

**Segmentação de Imagens pela Transformada
Imagem-Floresta**

Paulo André Vechiatto de Miranda

Dissertação de Mestrado

Segmentação de Imagens pela Transformada Imagem-Floresta

Paulo André Vechiatto de Miranda¹

15 de janeiro de 2006

Banca Examinadora:

- Prof. Dr. Alexandre Xavier Falcão
Instituto de Computação - UNICAMP (Orientador)
- Prof. Dr. Antonio Alberto Fernandes de Oliveira
Programa de Engenharia de Sistemas e Computação (PESC) - COPPE/UFRJ
- Prof. Dr. Neucimar Jerônimo Leite
Instituto de Computação - UNICAMP
- Prof. Dr. Siome Klein Goldenstein (Suplente)
Instituto de Computação - UNICAMP

¹Financiado pela Fundação de Amparo a Pesquisa do Estado de São Paulo (FAPESP), processo número 03/09793-1

FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA CENTRAL DA UNICAMP

Bibliotecário: Helena Joana Flipsen – CRB-8ª / 5283

M672s

Miranda, Paulo André Vechiatto de.
Segmentação de imagens pela transformada imagem -
floresta / Paulo André Vechiatto de Miranda. -- Campinas,
SP : [s.n.], 2006.

Orientador: Alexandre Xavier Falcão.
Dissertação (mestrado) - Universidade Estadual de
Campinas, Instituto de Computação.

1. Dijkstra, Edsger W. 2. Processamento de imagens.
3. Cérebro \x Imagem de ressonância magnética. 4. Imagens
- Interpretação. I. Falcão, Alexandre Xavier. II. Universidade
Estadual de Campinas. Instituto de Computação. III. Título.

Tradução do título em inglês: Image segmentation by the image foresting transform.

Palavras-chave em inglês (Keywords): Image processing, Brain – Magnetic Resonance imaging, Picture interpretation.

Área de concentração: Ciência da Computação.

Titulação: Mestre em Ciência da Computação.

Banca examinadora: Alexandre Xavier Falcão, Antonio Alberto Fernandes, Neucimar Jerônimo Leite, Siome Klein Goldenstein.

Data da defesa: 20-02-2006.

Segmentação de Imagens pela Transformada Imagem-Floresta

Este exemplar corresponde à redação final da Dissertação devidamente corrigida e defendida por Paulo André Vechiatto de Miranda e aprovada pela Banca Examinadora.

Campinas, 20 de fevereiro de 2006.

Prof. Dr. Alexandre Xavier Falcão
Instituto de Computação - UNICAMP
(Orientador)

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

TERMO DE APROVAÇÃO

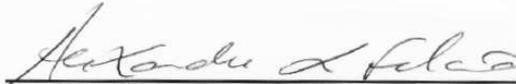
Tese defendida e aprovada em 20 de fevereiro de 2006, pela Banca examinadora composta pelos Professores Doutores:



Prof. Dr. Antônio Alberto Fernandes de Oliveira
PESC-COPPE/UFRJ.



Prof. Dr. Siome Klein Goldenstein
IC / UNICAMP.



Prof. Dr. Alexandre Xavier Falcão
IC / UNICAMP.

© Paulo André Vechiatto de Miranda, 2006.
Todos os direitos reservados.

Prefácio

Apesar do progresso das últimas duas décadas, o problema de segmentação de imagens continua sendo um dos mais difíceis desafios em análise de imagens. Mais especificamente, métodos de segmentação precisos são normalmente complexos, caros computacionalmente, dependentes de aplicação, e freqüentemente exigem interação humana.

Estamos pesquisando métodos de segmentação interativa de imagens que reduzem a intervenção do usuário a simples escolha de poucos pixels na imagem. Buscamos minimizar o número de interações de forma que a automação se torne possível em algumas aplicações onde esses pixels podem ser identificados automaticamente.

A metodologia adotada usa a *transformada imagem floresta* (*IFT - Image Foresting Transform*) - uma ferramenta geral para modelar, implementar e avaliar operadores de processamento de imagens baseados em conectividade. A IFT reduz o problema de processamento de imagem ao cálculo de uma floresta de caminhos de custo mínimo no grafo derivado da imagem. Nesse trabalho são apresentadas três novas técnicas de segmentação baseadas na IFT.

Abstract

Despite of the progress over the last two decades, image segmentation remains one of the most difficult challenges in image analysis. More specifically, accurate segmentation methods are likely to be complex, computationally expensive, application-dependent, and often require considerable human interaction.

We have investigated methods for interactive segmentation that reduce user intervention to simple selection of a few pixels in the image. We are interested in minimizing the user involvement such that automation becomes feasible for some applications where these pixels may be automatically identified.

The adopted methodology uses the framework of the *image foresting transform (IFT)* - a general tool for the design, implementation, and evaluation of image processing operators based on connectivity. The IFT reduces image processing problems into a minimum-cost path forest problem in a graph derived from the image. In this work, we present three new segmentation methods based on the IFT.

Agradecimentos

a todos que contribuíram diretamente no desenvolvimento deste trabalho: Dr. Alexandre Xavier Falcão pela orientação sempre presente e atuante, e corpo docente do Instituto de Computação da Unicamp, e também aos professores do Departamento de Neurologia da Faculdade de Ciências Médicas da Unicamp.

em especial aos professores Dr. Fernando Cendes (FCM-UNICAMP), Dr. Roberto Lotufo (FEEC-UNICAMP) e Dra. Helena Leitao (IC-UFF) pelas imagens fornecidas, e ao Dr. Ricardo da Silva Torres (IC-UNICAMP) e aos companheiros Felipe P.G. Bergo e Anderson Rocha pelos trabalhos conjuntos.

aos meus pais pelo total apoio que possibilitou os meus estudos e formação acadêmica e à Deus que colocou no meu caminho as pessoas certas nas horas certas.

à FAPESP pelo apoio financeiro e FUNCAMP pelo auxílio viagem.

Conteúdo

Prefácio	vii
Abstract	viii
Agradecimentos	ix
1 Introdução	1
1.1 Organização do trabalho	2
1.2 Imagens digitais	3
1.2.1 Imagem em tons de cinza	3
1.2.2 Amostragem e resolução espacial	3
1.2.3 Quantização e resolução radiométrica	4
1.2.4 Imagem multidimensional	4
1.2.5 Imagem multibanda	4
1.2.6 Imagem de rótulos	4
1.3 Noções de grafos	5
1.4 Mapeamento da imagem em um grafo	6
2 Transformada imagem-floresta	8
2.1 Definição	8
2.2 Algoritmo da IFT	10
2.3 Resolvendo empates	11
2.4 Fila de prioridade	12
2.5 Aplicações	12
2.6 A IFT diferencial	13
3 Pré-Processamento	14
3.1 Mapas de região	14
3.2 Imagens de gradiente realçadas	15
3.3 Exemplos	15

4	Segmentação por competição entre sementes κ-conexas	18
4.1	Introdução	18
4.2	Métodos relacionados	19
4.3	Estimativa automática de κ_s	20
4.4	Algoritmo	21
4.5	Funções de dissimilaridade avaliadas	23
4.6	Resultados experimentais	24
5	Segmentação por crescimento ótimo de regiões e corte em grafo	29
5.1	Introdução	29
5.2	Medidas de avaliação de corte em grafo	29
5.3	Algoritmo	31
5.4	Resultados experimentais	33
6	Segmentação por poda de árvores	40
6.1	Introdução	40
6.2	Métodos relacionados	40
6.3	Definição	41
6.4	Podando as árvores da floresta gerada pela IFT	42
6.5	Detecção automática de pixels de vazamento	42
6.6	Pré-processamento para a poda de árvores	45
6.7	Limitações	45
6.8	Algoritmos	48
6.9	Resultados experimentais	50
6.9.1	Rotulação de múltiplos objetos com texturas similares	50
6.9.2	Segmentação 3D de imagens RM do cérebro	51
6.9.3	Detecção de objetos baseada na forma	55
7	Conclusões	60
7.1	Visão geral	60
7.2	Discussão	62
7.3	Sugestões para trabalhos futuros	63
7.3.1	DIFT combinada com medidas de corte em grafo	63
7.3.2	Automatizar o que é ainda manual	63
7.3.3	Aplicações em medicina	63
	Bibliografia	64

Lista de Tabelas

4.1	Descrição, modalidade de imagem e número de fatias para cada objeto usado nos experimentos.	25
4.2	As funções de dissimilaridade usadas para cada combinação entre objeto e método.	27
4.3	O número médio de interações por usuário para cada objeto e método, e a média dos valores de similaridade entre ambos métodos para um mesmo objeto.	28
4.4	Número médio de sementes internas (NIS), sementes externas (NES), e detecções automáticas de κ_s (AKD).	28
6.1	Erros de segmentação dos <i>phantoms</i> do cérebro por poda de árvores e SPM2. Falsos positivos (FP) e falsos negativos (FN) são mostrados como uma porcentagem do volume de GM e WM obtida da máscara do gabarito (<i>ground truth</i>).	54

Lista de Figuras

1.1	Adjacências Euclidianas: (a) 2D de raio 1, (b) 2D de raio $\sqrt{2}$, (c) 2D de raio 2, (d) 3D de raio 1.	7
2.1	(a) Um grafo de uma imagem 2D em tons de cinza com vizinhança 4. Os números correspondem as intensidades $I(p)$ dos pixels e os pontos maiores denotam duas sementes (uma no interior do retângulo claro e uma no fundo mais escuro ao redor dele). O fundo também apresenta alguns pixels claros. (b) Uma floresta de caminhos de custo mínimo usando c_{\max} , com $\delta(p, q) = I(q) - I(p) $. Os números são os valores de custo e o retângulo é obtido como uma árvore enraizada na semente interna. As setas em (b) apontam para o predecessor no caminho ótimo.	9
3.1	(a) Imagem 2D em tons de cinza (seção 1.2.1) de um corte coronal do cérebro com três pixels de treinamento (os círculos indicam os raios de adjacência usados). (b) Mapa de região 2D do cérebro. (c) Gradiente realçado 2D baseado no mapa de região.	16
3.2	(a) Uma imagem colorida (seção 1.2.5) de pimentões com quatro pixels de treinamento (os círculos indicam seus raios de adjacência) e (b) o mapa de região do pimentão vermelho central. (c) Uma imagem colorida de duas vacas com dois pixels de treinamento e (d) o mapa de região da vaca mais escura.	17
4.1	Segmentação do globo ocular por competição entre sementes em imagem de tomografia de raios-X. (a) Uma semente interna e várias sementes externas são necessárias para a segmentação, usando c_{\max} com δ_4 (equação 4.4). (b) A segmentação falha quando algumas sementes externas são removidas. (c) Um limiar κ limita a zona de influência da semente interna onde a competição entre sementes falha. Observe a redução do número de sementes em relação a (a).	19

4.2	Uma imagem TC de um joelho onde a patela pode ser segmentada pela união das zonas de influência (sem competição) de dois pixels sementes, s_1 e s_2 , usando c_{\max} com δ_3 (equação 4.3). (a) O resultado com um único valor de κ para ambas sementes. (b) A segmentação com um único valor de κ falha quando nós alteramos a localização de s_1 , porque s_1 agora requer um valor de κ mais elevado para obter a parte mais clara do osso e s_2 invade o fundo para esse valor de κ maior. (c) O resultado pode ser corrigido com o uso de valores distintos de κ para cada semente.	20
4.3	Uma imagem de tomografia de raios-X do globo ocular com uma semente no seu interior. (a) A frente de onda de custo κ que proporciona a máxima extensão contida no objeto. (b) A frente de onda de custo $\kappa + 1$ apresenta um grande aumento de área quando esta invade o fundo.	21
4.4	(a)-(g) Resultados das segmentações de fatias dos objetos de 1 à 7, respectivamente, sobrepostos às imagens pré-processadas.	26
5.1	Resultados das segmentações onde as sementes são indicadas por pontos, usando (a) c_{\max} , $w(p, q)$ pela equação 3.3 e <i>mean cut</i> , (b) c_{sum} , $w(p, q)$ pela equação 3.3 e corte normalizado, (c) c_{\max} , $w(p, q)$ pela equação 3.3 e <i>mean cut</i> , e (d-e) c_{\max} , $w(p, q)$ pela equação 3.3 e a função de energia da equação 5.6.	34
5.2	A curva do <i>mean-cut</i> versus a ordem de propagação dos pixels para uma imagem RM do cérebro (figura 5.1a). O mínimo desejado só pode ser detectado quando nós limitamos a busca até uma ordem de propagação o , tal que $o < n$ e que seja maior que o tamanho do objeto.	35
5.3	(a-b) Curvas do corte normalizado e função de energia versus a ordem de propagação para a imagem RM do cérebro. (c-d) Suas respectivas segmentações usando a mesma semente, funções de custo de caminho c_{\max} , e mapas de região.	36
5.4	Detecção de fragmentos arqueológicos. (a) A imagem original. (b) O centro de cada disco representa um pixel semente sobre a imagem filtrada. (c-d) Exemplos de detecções corretas e incorretas, respectivamente.	38

6.1	(a) Grafo com vizinhança 4 onde os números são intensidades do gradiente e o vértice maior denota uma semente no interior do objeto. (b) Uma árvore de caminhos ótimos para c_{\max} enraizada nessa semente. (c) O objeto pode ser detectado através da eliminação da subárvore do pixel de vazamento na posição (4,4). (d) O número de descendentes na borda da imagem B para a árvore de caminhos ótimos da figura 6.1b. O primeiro máximo no caminho de volta a partir de qualquer pixel em B para a raiz encontra-se no pixel (4,4).	43
6.2	(a) A magnitude do gradiente de Sobel de uma imagem para a detecção da placa do carro. (b)–(d) A placa é automaticamente detectada para diferentes localizações da semente (pontos amarelos) usando (a) como imagem de gradiente. O pixel de vazamento está sempre no mesmo lugar (ponto na cor ciano).	44
6.3	Uma imagem RM de um pulso onde sua borda pode ser detectada mediante duas iterações do método para uma única semente (ponto amarelo). (a) A primeira iteração usa a borda da imagem como conjunto B e automaticamente detecta uma borda externa com um único pixel de vazamento (ponto na cor ciano). (b) A borda externa é então usada como conjunto B e o objeto desejado é detectado com um único pixel de vazamento (ponto na cor ciano) na segunda iteração.	45
6.4	(a)–(b) Imagens de gradiente realçadas baseadas em mapas de região para a imagem de um pimentão e uma imagem RM do cérebro. (c)–(d) Resultados de segmentação para ambas imagens, onde as sementes são selecionadas manualmente (pontos amarelos) e os pixels de vazamento (pontos na cor ciano) são automaticamente detectados.	46
6.5	(a) Imagem com objeto de borda perfeita e (b) imagem com objeto de borda contendo aberturas perfeitas, após quatro podas interativas indicadas pelas setas. (c) A magnitude do gradiente de Sobel aplicada em uma imagem RM de uma veia do pulso. (d) Resultado da segmentação para uma semente interna (ponto amarelo) usando (c) como imagem de gradiente. Dois pixels de vazamento são automaticamente detectados (pontos em ciano).	47
6.6	Imagem com um fragmento marcado para detecção. (a) O pixel de vazamento detectado está fora do fragmento devido a proximidade com outros fragmentos. (b) O pixel de vazamento correto é automaticamente detectado usando a variante do gradiente máximo.	48
6.7	Tecidos em imagens RM-T1 do cérebro. Substância branca (<i>white matter</i>), substância cinza (<i>gray matter</i>), dura mater, sulcos (<i>sulci</i>), osso (<i>bone</i>), pele (<i>skin</i>), ar (<i>air</i>) e líquido cérebro-espinhal (CSF).	51

6.8	(a) Corte sagital de um volume RM-T1 do cérebro. (b) Visão sobreposta da limiarização de Otsu aplicada no volume para o corte em (a). (c) Corte sagital do gradiente 3D final. (d) Visão dos voxels sementes para o corte em (a).	52
6.9	Amostras de fatias do experimento envolvendo segmentação de <i>phantoms</i> do cérebro: (a) Gabarito do <i>phantom</i> (<i>ground truth</i>) e a segmentação obtida com (b) SPM2, e (c) com a técnica de poda de árvores.	54
6.10	Renderizações 3D de segmentações incorretas de cérebro obtidas com o SPM2.	55
6.11	Renderizações 3D de segmentações do cérebro obtidas pela poda automática de árvores. As duas colunas da esquerda correspondem a controles e as outras duas colunas são de pacientes com epilepsia.	55
6.12	(a) Imagem original. (b) Imagem de gradiente. (c) Imagem (a) após limiarização. (d) Imagem (c) após erosão e dilatação. (e) Imagem com marcadores rotulados. (f) Os melhores retângulos para três objetos candidatos falsos.	57
6.13	(a)-(d) Detecções corretas com <i>scores</i> 0.937, 0.913, 0.902, e 0.767, respectivamente. (e)-(f) Detecções erradas com <i>scores</i> 0.917 e 0.801, respectivamente.	59

Capítulo 1

Introdução

Apesar do progresso das últimas duas décadas, o problema de segmentação de imagens continua sendo um dos mais difíceis desafios em análise de imagens [21]. Mais especificamente, métodos de segmentação precisos são normalmente complexos, caros computacionalmente, dependentes de aplicação, e freqüentemente exigem interação humana.

Segmentar uma imagem consiste em particioná-la em regiões relevantes para uma dada aplicação (e.g. objetos e fundo). A segmentação é uma das principais etapas na maioria das aplicações e representa um dos maiores desafios em processamento de imagens. A principal razão desta dificuldade está na falta de informação sobre os objetos nas imagens. Métodos de segmentação podem ser classificados pelo tipo de representação que extraem, como baseados em borda [42, 38, 56, 55, 12, 15, 32] ou em região [67, 27, 66, 72, 63]. Métodos baseados em região extraem o conjunto de pixels que representa o interior do objeto, incluindo os pixels de fronteira. Métodos baseados em borda possuem algumas complicações relacionadas à representação das bordas (e.g. contornos fechados) e ao casamento dessa representação com o espaço discreto da imagem. Nesse trabalho exploramos apenas métodos de segmentação por região, uma vez que esses são facilmente extensíveis para imagens multidimensionais (seção 1.2.4).

A segmentação possui importantes aplicações em medicina. A análise quantitativa de estruturas cerebrais a partir de imagens de ressonância magnética (RM) tem desempenhado um papel importante para a pesquisa em neurologia [52, 4, 10, 60] e pode ser muito útil no diagnóstico e tratamento de doenças relacionadas com alterações na anatomia do cérebro humano. No entanto a extração de características das estruturas (textura, volume, assimetria de forma/textura) só é possível após a segmentação. Métodos automáticos e eficientes permitiriam a análise em larga escala, possibilitando o acompanhamento de alterações entre controles e pacientes, ou mesmo entre diferentes fases do tratamento de um indivíduo. Portanto o avanço do estado da arte em segmentação é uma contribuição em ciência da computação mas também com reflexos na medicina.

A nossa meta durante o projeto foi desenvolver métodos de segmentação interativa de imagens que reduzissem a intervenção do usuário a simples escolha de poucos pixels na imagem, buscando minimizar o número de interações. Dessa forma a automação se torna possível em algumas aplicações onde esses pixels podem ser identificados automaticamente. Por exemplo, podemos usar abordagens para localizar o objeto na imagem [68] para estimar sementes. Também podemos usar sementes candidatas para obter um conjunto de possíveis objetos, sendo o objeto desejado selecionado a partir de funções objetivo [63, 43] ou baseado em características. No pior caso, objetos candidatos podem ser usados como inicializações efetivas para modelos deformáveis [42, 12, 15, 14, 37, 56].

A metodologia adotada usa a *transformada imagem floresta (IFT - Image Foresting Transform)* - uma ferramenta geral para modelar, implementar e avaliar operadores de processamento de imagens baseados em conectividade [30]. A IFT reduz o problema de processamento de imagem ao cálculo de uma floresta de caminhos de custo mínimo no grafo derivado da imagem (seção 1.4).

Nesse trabalho serão apresentadas três novas técnicas de segmentação baseadas na IFT: (i) *segmentação por competição entre sementes κ -conexas* [25, 26], (ii) *segmentação por crescimento ótimo de regiões e corte em grafo* [24]¹, e (iii) *segmentação por poda de árvores* [22, 23]. Estas técnicas possuem extensões diretas para 3D, reduzem o problema de segmentação à seleção de alguns pixels sementes nas imagens e têm sido avaliadas com sucesso em algumas aplicações, onde as sementes são selecionadas interativamente e automaticamente. A técnica (i) ainda requer a escolha de um parâmetro pelo usuário, mas as técnicas (ii) e (iii) podem ser implementadas sem depender de parâmetros *ad-hoc*. Em particular, a extensão 3D da técnica (iii) tem se mostrado bastante promissora para isolar automaticamente o cérebro humano (substâncias cinza e branca) em imagens de RM ponderadas em T1 (RM-T1) [23].

1.1 Organização do trabalho

Teoria dos grafos é uma área bem estabelecida que conta com diversas aplicações nas ciências e engenharias. A interpretação de uma imagem digital como um grafo (seção 1.4) possibilita explorar algoritmos eficientes em grafo na solução de diversos problemas de processamento de imagens, tais como a IFT [30]. Logo, no decorrer deste capítulo serão apresentados conceitos básicos e notações referentes a imagens digitais e teoria dos grafos que serão usados ao longo dos capítulos seguintes. No capítulo 2 introduzimos a transformada imagem-floresta. A eficácia dos métodos de segmentação baseados nessa técnica requer um bom contraste nas bordas dos objetos. Essa questão é tratada no capítulo 3 que

¹Technical report: <http://www.ic.unicamp.br/ic-tr/Welcome-p.html>

apresenta técnicas de pré-processamento. Os próximos capítulos apresentam as técnicas desenvolvidas. No capítulo 4 é apresentada a técnica de *segmentação por competição entre sementes κ -conexas*. A seguir, no capítulo 5 é apresentada a técnica que combina *crescimento ótimo de regiões com medidas de avaliação de corte em grafo*. E no capítulo 6 são apresentados a teoria e resultados do método de *segmentação por poda de árvores*. Os métodos serão apresentados para imagens 2D, mas suas extensões para 3D são diretas. Na seção 6.9.2 um exemplo em 3D será mostrado em detalhes. Conclusões e trabalhos futuros são discutidos no capítulo 7.

1.2 Imagens digitais

Uma imagem digital genérica \hat{I} é um par (D_I, \vec{I}) , onde D_I é um conjunto de pontos do Z^n (domínio da imagem), denominados *spels* (*space elements*), e \vec{I} é um mapeamento vetorial que associa a cada spel p em D_I um conjunto $\{I_1(p), I_2(p), \dots, I_k(p)\}$ de valores escalares, associados com alguma propriedade física. O valor de n refere-se à dimensão da imagem e o valor de k ao número de bandas.

1.2.1 Imagem em tons de cinza

Uma imagem $\hat{I} = (D_I, I)$ em tons de cinza (e.g. foto, imagem de ultrassom, fatia tomográfica) e bidimensional ($D_I \subset Z^2$) possui apenas uma banda I ($k = 1$), onde os spels são chamados pixels (*picture elements*). A imagem bidimensional é portanto uma matriz de tamanho $N \times M$ pixels (N linhas e M colunas).

Os valores $I(p)$ de cada pixel p são obtidos por amostragem e quantização de uma função contínua $I_c(x, y)$ que descreve a propriedade física correspondente em uma dada região do espaço. No caso de uma foto temos o brilho, e no caso de uma tomografia de raios-X, temos a densidade do tecido. Valores altos são apresentados na tela como pixels claros e valores baixos como pixels escuros.

1.2.2 Amostragem e resolução espacial

Cada pixel é amostrado a intervalos (Δ_x, Δ_y) (e.g. $\Delta_x = \Delta_y = 1mm$). Quanto menor for o intervalo de amostragem para uma mesma região do espaço, maior será a resolução espacial da imagem. Observe que neste caso, o tamanho $N \times M$ da imagem também é maior, mas se uma imagem tem mais pixels que outra, não implica que tenha maior resolução.

1.2.3 Quantização e resolução radiométrica

Os valores de $I_c(x, y)$ amostrados são quantizados em 2^b níveis de cinza, onde b é chamado profundidade da imagem em bits (e.g. $b = 8$, profundidade de 8 bits). Quanto menor o intervalo de quantização, maior é a resolução radiométrica da imagem.

1.2.4 Imagem multidimensional

Uma imagem $\hat{I} = (D_I, I)$ em tons de cinza e multidimensional define domínio de amostragem $D_I \subset Z^n$, para $n > 2$. Por exemplo, uma seqüência espacial de fatias tomográficas é uma imagem tridimensional ($n = 3$), e uma seqüência espacial e temporal de fatias tomográficas é uma imagem tetradimensional ($n = 4$). No primeiro caso, os spels são chamados de *voxels* (*volume element*) e a imagem é chamada de *volume*.

O intervalo de amostragem ao longo de eixo temporal define a resolução temporal da imagem. Quanto menor o intervalo, maior é a resolução.

1.2.5 Imagem multibanda

Uma imagem $\hat{I} = (D_I, \vec{I})$ é multibanda quando o mapeamento vetorial \vec{I} associa a cada spel p obrigatoriamente mais de um valor escalar ($\vec{I}(p) = \{I_1(p), \dots, I_k(p)\}$, para $k > 1$). Imagens de satélite, por exemplo, possuem bandas que correspondem a imagens cinza obtidas nos comprimentos de onda do azul, vermelho, verde, infravermelho, infravermelho próximo, termal, e ultravioleta. O intervalo de amostragem define a resolução espectral. No caso de uma foto colorida temos $k = 1, 2, 3$ correspondendo aos componentes vermelho, verde e azul. Observe que o vídeo colorido é uma imagem multidimensional e multibanda.

1.2.6 Imagem de rótulos

A segmentação de uma imagem $\hat{I} = (D_I, \vec{I})$ baseada em região pode ser vista como um mapeamento que associa para todo pixel $p \in D_I$ um inteiro $L(p)$, denominado rótulo, cujo valor é um identificador único de cada objeto (incluindo o fundo). Neste caso, a segmentação é dita *hard* porque cada pixel p só pertence a um único objeto. Algumas abordagens estendem este conceito para *fuzzy*, onde cada pixel p pertence a todos os objetos com diferentes graus de pertinência (e.g. atlas do cérebro humano [36, 48]). Técnicas de análise, porém, requerem uma definição precisa da extensão espacial das estruturas em estudo. Portanto nesse trabalho visamos a segmentação *hard* das estruturas. A segmentação *hard* pode ser representada como uma imagem em tons de cinza $\hat{L} = (D_I, L)$ chamada imagem de rótulos.

1.3 Noções de grafos

Um grafo $G(V, E)$ é um conjunto finito não vazio V e um conjunto E composto por pares de elementos distintos de V . G é chamado *trivial* quando $|V| = 1$. Dizemos que o grafo é *direcionado* (*dígrafo*) ou *orientado* se E for um conjunto de pares ordenados de elementos de V . Caso contrário o grafo é dito *não direcionado*. Os elementos de V são os *vértices* ou *nós* e os de E são as *arestas* de G , respectivamente. Cada aresta $e \in E$ será denotada pelo par de vértices $e = (v, w)$ que a forma. No caso de grafos não direcionados, a aresta e é dita *incidente* a ambos v, w . Em dígrafos a aresta $e = (v, w)$ é dita *divergente* de v e *convergente* a w . Dizemos que w é *adjacente* a v se $(v, w) \in E$. A cada vértice v de um grafo está associada uma *lista de adjacências* $A(v)$ que define uma vizinhança composta pelos vértices w adjacentes a v , ou seja, $A(v) = \{w \in V | (v, w) \in E\}$. Define-se *grau* (ou *grau de saída* no caso de dígrafos) de um vértice $v \in V$, denotado por $\text{grau}(v)$, como sendo o número de vértices adjacentes a v . Utilizaremos a notação $n = |V|$ e $m = |E|$ para denotar a cardinalidade dos conjuntos V e E .

Uma seqüência de vértices v_1, \dots, v_k tal que $(v_j, v_{j+1}) \in E$, $1 \leq j \leq |k - 1|$, é denominado *caminho* de v_1 a v_k . Diz-se então que v_1 *alcança* ou *atinge* v_k . Um caminho de k vértices é formado por $k - 1$ arestas $(v_1, v_2), (v_2, v_3), \dots, (v_{k-1}, v_k)$. O valor $k - 1$ é o *comprimento* do caminho. Se todos os vértices do caminho v_1, \dots, v_k forem distintos, a seqüência recebe o nome de *caminho simples* ou *elementar*. Um *ciclo* é um caminho v_1, \dots, v_k, v_{k+1} sendo $v_1 = v_{k+1}$ e $k \geq 3$ no caso de grafos não direcionados. Se o caminho v_1, \dots, v_k for simples, o ciclo v_1, \dots, v_k, v_{k+1} também é denominado simples ou elementar. Um grafo que não possui ciclos simples é *acíclico*.

Um *subgrafo* $G_2(V_2, E_2)$ de um grafo $G_1(V_1, E_1)$ é um grafo tal que $V_2 \subseteq V_1$ e $E_2 \subseteq E_1$. Se $V_2 = V_1$ então G_2 é chamado de *subgrafo gerador* ou *subgrafo de espalhamento*. Um grafo $G(V, E)$ não direcionado é denominado *conexo* quando existe caminho entre cada par de vértices de G . Caso contrário G é *desconexo*. Denominam-se *componentes conexos* de um grafo G não direcionado aos subgrafos maximais de G que sejam conexos.

Denomina-se *árvore* a um grafo $T(V, E)$ não direcionado que seja acíclico e conexo. Se um vértice v da árvore T possuir $\text{grau} \leq 1$ então v é uma *folha*. Caso contrário, se $\text{grau}(v) > 1$ então v é um *vértice interior*. Um conjunto de árvores é denominado *floresta*. Assim sendo, todo grafo não direcionado acíclico é uma floresta. Um grafo não direcionado G é uma árvore se e somente se existir um único caminho entre cada par de vértices de G . Uma árvore é chamada de *árvore geradora* ou de *espalhamento* se ela for um subgrafo gerador de outro grafo. Uma árvore é denominada *enraizada* quando algum vértice $v \in V$ é escolhido como especial. Este vértice é então chamado de *raiz* da árvore. Podemos adicionar orientação nas arestas de uma árvore enraizada de modo que todos vértices sejam alcançáveis a partir da raiz. A esse dígrafo se dá o nome de *árvore*

direcionada enraizada.

1.4 Mapeamento da imagem em um grafo

Uma imagem $\hat{I} = (D_I, \vec{I})$ pode ser vista como um grafo $G(V, E)$, onde os vértices em V correspondem aos *spels* em D_I e as arestas em E são definidas por uma *relação de adjacência*. Uma relação de adjacência A é uma relação binária entre vértices do grafo. Esta relação genérica pode ser descrita por um conjunto de n listas de adjacências $A(v)$, uma específica para cada vértice $v \in V$. O espaço utilizado por essa representação é $O(n + m)$. Porém, nós estamos interessados em relações que sejam irreflexivas, simétricas e invariantes a translação. Para um dado *spel*, os vértices adjacentes são obtidos por deslocamentos vetoriais fixos no espaço da imagem. Logo, podemos usar uma estrutura única que armazena as arestas de forma relativa. Isto permite uma redução do espaço de armazenamento para $O(n)$.

Por exemplo, podemos usar A consistindo de todos pares de pixels $(p, q) \in D_I \times D_I$ tais que $p \neq q$ e $d(p, q) \leq \rho$, onde $d(p, q)$ denota a distância Euclideana e ρ é uma constante especificada. Essas adjacências são chamadas de *adjacências Euclidianas*. As figuras 1.1a–c mostram alguns exemplos em 2D, $\rho = 1$ corresponde a vizinhança 4 e $\rho = \sqrt{2}$ corresponde a vizinhança 8. Na figura 1.1d é ilustrado o caso 3D com $\rho = 1$, que corresponde a vizinhança 6 em 3D. A adjacência pode então ser armazenada em uma única estrutura que guarda os deslocamentos relativos dx e dy , e também dz no caso 3D. Por exemplo, na figura 1.1a um pixel $q(x_1, y_1)$ é adjacente a um outro pixel $p(x_2, y_2)$ se e somente se $q - p = (x_1 - x_2, y_1 - y_2) \in \{(-1, 0), (0, -1), (1, 0), (0, 1)\}$.

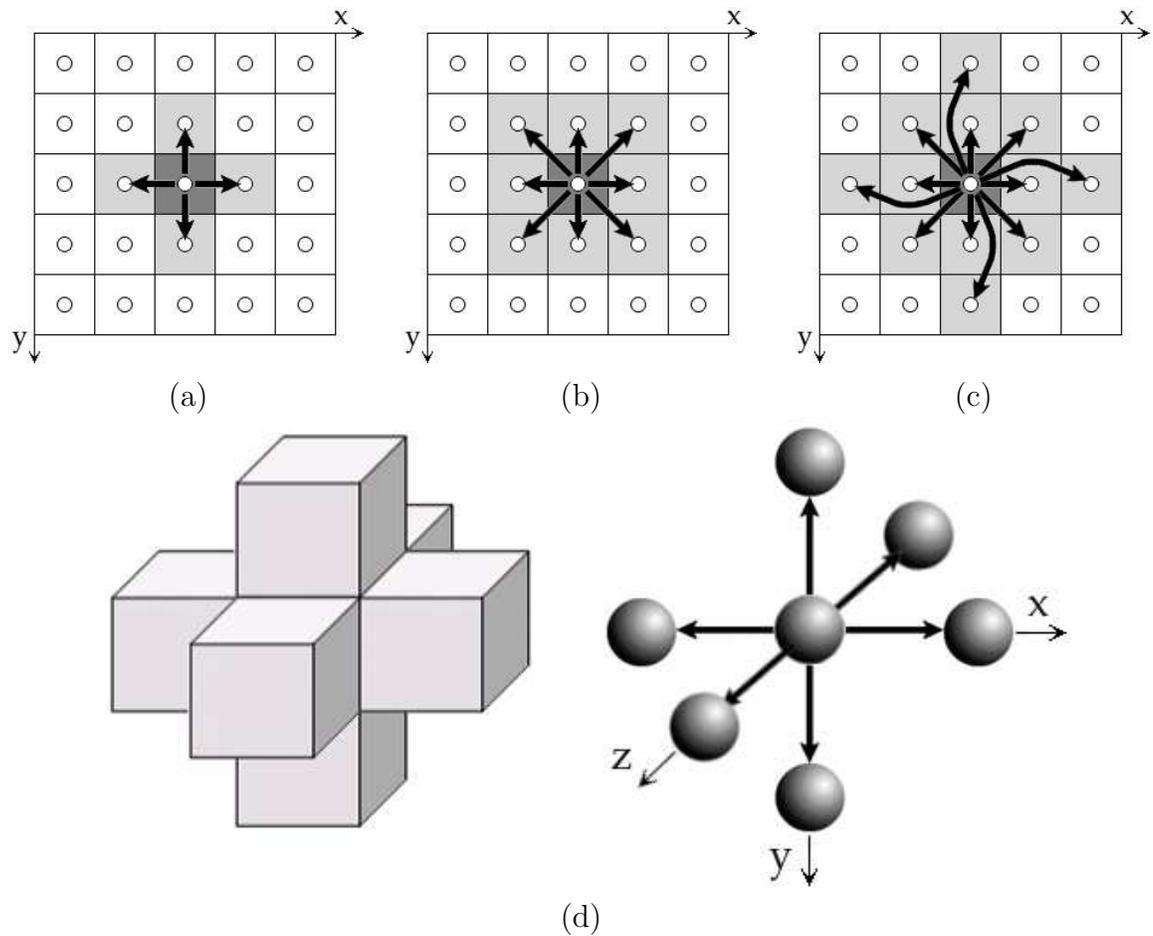


Figura 1.1: Adjacências Euclidianas: (a) 2D de raio 1, (b) 2D de raio $\sqrt{2}$, (c) 2D de raio 2, (d) 3D de raio 1.

Capítulo 2

Transformada imagem-floresta

A transformada imagem floresta (*IFT-Image Foresting Transform*) é uma ferramenta geral para modelar, implementar e avaliar operadores de processamento de imagens 2D/3D baseados em conectividade [30]. A IFT reduz o problema de processamento de imagem ao cálculo de uma floresta de caminhos de custo mínimo no grafo derivado da imagem.

Nesse capítulo, a IFT será apresentada na sua forma mais específica para a segmentação baseada em regiões. Na sua forma genérica [30], a IFT possui todas as aplicações descritas na seção 2.5, não se limitando apenas a esse problema.

2.1 Definição

Uma imagem 2D/3D pode ser vista como um grafo onde os nós são os pixels/voxels e as arestas são definidas por uma relação de adjacência A entre nós (ver seção 1.4). Um caminho π simples nesse grafo é uma seqüência de nós distintos e adjacentes. Cada caminho π possui um valor de custo associado $c(\pi)$, definido por uma *função de custo de caminho* c , que normalmente depende de propriedades locais da imagem, tais como cor, gradiente e posição do nó ao longo do caminho. Um caminho π é ótimo se $c(\pi) \leq c(\tau)$ para qualquer outro caminho τ com o mesmo destino de π . Em aplicações típicas da IFT, normalmente restringimos a busca por caminhos que se originam em um conjunto dado S de nós *sementes*. Para facilitar, podemos embutir o conjunto S na definição da função de custo. Nesse trabalho, duas funções de custo de caminho serão avaliadas:

$$c_{\max}(\pi) = \begin{cases} \max_{\forall(p,q) \in \pi} \{\delta(p,q)\} & \text{se a origem de } \pi \in S \\ +\infty & \text{senão} \end{cases} \quad (2.1)$$

$$c_{\text{sum}}(\pi) = \begin{cases} \sum_{\forall(p,q) \in \pi} \delta(p,q) & \text{se a origem de } \pi \in S \\ +\infty & \text{senão} \end{cases} \quad (2.2)$$

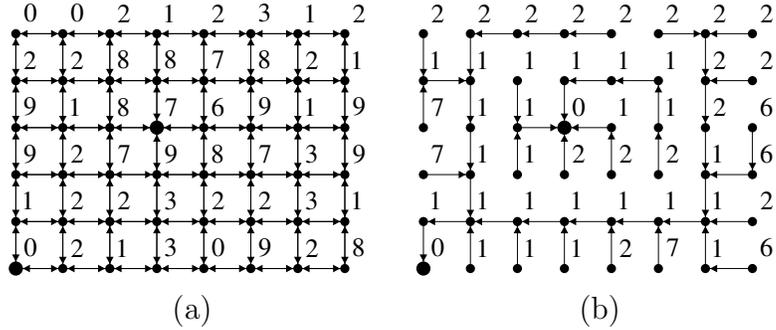


Figura 2.1: (a) Um grafo de uma imagem 2D em tons de cinza com vizinhança 4. Os números correspondem as intensidades $I(p)$ dos pixels e os pontos maiores denotam duas sementes (uma no interior do retângulo claro e uma no fundo mais escuro ao redor dele). O fundo também apresenta alguns pixels claros. (b) Uma floresta de caminhos de custo mínimo usando c_{\max} , com $\delta(p, q) = |I(q) - I(p)|$. Os números são os valores de custo e o retângulo é obtido como uma árvore enraizada na semente interna. As setas em (b) apontam para o predecessor no caminho ótimo.

onde $\delta(p, q)$ mede a dissimilaridade entre nós adjacentes (ver figuras 2.1a e 2.1b).

Para funções de custo apropriadas (*funções suaves* [30]), a IFT atribui um caminho de custo mínimo indo do conjunto de sementes para cada nó do grafo, de tal maneira que a união desses caminhos forma uma floresta orientada, estendendo-se por toda a imagem (ver figuras 2.1a e 2.1b). A IFT produz três atributos para cada pixel/voxel: seu predecessor no caminho ótimo, o custo desse caminho, e o nó raiz correspondente (ou algum rótulo associado a ele). Uma grande variedade de operadores de imagem pode ser implementada através de simples processamento local desses atributos.

Em geral, podem existir muitos caminhos de custo mínimo levando a um dado pixel, somente o custo ótimo para um dado pixel destino é unicamente definido. Observe que, se nós pegarmos um caminho ótimo para cada pixel, a união desses caminhos pode não ser uma floresta. Por construção, o algoritmo da IFT sempre produz uma floresta de espalhamento, embora para certos grafos e funções de custo a otimalidade da floresta pode não ser garantida [30]. A ambigüidade entre as diversas possíveis florestas ótimas é resolvida, por exemplo, atribuindo-se o pixel para o caminho de menor custo que o encontra primeiro durante o crescimento da floresta (ver seção 2.3). Condições suficientes para a existência da otimalidade da IFT são apresentadas na referência [30]. A seguir é apresentada uma instância particular da IFT com *imposição de sementes*, isto é todas sementes se tornarão obrigatoriamente raízes da floresta, e funções de custo de caminho dadas pelas equações 2.1 e 2.2. Para valores não negativos e fixos de $\delta(p, q)$ a otimalidade

da floresta gerada pelo algoritmo é garantida [30].

2.2 Algoritmo da IFT

Um *mapa de predecessores* P é uma função que atribui para cada pixel p da imagem algum outro pixel no domínio da imagem, ou uma marca distinta *nil* indicando ausência de predecessor. Nesse último caso, p é dito ser uma *raiz* do mapa. Uma floresta de espalhamento pode ser representada em memória através de um mapa de predecessores que não contenha ciclos. O algoritmo da IFT retorna um mapa de predecessores P representando a floresta ótima, um mapa de custos C e um mapa de raízes R . Os dois últimos mapas podem ser derivados a partir de P , porém são usados diretamente por questões de eficiência e praticidade. O mapa de raízes R é usado para acessar em tempo constante a raiz em S de cada pixels da floresta e o mapa C armazena para cada pixel o custo do caminho ótimo que o alcança a partir do conjunto de sementes.

O algoritmo da IFT abaixo é essencialmente o procedimento de Dijkstra para o cálculo de caminhos de custo mínimo a partir de uma única fonte [19, 1], ligeiramente modificado para permitir fontes múltiplas e função mais geral de custo (equações 2.1 e 2.2). O algoritmo também apresenta algumas otimizações que não afetam a sua exatidão.

Algoritmo 1 IFT

ENTRADA: Uma imagem, uma relação de adjacência A e um conjunto de nós sementes S .
 SAÍDA: Mapa de custo C , mapa de predecessores P e mapa de raízes R .
 AUXILIARES: Uma fila de prioridades Q inicialmente vazia.

1. **Para Cada** nó p do grafo derivado da imagem, **Faça**
2. $C(p) \leftarrow +\infty, R(p) \leftarrow p, P(p) \leftarrow nil.$
3. **Para Cada** nó $p \in S$ **Faça**
4. $C(p) \leftarrow 0$
5. $Insira\ p\ em\ Q.$
6. **Enquanto** Q não estiver vazia **Faça**
7. $Remova\ p\ de\ Q\ tal\ que\ C(p)$ seja mínimo.
8. **Para Cada** nó $q \in A(p)$ tal que $C(q) > C(p)$ **Faça**
9. $cst \leftarrow OP(C(p), \delta(p, q)).$
10. **Se** $cst < C(q)$ **Então**
11. **Se** $C(q) \neq +\infty$ **Então**
12. $Remova\ q\ de\ Q.$
13. $C(q) \leftarrow cst, R(q) \leftarrow R(p), P(q) \leftarrow p.$
14. $Insira\ q\ em\ Q.$
15. **Retorne** $\{C, P, R\}$

As linhas 1–2 inicializam a floresta como um conjunto de árvores triviais, nós isolados a serem conquistados durante o processo. Os custos são iniciados com $+\infty$ refletindo que nenhum caminho a partir das sementes ainda foi processado. Nas linhas 3–5 os caminhos triviais a partir das sementes são avaliados. Estes caminhos possuem o custo mínimo 0 de forma que todas as sementes se tornarão obrigatoriamente raízes da floresta. As sementes são inseridas na fila de prioridades Q . Os pixels presentes na fila de prioridades representam a fronteira da floresta em crescimento. Eles correspondem a nós da floresta atingidos por caminhos não necessariamente ótimos. A cada iteração do algoritmo (linha 6) um caminho ótimo é selecionado. Este corresponde ao caminho de menor custo entre os que atingem nós de fronteira (linha 7). O extremo do caminho ótimo recém formado deixa então a fronteira (linha 7) e os seus vértices adjacentes são avaliados (linha 8). A fronteira pode ser ampliada pela aquisição de novas conexões, ou melhores rotas podem ser encontradas para pixels de fronteira já existentes. Na linha 9 é calculado o custo cst de uma nova possível rota, onde a função OP pode assumir as seguintes formas:

$$\text{OP}(C(p), \delta(p, q)) = \begin{cases} \max\{C(p), \delta(p, q)\} & \text{para função de custo } c_{\max} \\ C(p) + \delta(p, q) & \text{para função de custo } c_{\text{sum}} \end{cases} \quad (2.3)$$

Este custo cst é comparado com o custo do caminho atual (linha 10). Os mapas C, R e P devem ser atualizados de forma a refletirem o melhor caminho encontrado (linha 13). A condição $C(q) > C(p)$ na linha 8 é uma otimização que assume que o custo ao longo do caminho ótimo é não decrescente.

Portanto, quando temos várias sementes em S , estas serão propagadas ao mesmo tempo e teremos um processo competitivo. Cada semente irá definir uma *zona de influência* composta por pixels conexos a ela por caminhos mais baratos do que os fornecidos por qualquer outra semente em S .

Uma descrição mais completa e genérica da IFT com provas de correteude pode ser encontrada em [30].

2.3 Resolvendo empates

Um dos principais problemas da IFT é estabelecer uma regra de desempate, quando um pixel é alcançado por duas ou mais sementes por caminhos de mesmo custo. Uma vez que os custos dos caminhos assumem normalmente valores discretos, múltiplas soluções são de fato muito comuns na prática. O algoritmo 1 da seção 2.2 já resolve algumas dessas ambigüidades. Quando um pixel é atingido por dois caminhos de mesmo custo, o algoritmo irá atribuí-lo ao caminho que for encontrado primeiro. A condição de estritamente inferior da linha 10 impede que ele seja atribuído a caminhos de mesmo custo posteriormente encontrados. A única ambigüidade restante é a escolha do pixel de custo mínimo em Q ,

no caso de empates. Nesse trabalho será adotada a regra de desempate FIFO (*First-In-First-Out*), ou seja, selecionamos o pixel que primeiro entrou em Q . Outra forma, com vantagens em algumas aplicações, é o desempate LIFO (*Last-In-First-Out*). Por consistência, nesse caso, o algoritmo deve também escolher sempre o último caminho ótimo encontrado no caso de empates.

Com a política FIFO, qualquer conjunto conexo de pixels que pode ser encontrado por duas ou mais raízes através de caminhos ótimos de mesmo custo será particionado entre as respectivas árvores. No caso de política LIFO, esses pixels serão associados a uma mesma árvore [30].

2.4 Fila de prioridade

A implementação mais fácil para a fila Q usa um *heap binário* [1]. Neste caso o algoritmo 1 da seção 2.2 terá complexidade $O(m + n \log n)$, onde n é o número de nós (pixels) e m o número de arestas do grafo. Na maioria das aplicações, porém, podemos usar funções de custo de caminho com incrementos de custo inteiros e limitados a uma constante K ao longo do caminho. Isto permite a utilização da fila circular de Dial com $K + 1$ posições [18]. Cada posição i , $i = 0, 1, \dots, K$, deve armazenar uma lista duplamente ligada de todos os pixels p com custo $C(p)$ tal que $i = C(p) \% K$. Como sabemos o tamanho máximo n do grafo, essas listas podem ser implementadas em uma única matriz X de ponteiros $X.next(p)$ e $X.prev(p)$ com n elementos. Neste caso, o algoritmo da IFT terá complexidade $O(m + nK)$. Se a adjacência definir um grafo esparso $m \ll n^2$, a IFT levará tempo proporcional ao número n de pixels, pois o número de pixels adjacentes pode ser considerado uma constante pequena, sendo desconsiderado na análise assintótica [30]. Note que a cada instante existe um valor mínimo C_{min} e um valor máximo C_{max} de custo para os pixels armazenados em Q . A diferença $C_{max} - C_{min} \leq K$ deve ser mantida para garantir a corretude da fila. Em algumas aplicações sabemos que os incrementos são inteiros e limitados, mas não conhecemos o valor de K . Neste caso, a fila circular inicia com um dado tamanho K , mas antes de inserir um novo pixel devemos verificar a necessidade de realocar ou não mais elementos para a fila.

2.5 Aplicações

A IFT fornece um comum e eficiente framework em grafos para o desenvolvimento de métodos de segmentação de imagens 2D/3D baseados em borda [33, 32, 31], bem como para métodos baseados em região [27, 46, 29, 11, 2, 22]. As aplicações deste algoritmo, na sua forma mais genérica [30], também incluem diversos operadores de filtragem e análise de

imagens: *caminhos geodésicos*, *transformadas de distância* [28], *esqueletos multiescala* [28], *dimensão fractal multiescala* [65], *saliências de formas* [64], *reconstruções morfológicas* e outras operações conexas [29]. Observe que alguns casos não são facilmente relacionados com um problema de partição da imagem (e.g. reconstrução morfológica, perseguição de bordas, pontos de saliência). Porém, na maioria dos casos, a solução é obtida pela simples escolha de parâmetros da IFT seguida de um processamento local dos mapas C, P e R , em tempo proporcional ao número de pixels.

Portanto, a IFT unifica e estende várias técnicas de análise de imagens que, muito embora baseadas em conceitos similares, são normalmente apresentadas como métodos não relacionados [30].

2.6 A IFT diferencial

Métodos clássicos de segmentação por partição em zonas de influência, tais como a *transformada de watershed* [46, 47, 3, 67, 50] e a *conexidade fuzzy relativa* [59, 41], podem ser reduzidos a uma IFT usando função de custo de caminho c_{\max} com competição entre sementes internas e externas. O objeto é definido pela união das zonas de influência de suas sementes internas. Estes métodos são geralmente usados de forma interativa, pois muito freqüentemente o usuário precisa corrigir a segmentação inserindo novas sementes e/ou removendo raízes (árvores), formando uma seqüência de IFTs. No trabalho de mestrado do Felipe Bergo foi proposto como melhoria a DIFT [27]. Ao invés de recalcular uma nova IFT toda vez que o conjunto de sementes é alterado, o algoritmo da DIFT atualiza o resultado da segmentação de forma diferencial, em tempo proporcional ao número de nós nas regiões modificadas da floresta. No caso de segmentação 3D de estruturas cerebrais (e.g. ventrículos, pons-medula, cerebelo), o método reduz o tempo das interações tipicamente da ordem de 30 segundos para margens toleráveis em torno de 3 segundos em computadores pessoais comuns (e.g. Pentium-IV 1.5GHz). No entanto, este método ainda pode ser bastante melhorado, no sentido de reduzir o número de sementes e automatizar o pré-processamento, pois a eficácia dos métodos de segmentação baseados na IFT requer um bom contraste nas bordas dos objetos. Nesta direção, os resultados do presente trabalho constituem importantes avanços, principalmente com relação ao primeiro ponto levantado.

Capítulo 3

Pré-Processamento

Pré-processamento é algumas vezes necessário para realçar as bordas de interesse em detrimento de outras bordas. Para tanto, nós primeiramente realizamos uma classificação atribuindo um valor de pertinência para cada pixel com relação ao objeto baseado em características da imagem.

3.1 Mapas de região

Seja \mathbf{x}_p um vetor de características calculado em um dado pixel p ; μ_p e Σ_p são as matrizes de média e de covariância do vetor de características \mathbf{x}_q calculado em todos pixels q dentro de uma adjacência circular ao redor de p ; e seja T um conjunto de pixels de treinamento, selecionados em regiões do objeto com diferentes características (figura 3.1a). Para cada pixel $s \in T$, um valor de pertinência $R_s(p)$ é calculado para todos pixels p da imagem.

$$R_s(p) = \exp\left(-\frac{1}{2d}(\mathbf{x}_p - \mu_s)^t \Sigma_s^{-1}(\mathbf{x}_p - \mu_s)\right) \quad (3.1)$$

Onde $d > 1$ leva em conta a ausência de informação estatística. Em todas as aplicações nós usamos $d = 10$ fixo, o que foi possível uma vez que o vetor de características usado estava normalizado no intervalo de $[0, 1]$. Nós também usamos um raio de adjacência distinto para cada semente $s \in T$, o tornando o maior possível de forma a obtermos uma melhor estimativa para μ_s e Σ_s . O mapa de região R é obtido como:

$$R(p) = \max_{\forall s \in T} \{R_s(p)\}. \quad (3.2)$$

Finalmente, nós aplicamos um filtro mediana no mapa R para torná-lo mais homogêneo. O mapa de região é uma imagem bidimensional em tons de cinza (figura 3.1b).

3.2 Imagens de gradiente realçadas

Uma imagem das bordas realçadas pode ser obtida através do gradiente do mapa de região (figura 3.1c). Outra idéia é calcular a dissimilaridade $\delta(p, q)$ entre nós adjacentes p e q diretamente a partir do mapa de regiões. Isto gera um gradiente com maior resolução, no nível de arestas.

$$w(p, q) = \exp\left(-\frac{(R(p) - R(q))^2}{2d}\right). \quad (3.3)$$

$$\delta(p, q) = K(1 - w(p, q)) \quad (3.4)$$

Onde $w(p, q)$ é uma medida de similaridade no intervalo de $[0, 1]$ e K é um inteiro representando a máxima dissimilaridade entre pixels, usado para fins de quantização (e.g., $K = 1023$, ver seção 1.2.3).

3.3 Exemplos

A figura 3.2 mostra outros exemplos de mapas de região criados pelos procedimentos da seção 3.1. Os pixels de treinamento e seus raios de adjacência são indicados pelos círculos sobre as imagens originais (figuras 3.2a e 3.2c) Os respectivos mapas de região são mostrados nas figuras 3.2b e 3.2d.

Em todos exemplos, foram usados dois atributos normalizados no intervalo de $[0, 1]$ para o vetor de características da equação 3.1. Os atributos usados foram respectivamente: brilho e magnitude do gradiente (figura 3.1b); e bandas do vermelho e verde (figuras 3.2b e 3.2d).

Note que a escolha desses atributos é um problema à parte, e que a segmentação geralmente não pode ser solucionada por limiarização do mapa de região seguida pela extração de componentes binários, que estão conexos com as sementes internas (e.g., figura 3.1b).

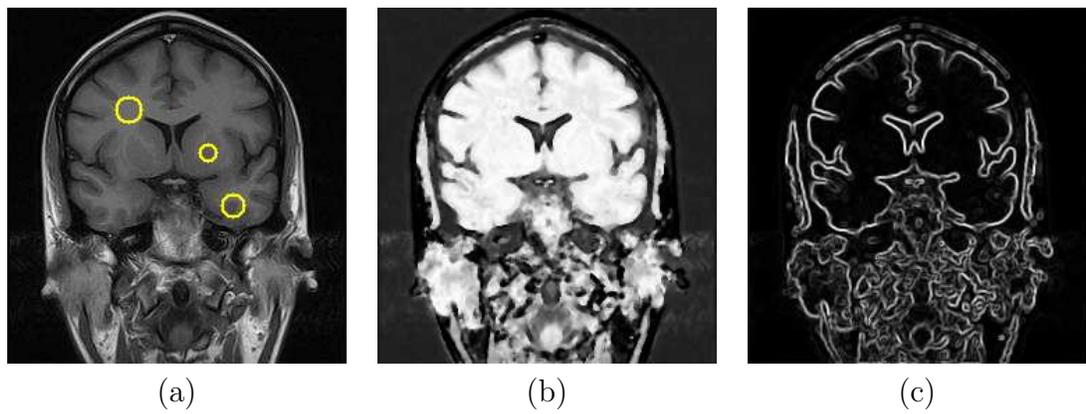


Figura 3.1: (a) Imagem 2D em tons de cinza (seção 1.2.1) de um corte coronal do cérebro com três pixels de treinamento (os círculos indicam os raios de adjacência usados). (b) Mapa de região 2D do cérebro. (c) Gradiente realçado 2D baseado no mapa de região.

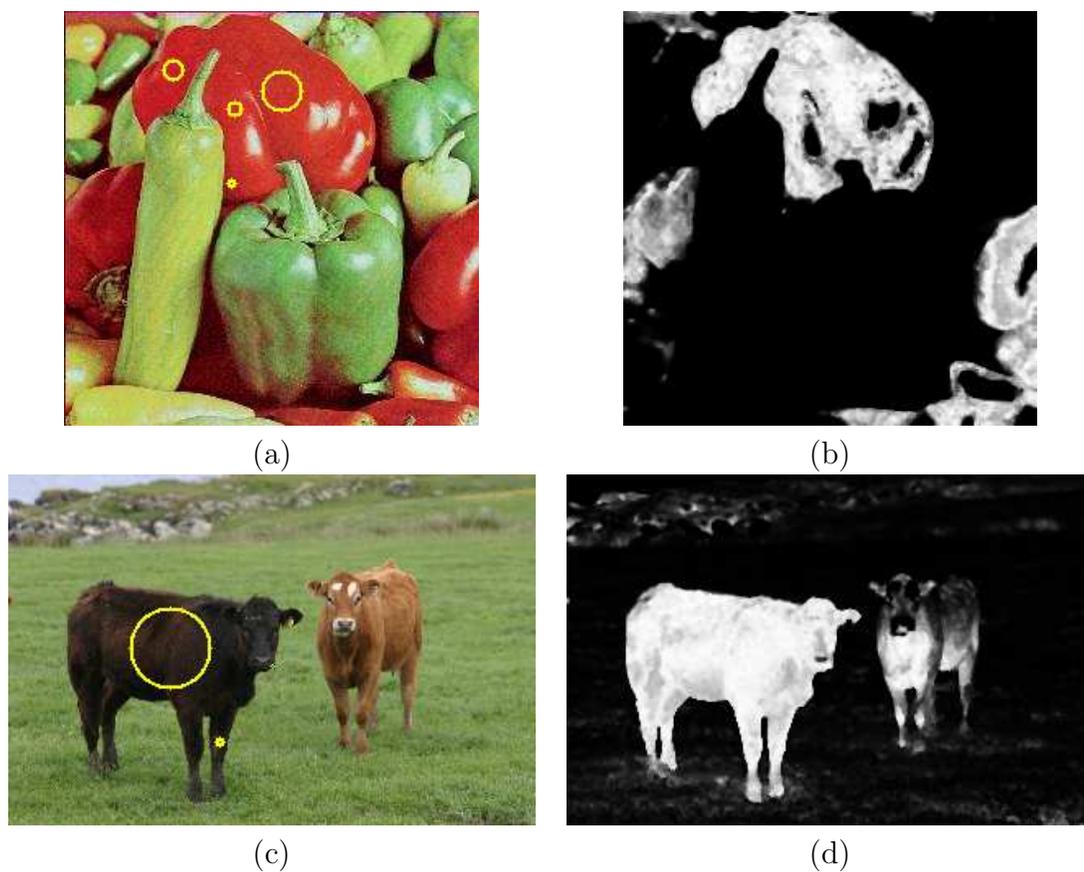


Figura 3.2: (a) Uma imagem colorida (seção 1.2.5) de pimentões com quatro pixels de treinamento (os círculos indicam seus raios de adjacência) e (b) o mapa de região do pimentão vermelho central. (c) Uma imagem colorida de duas vacas com dois pixels de treinamento e (d) o mapa de região da vaca mais escura.

Capítulo 4

Segmentação por competição entre sementes κ -conexas

4.1 Introdução

Chamamos um conjunto de pixels de *componente κ -conexo* com relação a uma dada semente, quando estes são alcançados por caminhos ótimos com custos inferiores ou iguais a um valor κ [25]. No *framework* da IFT, a *força de conectividade* entre dois pixels é inversamente relacionada com o custo do caminho ótimo interligando eles no grafo. Ou seja, quanto menor o custo maior será a força de conectividade entre eles. Na literatura o termo força de conectividade é mais empregado [59, 41]. Logo, um componente κ -conexo com relação a um pixel semente pode ser redefinido como um conjunto de pixels com forças de conectividade em relação a semente maiores ou iguais a um limiar derivado de κ . Nesse sentido, quando uma semente é selecionada no interior de um objeto, sua maior zona de influência possível contida no objeto pode ser descrita por um componente κ -conexo.

Métodos baseados em competição entre sementes juntamente com o uso de limiares nunca foram explorados na literatura. Descobrimos que essa abordagem híbrida revelou ser mais efetiva do que as anteriores em muitas situações (ver figura 4.1) [25, 26]. O método restringe a competição entre sementes em regiões com pixels que são κ_s -conexos a alguma semente s , pixels não alcançados por nenhuma semente são considerados como pertencentes ao fundo e sementes externas são somente necessárias quando a extensão de uma semente não está totalmente contida no objeto. Evidentemente temos agora o problema de determinar os valores de κ_s para cada semente s .

Em [26] foram também apresentados alguns resultados preliminares que envolviam componentes κ_s -conexos sem competição, além de uma solução interativa para a estimativa de κ_s . No entanto, essa solução interativa não é extensível para imagens multidimensionais. Nesse capítulo serão apresentados somente os resultados mais relevantes dados

pela competição entre sementes κ_s -conexas.

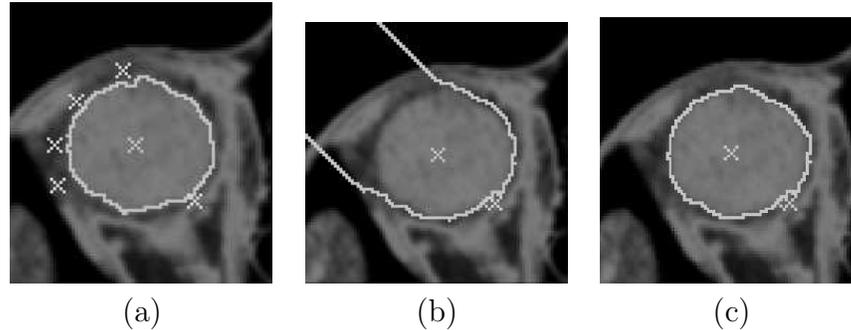


Figura 4.1: Segmentação do globo ocular por competição entre sementes em imagem de tomografia de raios-X. (a) Uma semente interna e várias sementes externas são necessárias para a segmentação, usando c_{\max} com δ_4 (equação 4.4). (b) A segmentação falha quando algumas sementes externas são removidas. (c) Um limiar κ limita a zona de influência da semente interna onde a competição entre sementes falha. Observe a redução do número de sementes em relação a (a).

4.2 Métodos relacionados

Conexidade fuzzy e transformada de watershed são métodos de segmentação de imagens baseados em pixels sementes, ainda estão em andamento, e foram usados com sucesso em várias aplicações [44, 51, 53, 39]. Essas abordagens tem sido usadas sob dois diferentes paradigmas, com e sem competição entre sementes e podem ser implementadas a partir da IFT com função de custo de caminho c_{\max} [46, 47, 54, 17]. Em detecção de objetos com competição entre sementes [59, 41, 67, 3], as sementes são especificadas dentro e fora do objeto, cada semente define uma zona de influência composta por pixels mais fortemente conexos com essa semente do que com qualquer outra (floresta de caminhos de custo mínimo), e o objeto é definido pela união das zonas de influência de suas sementes internas (árvores ótimas enraizadas em suas sementes internas). Na definição de objetos sem competição de sementes, uma semente é especificada dentro do objeto e a força de conexidade de cada pixel em relação a semente é calculada, tal que o objeto é obtido por limiarização da imagem de conexidade resultante. Em [66], o objeto é definido sem competição, como a união de todos componentes κ -conexos (árvores de caminhos de custo mínimo) criados a partir de cada semente interna separadamente (o que requer uma execução da IFT para cada semente). O algoritmo descrito em [66] assume que o objeto pode ser definido por um único valor de κ para todas sementes em S . A figura 4.2a ilustra um caso onde essa técnica funciona. No entanto, uma simples mudança de posição de

uma das sementes pode levar a falhas de segmentação (figura 4.2b), porque a zona de influência de cada semente no interior do objeto é limitada por um valor distinto de custo κ (figura 4.2c). Claramente, o apelo inicial em favor da competição entre sementes é a possibilidade de detectar múltiplos objetos através de uma única IFT e sem a dependência de limiares.

A técnica aqui apresentada combina as anteriores, sendo mais geral. Além disso, ela permite o uso de diferentes valores de κ para cada semente. O tratamento personalizado que se dá a cada semente permite a seleção da sua maior zona de influência possível contida no objeto. Esse melhor aproveitamento que temos das sementes leva a uma redução do número total de sementes necessárias para se completar uma segmentação [26].

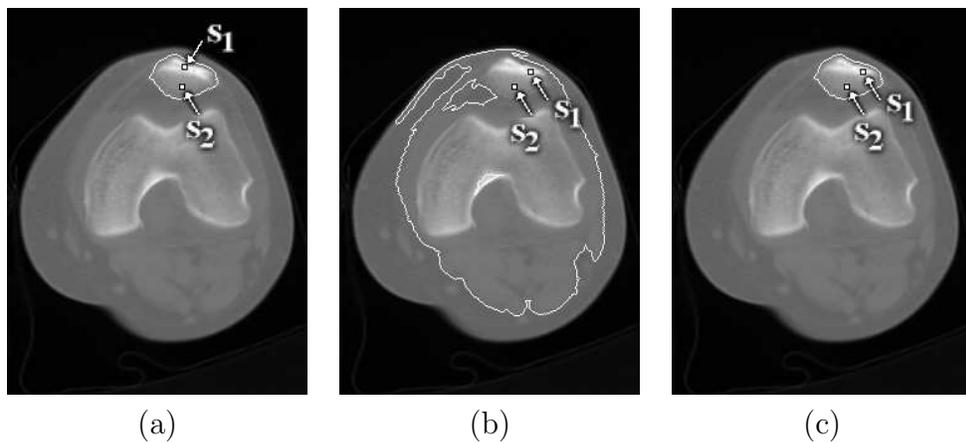


Figura 4.2: Uma imagem TC de um joelho onde a patela pode ser segmentada pela união das zonas de influência (sem competição) de dois pixels sementes, s_1 e s_2 , usando c_{\max} com δ_3 (equação 4.3). (a) O resultado com um único valor de κ para ambas sementes. (b) A segmentação com um único valor de κ falha quando nós alteramos a localização de s_1 , porque s_1 agora requer um valor de κ mais elevado para obter a parte mais clara do osso e s_2 invade o fundo para esse valor de κ maior. (c) O resultado pode ser corrigido com o uso de valores distintos de κ para cada semente.

4.3 Estimativa automática de κ_s

Nós podemos especificar um κ_s fixo para uma aplicação particular, mas o resultado seria suscetível as variações de intensidade. Uma outra alternativa é buscar por possíveis casamentos entre a forma do objeto desejado e as formas dos componentes κ -conexos para os diversos valores de κ . Uma desvantagem dessa abordagem seria a baixa velocidade da

segmentação. Uma situação mais complexa ocorre quando a definição do objeto requer mais de uma semente. Cada semente possui sua própria extensão máxima dentro do objeto, sendo necessário avaliar a forma composta pela união de suas zonas de influência. A abordagem apresentada aqui é muito mais simples e ainda efetiva.

A estimativa automática de κ_s vem da observação de que quando as arestas possuem valores atribuídos baixos dentro e fora do objeto e valores altos nas bordas, as *frentes de custo* (conjunto de pixels atingidos por caminhos ótimos de mesmo custo) da IFT usando c_{\max} normalmente apresentam um grande aumento no número de pixels quando elas cruzam as bordas dos objetos (figuras 4.3a e 4.3b). Isto é, muitos pixels do fundo são alcançados por caminhos ótimos cujo custo é o valor de alguma aresta que cruza a borda. A idéia é limitar o máximo tamanho das frentes de custo. Em outras palavras, o processo de crescimento de região de uma dada semente deve parar quando sua frente de onda de mesmo custo for maior que um dado limiar. Seleccionamos então apenas as frentes de custo anteriores que possuem tamanhos inferiores ao limiar. Assim, nós reduzimos o problema de achar um κ_s para cada semente por um único limiar de área aplicado as frentes de custo [25].

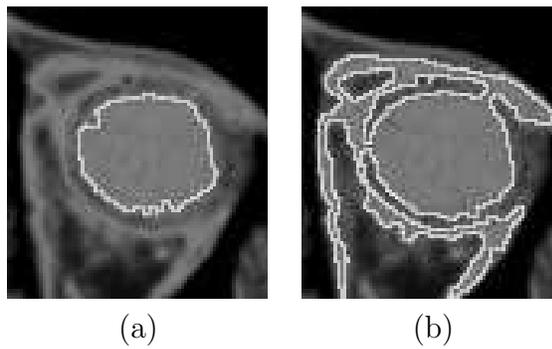


Figura 4.3: Uma imagem de tomografia de raios-X do globo ocular com uma semente no seu interior. (a) A frente de onda de custo κ que proporciona a máxima extensão contida no objeto. (b) A frente de onda de custo $\kappa + 1$ apresenta um grande aumento de área quando esta invade o fundo.

4.4 Algoritmo

A IFT calcula três atributos para cada pixel $p \in D_I$ (seção 2): seu predecessor $P(p)$ no caminho ótimo, o custo $C(p)$ desse caminho, e a raiz correspondente $R(p)$. No algoritmo apresentado nessa seção, não será necessário criar o mapa de predecessores P . A IFT propaga frentes de onda W_{cst} de mesmo custo cst ao redor de cada semente, seguindo a

ordem crescente de custos $cst = 0, 1, \dots, K$. Esse processo é explorado para calcular os valores κ_s de cada semente $s \in S$ automaticamente.

Em primeiro lugar, considere as frentes de onda ao redor de uma semente s selecionada no interior de um dado objeto. Todos pixels p na frente de onda W_{cst} ao redor de s possuem custos ótimos $C(p) = cst$, $0 \leq cst \leq K$. Se o objeto for um único componente κ -conexo em relação a s , então existe um limiar κ_s , $0 \leq \kappa_s \leq K$, tal que o objeto pode ser definido pela união de todas as frentes de onda W_{cst} , para $cst = 0, 1, \dots, \kappa_s$.

A eficácia da segmentação usando c_{\max} depende da atribuição de pesos baixos $\delta(p, q)$ para arestas dentro (e fora) dos objetos e pesos elevados para arestas sobre suas bordas. Como consequência, as frentes de onda normalmente apresentam um considerável aumento no *número de pixels* quando elas cruzam as bordas do objeto (figuras 4.3a e 4.3b). Isso é, muitos pixels do fundos são alcançados por caminhos ótimos com custos iguais ao valor $\delta(p, q)$ de alguma aresta (p, q) que atravessa a borda.

Nós trocamos a escolha de um valor κ_s para cada semente $s \in S$ por um único limiar T (e.g. uma porcentagem do número total de pixels dividida pelo número de sementes internas), que impõe um limite para o tamanho máximo das frentes de onda. O crescimento de região de uma semente s deve parar quando o tamanho de sua frente de onda de custo cst for maior que T , e o valor de κ_s é determinado como $\max\{cst - 1, 0\}$. O algoritmo apresentado abaixo calcula κ_s para definição de objetos múltiplos com competição entre sementes.

Algoritmo 2 COMPETIÇÃO ENTRE SEMENTES COM DETECÇÃO AUTOMÁTICA DE κ_s

ENTRADA: Imagem $\hat{I} = (D_I, \vec{I})$, relação de adjacência A , limiar de tamanho T , e uma imagem de rótulos $\hat{L} = (D_I, L)$, inicialmente com $L(p) = i$ para sementes no interior do objeto $0 < i \leq k$, $L(p) = 0$ para sementes de fundo e $L(p) = -1$ caso contrário.

SAÍDA: Uma imagem rotulada $\hat{L} = (D_I, L)$, onde $L(p) = i$, $0 \leq i \leq k$.

AUXILIARES: Uma fila de prioridades Q e mapas C , R , κ , $size$, e cst definidos em D_I para armazenar o custo e a raiz de cada pixel e limiar, tamanho da frente de onda e custo de frente de onda para cada semente, respectivamente.

1. **Para Cada** *pixel* $p \in D_I$, **Faça**
2. $R(p) \leftarrow p$, $size(p) \leftarrow 0$, $cst(p) \leftarrow 0$, $\kappa(p) \leftarrow +\infty$.
3. **Se** $L(p) = -1$, **Então** $C(p) \leftarrow +\infty$, $L(p) \leftarrow 0$.
4. **Senão**, $C(p) \leftarrow 0$ e *insira* p em Q .
5. **Enquanto** $Q \neq \emptyset$, **Faça**
6. *Remova um pixel* p de Q *tal que* $C(p)$ *seja mínimo*.
7. **Se** $\kappa(R(p)) = +\infty$ e $L(R(p)) \neq 0$, **Então**
8. **Se** $C(p) \neq cst(R(p))$, **Então**
9. $size(R(p)) \leftarrow 1$, $cst(R(p)) \leftarrow C(p)$.

```

10.   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
11.   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
12.   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
13.   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
14.   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
15.   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
16.   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
17.   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
18.   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
19.   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
20.   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
21.   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
22.   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |

```

10. | | | | | | | | | | | | | | | | | | | | | | |

11. | | | | | | | | | | | | | | | | | | | | | | |

12. | | | | | | | | | | | | | | | | | | | | | | |

13. | | | | | | | | | | | | | | | | | | | | | | |

14. | | | | | | | | | | | | | | | | | | | | | | |

15. | | | | | | | | | | | | | | | | | | | | | | |

16. | | | | | | | | | | | | | | | | | | | | | | |

17. | | | | | | | | | | | | | | | | | | | | | | |

18. | | | | | | | | | | | | | | | | | | | | | | |

19. | | | | | | | | | | | | | | | | | | | | | | |

20. | | | | | | | | | | | | | | | | | | | | | | |

21. | | | | | | | | | | | | | | | | | | | | | | |

22. | | | | | | | | | | | | | | | | | | | | | | |

O mapa de raízes é usado para acessar em tempo constante a raiz em S de cada pixels da floresta. A zona de influência de uma semente $s \in S$ é limitada por competição quando se depara com a zona de influência de outra semente ou quando o valor κ_s de s for encontrado.

Uma vantagem desse algoritmo em relação a métodos clássicos de segmentação baseados em competição entre sementes ocorre quando o objeto contém várias partes de fundo (buracos) no seu interior. Nesse caso, o uso de κ_s normalmente elimina a necessidade de pelo menos uma semente de fundo em cada buraco. Por outro lado, algumas pequenas regiões ruidosas do objeto podem não ser conquistadas pelas sementes internas devido ao uso do κ_s . A imagem rotulada pode ser pós-processada, tal que buracos com área abaixo de um limiar são fechadas [49]. O operador de fechamento de área tem mostrado ser um complemento muito efetivo para o presente algoritmo. Em muitas situações, o objeto não tem buracos e um limiar de área elevado pode ser usado de forma a reduzir o número de sementes internas. O algoritmo é comparado com abordagens de segmentação tradicionais baseadas em competição entre sementes nas próximas seções.

4.5 Funções de dissimilaridade avaliadas

Durante a validação foram usadas uma relação de adjacência A de vizinhança 8, função de custo de caminho c_{\max} e foram avaliados pesos de arestas $\delta(p, q)$ dados pelas equações a seguir. Essas funções de dissimilaridade são apresentadas para imagens em tons de cinza (seção 1.2.1) mas podem ser estendidas para imagens multibanda (seção 1.2.5).

$$\delta_1(p, q) = K(1 - \text{affinity}(I(p), I(q))); \quad (4.1)$$

$$\delta_2(p, q) = G(q); \quad (4.2)$$

$$\delta_3(p, q) = K \left(1 - \text{affinity} \left(\frac{I(p) + I(q)}{2}, I(R(p)) \right) \right); \quad (4.3)$$

$$\delta_4(p, q) = \min_{s \in S} \left\{ K \left(1 - \text{affinity} \left(\frac{I(p) + I(q)}{2}, I(s) \right) \right) \right\}; \quad (4.4)$$

$$\delta_5(p, q) = a\delta_1(p, q) + b\delta_3(p, q); \quad (4.5)$$

onde K é um inteiro positivo usado para fins de quantização, $I(p)$ é o brilho do pixel p , $G(q)$ é uma magnitude de gradiente sobel calculada em q , $R(p)$ é a origem do caminho até o pixel p (mapa de raízes), a e b são constantes tais que $a + b = 1$. A função de afinidade usada deve produzir um número real no intervalo $[0,1]$, onde 1 representa a máxima afinidade. Uma forma geralmente empregada na literatura segue abaixo:

$$\text{affinity}(u, v) = \exp \frac{-(u-v)^2}{2\sigma^2} \quad (4.6)$$

onde σ é uma variação de intensidade permitida.

As funções δ_1 e δ_2 assumem alta homogeneidade interna nos objetos. Elas representam magnitudes de gradiente com diferentes resoluções de imagem e produzem resultados ótimos pela IFT quando usadas nas equações 2.1 e 2.2. A IFT com c_{\max} e esses custos de arestas se torna a transformada de watershed [46]. A função δ_3 explora a dissimilaridade entre objeto e a intensidade dos pixels, sendo o objeto representado pelas suas sementes internas, mas nesse caso c_{\max} não garante a otimalidade da IFT quando usada com múltiplas sementes [30] (a IFT gera uma floresta de espalhamento que pode não ser ótima). Esse problema foi a principal motivação para δ_4 [59]. No entanto, algumas vezes δ_3 proporciona melhores resultados que δ_4 (veja seção 4.6). A função δ_3 pode também limitar as zonas de influência das sementes, quando as intensidades dentro do objeto variam linearmente em direção ao fundo. A função δ_5 reduz esse problema, e δ_3 pode ser trocado por δ_4 na equação 4.5 de forma a garantir a otimalidade da floresta. Outras idéias interessantes para funções de dissimilaridade são apresentadas em [66, 41, 58, 59].

4.6 Resultados experimentais

Nos experimentos foram usadas 100 imagens em tons de cinza 2D de ressonância magnética e tomografia de raios-X extraídas de 7 diferentes objetos (veja tabela 4.1 e figura 4.4). As imagens originais foram pré-processadas para aumentar a similaridade entre pixels internos e o contraste entre objeto e fundo. Quatro usuários distintos executaram as segmentações 2D nas 100 imagens para dois diferentes métodos:

- M1. Detecção de objetos com competição entre sementes e cálculo de κ_s automático (algoritmo 2);

Objeto	Descrição	Modalidade	# de fatias
O1	globo ocular esquerdo	TC-órbita	15
O2	núcleo caudado esquerdo	RM-cérebro	15
O3	ventrículos laterais	RM-cérebro	15
O4	corpo caloso	RM-cérebro	10
O5	patela	TC-joelho	15
O6	fêmur	TC-joelho	15
O7	substância branca	RM-cérebro	15

Tabela 4.1: Descrição, modalidade de imagem e número de fatias para cada objeto usado nos experimentos.

M2. Detecção de objetos com competição entre sementes e sem cálculo de κ_s (algoritmo 1 com função de custo c_{\max}).

Os experimentos visam comparar esses métodos com relação ao número de interações de usuário necessárias para completar a segmentação.

A fim de mostrar a robustez da nova abordagem M1, nós selecionamos as melhores funções de dissimilaridade para cada situação e fixamos os parâmetros de segmentação. Nós usamos uma adjacência de vizinhança 8. O valor do limiar T foi de 1% do tamanho da imagem para todos objetos, exceto para O2 onde usamos $T = 0.2\%$. Visto que os objetos de O1 à O6 não apresentam buracos, nós usamos um valor para o fechamento de área arbitrariamente elevado (e.g. 500 pixels). A única exceção foi O7, para o qual o limiar de área não pode ser maior que 3 pixels devido aos seus buracos. Tabela 4.2 mostra a função de dissimilaridade mais apropriada encontrada para cada par de objeto e método. Na função δ_2 , nós usamos a magnitude do gradiente de Sobel. Note que nós usamos δ_3 em algumas situações, apesar da otimalidade da floresta não ser garantida.

Em análise de imagens médicas, é comum usarmos como gabarito o resultado de uma segmentação manual feita por um especialista. Essa metodologia é questionável, porque os especialistas normalmente cometem equívocos quando eles fazem o delineamento do mesmo objeto duas vezes. Na maioria dos casos, os resultados parecem os mesmos, porém existem pequenas diferenças ao longo do contorno do objeto. Essas pequenas diferenças, no entanto, parecem ser aceitáveis em muitas aplicações. Usamos a medida de similaridade definida abaixo.

Cada objeto foi representado por um conjunto de l fatias binárias $\hat{L}_i = (D_I, L_i)$, $i = 1, 2, \dots, l$, onde $L_i(p) = 1$ para pixels de objeto e 0 caso contrário. Sejam \hat{L}_i e \hat{L}'_i as imagens binárias resultantes das segmentações de uma fatia de um mesmo objeto usando

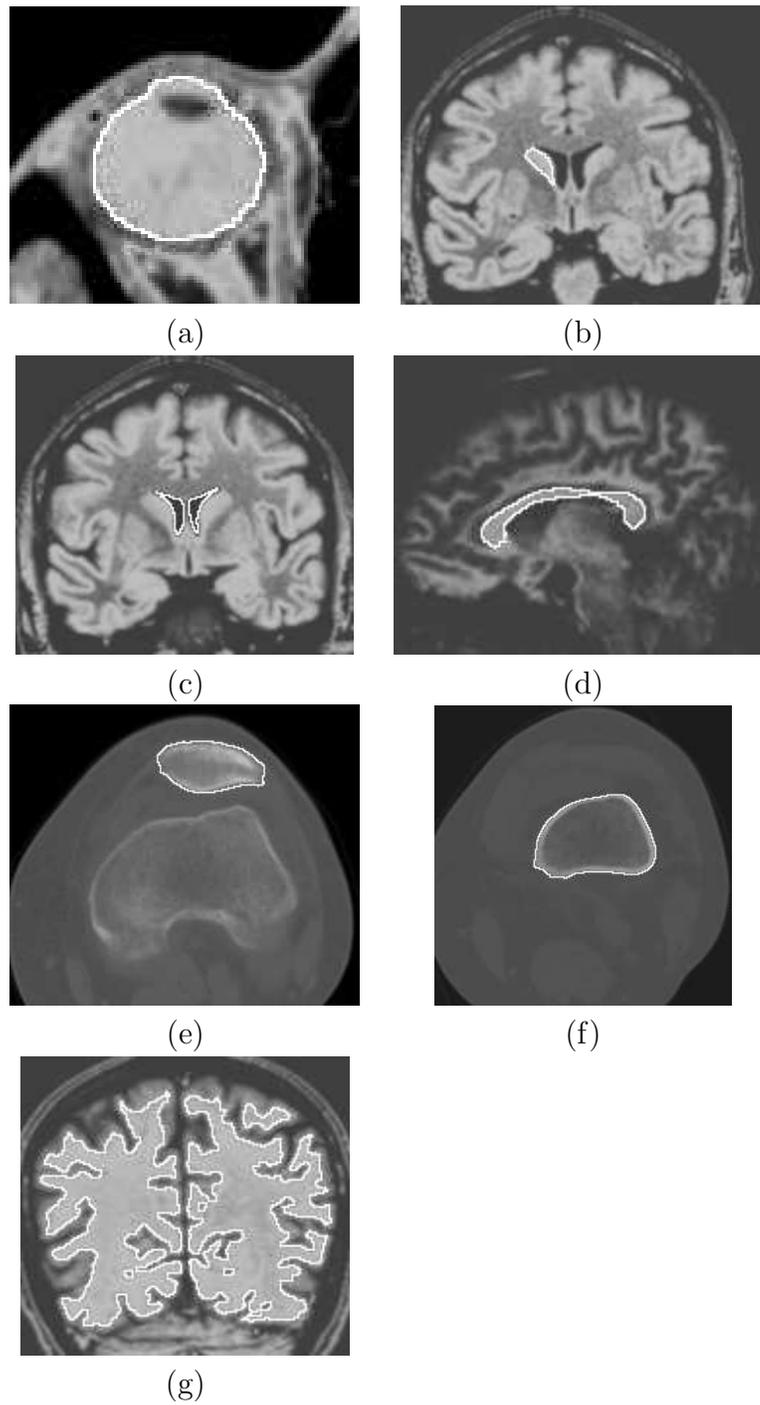


Figura 4.4: (a)-(g) Resultados das segmentações de fatias dos objetos de 1 à 7, respectivamente, sobrepostos às imagens pré-processadas.

Objeto	M1	M2
O1	δ_2	δ_2
O2	δ_4	δ_2
O3	δ_3	δ_3
O4	δ_4	δ_2
O5	δ_4	δ_4
O6	δ_3	δ_2
O7	δ_3	δ_3

Tabela 4.2: As funções de dissimilaridade usadas para cada combinação entre objeto e método.

diferentes métodos. A similaridade entre esses resultados foi medida por:

$$1 - \frac{\sum_{i=1}^{i=l} \sum_{\forall p \in D_I} L_i(p) \oplus L'_i(p)}{\sum_{i=1}^{i=l} \sum_{\forall p \in D_I} L_i(p) + \sum_{i=1}^{i=l} \sum_{\forall p \in D_I} L'_i(p)} \quad (4.7)$$

onde \oplus denota a operação de “or exclusivo”.

No caso de segmentação manual por um especialista, foi mostrado que os valores de similaridade estão em torno de 0.96 [33]. Uma vez que nenhum dos usuários envolvidos nos experimentos é um perito, nós exigimos deles resultados com no mínimo 0.90 de similaridade entre segmentações de um mesmo objeto, usando diferentes métodos.

Método M2 representa as abordagens clássicas baseadas na conexidade *fuzzy* relativa e transformada de watershed [59, 46]. O número de interações de usuário em ambos métodos corresponde ao número total de sementes selecionadas no interior (NIS - *Number of Internal Seeds*) e no exterior (NES - *Number of External Seeds*) do objeto. Método M1 é a variante proposta. O número de sementes é esperado ser muito menor em M1 do que em M2, devido a detecção automática de κ_s .

A tabela 4.3 mostra o número médio de interações e valores de similaridade entre todos usuários, para ambos métodos. Em média, M2 precisou de 2.8 vezes mais interações de usuário do que M1.

A tabela 4.4 mostra em detalhes os valores médios de NIS, NES, bem como o número de detecções automáticas de κ_s (AKD - *Automatic κ_s Detections*) para cada objeto e método. Note que o número de detecções automáticas de κ_s variou de 59% até 100% do NIS (88% em média). Isto demonstra a eficácia da abordagem proposta para a detecção automática de κ_s e explica a redução do número de interações e sementes externas de M1 em relação a M2. Isso é um importante resultado para automação futura, já que a competição entre sementes M2 é sensível à localização das sementes externas devido a heterogeneidade do fundo.

	M1	M2	M1,M2
O1	29.5	77.6	0.962
O2	29.3	38.8	0.915
O3	31.3	61.3	0.935
O4	27.5	46.8	0.918
O5	15.0	61.0	0.946
O6	26.3	37.8	0.981
O7	46.3	284.8	0.930

Tabela 4.3: O número médio de interações por usuário para cada objeto e método, e a média dos valores de similaridade entre ambos métodos para um mesmo objeto.

	M1			M2	
	NIS	NES	AKD	NIS	NES
O1	18.0	11.5	13.0	26.8	50.8
O2	25.3	4.0	24.3	18.8	20.0
O3	30.3	1.0	30.3	30.3	31.0
O4	22.3	5.2	19.8	22.3	24.5
O5	15.0	0.0	15.0	44.0	17.0
O6	26.3	0.0	15.5	22.8	15.0
O7	46.0	0.3	46.0	66.0	218.8

Tabela 4.4: Número médio de sementes internas (NIS), sementes externas (NES), e detecções automáticas de κ_s (AKD).

Capítulo 5

Segmentação por crescimento ótimo de regiões e corte em grafo

5.1 Introdução

Segmentação de imagens usando corte em grafos se tornou muito popular nos últimos anos. Esses métodos são normalmente caros computacionalmente, mesmo com o uso de nós terminais (fonte/destino) e restrições fortes (pixels sementes) (e.g. $O(mn^2)$ onde m é o número de arestas e n o número de nós do grafo) [72, 6]. Nós usamos a IFT no grafo derivado da imagem de forma a reduzir o espaço de busca por bordas de corte para o número de pixels n da imagem de forma a encontrar uma solução em $O(n)$. Nessa solução utilizamos somente sementes internas. A IFT calcula um crescimento de região a partir do conjunto de sementes onde a ordem de propagação de cada pixel é proporcional ao custo do caminho ótimo que o conecta ao conjunto de sementes. Nesse processo, cada pixel define uma região que inclui ele e todos os pixels com ordem de propagação inferior. A borda de cada região é uma possível borda de corte, cuja medida de avaliação de corte é também calculada e atribuída para o pixel correspondente durante o processo. O objeto é obtido através da seleção do pixel de corte de valor mínimo e todos os pixels em sua respectiva borda de corte. O método pode ser combinado com as diversas medidas de avaliação de corte presentes na literatura.

5.2 Medidas de avaliação de corte em grafo

A estratégia de segmentação por corte em grafos envolve dois desafios: (i) atribuir pesos de aresta mais altos no interior dos segmentos e mais baixos caso contrário, e (ii) definir alguma função objetivo baseada nesses pesos, tal que o seu mínimo corresponda a seg-

mentação desejada. A segmentação é um particionamento ótimo do grafo, onde a *borda de corte* é o conjunto de arestas que conectam vértices de segmentos distintos.

Wu e Leahy [71] foram os primeiros a introduzir essas abordagens usando como função objetivo a soma dos pesos das arestas na borda de corte. A função objetivo deles no entanto possui uma inclinação por pequenas bordas e outras funções objetivo, tais como *average cut* [16], *mean cut* [70], *average association* [61], *normalized cut* [63], *ratio cut* [69], e *funções de energia* [72, 7, 6, 43] tem sido propostas para contornar esse problema.

O problema de se encontrar o mínimo de uma função objetivo através de corte em grafo é NP-difícil para um grafo genérico e muito freqüentemente soluções requerem restrições fortes. Soluções heurísticas tem sido propostas em tempo polinomial [35], mas com baixo desempenho computacional, e os resultados estão algumas vezes longe das segmentações desejadas [9]. Em detecção de objetos [72, 7, 6], nós terminais (*fonte* e *destino*) são adicionados ao grafo da imagem, representando objeto e fundo respectivamente. Os pesos das arestas entre cada pixel e um nó terminal (fonte/destino) são penalidades de se atribuir tal pixel para o objeto/fundo. Um algoritmo de corte-mínimo/fluxo-máximo entre fonte e destino [34, 40] calcula uma borda de corte mínimo que representa a fronteira do objeto. Se o método falha, o usuário pode impor restrições fortes para corrigir a segmentação através da inserção de pixels sementes que obrigatoriamente devem fazer parte do objeto/fundo [72]. No entanto, o tempo de execução do algoritmo de corte-mínimo/fluxo-máximo continua elevado demais para aplicações práticas [6] (e.g., tipicamente $O(mn^2)$ onde m é o número de arestas e n é o número de vértices).

As medidas de avaliação de corte em grafo normalmente assumem pesos altos para arestas no interior e no exterior dos objetos e pesos baixos nas bordas. Podemos usar $w(p, q)$ como definido pela equação 3.3 (capítulo 3).

Devido a heterogeneidade do fundo, mesmo com o pré-processamento é muito difícil obter pesos de arestas $w(p, q)$ altos fora do objeto. Isto afeta algumas medidas de corte em grafo mais do que outras. Portanto, nós vamos considerar o corte normalizado [63], o *mean cut* [70] e uma função de energia [72, 43] para avaliar esse aspecto na seção 5.4.

Seja I e E o interior e exterior de uma borda de corte IE , composta por arestas (p, q) tais que $p \in I$ e $q \in E$. O corte normalizado é definido como

$$\frac{cut(I, E)}{asso(I) + cut(I, E)} + \frac{cut(I, E)}{asso(E) + cut(I, E)} \quad (5.1)$$

onde

$$cut(I, E) = \sum_{\forall (p,q) | p \in I, q \in E} w(p, q) \quad (5.2)$$

$$asso(I) = \sum_{\forall (p,q) | p \in I, q \in I} w(p, q) \quad (5.3)$$

$$asso(E) = \sum_{\forall(p,q) | p \in E, q \in E} w(p, q). \quad (5.4)$$

O *mean cut* é definido como

$$\frac{cut(I, E)}{|IE|} \quad (5.5)$$

onde $|IE|$ é o número de arestas em IE .

Nós escolhemos uma função de energia similar aquela proposta em [72] e consistente com a formulação geral apresentada em [43].

$$\lambda \left(\sum_{\forall p \in I} (1 - Ro(p)) + \sum_{\forall q \in E} (1 - Rb(q)) \right) + \sum_{\forall(p,q) | p \in I, q \in E} w(p, q) \quad (5.6)$$

onde Ro e Rb são mapas de região calculados pela equação 3.2 usando pixels de treinamento dentro e fora do objeto respectivamente; e $\lambda > 0$ representa a importância do primeiro termo em relação ao segundo (e.g., nós usamos $\lambda = 40$).

5.3 Algoritmo

A IFT nos fornece um crescimento de região ordenado a partir de S para todos pixels p , de modo que o custo $C(p)$ de um caminho ótimo indo de S até p é função não decrescente da ordem de propagação $O_d(p)$. Quando p é removido de Q , p e todos pixels com ordem de propagação inferior definem uma região I . Nesse momento, o algoritmo já encontrou os caminhos ótimos de S para todos pixels em I , os pixels remanescentes definem uma região E , e a medida de corte $M(p)$ (*normalized cut*, *mean cut*, ou função de energia) para IE pode ser calculada *on-the-fly*. O objeto é obtido através da seleção de um pixel m com valor de corte mínimo, seguida pela limiarização do mapa O_d para valores menores ou iguais a $O_d(m)$. O algoritmo abaixo mostra uma instância do método para c_{sum} e corte normalizado.

Algoritmo 3 CÁLCULO DO MAPA DA ORDEM DE PROPAGAÇÃO O_d E MAPA DO CORTE NORMALIZADO M

ENTRADA: Uma imagem, uma relação de adjacência A e um conjunto de sementes S .
 SAÍDA: Mapas O_d e M .
 AUXILIARES: Uma fila de prioridades Q e variáveis o , ai , ie , e ae que armazenam a ordem e os valores das equações 5.2- 5.4 para o corte IE .

1. $o \leftarrow 1$, $ai \leftarrow 0$, $ie \leftarrow 0$, $ae \leftarrow 0$.
2. **Para Cada** nó p do grafo derivado da imagem, **Faça**

```

3.   |  $C(p) \leftarrow +\infty, O_d(p) \leftarrow +\infty.$ 
4.   | Para Cada nó  $q \in A(p)$  Faça
5.   |   |  $ae \leftarrow ae + w(p, q)/2.$ 
6.   | Para Cada nó  $p \in S$  Faça
7.   |   |  $C(p) \leftarrow 0$ 
8.   |   | Insira  $p$  em  $Q.$ 
9.   | Enquanto  $Q$  não estiver vazia Faça
10.  |   | Remova  $p$  de  $Q$  tal que  $C(p)$  seja mínimo.
11.  |   | Para Cada nó  $q \in A(p)$  Faça
12.  |   |   | Se  $O_d(q) < O_d(p)$ , Então
13.  |   |   |   |  $ie \leftarrow ie - w(p, q), ai \leftarrow ai + w(p, q).$ 
14.  |   |   |   | Senão
15.  |   |   |   |   |  $ie \leftarrow ie + w(p, q), ae \leftarrow ae - w(p, q).$ 
16.  |   |   |   |   | Compute  $cst \leftarrow C(p) + \delta(p, q).$ 
17.  |   |   |   |   | Se  $cst < C(q)$  Então
18.  |   |   |   |   |   | Se  $C(q) \neq +\infty$  Então
19.  |   |   |   |   |   |   | Remova  $q$  de  $Q.$ 
20.  |   |   |   |   |   |   |  $C(q) \leftarrow cst.$ 
21.  |   |   |   |   |   |   | Insira  $q$  em  $Q.$ 
22.  |   |   |  $O_d(p) \leftarrow o, o \leftarrow o + 1.$ 
23.  |   |  $M(p) \leftarrow \frac{ie}{ie+ai} + \frac{ie}{ie+ae}.$ 
24. Retorne  $\{O_d, M\}$ 

```

Note que, é fácil modificar o algoritmo acima para qualquer combinação entre $\{c_{\max}, c_{\text{sum}}\}$ e $\{\text{corte normalizado, mean cut e função de energia}\}$. Essa observação é também válida para muitas outras medidas de avaliação de corte em grafo (e.g., [16, 61, 43]).

Linhas 1–8 inicializam os mapas, variáveis e inserem pixels sementes na fila de prioridades Q . A divisão por 2 na linha 5 leva em conta o fato de que o grafo é não direcionado (e.g., $w(p, q) = w(q, p)$ deve ser considerado somente uma vez). Desta maneira, a variável ae é inicializada com a soma de todos os pesos das arestas do grafo. Linhas 9–23 calculam os mapas M e O_d durante a execução da IFT. Quando p é removido de Q (linha 10), este deixa E e passa a pertencer para I . Nesse momento, todas arestas incidentes a p devem ser reavaliadas. A condição declarada na linha 12 indica que $q \in I$, logo a aresta (p, q) está sendo removida de IE e o seu peso deve ser considerado para atualizar ie e ai . De outra maneira se $q \in E$, então a aresta (p, q) está sendo inserida em IE e o seu peso deve ser usado para atualizar ie e ae . Linhas 16–21 avaliam se o caminho que alcança q através de p é melhor que o caminho corrente com término em q e atualiza Q e $C(q)$ em caso de mudança. Finalmente, as linhas 22–23 calculam a ordem de propagação de p e a medida de corte correspondente ao corte IE .

Nós podemos observar pelas figuras 3.1b e 3.2b que a borda de corte pode conter múltiplos contornos devido aos “buracos” (regiões escuras) no interior do mapa de região. Na figura 3.1b, os buracos são ventrículos laterais e não pertencem ao objeto. Já os buracos da figura 3.2b são resultantes de efeitos de iluminação e pertencem para o objeto. Note que, esse problema pode ocorrer em qualquer abordagem de segmentação por corte em grafo. No nosso método, nós fechamos os buracos presentes nas imagens binárias e consideramos somente o contorno externo como borda do objeto.

A figura 5.1 ilustra alguns resultados da combinação da IFT com medidas de corte. A figura 5.1a apresenta a segmentação do cérebro usando uma semente, função de custo de caminho c_{\max} , *mean cut* e o mapa de região da figura 3.1b. A figura 5.1b apresenta a segmentação do cérebro usando a mesma semente e mapa de região, porém com função de custo de caminho c_{sum} e corte normalizado. Note que, c_{sum} normalmente impõe mais regularidade para as bordas de corte do que c_{\max} . Por outro lado, c_{\max} é mais robusto com relação a localização inicial da semente e além disso a borda de corte se adapta melhor a saliências e concavidades.

Para a segmentação do pimentão vermelho mostrada na figura 5.1c, usamos c_{\max} , *mean cut* e o mapa de região da figura 3.2b. O mapa de região da figura 3.2d é usado como R_o para calcular $w(p, q)$ na equação 5.6. Nós usamos sementes de fundo para calcular o outro mapa de região R_b . Figuras 5.1d e 5.1e mostram as segmentações da figura 3.2c para uma e duas sementes, respectivamente. Em ambos casos, nós usamos a função de energia da equação 5.6 e c_{\max} .

5.4 Resultados experimentais

É difícil evitar falsas bordas de corte fora do objeto, quando o fundo contém muitas partes com características de imagem similares àquelas do objeto. A medida *mean-cut* é a mais sensível a esse problema. Os resultados na figura 5.1 foram obtidos pela seleção dos cortes mínimos em todos os casos, exceto para a medida *mean-cut* (figuras 5.1a e 5.1c). A figura 5.2 ilustra a curva do *mean-cut* versus a ordem de propagação dos pixels para uma imagem RM do cérebro (figura 5.1a). O mínimo desejado encontra-se na ordem 13.251 e ele pode somente ser detectado quando nós limitamos a busca até alguma ordem de propagação o , tal que $o < n$ e que seja maior que o tamanho do objeto procurado. Os resultados nas figuras 5.1a e 5.1c foram obtidos com $o = 0.7n$. Para comparação, a figura 5.3 mostra a medida de corte normalizado e a função de energia para a mesma imagem RM do cérebro, com seus respectivos resultados de segmentação usando a mesma semente, função de custo de caminho c_{\max} , e mapa de região da figura 3.1b (um mapa de região adicional para o fundo foi criado para a função de energia e foi necessário aumentar λ para 80 na equação 5.6).

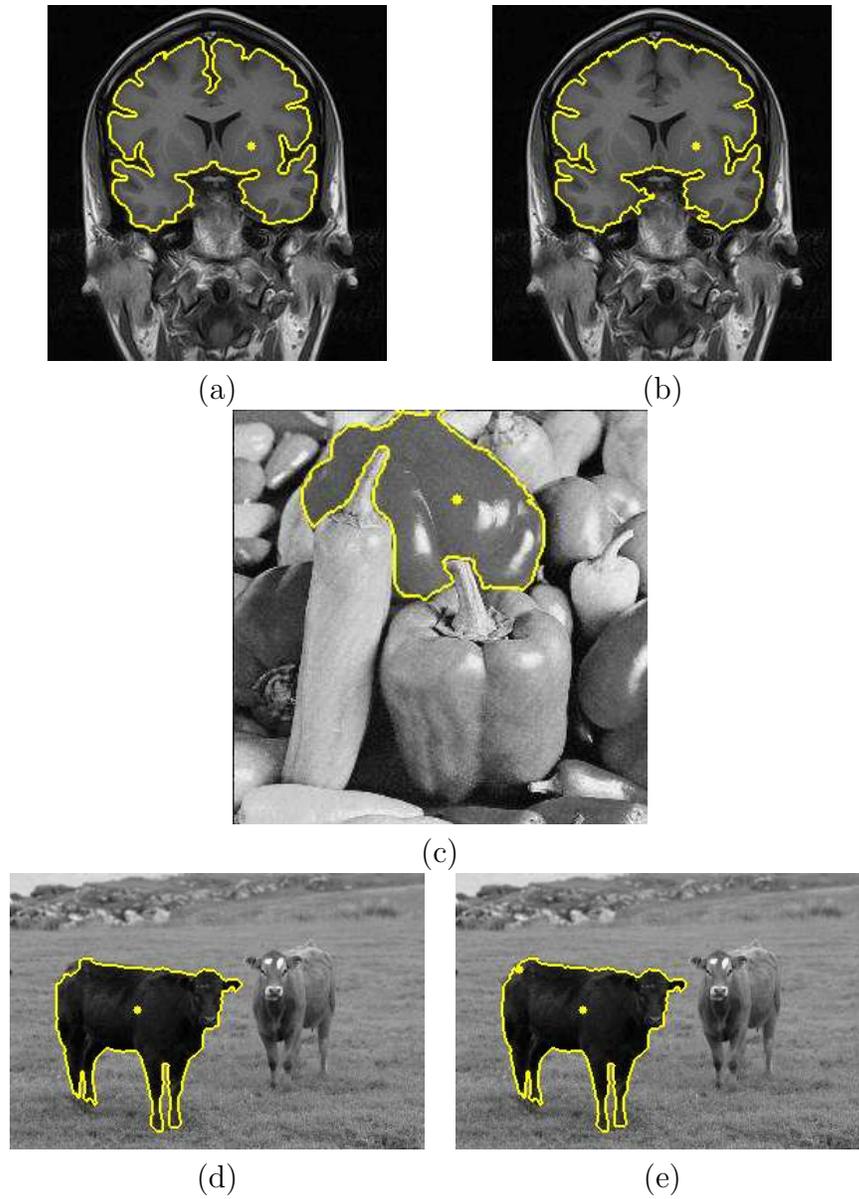


Figura 5.1: Resultados das segmentações onde as sementes são indicadas por pontos, usando (a) c_{\max} , $w(p, q)$ pela equação 3.3 e *mean cut*, (b) c_{sum} , $w(p, q)$ pela equação 3.3 e corte normalizado, (c) c_{\max} , $w(p, q)$ pela equação 3.3 e *mean cut*, e (d–e) c_{\max} , $w(p, q)$ pela equação 3.3 e a função de energia da equação 5.6.

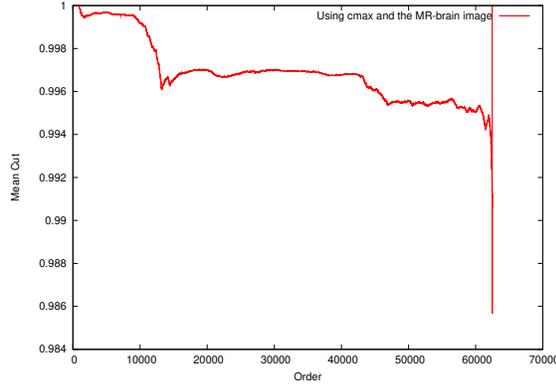


Figura 5.2: A curva do *mean-cut* versus a ordem de propagação dos pixels para uma imagem RM do cérebro (figura 5.1a). O mínimo desejado só pode ser detectado quando nós limitamos a busca até uma ordem de propagação o , tal que $o < n$ e que seja maior que o tamanho do objeto.

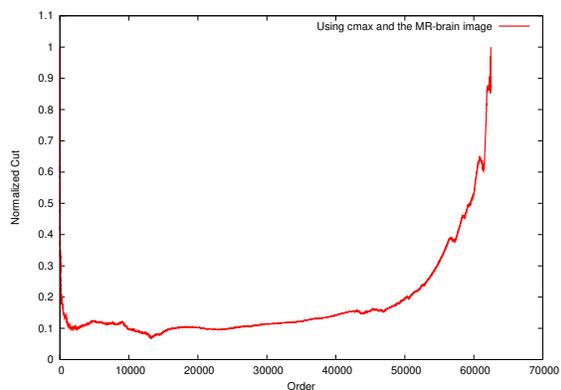
Isto mostra que qualquer abordagem para separar objeto e fundo usando corte em grafo provavelmente precisará de restrições fortes, porque o problema não pode ser simplesmente reduzido à busca de um mínimo global de uma função objetivo. Visto que falsas bordas de corte devido as similaridades existentes entre objeto e fundo são muito comuns na prática, nós escolhemos uma aplicação que representa o pior caso para avaliar o nosso método.

Nós selecionamos 6 imagens de fragmentos arqueológicos, similares as mostradas na figura 5.4a. Nessa aplicação, as bordas de cada fragmento devem ser perfeitamente detectadas de forma a permitir a reconstrução dos objetos originais [45]. Portanto, qualquer pequena falha na detecção da borda, é considerada um erro de segmentação. As imagens possuem 512×384 pixels ($n = 196.608$) e um total de 211 fragmentos. Nós aplicamos operações morfológicas para reduzir o ruído interno, eliminar o padrão em forma de grade do fundo, e estimar um pixel semente em cada fragmento. Essa abordagem foi capaz de encontrar sementes dentro de 201 dos 211 fragmentos (figura 5.4b). Portanto, o nosso experimento consiste em usar o método proposto para detectar as bordas de 201 fragmentos com sementes internas, nas imagens filtradas.

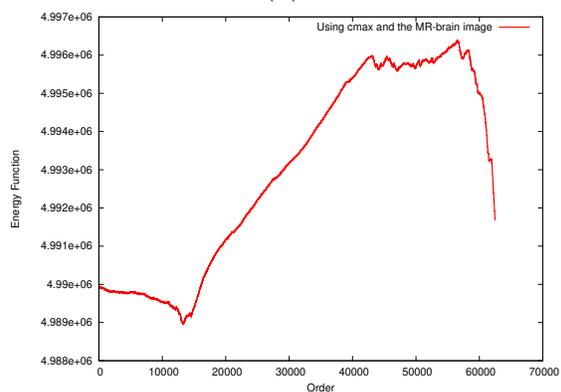
Foram usadas uma relação de adjacência A de vizinhança 4 e função de custo de caminho c_{\max} . Com relação a dissimilaridade e peso de arestas, ao invés de calcular um mapa de região para cada fragmento, nós decidimos usar funções mais simples.

$$\delta(p, q) = |f(p) - f(q)| \quad (5.7)$$

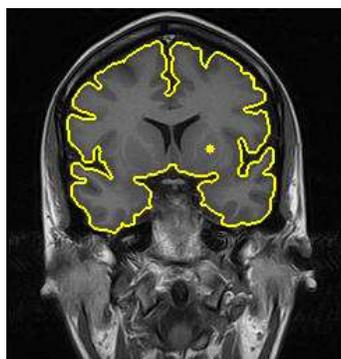
$$w(p, q) = 1.0 - \frac{\delta(p, q)}{K} \quad (5.8)$$



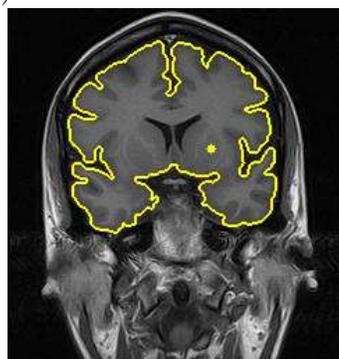
(a)



(b)



(c)



(d)

Figura 5.3: (a-b) Curvas do corte normalizado e função de energia versus a ordem de propagação para a imagem RM do cérebro. (c-d) Suas respectivas segmentações usando a mesma semente, funções de custo de caminho c_{max} , e mapas de região.

onde $f(p)$ é o brilho do pixel p e K é o máximo valor de brilho da imagem filtrada. No entanto, o mapa de região Ro e Rb foram calculados para a imagem inteira, levando em conta que fragmentos e não fragmentos possuem características dissimilares e foram usados na seguinte função de energia.

$$\lambda \left(\sum_{\forall p \in I} (1 - Ro(p)) + \sum_{\forall q \in E} (1 - Rb(q)) \right) + \sum_{\forall (p,q) | p \in I, q \in E} \alpha(p, q) w(p, q) \quad (5.9)$$

onde

$$\alpha(p, q) = \begin{cases} 0 & \text{se } Ro(p) > Rb(p) \text{ e} \\ & Ro(q) < Rb(q) \\ 1 & \text{caso contrário.} \end{cases} \quad (5.10)$$

Isto é, os mapas de região Ro e Rb podem ser usados para restringir o cálculo de $w(p, q)$ no interior de regiões não seguras, como sugerido em [72].

Nossa estratégia é atribuir um número distinto para cada semente, detectar cada fragmento separadamente, e rotular ele com seu correspondente número. Alguns fragmentos tocam outros, mas o algoritmo pode separar eles. Nesse caso, quando o algoritmo falha, ele normalmente gera a união de dois fragmentos encostados duas vezes, uma para cada semente. Essa situação é automaticamente detectada e os fragmentos são separados por uma transformada de watershed restrita a união deles [30]. Figuras 5.4c e 5.4d ilustram exemplos de detecções corretas e incorretas, respectivamente.

A IFT com c_{\max} e corte normalizado corretamente detectou somente 52 fragmentos (25.87%). A fim de confirmar que esse péssimo resultado não foi devido ao crescimento de região da IFT, nós repetimos o experimento com c_{\max} e *mean cut*, porém limitamos a busca pelo valor de corte mínimo até a ordem de propagação $o = 0.7n$. O método corretamente detectou 172 (85.57%) fragmentos. De fato, os fragmentos não são muito grandes e nós podemos reduzir ainda mais esse limiar para $o = 0.05n$. Nesse caso, a IFT com c_{\max} e *mean cut* corretamente detectaram 190 (94.53%) fragmentos. Nós repetimos esse último experimento com c_{sum} , porém o número de detecções corretas foi reduzido para 178 (88.56%).

A IFT com c_{\max} e função de energia corretamente detectou 182 (90.50%) dos fragmentos com sementes. Embora tendo um número de detecções corretas inferior ao obtido com *mean cut* e $o = 0.05n$, observamos que as funções de energia são normalmente mais robustas que as demais medidas avaliadas, que ou requerem restrições adicionais (*Mean cut* [70, 69]) ou são afetadas pela borda da imagem (*Normalized cut* [62, 63]).

Finalmente, o tempo médio de execução gasto pelo método em imagens com 512×384 pixels foi de 161 ms, usando um PC Pentium IV de 2.8GHz.

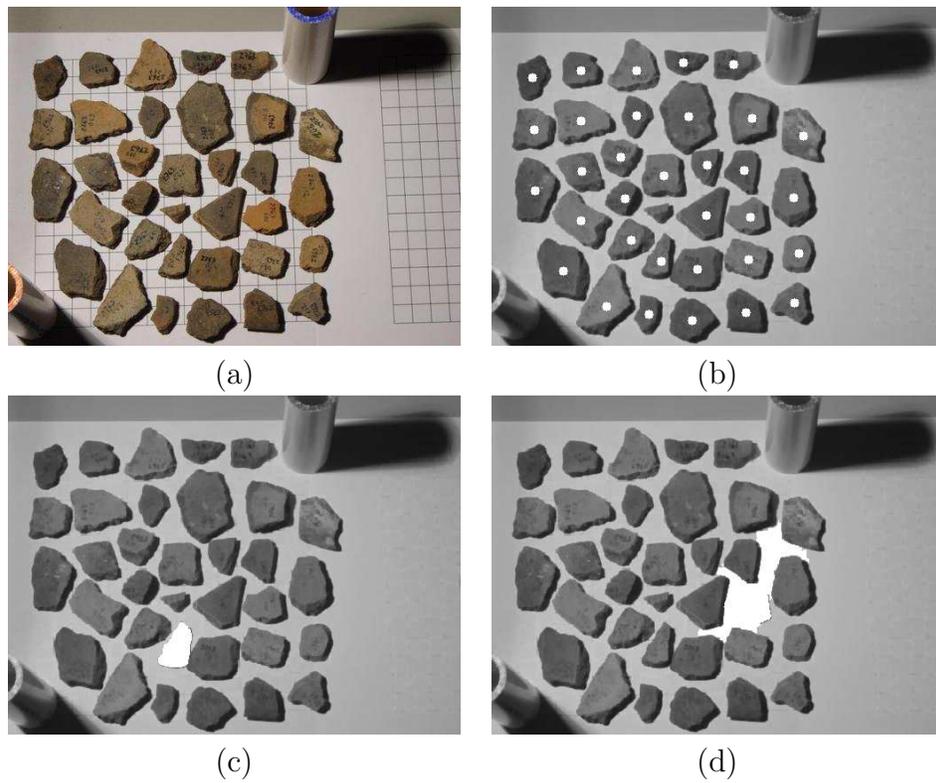


Figura 5.4: Detecção de fragmentos arqueológicos. (a) A imagem original. (b) O centro de cada disco representa um pixel semente sobre a imagem filtrada. (c-d) Exemplos de detecções corretas e incorretas, respectivamente.

Esses experimentos mostram que, até mesmo no pior caso, o método proposto proporciona bons resultados (exatidão acima de 90%) sob certas restrições fortes.

Capítulo 6

Segmentação por poda de árvores

6.1 Introdução

Para uma dada imagem com pixels sementes dentro do objeto de interesse, a IFT calcula uma floresta de caminhos ótimos no grafo derivado da imagem, tal que objeto e fundo estão conectados por poucos caminhos ótimos, que denominamos de *caminhos de vazamento*. As sementes internas competem entre si e apenas algumas se tornam raízes de caminhos de vazamento. A escolha da função de custo visa conectar objeto e fundo por poucos caminhos de vazamento, que cruzam a borda do objeto através de suas partes “mais fracamente conexas” chamadas de *pixels de vazamento*. A topologia da floresta é explorada para identificar os pixels de vazamento e eliminar suas subárvores, de modo que a floresta remanescente define o objeto. O método é chamado *segmentação por poda de árvores*.

A *segmentação por poda de árvores* foi primeiramente apresentada pelo nosso grupo de pesquisadores em [22], com duas abordagens para detecção de pixels de vazamento. Uma interativa, não facilmente extensível para imagens multidimensionais, onde o usuário pode visualmente identificar os pixels de vazamento e selecioná-los usando o ponteiro do mouse. A outra é automática, porém depende de parâmetros *ad-hoc*. Nesse trabalho o método é apresentado com uma abordagem automática para a detecção dos pixels de vazamento, que é livre de parâmetros; vários exemplos são mostrados; as limitações e vantagens são discutidas; e o método é avaliado em aplicações reais.

6.2 Métodos relacionados

Em comparação com abordagens de crescimento de regiões que usam caminhos ótimos a partir de sementes internas [66], o critério para separar objeto e fundo não é baseado nos custos dos caminhos ótimos mas sim na topologia da floresta. Assume-se que caminhos

ótimos com extremos em pixels do objeto não passam através do fundo, ao invés de terem custos estritamente inferiores aos custos de caminhos que alcançam o fundo.

No caso de outras abordagens com competição entre sementes por zonas de influência [3, 67, 59], a ausência de informação de borda e/ou a heterogeneidade do fundo normalmente causam a invasão de regiões do fundo (vazamento) por parte das zonas de influência das sementes de objeto e/ou o oposto. A *segmentação por poda de árvores* difere dessas abordagens em dois aspectos: ela não requer sementes externas (simplificando a etapa de seleção de sementes) e explora o problema de vazamento para resolver a segmentação.

Em comparação com técnicas de corte em grafo que usam nós terminais e restrições fortes (pixels sementes) [72], o método de segmentação por poda de árvores não requer sementes externas, usa um grafo derivado da imagem mais simples, e roda em tempo proporcional ao número de pixels.

6.3 Definição

Nós iniciamos a partir de um conceito similar ao da transformada de watershed [3, 67]. Na transformada de watershed por marcadores, a borda de um objeto é uma linha de watershed resultante de uma simulação de inundação. O método assume sementes internas e externas (marcadores) com uma fonte de água na localização de cada semente; simula um processo de inundação sobre a superfície topográfica de uma imagem de gradiente; e levanta barreiras (linhas de watershed) sempre que dois volumes de água provenientes de fontes internas e externas se encontram.

Na segmentação por poda de árvores nós temos somente fontes internas e deixamos a água vazar para o fundo através dos pixels mais baixos na fronteira do objeto. Na prática, a borda normalmente não possui a mesma altura e esses vazamentos não são muitos. Isto é, os fluxos de água que alcançam o fundo normalmente passam através de poucos pixels de vazamento. A identificação dos pixels de vazamento permite o erguimento de barreiras em suas localizações, separando a água de dentro da água de fora do objeto.

A imagem de gradiente usada deve ser mais alta na fronteira do objeto do que no seu interior e, ao menos, em uma vizinhança nas imediações externas. Como veremos, essa *condição do gradiente* é importante para que o método seja eficaz independentemente da localização das sementes.

Nas seções que se seguem, iremos instanciar a IFT [30] para a segmentação por poda de árvores, apresentar uma abordagem para a detecção automática dos pixels de vazamento, discutir como criar gradientes que satisfazem as condições apontadas acima, e mostrar vários exemplos que ilustram as vantagens e limitações do método.

6.4 Podando as árvores da floresta gerada pela IFT

Nesse método usamos a IFT com relação de adjacência A de vizinhança 4, com função de custo de caminho c_{\max} e peso de aresta dado por $\delta_2(p, q)$ (equação 4.2) sobre o gradiente Sobel ou algum outro gradiente com as bordas do objeto realçadas (ver capítulo 3). Esta configuração simula um processo de inundação sobre o gradiente visto como uma superfície topográfica. Durante essa inundação a água irá vazar para o fundo através dos pixels mais baixos da borda. Na prática, devido a variabilidade e ruídos presentes, as bordas normalmente não apresentam a mesma altura e conseqüentemente esses pixels de vazamento não são muitos.

A figura 6.1b mostra um exemplo onde o mapa de predecessores representa uma única árvore de caminhos ótimos enraizada na semente indicada na figura 6.1a.

Objeto e fundo na figura 6.1b estão conexos através do pixel (4,4), que representa a mais baixa intensidade na fronteira do objeto. Claramente, esse resultado irá depender da condição do gradiente. Se ela for satisfeita, o objeto pode ser obtido através da eliminação da subárvore do pixel de vazamento (figura 6.1c).

Quando dois caminhos ótimos alcançam um pixel p ambíguo com o mesmo custo mínimo, p é atribuído para o primeiro caminho que o atingiu. Isto é, os empates são quebrados usando política *first-in-first-out* no algoritmo da IFT (ver seção 2.3). Após o vazamento, essa política trata os pixels ambíguos nas imediações externas de modo que os caminhos de vazamento se ramificam em vários ramos (busca em largura), evitando que outros caminhos alcancem o exterior (veja figura 6.1b). Isto permite explorar a topologia da floresta para a detecção de pixels de vazamento.

6.5 Detecção automática de pixels de vazamento

A detecção automática dos pixels de vazamento pode ser feita sem o uso de parâmetros *ad-hoc* da seguinte forma. Seja B um conjunto de pixels pertencentes ao fundo. Por exemplo, a borda da imagem. Todos caminhos que atingem pixels do conjunto B passam através dos pixels de vazamento (figura 6.1b). Nós calculamos o número de descendentes que cada pixel da floresta possui no conjunto B obtendo assim um mapa de descendentes D (figura 6.1d). Para cada caminho π que atinge pixels do conjunto B teremos um número de descendentes máximo $d_{\max}(\pi)$ no mapa D ao longo do caminho (sem considerar a contagem do pixel raiz). Percorrendo cada caminho ótimo π no sentido inverso a partir de qualquer pixel em B de volta para a sua respectiva raiz, teremos que a primeira ocorrência de uma contagem igual ao máximo $d_{\max}(\pi)$ corresponderá a um pixel de vazamento (e.g., $p = (4, 4)$ com $D(p) = 36$ na figura 6.1d). Além disso, a soma das contagens do mapa D para pixels de vazamento distintos detectados será igual a cardinalidade do conjunto B .

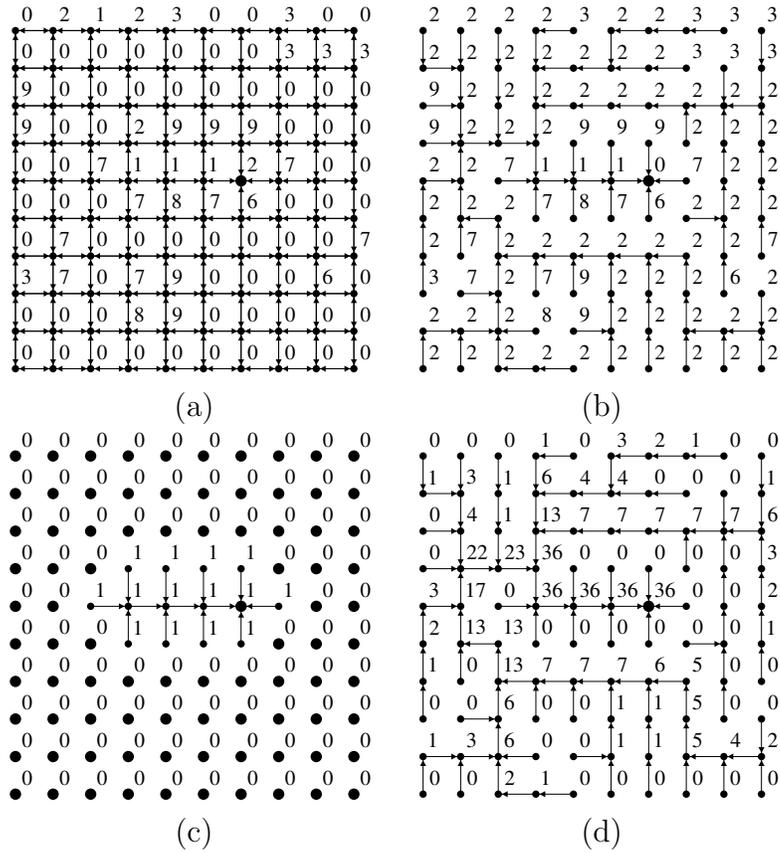


Figura 6.1: (a) Grafo com vizinhança 4 onde os números são intensidades do gradiente e o vértice maior denota uma semente no interior do objeto. (b) Uma árvore de caminhos ótimos para c_{\max} enraizada nessa semente. (c) O objeto pode ser detectado através da eliminação da subárvore do pixel de vazamento na posição (4,4). (d) O número de descendentes na borda da imagem B para a árvore de caminhos ótimos da figura 6.1b. O primeiro máximo no caminho de volta a partir de qualquer pixel em B para a raiz encontra-se no pixel (4,4).

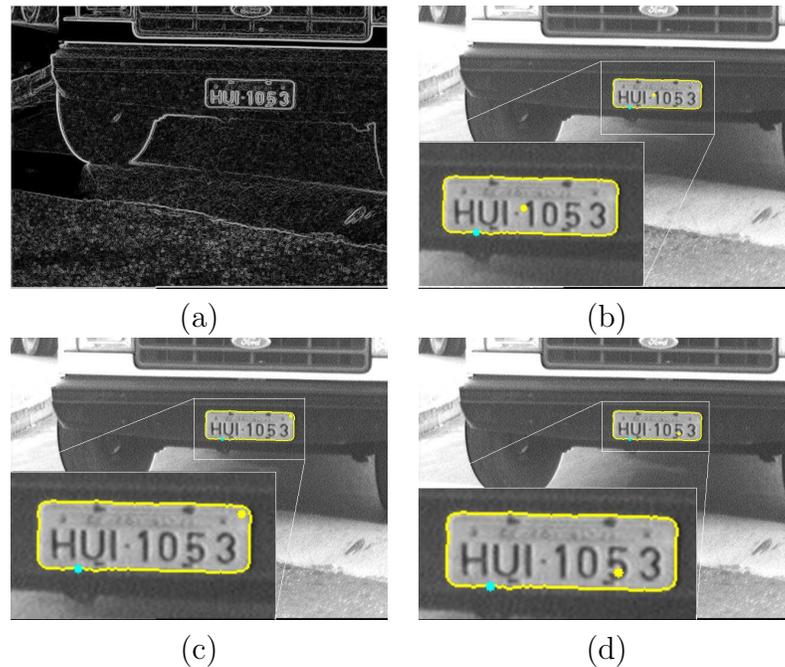


Figura 6.2: (a) A magnitude do gradiente de Sobel de uma imagem para a detecção da placa do carro. (b)–(d) A placa é automaticamente detectada para diferentes localizações da semente (pontos amarelos) usando (a) como imagem de gradiente. O pixel de vazamento está sempre no mesmo lugar (ponto na cor ciano).

Logo podemos parar o processo assim que a igualdade entre a soma e a cardinalidade de B for satisfeita.

Figura 6.2 ilustra um exemplo de detecção de placas de carro, onde a imagem de gradiente é a magnitude do gradiente de Sobel. Note que a condição do gradiente não é satisfeita, devido a presença de altos gradientes internos, mas o método funciona contanto que a semente selecionada não esteja dentro de uma letra ou dígito da placa.

A figura 6.3 ilustra outra situação onde a fronteira do objeto desejado encontra-se no interior de (confinada por) uma falsa borda e todos caminhos que alcançam a borda da imagem passam através de um pixel de vazamento (ponto na cor ciano) da borda externa. Nesse caso, nós não podemos usar a borda da imagem como conjunto B para detectar o objeto, mas podemos executar o método duas vezes. Na primeira iteração, nós usamos a borda da imagem para detectar a borda externa, e o objeto é então detectado na segunda iteração usando a falsa borda externa como conjunto B .

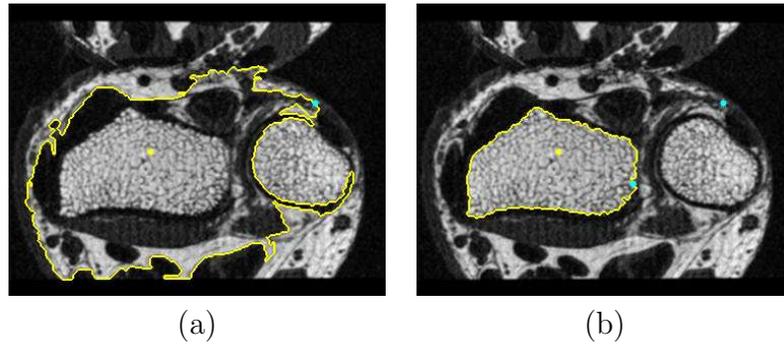


Figura 6.3: Uma imagem RM de um pulso onde sua borda pode ser detectada mediante duas iterações do método para uma única semente (ponto amarelo). (a) A primeira iteração usa a borda da imagem como conjunto B e automaticamente detecta uma borda externa com um único pixel de vazamento (ponto na cor ciano). (b) A borda externa é então usada como conjunto B e o objeto desejado é detectado com um único pixel de vazamento (ponto na cor ciano) na segunda iteração.

6.6 Pré-processamento para a poda de árvores

Em ordem para satisfazer as condições de gradiente exigidas, pré-processamento é algumas vezes necessário para realçar as bordas de interesse em detrimento de outras. Por exemplo, podemos primeiramente proceder com uma classificação dos pixels baseada em características dos objetos procurados (e.g., cor, distribuição de brilho, textura) gerando um mapa de pertinência. A partir desse mapa é gerada então uma imagem de gradiente realçada. O capítulo 3 apresenta em detalhes essa técnica. A figura 6.4 ilustra alguns exemplos de segmentações obtidas a partir desses gradientes realçados.

A fim de demonstrar a eficácia da segmentação por poda de árvores, em algumas aplicações da seção 6.9 iremos usar apenas gradientes simples tais como a magnitude do gradiente de Sobel da imagem original.

6.7 Limitações

Curiosamente, o método falha em situações ideais, tais como a mostrada na figura 6.5a, onde temos uma borda perfeita. O método falha porque quase todos pixels da borda são pixels de vazamento. Um outro problema ocorre quando as partes fracas da borda são aberturas perfeitas (figura 6.5b). No entanto, na presença de ruído, o método normalmente funciona mesmo para bordas com aberturas (figuras 6.5c e 6.5d).

Um outro problema pode ocorrer quando objetos múltiplos com intensidades de gra-

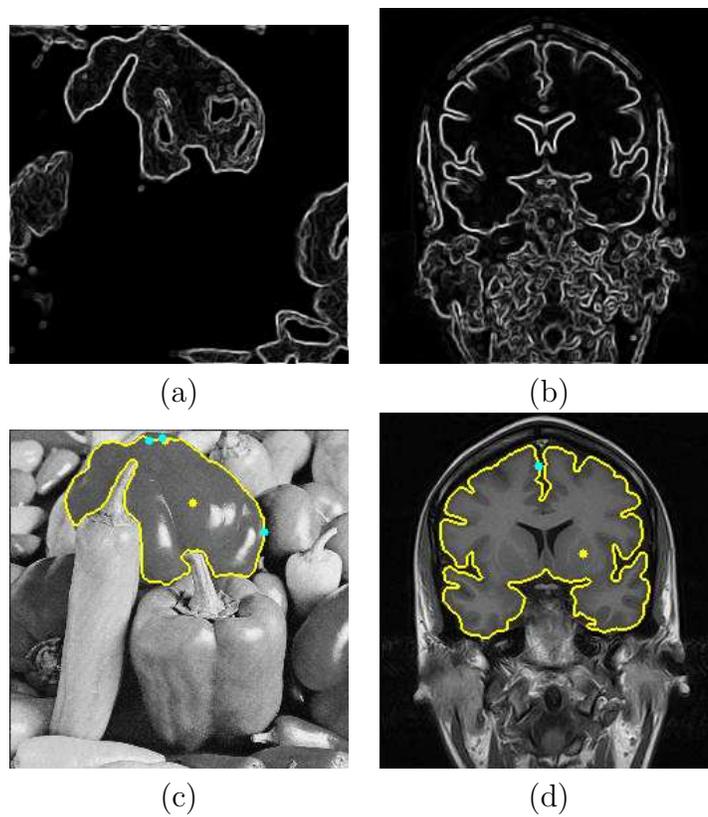


Figura 6.4: (a)–(b) Imagens de gradiente realçadas baseadas em mapas de região para a imagem de um pimentão e uma imagem RM do cérebro. (c)–(d) Resultados de segmentação para ambas imagens, onde as sementes são selecionadas manualmente (pontos amarelos) e os pixels de vazamento (pontos na cor ciano) são automaticamente detectados.

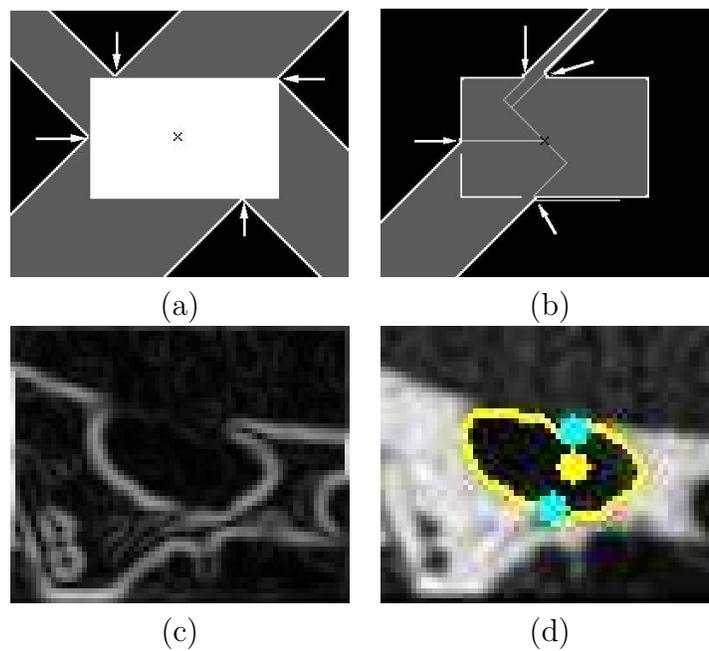


Figura 6.5: (a) Imagem com objeto de borda perfeita e (b) imagem com objeto de borda contendo aberturas perfeitas, após quatro podas interativas indicadas pelas setas. (c) A magnitude do gradiente de Sobel aplicada em uma imagem RM de uma veia do pulso. (d) Resultado da segmentação para uma semente interna (ponto amarelo) usando (c) como imagem de gradiente. Dois pixels de vazamento são automaticamente detectados (pontos em ciano).

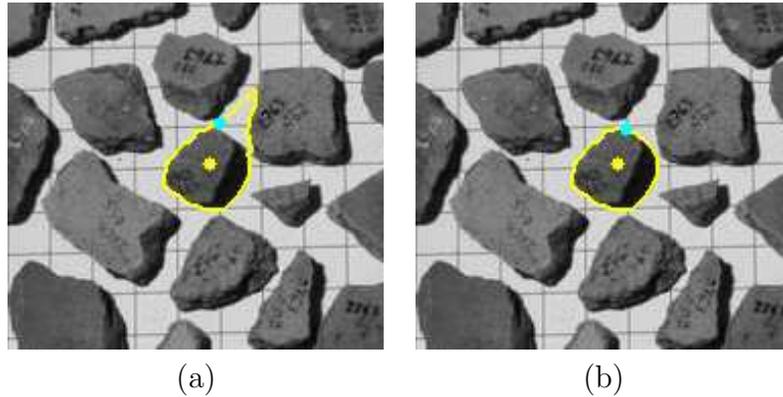


Figura 6.6: Imagem com um fragmento marcado para detecção. (a) O pixel de vazamento detectado está fora do fragmento devido a proximidade com outros fragmentos. (b) O pixel de vazamento correto é automaticamente detectado usando a variante do gradiente máximo.

diente similares estão muito próximos entre si. A ausência de espaço entre os objetos impede o surgimento das ramificações do caminho de vazamento, sendo que essas só ocorrerão mais adiante gerando um falso pixel de vazamento fora do objeto (figura 6.6a). No entanto, se a condição de gradiente for satisfeita, podemos então assumir que a correta localização de um pixel de vazamento estará sempre na posição de gradiente máximo ao longo do trecho entre o pixel detectado e a sua raiz. Nós chamamos essa solução de *variante do gradiente máximo*. Note que ela não afeta a localização de pixels de vazamento reais que já se encontram sobre a borda do objeto, mas corrige o problema dos falsos pixels de vazamento como ilustrado na figura 6.6b.

6.8 Algoritmos

Algoritmo 4 CONTAGEM DO NÚMERO DE DESCENDENTES QUE PERTENCEM AO CONJUNTO B

ENTRADA: Floresta de caminhos ótimos P e conjunto de nós B .

SAÍDA: Mapa D com a contagem de descendentes no conjunto B .

1. **Para Cada** *pixel* p da imagem, **Faça** $D(p) \leftarrow 0$.
2. **Para Cada** *pixel* p tal que $p \in B$, **Faça**
3. $q \leftarrow p$.
4. **Enquanto** $P(q) \neq nil$, **Faça**
5. $D(P(q)) \leftarrow D(P(q)) + 1$, $q \leftarrow P(q)$.
6. **Retorne** $\{D\}$

Esse algoritmo percorre cada caminho ótimo que atinge nós em B no sentido inverso, incrementando a contagem de descendentes ao longo do trecho entre o nó em B e sua raiz.

O mapa D pode ser usado para detectar os pixels de vazamento como se segue.

Algoritmo 5 DETECÇÃO DOS PIXELS DE VAZAMENTO

ENTRADA: Floresta de caminhos ótimos P e mapa D do número de descendentes em B .

SAÍDA: Conjunto V_z de pixels de vazamento.

1. $V_z \leftarrow \emptyset$
2. **Para Cada** *pixel* $p \in B$, **Faça**
3. $q \leftarrow p, d_{max} \leftarrow -\infty.$
4. **Enquanto** $P(q) \neq nil$, **Faça**
5. **Se** $D(q) > d_{max}$ **Então** $d_{max} \leftarrow D(q), r \leftarrow q.$
6. $q \leftarrow P(q).$
7. $V_z \leftarrow V_z \cup \{r\}.$
8. **Retorne** $\{V_z\}$

Para cada pixel p pertencente a B , um pixel de vazamento r é localizado na primeira ocorrência do valor máximo d_{max} em D no caminho voltando de p até a sua raiz (veja figura 6.1d). Considerando que um mesmo pixel de vazamento pode ser detectado em mais de um caminho e que a soma das contagens de descendentes para pixels de vazamento distintos é a cardinalidade do conjunto B , o algoritmo 5 pode encerrar assim que essa observação for detectada.

Como discutido na seção 6.7 e ilustrado na figura 6.6a, a proximidade entre objetos pode causar falsos pixels de vazamento no fundo. A correção pela *variante do máximo gradiente* é apresentada a seguir. Se a condição de gradiente for satisfeita, então as posições dos pixels de vazamento que já se encontrarem sobre a borda do objeto não serão afetadas.

Algoritmo 6 CORREÇÃO PELO GRADIENTE MÁXIMO

ENTRADA: Imagem de gradiente \hat{G} , floresta de caminhos ótimos P , conjunto de pixels de vazamento V_z .

SAÍDA: Novo conjunto de pixels de vazamento V'_z .

1. $V'_z \leftarrow \emptyset$
2. **Para Cada** *pixel* $r \in V_z$, **Faça**
3. $q \leftarrow r, G_{max} \leftarrow G(q).$
4. **Enquanto** $P(q) \neq nil$, **Faça**

5. $\left[\begin{array}{l} \text{Se } G(q) > G_{max} \text{ Então} \\ \quad \left[\begin{array}{l} \text{L } r \leftarrow q, G_{max} \leftarrow G(q). \\ \quad q \leftarrow P(q). \\ \quad V'_z \leftarrow V'_z \cup \{r\}. \end{array} \right. \end{array} \right.$
- 6.
- 7.
- 8.
9. **Retorne** $\{V'_z\}$

6.9 Resultados experimentais

Nós testamos a eficácia do método de segmentação por poda de árvores em três diferentes situações: (i) rotulação de múltiplos objetos com texturas similares, (ii) definição de objetos 3D, e (iii) a detecção de objetos baseada na forma [23].

Em (i), texturas similares facilitam a seleção automática de sementes e a estimativa de gradiente, porém objetos próximos podem causar falsos pontos de vazamento no exterior (seção 6.7) e objetos que se tocam precisam ser separados uns dos outros. Nós repetimos o mesmo experimento envolvendo fragmentos arqueológicos apresentado no capítulo 5 porém agora usando a poda de árvores. Em (ii), a idéia é verificar a extensão direta para 3D do método. A segmentação 3D do cérebro foi a aplicação escolhida. Em (iii), mudando a localização das sementes na imagem, podemos gerar vários objetos candidatos pela técnica de poda de árvore e explorar alguma característica global para selecionar o objeto correto. Testamos esse procedimento na segmentação automática de placas de carros, devido ao formato simples retangular. Note que, nós não pretendemos propor as melhores soluções para cada uma dessas aplicações. Nosso objetivo é mostrar que algumas soluções preliminares e simples já fornecem bons resultados. Em todos os casos apresentados nós usamos a variante do gradiente máximo (algoritmo 6).

Alguns dos experimentos comparam os resultados com gabaritos (*ground truth*). Nós definimos FN como a *porcentagem de falsos negativos* – o número de pixels que pertencem ao objeto, mas foram classificados como fundo, dividido pelo número de pixels do objeto; e FP como a *porcentagem de falsos positivos* – o número de pixels classificados como objeto, mas que pertencem ao fundo, dividido pelo número de pixels do objeto.

6.9.1 Rotulação de múltiplos objetos com texturas similares

O padrão de grade do fundo (figura 6.6a) pode ser eliminado por um fechamento morfológico por um disco de raio 1. Nós calculamos o gradiente de Sobel da imagem filtrada e usamos sua magnitude como função de dissimilaridade ($\delta_2(p, q) = G(q)$ na equação 2.1). O resultado obtido foi de 96.52% de acerto. Este resultado mostra que a variante do gradiente máximo é uma forma eficaz para corrigir os falsos pixels de vazamento (seção 6.7).

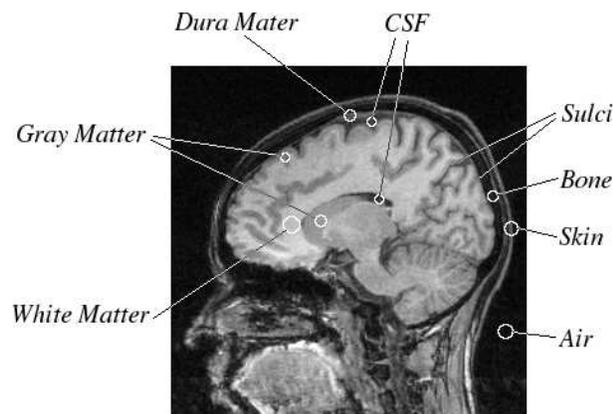


Figura 6.7: Tecidos em imagens RM-T1 do cérebro. Substância branca (*white matter*), substância cinza (*gray matter*), dura mater, sulcos (*sulci*), osso (*bone*), pele (*skin*), ar (*air*) e líquido cérebro-espinhal (CSF).

6.9.2 Segmentação 3D de imagens RM do cérebro

A segmentação do cérebro humano a partir de imagens de ressonância magnética (RM) é um problema em aberto, que tem sido tratado de vários modos diferentes, cada um com seus pontos positivos e negativos[39]. A figura 6.7 mostra uma fatia RM-T1 do cérebro, onde algumas estruturas estão indicadas. Nós desejamos separar a substância cinza (GM) e a substância branca (WM), como um único objeto, do restante da imagem. Nós avaliamos a segmentação por poda de árvores usando imagens de *phantoms* RM-T1 (que encontram-se disponíveis no site BrainWeb¹ [13] juntamente com seus gabaritos) e imagens reais RM-T1. Nós comparamos os resultados com os obtidos pelo algoritmo baseado em *template* fornecido pelo SPM2² [36].

Estimação do gradiente 3D

No caso de imagens de *phantoms*, a extensão 3D do gradiente de Sobel é suficiente para proporcionar bons resultados de segmentação. Por outro lado, imagens reais apresentam outros artefatos e necessitam de realce da borda entre GM e líquido cérebro-espinhal (CSF) a fim de satisfazer a condição do gradiente.

Imagens RM-T1 do cérebro formam um volume no espaço 3D que contém dois grandes *clusters*: o primeiro composto por ar, osso e líquido cérebro-espinhal (CSF), caracterizados por voxels mais escuros, e o segundo representado por voxels claros, consistindo do GM, WM, pele, gordura e músculos. O limiar ótimo de Otsu [57] pode separar esses dois grupos

¹URL: <http://www.bic.mni.mcgill.ca/brainweb/>

²URL: <http://www.fil.ion.ucl.ac.uk/spm/software/spm2/>

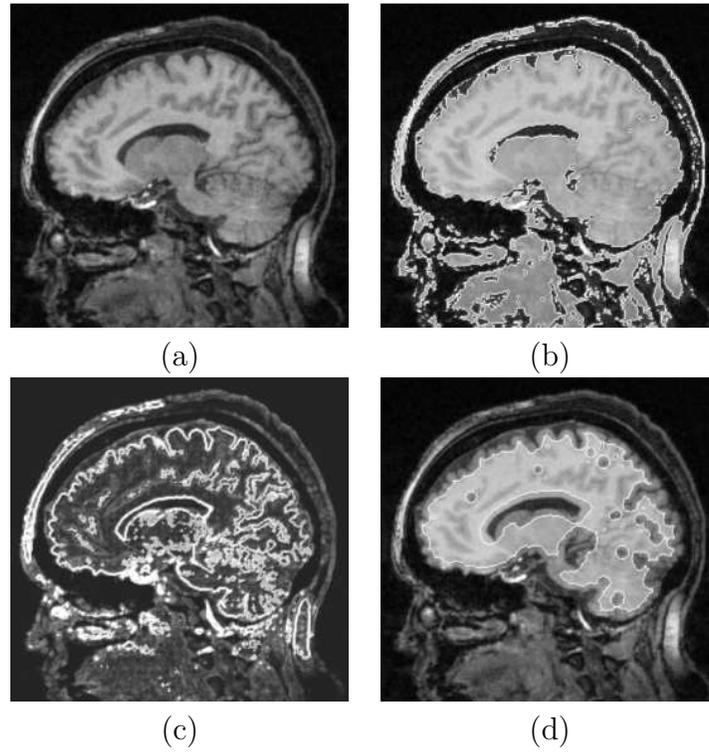


Figura 6.8: (a) Corte sagital de um volume RM-T1 do cérebro. (b) Visão sobreposta da limiarização de Otsu aplicada no volume para o corte em (a). (c) Corte sagital do gradiente 3D final. (d) Visão dos voxels sementes para o corte em (a).

(figuras 6.8a e 6.8b), de forma que a borda GM/CSF se torne parte da borda entre eles. Para realçar a borda GM/CSF na imagem 3D original, nós multiplicamos a intensidade de cada voxel $I(p)$ por um peso $w(p)$ como se segue:

$$w(p) = \begin{cases} 0 & I(p) \leq m_1 \\ 2 \left(\frac{I(p)-m_1}{m_2-m_1} \right)^2 & m_1 < I(p) \leq \tau \\ 1 - 2 \left(\frac{I(p)-m_2}{m_2-m_1} \right)^2 & \tau < I(p) \leq m_2 \\ 2 & I(p) > m_2 \end{cases} \quad (6.1)$$

Onde τ é o limiar de Otsu, e m_1 e m_2 são as médias de intensidade calculadas em cada grupo. Nós então aplicamos um gradiente morfológico 3D [20] usando um elemento estruturante de vizinhança 6 (figura 6.8c).

Seleção de sementes

O *cluster* mais claro contém muitos voxels fora do cérebro (figura 6.8b). Para obter um conjunto de sementes dentro do cérebro, nós aplicamos uma erosão morfológica por um elemento estruturante esférico de raio 5 na imagem binária do grupo mais claro. Essa operação desconecta o interior do cérebro das estruturas vizinhas. Nós então selecionamos o maior componente conexo 3D como o conjunto de sementes (figura 6.8d).

A escolha do raio de erosão depende do tamanho dos voxels. O valor aqui apresentado foi ajustado para imagens RM com voxels isotrópicos de 1mm. O valor usado garante a separação das regiões cerebrais, gerando um número razoável de sementes no interior do cérebro. A resolução do voxel é sempre constante para uma dada aplicação, logo é sempre possível a sua configuração segura. É importante notar que a segmentação automática do cérebro não pode ser obtida apenas com o uso de operações morfológicas.

Resultados

O conjunto de sementes internas obtido é então usado para a segmentação 3D do cérebro pela técnica de poda de árvores. Usamos a IFT com relação de adjacência de vizinhança 6 entre voxels e, durante a detecção automática dos voxels de vazamento, usamos as faces do volume de dados como conjunto B .

No primeiro conjunto de experimentos, nós geramos 8 *phantoms* RM-T1 com diferentes configurações de ruído (INU) (tabela 6.1). Visto que o nosso método detecta a borda externa do cérebro, os objetos resultantes ainda contém pequenas regiões de CSF (sulcos e ventrículos). Para adequadamente classificar eles como fundo, nós tomamos a intersecção entre o objeto podado e os voxels com intensidades acima de τ .

Tabela 6.1 mostra os erros de segmentação para cada método quando comparados com o gabarito (*ground truth*). Note que os métodos apresentaram resultados similares estando o SPM2 ligeiramente mais preciso (veja também a figura 6.9). Porém para imagens reais o mesmo não se observa.

Para o segundo conjunto de experimentos, nós selecionamos 5 volumes RM de pacientes com epilepsia e 10 volumes RM de controles com nenhuma anomalia conhecida. Todas imagens foram adquiridas com um scanner RM *Elscint 2T* com tamanho de voxel de $0.98 \times 0.98 \times 1.00 \text{mm}^3$ e foram interpoladas para voxels isotrópicos de 1.00 mm antes de qualquer processamento.

O uso de *templates* (atlas do cérebro humano) não é garantia de sucesso na segmentação automática do cérebro humano em imagens de ressonância magnética [48]. A figura 6.10 ilustra resultados de segmentações 3D obtidos usando o algoritmo baseado em *template* fornecido no SPM2³ [36] para imagens RM-T1 reais de controles. Este software

³URL: <http://www.fil.ion.ucl.ac.uk/spm/software/spm2/>

Parâmetros		Seg. por poda de árvores		SPM2	
Ruído (%)	INU (%)	FP (%)	FN (%)	FP (%)	FN (%)
3	20	3.57	1.82	1.63	1.46
5	20	3.52	1.80	1.60	1.43
7	20	3.10	2.18	2.28	1.66
9	20	2.99	3.03	2.83	2.30
3	40	3.61	1.86	0.90	1.64
5	40	3.34	2.14	1.61	1.40
7	40	3.01	2.33	2.31	1.57
9	40	2.71	3.11	2.78	2.29

Tabela 6.1: Erros de segmentação dos *phantoms* do cérebro por poda de árvores e SPM2. Falsos positivos (FP) e falsos negativos (FN) são mostrados como uma porcentagem do volume de GM e WM obtida da máscara do gabarito (*ground truth*).

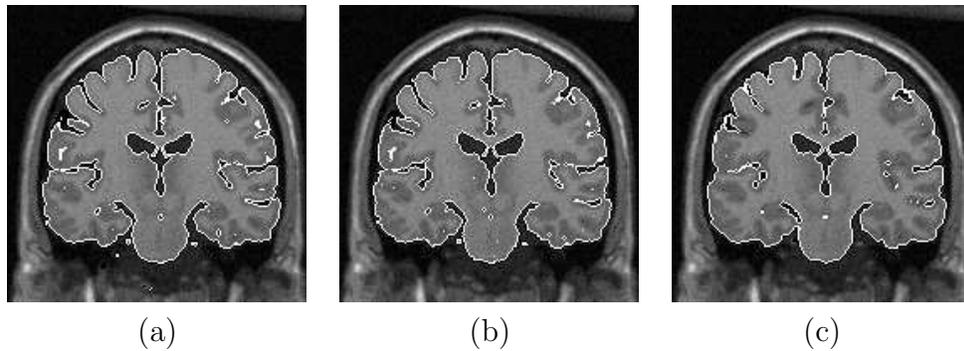


Figura 6.9: Amostras de fatias do experimento envolvendo segmentação de *phantoms* do cérebro: (a) Gabarito do *phantom* (*ground truth*) e a segmentação obtida com (b) SPM2, e (c) com a técnica de poda de árvores.

é amplamente usado em pesquisas de neurologia para segmentação e análise do cérebro e no entanto como podemos ver, os resultados estão longe do esperado [23].

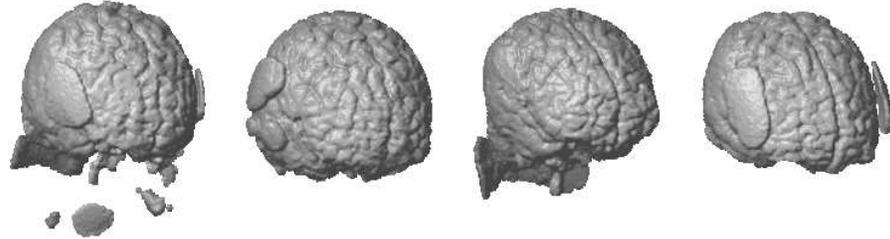


Figura 6.10: Renderizações 3D de segmentações incorretas de cérebro obtidas com o SPM2.

A figura 6.11 mostra os resultados obtidos pela técnica de poda de árvores que são nitidamente superiores aos dados pelo SPM2 da figura 6.10. Além disso, o procedimento proposto é nove vezes mais rápido que o SPM2 para a mesma estação de trabalho.

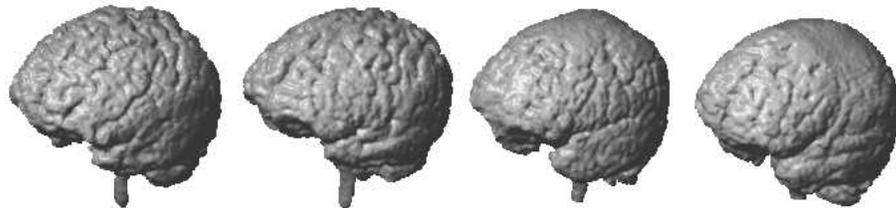


Figura 6.11: Renderizações 3D de segmentações do cérebro obtidas pela poda automática de árvores. As duas colunas da esquerda correspondem a controles e as outras duas colunas são de pacientes com epilepsia.

Em ambos casos, *phantoms* e imagens reais, nós consideramos como voxels de objeto no SPM2 aqueles com probabilidade maior que 50% de serem GM ou WM. Algumas estruturas fora do cérebro podem ser eliminadas com limiares mais altos, porém o SPM2 passa a perder alguns voxels de GM nesses casos. Por outro lado, o SPM2 elimina a espinha e um pouco mais de sulcos que a poda de árvores.

6.9.3 Detecção de objetos baseada na forma

Nós avaliamos a eficácia da segmentação por poda de árvores para a detecção automática de placas de carros em uma base de dados com 990 imagens em tons de cinza com 352×240

pixels cada (figura 6.12a). A magnitude do gradiente de Sobel é usada como imagem de dissimilaridade (figura 6.12b).

Nessa aplicação, a seleção automática de sementes é uma tarefa difícil porque qualquer tentativa de estimar sementes no interior da placa provavelmente irá encontrar também sementes em outras partes da imagem. Uma estratégia natural é selecionar marcadores como conjuntos de sementes (componentes conexos); rodar a poda de árvores para cada marcador separadamente; calcular um *score* para cada objeto candidato; e escolher o de melhor *score*. O *score* de cada objeto é obtido baseado em características de forma, visto que as placas são retângulos ligeiramente deformados.

A seleção de marcadores e o cálculo do *score* aqui apresentados são procedimentos *ad-hoc*, mas seus parâmetros podem ser fixados usando o nosso conhecimento prévio do problema. Melhores procedimentos para essas duas etapas são deixados para trabalhos futuros.

Seleção de marcadores

A seleção de marcadores visa isolar alguns pixels ao redor dos números das placas, dado que eles são compostos por pixels escuros na nossa base de dados. A figura 6.12c mostra a imagem binária dos pixels da figura 6.12a com valores abaixo de 30% da intensidade máxima. Caso nenhuma placa seja detectada para esse limiar, o algoritmo reduz o limiar para 15% da intensidade máxima e repete o processo de segmentação. Nós aplicamos uma erosão por um disco de raio 5.0, seguida por uma dilatação por um disco de raio 7.0 (figura 6.12d), e consideramos como marcadores os componentes da figura 6.12c que não pertencem à figura 6.12d. Esses parâmetros são muito pequenos para comprometer a eficácia do método (e.g., 93% das placas possuem tamanhos em torno de 2936 pixels). O raio de dilatação maior foi usado para evitar sementes sobre as bordas das placas. Nós também eliminamos componentes que tocam a borda da imagem e componentes com menos de 6 pixels. Marcadores presentes na região superior da imagem (30% da altura) são rejeitados visto que não há placas nessas áreas. Os marcadores resultantes são finalmente dilatados por um disco de raio 1 a fim de assegurar algumas sementes ao redor dos números das placas, e rotulados com diferentes números (figura 6.12e).

Seleção de objetos pela forma

A segmentação por poda de árvores é aplicada para cada marcador com rótulo distinto e os objetos resultantes recebem um *score* proporcional a sua similaridade em relação ao formato retangular. No final, o objeto com maior *score* é selecionado como a placa do carro. Esse processo é restrito a uma região de interesse de 200×100 pixels ao redor de cada marcador. Nós também rejeitamos objetos com menos de 1200 pixels e mais de 8200

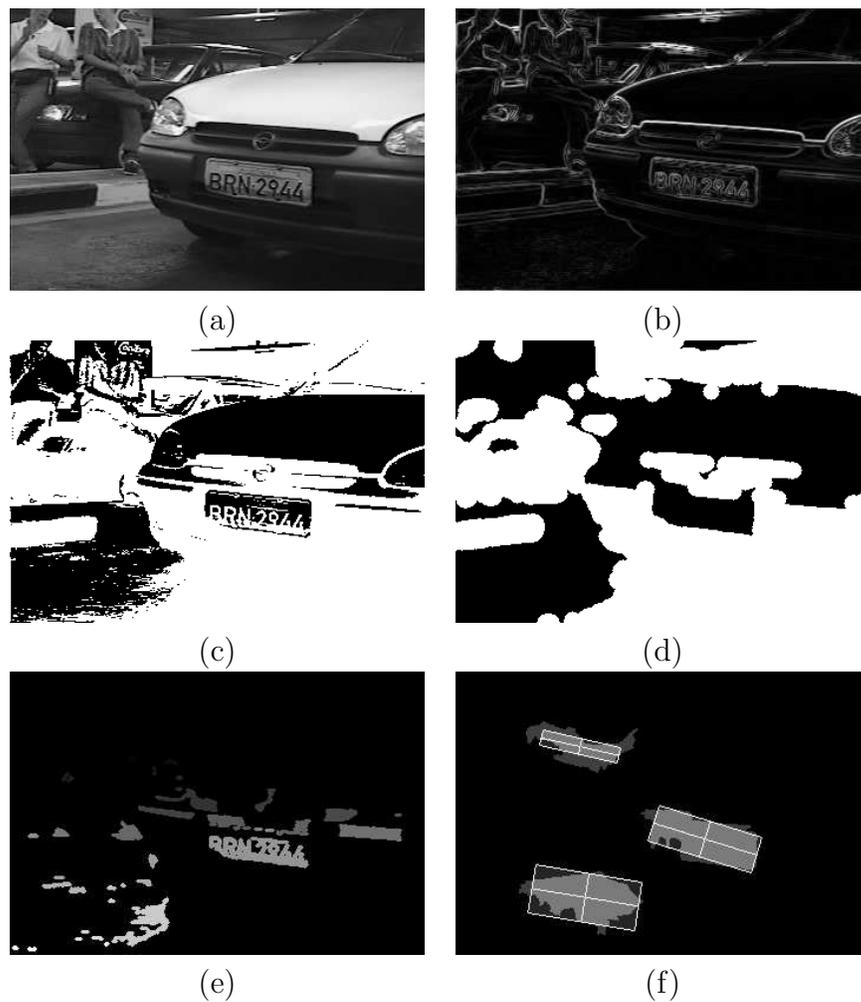


Figura 6.12: (a) Imagem original. (b) Imagem de gradiente. (c) Imagem (a) após limiarização. (d) Imagem (c) após erosão e dilatação. (e) Imagem com marcadores rotulados. (f) Os melhores retângulos para três objetos candidatos falsos.

pixels, visto que o tamanho das placas varia entre 1559 e 7753 pixels.

Primeiro, nós encontramos o melhor ajuste entre o objeto candidato e um retângulo. Para garantir invariância a rotação, nós determinamos o semi-eixo principal do objeto através de análise de componentes principais. Então a partir do centro geométrico, nós percorremos o eixo principal em ambos sentidos, alternadamente e ao mesmo tempo, até nós atingirmos o fundo em alguma direção. O mesmo processo é repetido para o semi-eixo ortogonal. Esses pontos de borda nos proporcionam toda informação necessária para traçarmos um modelo retangular. A figura 6.12f mostra três desses retângulos para falsos candidatos.

Considerando cada retângulo como um pseudo-gabarito, nós calculamos um *score* SC para o objeto correspondente baseado nas porcentagens de falsos positivos FP e falsos negativos FN , como se segue.

$$SC = (1.0 - \min\{1.0, FP + FN\}) * W \quad (6.2)$$

$$W = \begin{cases} 1.0 & \text{se } 2.4 \leq R \leq 3.0, \\ \exp(-(2.4 - R)^2/3.0) & \text{se } R < 2.4, \\ \exp(-(3.0 - R)^2/3.0) & \text{se } R > 3.0. \end{cases}$$

onde R é a razão entre os lados do retângulo e o intervalo $[2.4, 3.0]$ foi escolhido entre o mínimo e o máximo valor de razão de proporção esperados para as placas.

Resultados

O método corretamente detectou 931 (94.04%) placas das 990. Alguns exemplos de segmentações corretas e incorretas são mostrados na figura 6.13. Em todos os casos foi possível estimar marcadores no interior das placas. Entre as 59 placas não detectadas, o método falhou devido ao processo de *score* de objeto para 28 placas, e o método de segmentação por poda de árvores falhou para as 31 placas restantes (somente em 3.13% da base).

Conforme uma publicação recente [73], onde os autores usam uma base de dados com 784 imagens de 384×288 pixels, nossos resultados parecem ser bons. As imagens dessa base possuem resolução ligeiramente superior e as placas parecem ter formas menos deformadas. Nesse artigo [73] foram avaliadas quatro abordagens distintas com uma média de acerto de 88.83%. O método deles obteve 99.7% de detecções corretas e a taxa mínima de acerto foi de 82.0%.

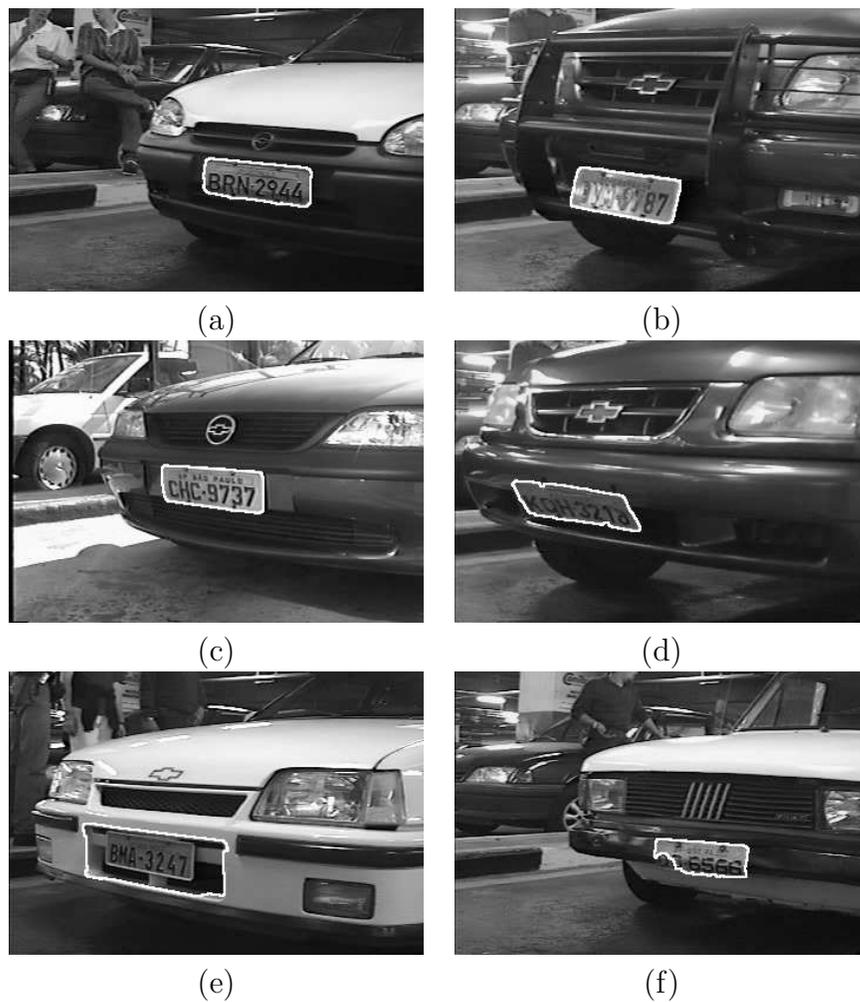


Figura 6.13: (a)-(d) Detecções corretas com *scores* 0.937, 0.913, 0.902, e 0.767, respectivamente. (e)-(f) Detecções erradas com *scores* 0.917 e 0.801, respectivamente.

Capítulo 7

Conclusões

7.1 Visão geral

A nossa meta durante o projeto foi desenvolver métodos de segmentação interativa de imagens que reduzissem a intervenção do usuário a simples escolha de poucos pixels na imagem, buscando minimizar o número de interações. Dessa forma a automação se torna possível em algumas aplicações onde esses pixels podem ser identificados automaticamente.

Nesse trabalho foram apresentadas três novas técnicas de segmentação, com extensões diretas para 3D, baseadas na IFT: (i) *segmentação por competição entre sementes κ -conexas* [25, 26], (ii) *segmentação por crescimento ótimo de regiões e corte em grafo* [24], e (iii) *segmentação por poda de árvores* [22, 23]. A técnica (i) ainda requer a escolha de um parâmetro pelo usuário, mas as técnicas (ii) e (iii) podem ser implementadas sem depender de parâmetros *ad-hoc*.

Em (i), nós apresentamos um algoritmo para a detecção de objetos baseado em componentes κ -conexos com competição entre sementes. Esse método proposto difere dos anteriores [66, 59, 41, 67, 3] nos seguintes aspectos: uso de diferentes valores de κ para cada semente e detecção automática dos valores de κ . O novo método reduziu consideravelmente o número de interações por parte de usuário necessárias para a segmentação de imagens médicas quando comparado com as abordagens anteriores.

Em (ii) foram discutidas algumas limitações da segmentação por corte em grafo e foi apresentada uma solução eficiente que usa a IFT para reduzir o problema geral de corte em grafo para $O(n)$ em imagens com n pixels. O método foi avaliado usando várias medidas de corte [63, 70, 72, 43] e experimentos. Os experimentos mostraram que a busca por cortes de valor mínimo normalmente requer restrições fortes e algum conhecimento prévio sobre o objeto desejado. Os resultados indicam que o método proposto pode ser eficaz para a detecção de objetos.

Diferentemente de outras abordagens por corte em grafos [6, 69], nosso método usa um grafo derivado da imagem mais simples, permitindo o uso de diferentes medidas de avaliação de corte, independentemente do grafo da imagem e roda em tempo linear. Por outro lado, a IFT tem obrigatoriamente que gerar a fronteira do objeto durante a propagação dos pixels. Nesse sentido, o método é mais restritivo do que o (iii), porque esse último requer apenas que caminhos ótimos que atingem pixels de objeto não passem através do fundo. Em relação ao método (i), o uso da limiarização pela ordem de propagação ao invés de pelo custo dos pixels é importante para criar transições mais suaves entre frentes de custo consecutivas, permitindo a avaliação de um número maior de objetos candidatos, o que pode melhorar a qualidade das segmentações.

Em (iii), nós introduzimos a *segmentação por poda de árvores* com detecção automática dos pixels de vazamento e mostramos através de vários exemplos que o método pode reduzir a detecção de objetos a simples escolha de algumas sementes internas. Essa técnica introduz uma abordagem inovadora dentro do paradigma de segmentação por região baseada em sementes. Ao contrário dos métodos tradicionais de crescimento de região que param o crescimento quando um certo critério é atingido [66], na *segmentação por poda de árvores* deixamos o processo de crescimento, dado pela floresta de caminhos de custos mínimos da IFT, dominar toda a imagem e exploramos a topologia da floresta ao final do processo para identificar o objeto. No contexto da IFT, temos que os métodos tradicionais de segmentação por competição entre sementes [59, 41, 67, 3] exploram o mapa de raízes e o método sem competição [66] explora o mapa de custos. Na *segmentação por poda de árvores* utilizamos o mapa de predecessores, o que é original para segmentação por região baseada na IFT. No contexto dos métodos clássicos de competição [59, 41, 67, 3], uma segmentação automática era dada pela seleção automática de sementes de objeto e de fundo. Já na *segmentação por poda de árvores* a escolha de sementes externas é substituída por uma estimativa automática de pontos para poda. A escolha de sementes internas sempre depende da aplicação, mas os pontos de poda podem ser estimados de maneira independente (seção 6.5).

O desempenho do método foi avaliado em três diferentes situações envolvendo a detecção automática de objetos: rotulação de fragmentos arqueológicos, segmentação 3D do cérebro e detecção de placas de carros. Em todos casos, foram obtidos bons resultados de segmentação (ver seção 6.9). A extensão 3D da técnica tem se mostrado bastante promissora para isolar automaticamente o cérebro humano (substâncias cinza e branca) em imagens de RM ponderadas em T1 (RM-T1), proporcionando resultados superiores aos fornecidos pelo SPM2 [36] (ver seção 6.9.2). Atualmente estamos buscando novas aplicações. Também pretendemos melhorar o método para detecção de placas de carros, além de comparar, no caso da segmentação 3D do cérebro, as segmentações obtidas pela poda de árvores com as dadas pelo SPM2 usando uma base de dados maior.

7.2 Discussão

Podemos concluir que o método (ii) é superior ao método (i) por utilizar um critério de seleção global e envolver otimalidade, ao contrário de (i) que utiliza um critério local (tamanho das frentes de custo) e requer um parâmetro *ad-hoc* (limiar T). No entanto em (i) as zonas de influência de cada semente são analisadas separadamente. Esse tratamento personalizado permite melhores ajustes, sendo menos restritivo do que (ii) que exige que em algum instante o crescimento de região conjunto das sementes compreenda totalmente e somente o objeto.

Baseado nessas observações, acreditamos que a combinação das técnicas (i) e (ii) é uma opção promissora. Analisaríamos as zonas de influência das sementes separadamente como em (i), porém segundo as medidas de avaliação de corte em grafo de (ii) e levando em conta a ordem de propagação ao invés do custo. Apesar de não termos utilizado sementes externas em (ii), elas podem ser usadas como restrições fortes que limitam o espaço de busca das sementes internas por bordas de corte mínimo. Esse método implementado de forma diferencial (seção 7.3.1) nos proporcionaria uma ferramenta eficaz para a segmentação interativa de volumes 3D. Já o método de segmentação por poda de árvores possui uma vocação natural para aplicações automáticas, uma vez que a detecção dos pontos de vazamento para poda se dá de forma automática. Ferramentas para detecção interativa de pontos de vazamento como a apresentada em [22] são de difícil extensão para imagens multidimensionais. Logo os métodos (i)+(ii) e (iii) podem coexistir com diferentes enfoques, interativo e automático.

Com relação ao pré-processamento, diferentemente de outros trabalhos [66, 59, 41, 58, 72, 6] que o tratam de forma embutida, nós optamos por tratar o pré-processamento separadamente seguindo a estratégia de divisão e conquista. O uso de pré-processamento implícito leva a funções de dissimilaridade mais complexas [66, 59, 41, 58] para pesos de arestas (e.g. equações 4.3, 4.4 e 4.5), com impactos no desempenho, ou torna o grafo mais complexo [72, 6] tendo arestas adicionais com pesos dados por um processo de classificação interno (escondido).

Tratando o pré-processamento como um procedimento distinto temos as seguintes vantagens: conseguimos depurar melhor os programas evitando erros de implementação, podemos usar um pré-processamento específico mais adequado para cada aplicação, podemos usar funções de dissimilaridade mais simples e rápidas (e.g. equações 4.2, 5.7 e 5.8), é possível visualizar os resultados do pré-processamento (e.g. mapas de região e gradientes do capítulo 3) e nos permite um melhor entendimento sobre a real contribuição que cada etapa exerce sobre os resultados finais da segmentação.

Esse procedimento não foi explorado no capítulo 4 (ver tabela 4.2) porque a segmentação por competição entre sementes κ -conexas foi um dos primeiros trabalhos reali-

zados e também por causa das comparações feitas com abordagens tradicionais que usam funções de dissimilaridade complexas [66, 59, 41].

7.3 Sugestões para trabalhos futuros

7.3.1 DIFT combinada com medidas de corte em grafo

Consiste na extensão da *segmentação por crescimento ótimo de regiões e corte em grafo* discutida na seção 7.2 estendida de forma a permitir correções em tempo sublinear (veja [27]). Esse método pode então ser usado para a segmentação interativa 3D, sendo esperada uma redução do número de interações.

7.3.2 Automatizar o que é ainda manual

Métodos de pré-processamento automático e escolha automática de marcadores (sementes) são geralmente dependentes da aplicação e logo não foram o enfoque desse trabalho. Pesquisas futuras são necessárias para aprimorar os métodos de seleção de sementes aqui apresentados (seção 6.9.3), bem como para permitir a aplicação das técnicas aqui defendidas em outros domínios de aplicação.

7.3.3 Aplicações em medicina

A análise quantitativa de estruturas cerebrais a partir de imagens de ressonância magnética (RM) tem desempenhado um papel importante para a pesquisa em neurologia e pode ser muito útil no diagnóstico e tratamento de doenças relacionadas com alterações na anatomia do cérebro humano [52, 4, 10, 60]. Muito embora existam algumas técnicas propostas para segmentação de pequenas estruturas (e.g., hipocampos) com o mínimo de intervenção do usuário [8], técnicas de delineamento manual ainda são bastante adotadas [4, 5]. A tarefa é laboriosa e inviável para estudos em larga escala. Um possível trabalho futuro consiste em avaliar as técnicas propostas para a segmentação automática de estruturas cerebrais, de forma a permitir a análise em larga escala. Para isso é necessário investigar o pré-processamento mais adequado para cada estrutura. Como resultado, teríamos o avanço não só do estado da arte em ciência de computação, mas também da medicina.

Bibliografia

- [1] R.K. Ahuja, T.L. Magnanti, and J.B. Orlin. *Network Flows: Theory, Algorithms and Applications*. Prentice-Hall, Englewood Cliffs, NJ, 1993.
- [2] R. Audigier, R.A. Lotufo, and A. X. Falcão. On integrating iterative segmentation by watershed with tridimensional visualization of MRIs. In *Proc. of XVII Brazilian Symposium on Computer Graphics and Image Processing*, Curitiba, PR, Brazil, Oct 2004. IEEE. to appear.
- [3] S. Beucher and F. Meyer. The morphological approach to segmentation: The watershed transformation. In Edward R. Dougherty, editor, *In Mathematical Morphology in Image Processing*, chapter 12, pages 433–481. Marcel Dekker, Inc., New York, NY, 1993.
- [4] L. Bonilha. *Dano neuronal em pacientes com epilepsia do lobo temporal medial refratária a tratamento clínico: Estudo quantitativo por ressonância magnética*. PhD thesis, Faculty of Medical Sciences, State University of Campinas, Campinas, SP, Apr 2004. in Portuguese.
- [5] L Bonilha, E Kobayashi, G Castellano, G Coelho, E Tinois, F Cendes, and LM Li. Texture analysis of hippocampal sclerosis. *Epilepsia*, 44(12):1546–1550, 2003.
- [6] Yuri Boykov and Vladimir Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 26(9):1124–1137, Sep 2004.
- [7] Yuri Boykov, Olga Veksler, and Ramin Zabih. Fast approximate energy minimization via graph cuts. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 23(11):1222–1239, Nov 2001.
- [8] G. Bueno, O. Musse, F. Heitz, and J. P. Armspach. Three-dimensional segmentation of anatomical structures in MR images on large data bases. *Magnetic Resonance Imaging*, 19:73–88, 2001.

- [9] Julio Carballido-Gamio, Serge J. Belongie, and Sharmila Majumdar. Normalized cuts in 3D for spinal MRI segmentation. *IEEE Trans. on Medical Imaging*, 23(1):36–44, Jan 2004.
- [10] G Castellano, L Bonilha, LM Li, and F Cendes. Texture analysis in medical images. *Clinical Radiology*, in print, 2004.
- [11] G. Castellano, R.A. Lotufo, A.X. Falcão, and F. Cendes. Sulcal lines extraction for cortex characterization in cerebral MR images using the image foresting transform. In *XVI Brazilian Symposium on Computer Graphics and Image Processing*, pages 355–362, São Carlos, SP, 2003. IEEE.
- [12] L.D. Cohen. On active contour models and ballons. *Computer Vision, Graphics, and Image Processing: Image Understanding*, 53(2):211–218, 1991.
- [13] D.L. Collins, A.P. Zijdenbos, V. Kollokian, J.G. Sled, N.J. Kabani, C.J. Holmes, and A.C. Evans. Design and construction of a realistic digital brain phantom. *IEEE Trans. on Medical Imaging*, 17(3):463–468, June 1998.
- [14] T. Cootes, G. Edwards, and C.J. Taylor. Active appearance models. In *European Conference on Computer Vision (ECCV)*, volume 2, pages 484–498, 1998.
- [15] T. Cootes, C. Taylor, D. Cooper, and J. Graham. Active shape models – their training and application. *Computer Vision and Image Understanding*, 61(1):38–59, 1995.
- [16] I. J. Cox, S. B. Rao, and Y. Zhong. Ratio regions: a technique for image segmentation. In *Intl. Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 557–564, 1996.
- [17] B.S. da Cunha. Projeto de operadores de processamento e análise de imagens usando a transformada imagem-floresta. Master’s thesis, Instituto de Computação - UNICAMP, Jun 2001.
- [18] R.B. Dial. Shortest-path forest with topological ordering. *Communications of the ACM*, 12(11):632–633, Nov 1969.
- [19] E.W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1:269–271, 1959.
- [20] E.R. Dougherty and R.A. Lotufo. *Hands-on Morphological Image Processing*. SPIE Press, Bellingham, WA, 2003.

- [21] J.S. Duncan and N. Ayache. Medical image analysis: Progress over two decades and the challenges ahead. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22(1):85–105, 2000.
- [22] A. X. Falcão, F. P. G. Bergo, and P. A. V. Miranda. Image segmentation by tree pruning. In *Proc. of XVII Brazilian Symposium on Computer Graphics and Image Processing*, pages 65–71, Curitiba, PR, Brazil, Oct 2004. IEEE.
- [23] A. X. Falcão, P. A. V. Miranda, and F. P. G. Bergo. Automatic object detection by tree pruning. Technical Report IC-05-19, Institute of Computing, State University of Campinas, Sep 2005.
- [24] A. X. Falcão, P. A. V. Miranda, and A. Rocha. A linear-time approach for graph-cut object detection. Technical Report IC-05-31, Institute of Computing, State University of Campinas, Dec 2005.
- [25] A. X. Falcão, P. A. V. Miranda, A. Rocha, and F. G. Bergo. Object detection by κ -connected seed competition. In *Proc. of XVIII Brazilian Symposium on Computer Graphics and Image Processing*, Natal, RN, Brazil, Oct 2005. IEEE. to appear.
- [26] A. X. Falcão, P. A. V. Miranda, A. Rocha, and F. P. G. Bergo. Object definition by κ -connected components. Technical Report IC-05-18, Institute of Computing, State University of Campinas, Sep 2005.
- [27] A.X. Falcão and F.P.G. Bergo. Interactive volume segmentation with differential image foresting transforms. *IEEE Trans. on Medical Imaging*, 23(9):1100–1108, 2004.
- [28] A.X. Falcão, L.F. Costa, and B.S. da Cunha. Multiscale skeletons by image foresting transform and its applications to neuromorphometry. *Pattern Recognition*, 35(7):1571–1582, Apr 2002.
- [29] A.X. Falcão, B. S. da Cunha, and R. A. Lotufo. Design of connected operators using the image foresting transform. In *Proceedings of SPIE on Medical Imaging*, volume 4322, pages 468–479, San Diego, CA, Feb 2001.
- [30] A.X. Falcão, J. Stolfi, and R.A. Lotufo. The image foresting transform: Theory, algorithms, and applications. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 26(1):19–29, 2004.
- [31] A.X. Falcão and J.K. Udupa. A 3D generalization of user-steered live wire segmentation. *Medical Imaging Analysis*, 4(4):389–402, Dec 2000.

- [32] A.X. Falcão, J.K. Udupa, and F.K. Miyazawa. An ultra-fast user-steered image segmentation paradigm: Live-wire-on-the-fly. *IEEE Transactions on Medical Imaging*, 19(1):55–62, Jan 2000.
- [33] A.X. Falcão, J.K. Udupa, S. Samarasekera, S. Sharma, B.E. Hirsch, and R.A. Lotufo. User-steered image segmentation paradigms: Live-wire and live-lane. *Graphical Models and Image Processing*, 60(4):233–260, Jul 1998.
- [34] L. Ford and D. Fulkerson. *Flows in networks*. Princeton University Press, 1962.
- [35] C. Fowlkes, S. Belongie, and J. Malik. Efficient spatiotemporal grouping using the nystrom method. In *Intl. Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 231–238, 2001.
- [36] R.S.J. Frackowiak, K.J. Friston, C. Frith, R. Dolan, C.J. Price, S. Zeki, J. Ashburner, and W.D. Penny. *Human Brain Function*. Academic Press, 2nd edition, 2003.
- [37] G.A. Girdali, L.M.G. Goncalves, and A.A.F. Oliveira. Dual topologically adaptable snakes. In *(JCIS) Joint Conference on Information Sciences*, pages 103–106, 2000.
- [38] Gilson A. Girdali, Edilberto Strauss, and Antonio A. Oliveira. A boundary extraction method based on dual-t-snakes and dynamic programming. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 44–49, June 2000.
- [39] V. Grau, A.U.J. Mewes, M. Alcaniz, R. Kikinis, and S.K. Warfield. Improved watershed transform for medical image segmentation using prior information. *IEEE Trans. on Medical Imaging*, 23(4):447–458, Apr 2004.
- [40] D. Greig, B. Porteous, and A. Seheult. Exact maximum a posteriori estimation for binary images. *J. Royal Statistical Society, series B*, 51(2):271–279, 1989.
- [41] G.T. Herman and B.M. Carvalho. Multiseeded segmentation using fuzzy connectedness. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(5):460–474, May 2001.
- [42] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: active contour models. *International Journal of Computer Vision*, 1(4):321–331, 1987.
- [43] Vladimir Kolmogorov and Ramin Zabih. What energy functions can be minimized via graph cuts. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 26(2):147–159, Feb 2004.

- [44] T. Lei, J. K. Udupa, P. K. Saha, and D. Odhner. Artery-vein separation via MRA - An image processing approach. *IEEE Trans. on Medical Imaging*, 20(8), 2001.
- [45] H.C.G. Leitao and J. Stolfi. A multiscale method for the reassembly of two-dimensional fragmented objects. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(9):1239–1251, Sep 2002.
- [46] R.A. Lotufo and A.X. Falcão. The ordered queue and the optimality of the watershed approaches. In *Mathematical Morphology and its Applications to Image and Signal Processing*, volume 18, pages 341–350. Kluwer Academic Publishers, Palo Alto, USA, Jun 2000.
- [47] R.A. Lotufo, A.X. Falcão, and F.A. Zampiroli. IFT-watershed from gray scale marker. In *Proceedings of XV Brazilian Symposium on Computer Graphics and Image Processing*, pages 146–152, Fortaleza, CE, Oct 7–10 2002. IEEE Press.
- [48] J.L. Marroquin, B.C. Vemuri, S. Botello, E. Calderon, and A. Fernandez-Bouzas. An accurate and efficient bayesian method for automatic segmentation of brain MRI. *IEEE Trans. on Medical Imaging*, 21(8):934–945, Aug 2002.
- [49] A. Meijster and M. H. F. Wilkinson. A comparison of algorithms for connected set openings and closings. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(4):484–494, Apr 2002.
- [50] F. Meyer. Topographic distance and watershed lines. *Signal Processing*, 38:113–125, 1994.
- [51] G. Moonis, J. Liu, J. K. Udupa, and D. B. Hackney. Estimation of tumor volume with fuzzy-connectedness segmentation of MR images. *American Journal of Neuro-radiology*, 23:356–363, Mar 2002.
- [52] J. Natsume and et al. MRI volumetry of the thalamus in temporal, extratemporal, and idiopathic generalized epilepsy. *Neurology*, 60:1296–1300, 2003.
- [53] H. T. Nguyen, M. Worring, and R. van den Boomgaard. Watersnakes: energy-driven watershed segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(3):330–342, Mar 2003.
- [54] L.G. Nyúl, A.X. Falcão, and J.K. Udupa. Fuzzy-connected 3D image segmentation at interactive speeds. In *Proceedings of SPIE on Medical Imaging*, volume 3979, pages 212–223, San Diego, CA, Feb 2000.

- [55] A. Oliveira, S. Ribeiro, C. Esperanca, and G. Giraldi. Loop snakes: the generalized model. In *Information Visualisation. Proceedings. Ninth International Conference on*.
- [56] A. Oliveira, S. Ribeiro, R. Farias, and C. Esperanca. Loop snakes: snakes with enhanced topology control. In *Computer Graphics and Image Processing. Proceedings. 17th Brazilian Symposium on*. IEEE.
- [57] N. Otsu. A threshold selection method from gray level histograms. *IEEE Trans. Systems, Man and Cybernetics*, 9:62–66, March 1979.
- [58] P. Saha and J. K. Udupa. Scale-based fuzzy connected image segmentation: Theory, algorithms and validation. *Computer Vision and Image Understanding*, 77(2):145–174, 2000.
- [59] P.K. Saha and J.K. Udupa. Relative fuzzy connectedness among multiple objects: theory, algorithms, and applications in image segmentation. *Computer Vision and Image Understanding*, 82:42–56, 2001.
- [60] E.K. Sandock and et al. Significance of cerebellar atrophy in intractable temporal lobe epilepsy: a quantitative MRI study. *Epilepsia*, 41:1315–1320, 2000.
- [61] S. Sarkar and K. L. Boyer. Quantitative measures of change based on feature organization: eigenvalues and eigenvectors. In *Intl. Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 478–483, 1996.
- [62] Jiambo Shi and Jitendra Malik. Normalized cuts and image segmentation. In *Intl. Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 731–737, 1997.
- [63] Jiambo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22(8):888–905, Aug 2000.
- [64] R.S. Torres, A.X. Falcão, and L.F. Costa. Shape description by image foresting transform. In *14th International Conference on Digital Signal Processing*, pages 1089–1092, Santorini, Greece, Jul 2002. IEEE Press.
- [65] R.S. Torres, A.X. Falcão, and L.F. Costa. A graph-based approach for multiscale shape analysis. *Pattern Recognition*, 37(6):1163–1174, 2004.
- [66] J.K. Udupa and S. Samarasekera. Fuzzy connectedness and object definition: theory, algorithms, and applications in image segmentation. *Graphical Models and Image Processing*, 58:246–261, 1996.

- [67] L. Vincent and P. Soille. Watersheds in digital spaces: An efficient algorithm based on immersion simulations. *IEEE on Pattern Analysis and Machine Intelligence*, 13(6), Jun 1991.
- [68] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Intl Conf. on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages I-511–I-518, 2001.
- [69] Song Wang and Jeffrey Mark Siskind. Image segmentation with ratio cut. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 25(6):675–690, Jun 2003.
- [70] Song Wang and Jeffrey Mark Siskind. Image segmentation with minimum mean cut. In *Intl. Conf. on Computer Vision (ICCV)*, volume 1, pages 517–525, Jul 2001.
- [71] Z. Wu and R. Leahy. An optimal graph theoretic approach to data clustering: theory and its applications to image segmentation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 15(11):1101–1113, Nov 1993.
- [72] Marie-Pierre Jolly Yuri Y. Boykov. Interactive graph cuts for optimal boundary & region segmentation of objects in N-D images. In *Intl. Conf. on Computer Vision (ICCV)*, volume 1, pages 105–112, 2001.
- [73] D. Zheng, Y. Zhao, and J. Wang. An efficient method of license plate location. *Pattern Recognition Letters*, 2005. Article in press, available at www.sciencedirect.com.