões* e o *Zoom*, descritos na Seção B.2.1.

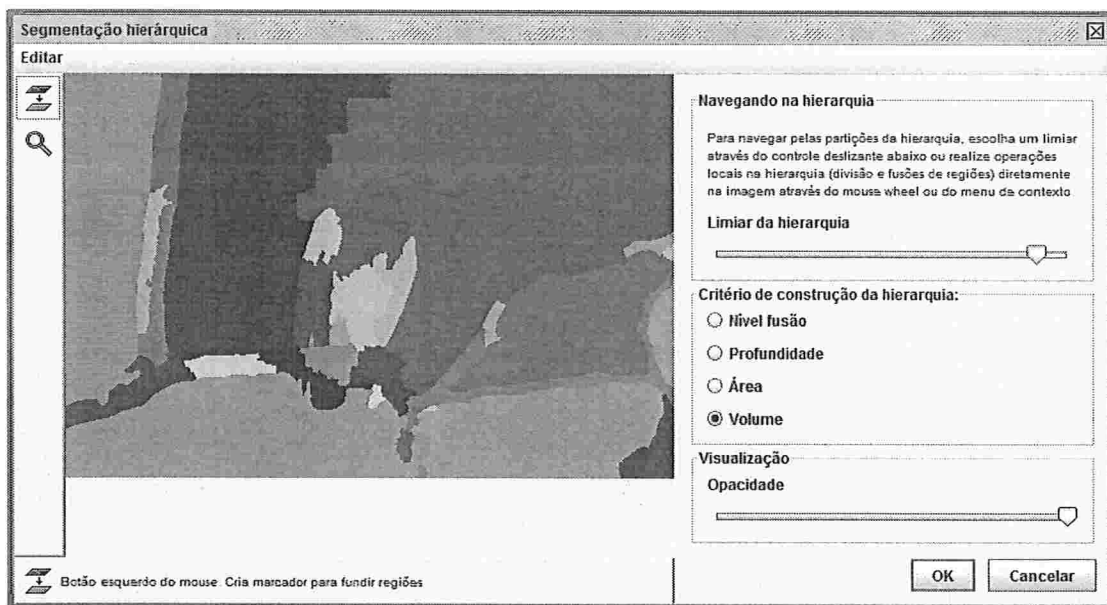


Figura B.12: Interface da abordagem hierárquica.

Na janela do filtro **gradiente com pesos** (exibida na Fig. B.13), é possível escolher

entre os pesos calculados automaticamente pelo método descrito em [10] ou escolher pesos arbitrários para as bandas de matiz, saturação e brilho.

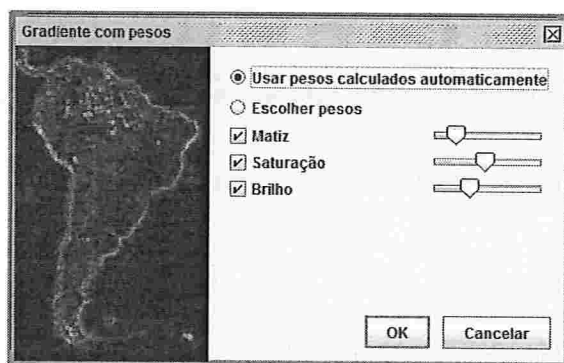


Figura B.13: Interface do filtro gradiente com pesos.

Referências Bibliográficas

- [1] AUDIGIER, R., AND LOTUFO, R. Seed-relative segmentation robustness of watershed and fuzzy connectedness approaches. In *SIBGRAPI '07: Proceedings of the XX Brazilian Symposium on Computer Graphics and Image Processing* (Washington, DC, USA, 2007), IEEE Computer Society, pp. 61–70. 34
- [2] AUDIGIER, R., LOTUFO, R. A., AND COUPRIE, M. The tie-zone watershed: Definition, algorithm and applications. In *Proceedings of IEEE International Conference on Image Processing (ICIP'05)* (2005), pp. 654–657. 20
- [3] BEUCHER, S. Watershed, hierarchical segmentation and waterfall algorithm. In *Mathematical Morphology and its Applications to Image Processing*, J. Serra and P. Soille, Eds. Eds. Kluwer Acad. Publ., 1994, pp. 69–76. 3
- [4] BEUCHER, S., AND LANTUEJOUL, C. Use of watersheds in contour detection. In *Proceedings of the International Workshop on Image Processing: Real-Time Edge and Motion Detection/Estimation* (September 1979). 15
- [5] BEUCHER, S., AND MEYER, F. The morphological approach to segmentation: the watershed transformation. In *Mathematical morphology in image processing*, E. Dougherty, Ed. M. Dekker, 1993, ch. 12, pp. 433–481. 3
- [6] DA SILVA, W. D. F. *Marcadores mínimos usando watershed*. PhD thesis, UNICAMP, dezembro 2001. 35
- [7] DOUGHERTY, E. R., AND LOTUFO, R. A. *Hands-on Morphological Image Processing*. SPIE Publications, 2003, ch. 7 – Morphological Segmentation – Watershed. x, 18

- [8] DOUGHERTY, E. R., AND LOTUFO, R. A. *Hands-on Morphological Image Processing*. SPIE Publications, 2003, ch. 6 – Morphological Processing of Gray-Scale Images. 14
- [9] FALCÃO, A. X., STOLFI, J., AND DE ALENCAR LOTUFO, R. The image foresting transform: Theory, algorithms, and applications. *IEEE Trans. Pattern Anal. Mach. Intell.* 26, 1 (2004), 19–29. 18
- [10] FLORES, F. C., POLIDORIO, A. M., AND LOTUFO, R. A. Color image gradients for morphological segmentation: The weighted gradient improved by automatic imposition of weights. In *SIBGRAPI '04: Proceedings of the Computer Graphics and Image Processing, XVII Brazilian Symposium* (Washington, DC, USA, 2004), IEEE Computer Society, pp. 146–153. 49, 66, 80
- [11] GONZALEZ, R. C., AND WOODS, R. E. *Digital Image Processing*, 2nd ed. Prentice Hall, January 2002. 1
- [12] Google Maps. <http://maps.google.com/>. x
- [13] HAHN, H. K., AND PEITGEN, H.-O. IWT - interactive watershed transform: A hierarchical method for efficient interactive and automated segmentation of multidimensional gray-scale images. In *Medical Imaging 2003: Image Processing. Proceedings of the SPIE* (may 2003), vol. 5032, pp. 643–653. 4
- [14] KLAVA, B., AND HIRATA, N. S. T. Watershed segmentation: Switching back and forth between markers and hierarchies. In *Proceedings of ISMM 2007 - 8th International Symposium on Mathematical Morphology* (São José dos Campos, October 10–13, 2007 2007), G. J. F. Banon, J. Barrera, U. d. M. Braga-Neto, and N. S. T. Hirata, Eds., vol. 2, Universidade de São Paulo (USP), Instituto Nacional de Pesquisas Espaciais (INPE), pp. 29–30. 61
- [15] KLAVA, B., AND HIRATA, N. S. T. Watershed transform based interactive image segmentation tool. In *Proceedings of the XXIst Brazilian Symposium on Computer Graphics and Image Processing* (October 2008), vol. Technical Posters, pp. 3–4. 61
- [16] KLAVA, B., AND HIRATA, N. S. T. Interactive image segmentation with integrated use of the markers and the hierarchical watershed approaches. In *Proceedings of the Fourth International Conference on Computer Vision Theory and Applications* (February 2009), vol. 1, pp. 186–193. 61
- [17] LOTUFO, R., AND SILVA, W. Minimal set of markers for the watershed. In *6th International Symposium on Mathematical Morphology* (2002), H. Talbot and R. Beare, Eds., CSIRO Publications, pp. 359–368. 34

- [18] MARCOTEGUI, B., AND ZANOQUERA, F. Image editing tools based on multi-scale segmentation. In *Proceedings of ISMM 2002 - 6th International Symposium on Mathematical Morphology* (2002), H. Talbot and R. Beare, Eds., pp. 127–135. 4
- [19] MEYER, F. Sequential algorithms cell segmentation: maximum efficiency? In *Proceedings of the International Symposium on Clinical Cytometry and Histometry* (1986). 15
- [20] MEYER, F. The dynamics of minima and contours. In *Mathematical Morphology and its Applications to Image and Signal Processing*, P. Maragos, R. W. Schafer, and M. A. Butt, Eds. Kluwer Academic Publishers, 1996, pp. 329–336. 25
- [21] MEYER, F. Hierarchies of partitions and morphological segmentation. In *Scale-Space '01: Proceedings of the Third International Conference on Scale-Space and Morphology in Computer Vision* (London, UK, 2001), Springer-Verlag, pp. 161–182. 29
- [22] MEYER, F. An overview of morphological segmentation. *International Journal of Pattern Recognition and Artificial Intelligence (IJPRAI)* 15, 7 (2001), 1089–1118. 4, 15
- [23] MEYER, F., AND BEUCHER, S. Morphological segmentation. *Journal of Visual Communication and Image Representation* 1, 1 (september 1990), 21–46. 15
- [24] The Berkeley Segmentation Dataset and Benchmark. <http://www.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/>. x, 3
- [25] UDUPA, J., LEBLANC, V., ZHUGE, Y., IMIELINSKA, C., SCHMIDT, H., CURRIE, L., HIRSCH, B., AND WOODBURN, J. A framework for evaluating image segmentation algorithms. *Computerized Medical Imaging and Graphics* 18, 2 (2006), 75–87. 3
- [26] VINCENT, L., AND SOILLE, P. Watersheds in digital spaces: An efficient algorithm based on immersion simulations. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 13, 6 (june 1991), 583–598. 16, 73
- [27] Wikimedia Commons. <http://commons.wikimedia.org/>. x
- [28] ZANOQUERA, F., MARCOTEGUI, B., AND MEYER, F. A toolbox for interactive segmentation based on nested partitions. In *Proceedings of the International Conference on Image Processing* (1999), vol. 1, pp. 21–25. 24