

# Reconhecimento e Delineamento Sinérgicos de Objetos em Imagens com Aplicações na Medicina

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

Campinas, 4 de Novembro de 2009.

  
Prof. Dr. Alexandre X. Falcão (Orientador)

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

**FICHA CATALOGRÁFICA ELABORADA PELA  
BIBLIOTECA DO IMECC DA UNICAMP**  
Bibliotecária: Maria Fabiana Bezerra Müller – CRB8 / 6162

Miranda, Paulo André Vechiatto de  
M672r Reconhecimento e delineamento sinérgicos de objetos em imagens  
com aplicações na medicina/Paulo André Vechiatto de Miranda --  
Campinas, [S.P. : s.n.], 2009.

Orientador : Alexandre Xavier Falcão

Tese (doutorado) - Universidade Estadual de Campinas, Instituto de  
Computação.

1.Análise de imagem. 2.Algoritmos. 3.Estruturas de dados  
(Computação). 4.Ressonância magnética. 5.Processamento de imagens .  
I. Falcão, Alexandre Xavier. II. Universidade Estadual de Campinas.  
Instituto de Computação. III. Título.

Título em inglês: Synergistic delineation and recognition of objects in images with  
applications in medicine

Palavras-chave em inglês (Keywords): 1. Image analysis. 2. Algorithms. 3. Data structures  
(Computer science). 4. Magnetic resonance. 5. Image processing.

Área de concentração: Ciência da Computação

Titulação: Doutor em Ciência da Computação

Banca examinadora: Prof. Dr. Alexandre Xavier Falcão (IC-UNICAMP)  
Prof. Dr. Roberto Marcondes Cesar Junior (IME – USP)  
Prof. Dr. Marcel P. Jackowski (IME – USP)  
Prof. Dr. Hélio Pedrini (IC – UNICAMP)  
Prof. Dr. Siome Klein Goldenstein (IC - UNICAMP)

Data da defesa: 04/11/2009

Programa de Pós-Graduação: Doutorado em Ciência da Computação

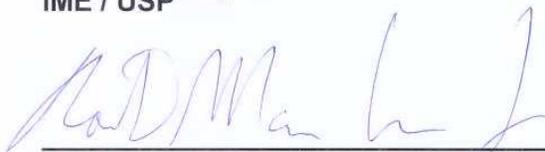
## TERMO DE APROVAÇÃO

Tese Defendida e Aprovada em 04 de novembro de 2009, pela Banca examinadora composta pelos Professores Doutores:



---

**Prof. Dr. Marcel Parolin Jackowski**  
IME / USP



---

**Prof. Dr. Roberto Marcondes Cesar Junior**  
IME / USP



---

**Prof. Dr. Hélio Pedrini**  
IC / UNICAMP



---

**Prof. Dr. Siome Klein Goldenstein**  
IC / UNICAMP



---

**Prof. Dr. Alexandre Xavier Falcão**  
IC / UNICAMP

# Reconhecimento e Delineamento Sinérgicos de Objetos em Imagens com Aplicações na Medicina

Paulo André Vechiatto de Miranda<sup>1</sup>

Novembro de 2009

## Banca Examinadora:

- Prof. Dr. Alexandre X. Falcão (Orientador)
- Prof. Dr. Roberto Marcondes Cesar Junior  
IME – Universidade de São Paulo
- Prof. Dr. Marcel P. Jackowski  
IME – Universidade de São Paulo
- Prof. Dr. Hélio Pedrini  
IC – Universidade Estadual de Campinas
- Prof. Dr. Siome Klein Goldenstein  
IC – Universidade Estadual de Campinas
- Prof. Dr. Ricardo da Silva Torres (Suplente)  
IC - Universidade Estadual de Campinas
- Prof. Dr. Sergio Shiguemi Furuie (Suplente)  
Escola Politécnica – Universidade de São Paulo

---

<sup>1</sup>Suporte financeiro de: Bolsa FAPESP 05/59808-0, Bolsa PED Unicamp 2007–2008.

# Resumo

Segmentar uma imagem consiste em particioná-la em regiões relevantes para uma dada aplicação (e.g., objetos e fundo). A segmentação de imagem é um dos problemas mais fundamentais e desafiadores em processamento de imagem e visão computacional. O problema da segmentação representa um desafio técnico importante na computação devido à dificuldade da máquina em extrair informações globais sobre os objetos nas imagens (e.g., forma e textura) contando apenas com informações locais (e.g., brilho e cor) dos pixels.

Segmentação de imagens envolve o *reconhecimento* de objetos e o *delineamento*. O reconhecimento é representado por tarefas cognitivas que determinam a localização aproximada de um objeto desejado em uma determinada imagem (detecção de objeto), e identificam um objeto desejado de entre uma lista de objetos candidatos (classificação de objeto). Já o delineamento consiste em definir de forma precisa a extensão espacial do objeto de interesse. No entanto, métodos de segmentação efetivos devem explorar essas tarefas de forma sinérgica. Esse tema constitui o foco central deste trabalho que apresenta soluções interativas e automáticas para segmentação. A automação é obtida mediante o uso de modelos discretos que são criados por aprendizado supervisionado. Esses modelos empregam reconhecimento e delineamento de uma maneira fortemente acoplada pelo conceito de *Clouds*. Estes modelos são demonstrados no âmbito da neurologia para a segmentação automática do cérebro (sem o tronco cerebral), do cerebelo, e de cada hemisfério cerebral a partir de imagens de ressonância magnética. Estas estruturas estão ligadas em várias partes, o que impõe sérios desafios para a segmentação. Os resultados indicam que estes modelos são ferramentas rápidas e precisas para eliminar as intervenções do usuário ou, pelo menos, reduzi-las para simples correções, no contexto da segmentação de imagens do cérebro.

# Abstract

Segmenting an image consists of partitioning it into regions relevant for a given application (e.g., objects and background). The image segmentation is one of the most fundamental and challenging problems in image processing and computer vision. The segmentation problem represents a significant technical challenge in computer science because of the difficulty of the machine in extracting global informations about the objects in the images (e.g., shape and texture) counting only with local information (e.g., brightness and color) of the pixels.

Image segmentation involves object *recognition* and *delineation*. Recognition is represented by cognitive tasks that determine the approximate location of a desired object in a given image (object detection), and identify a desired object among candidate ones (object classification), while delineation consists in defining the exact spatial extent of the object. Effective segmentation methods should exploit these tasks in a synergistic way. This topic forms the central focus of this work that presents solutions for interactive and automatic segmentation. The automation is achieved through the use of discrete models that are created by supervised learning. These models employ recognition and delineation in a tightly coupled manner by the concept of *Clouds*. We demonstrate their usefulness in the automatic MR-image segmentation of the brain (without the brain stem), the cerebellum, and each brain hemisphere. These structures are connected in several parts, imposing serious challenges for segmentation. The results indicate that these models are fast and accurate tools to eliminate user's intervention or, at least, reduce it to simple corrections, in the context of brain image segmentation.

# Agradecimentos

A todos que contribuíram diretamente no desenvolvimento deste trabalho: Prof. Dr. Alexandre Falcão (orientador) e o corpo docente do Instituto de Computação da Unicamp; Prof. Dr. Fernando Cendes e todos os pesquisadores do Laboratório de Neuro-Imagem do Hospital das Clínicas da Unicamp; Prof. Dr. Jayaram K. Udupa, Dr. K. Chris Ciesielski e demais pesquisadores do Departamento de Radiologia da Universidade da Pensilvânia.

A todos os colegas do Laboratório de Informática Visual do Instituto de Computação (LIV-IC) da Unicamp; e a todos colaboradores do Grupo de Processamento de Imagens Médicas (MIPG) da Universidade da Pensilvânia.

À minha família por todo o apoio e motivação e a Deus que colocou no meu caminho as pessoas certas nas horas certas.

À FAPESP e à Unicamp pelo apoio financeiro.

# Sumário

<b>Resumo</b>	<b>vii</b>
<b>Abstract</b>	<b>ix</b>
<b>Agradecimentos</b>	<b>xi</b>
<b>1 Introdução</b>	<b>1</b>
1.1 Introdução à segmentação de imagens . . . . .	1
1.2 Motivação . . . . .	3
1.3 Objetivos . . . . .	3
1.4 Visão Geral das Contribuições . . . . .	4
1.5 Organização da Tese . . . . .	8
1.5.1 Capítulo 2 . . . . .	9
1.5.2 Capítulo 3 . . . . .	9
1.5.3 Capítulo 4 . . . . .	10
1.5.4 Capítulo 5 . . . . .	10
<b>2 Synergistic Arc-Weight Estimation for Interactive Image Segmentation using Graphs</b>	<b>11</b>
2.1 Introduction . . . . .	11
2.2 Basic Concepts on Image Graphs . . . . .	13
2.3 Synergistic arc-weight estimation . . . . .	14
2.3.1 Object-based weight assignment . . . . .	15
2.3.2 Image-based weight assignment . . . . .	20
2.4 Interactive segmentation methods . . . . .	21
2.4.1 Image Foresting Transform . . . . .	21
2.4.2 Segmentation by differential IFT (DIFT) . . . . .	23
2.4.3 Segmentation by $\kappa$ -connected components . . . . .	25
2.4.4 Segmentation by tree pruning . . . . .	25
2.4.5 Segmentation by graph cut . . . . .	26

2.4.6	Segmentation by IFT with graph cut . . . . .	28
2.4.7	Segmentation by live wire . . . . .	29
2.5	Experiments and Results . . . . .	30
2.6	Conclusion . . . . .	38
<b>3</b>	<b>Links Between Image Segmentation based on Optimum-Path Forest and Minimum Cut in Graph</b>	<b>39</b>
3.1	Introduction . . . . .	39
3.2	Basic Concepts on Image Graphs . . . . .	40
3.3	Image Foresting Transform (IFT) . . . . .	42
3.4	Region-based segmentation using IFT . . . . .	43
3.4.1	IFT with only internal seeds . . . . .	44
3.4.2	IFT with internal and external seeds . . . . .	45
3.5	Graph-cut segmentation . . . . .	48
3.6	IFT-SC as a graph-cut approach . . . . .	51
3.7	IFT-CT as a graph-cut approach . . . . .	57
3.8	The link with the min-cut/max-flow algorithm . . . . .	59
3.9	Comparative analysis between the main paradigms . . . . .	63
3.10	Conclusions . . . . .	65
<b>4</b>	<b>Object delineation by <math>\kappa</math>-connected components</b>	<b>67</b>
4.1	Introduction . . . . .	67
4.2	Background . . . . .	69
4.3	Image segmentation by $\kappa$ -connectivity . . . . .	71
4.3.1	Object definition without seed competition . . . . .	74
4.3.2	Object definition with seed competition . . . . .	75
4.4	Algorithms . . . . .	77
4.4.1	Automatic computation of $\kappa_s$ . . . . .	78
4.4.2	Interactive computation of $\kappa_s$ . . . . .	81
4.5	Evaluation . . . . .	84
4.6	Conclusions . . . . .	88
<b>5</b>	<b>Cloud Models: Their Construction and Employment in Automatic MRI Segmentation of the Brain</b>	<b>91</b>
5.1	Introduction . . . . .	91
5.2	Object Cloud Models . . . . .	93
5.2.1	Grouping . . . . .	97
5.2.2	Locating objects . . . . .	98
5.2.3	Arc-weight estimation for delineation . . . . .	98

5.2.4	Delineation algorithm . . . . .	99
5.2.5	Functional for recognition . . . . .	100
5.3	Application to brain segmentation . . . . .	101
5.3.1	Preprocessing & grouping . . . . .	101
5.3.2	Arc-weight estimation . . . . .	102
5.3.3	Recognition functional & delineation algorithm . . . . .	103
5.3.4	Multiple object search . . . . .	104
5.4	Results . . . . .	105
5.5	Conclusion . . . . .	106
<b>6</b>	<b>Conclusões e Trabalhos Futuros</b>	<b>109</b>
6.1	Contribuições & Desafios . . . . .	109
6.2	Outras Contribuições . . . . .	111
6.3	Extensões Futuras . . . . .	112
	<b>Bibliografia</b>	<b>113</b>

# Lista de Tabelas

2.1	Description, imaging modality and number of slice images for each object used in the experiments. Objects O2 and O3 use sample slices from phantoms with different degrees of noise (N) and inhomogeneity (INU). . .	35
2.2	Segmentation results using DIFT with $w_o$ ( $\lambda = 0.5$ ). . . . .	36
2.3	Segmentation results using DIFT without $w_o$ ( $\lambda = 0$ ). . . . .	37
4.1	Description, imaging modality and number of slices for each object used in the experiments. . . . .	85
4.2	The dissimilarity functions used for each combination of object and method.	87
4.3	The percentages of different $\kappa_s$ values (PDK) in M1, the average numbers of user interactions for each object and method, and the average similarity values between different methods for a same object. . . . .	89
4.4	Average numbers of internal seeds (NIS), interactive $\kappa_s$ detections (IKD), external seeds (NES), and automatic $\kappa_s$ detections (AKD). . . . .	89
5.1	Mean (%) and standard deviation (%) of the Dice similarity. . . . .	105

# Lista de Figuras

1.1	Diagrama do sistema de segmentação proposto. . . . .	5
1.2	(a) Uma fatia de RM com marcadores para treinamento. (b) Visualização dos pesos das arestas inicial. (c) Um considerável realce é obtido para a substância branca através de aprendizado supervisionado dos marcadores. .	6
1.3	Avanços teóricos são obtidos pela análise das propriedades dos algoritmos nos grafos derivados das imagens. (a) Um grafo com vizinhança 4 a ser segmentado tendo duas restrições fortes (pontos maiores), onde o ponto branco deve pertencer ao objeto e o ponto maior preto deve pertencer ao fundo. (b-c) Dois possíveis resultados de segmentação. (d) A solução por floresta de caminhos ótimos conforme será discutido no Capítulo 3. . . . .	7
1.4	(a) Um sistema de <i>Clouds</i> para três objetos mostrado no espaço de cores RGB. (b-c) Valores obtidos pelo critério de reconhecimento para diversas posições do modelo sobre uma imagem de entrada e o resultado final de segmentação. . . . .	8
2.1	Euclidean adjacency relations for 2D images: (a) 4-neighborhood ( $\rho = 1$ ), (b) 8-neighborhood ( $\rho = \sqrt{2}$ ), and (c) extended adjacency to the 12 closest neighbors ( $\rho = 2$ ). . . . .	13
2.2	(a) A 2D image graph with 4-adjacent pixels $s$ and $t$ . (b) An extended graph obtained by adding two terminal nodes ( <i>source</i> $o$ and <i>sink</i> $b$ ), which represent object and background respectively. . . . .	14
2.3	(a) An image with markers selected inside and outside the object. (b) The weight image $\hat{W}$ considering only the image-based component ( $\lambda = 0$ ), and (c) by combining it with the object-based weight ( $\lambda = 0.8$ ), $c = 2$ . . . . .	16
2.4	Image $\hat{M}_1$ , where the desired object is a family of bears. (a-b) An initial marker selection and the corresponding membership image $\hat{M}_1$ . (c-d) The estimation improves as the user adds internal markers on the dark regions of the object and external markers on the bright regions of the background in $\hat{M}_1$ . . . . .	17

2.5	(a) Path $\pi_t = \pi_s \cdot \langle s, t \rangle$ indicates the extension of path $\pi_s$ by an arc $(s, t) \in \mathcal{A}$ . (b) A 4-neighborhood graph showing a path $\pi_t$ (dashed line) represented in backwards, where $P(t)$ is the predecessor node of $t$ and $R(t)$ is the root pixel. (c) A forest $P$ with two root nodes $r_1$ and $r_2$ . . . . .	22
2.6	(a) An initial marker selection for segmentation. (b-d) The IFT region growing. (d) The regions meet each other at the object's boundary. . . . .	23
2.7	(a) A weight image $\hat{W}$ obtained from Figure 2.4 using $\lambda = 0.4$ . (b) An initial marker selection and segmentation. (c) An additional seed is inserted in order to include the other bears. (d) The final segmentation after some small corrections. . . . .	24
2.8	(a-b) The IFT region growing from internal seeds. There is a burst in the size of the wavefront when an optimum path reaches the background. (c) The total wavefront area for each optimum-path value $u = 1, 2, \dots, K$ during propagation. (d) The resulting segmentation with $\kappa$ -connected components. . . . .	26
2.9	Example of license plate segmentation. (a) The original image overlaid by the number of descendants in the background set $\mathcal{B}$ . (b) The resulting segmentation with tree pruning. . . . .	27
2.10	Graph-cut segmentation with $\alpha = 20$ and $\lambda = 0.5$ . (a) Marker selection for training and the result of segmentation. (b) The membership image $\hat{M}_1$ used in $w(o, s)$ and $w(s, b)$ . (c) The weight image $\hat{W}$ that reflects $w(s, t)$ . . . . .	28
2.11	Segmentation example of a MR-brain image. (a) The normalized cut versus the pixel propagation order. (b) The respective segmentation. . . . .	29
2.12	Contour tracking with live wire. (a) Initial point $s_1$ is selected on the boundary and the user moves the mouse. (b) A second point $s_2$ is selected on the boundary. (c-d) Final contour with 7 segments. . . . .	30
2.13	(a) Segmentation result of the watershed transform from markers considering only the traditional image-based component (Figure 2.3b). (b) A better result is obtained with less seeds, by the use of object-based weights (Figure 2.3c). . . . .	31
2.14	(a) DIFT segmentation using the same markers for training and delineation. (b-c) The respective membership image $\hat{M}_1$ and weight image $\hat{W}$ of the training ( $\lambda = 0.5$ ). (d) The correct segmentation is obtained with additional markers, which should never be used to recompute weights. (e) The membership image $\hat{M}_1$ is destroyed if we recompute weights from the additional markers, affecting (f) the result of segmentation. . . . .	33

2.15	(a) The same training markers are used to delineate the left caudate nucleus by the DIFT algorithm. Additional markers are needed to refine segmentation. (b-c) The corresponding membership image $\hat{M}_1$ and weight image $\hat{W}$ for $\lambda = 0.5$ . (d-f) Segmentations by $\kappa$ -connected component, live wire and graph cut. . . . .	34
2.16	Sample slice images showing objects O1, O3 and O4. Object O2 is the same as object O3, but on images with less noise and inhomogeneity. . . .	35
2.17	(a) Sample slice of object O2 with training markers. (b-c) The corresponding weight images $\hat{W}$ for $\lambda = 0$ and $\lambda = 0.5$ , respectively. By using $w_o$ a suitable boundary enhancement is obtained for the white matter.	37
2.18	(a) Sample slice of object O1 with training markers. (b) The corresponding membership image $\hat{M}_1$ . (c) Graph-cut segmentation with $\alpha = 40$ and $\lambda = 0.5$ . The background bones are easily removed using 4 markers with a larger brush. . . . .	37
3.1	(a) Path $\pi_t = \pi_s \cdot \langle s, t \rangle$ indicates the extension of path $\pi_s$ by an arc $(s, t)$ . (b) A 4-neighborhood graph showing a path $\pi_t$ (dashed line) represented in backwards, where $P(t)$ is the predecessor node of $t$ and $R(\pi_t)$ is the root pixel. (c) A spanning forest $P$ with two root nodes, $r_1$ and $r_2$ . . . . .	41
3.2	(a) A 4-neighborhood graph, where the numbers indicate the arc weights and the object is the <i>shaded square</i> . Two seeds are selected inside the object (bigger white dots). (b) An optimum-path forest for the path-value function defined via Eq. 3.2. The numbers inside the nodes indicate the values of the optimum paths (Eq. 3.1). The object is obtained as the white dots after using a threshold $\kappa = 3$ . . . . .	45
3.3	(a) A 4-neighborhood graph, where the numbers indicate the arc weights and the object is the <i>shaded square</i> . Four seeds are selected, two are inside the object (white dots) and two are in the background (bigger black dots). (b) An optimum-path forest for the path-value function defined by Eq. 3.2. The numbers inside the nodes indicate the values of the optimum paths (Eq. 3.1). The label $L(s) = 0$ (black), or $L(s) = 1$ (white) of each seed $s$ is propagated to all pixels within its respective optimum-path tree. . . . .	46

3.4	(a) A 4-neighborhood graph with one object seed (white dot) and one background seed (bigger black dot). (b-c) Two possible optimum-path forests using Eq. 3.2. The numbers inside the nodes indicate the values of the optimum paths (Eq. 3.1). (d) Two tie zones having values 2 and 4 are shown in gray. The pixel (0, 3) with optimum value 0 is not a tie-zone pixel, since it is unambiguously surrounded by a black influence zone. Note that, any path leading to this node from the white dot also has value 0, but these paths are disregarded since they are not complete optimum. . . .	47
3.5	(a) An image of peppers. (b-c) The probability maps $\mathcal{P}_o$ and $\mathcal{P}_b$ for the bigger pepper at the top. (d) The segmentation by graph-cut using Eq. 3.5 with $\lambda = 2$ . (e-f) Graph-cut results using Eq. 3.6 with increasing power values, 2 and 5, respectively. . . . .	51
3.6	The dashed line represents a segmentation by an optimum-path forest with the arc $(a, b)$ in its cut boundary. The sets of nodes $A$ and $B$ represents internal and external seeds respectively. . . . .	52
3.7	A 1D example. All pixels in the tie zone are reached by paths from $A$ and $B$ with the same optimum value 3. If this tie zone is labeled to the background (set $B$ ) then $a$ is reached from $A$ with $f_{\min}(\pi_a) = 6$ and $b$ is reached from $B$ with $f_{\min}(\pi_b) = 3$ . Note that $w(a, b) = 3$ and the Equation 3.10 is valid ( $3 < 6$ and $3 \leq 3$ ). . . . .	53
3.8	(a) A 4-neighborhood graph with one internal seed (white dot) and one external seed (bigger black dot). (b-c) Two optimum segmentation candidates for $E_3$ (Eq. 3.11) satisfying the hard constraints. (d) The chosen solution by optimum-path forest using Eq. 3.2, where the numbers inside the nodes indicate the values of the optimum paths (Eq. 3.1). . . . .	55
3.9	Two possible optimum cut boundaries are shown, both with optimum value in Equation 3.11 given by $w(a, b)$ . The dashed lines represent the boundary pieces at issue, with greatest weights given by $w(u_1, v_1)$ and $w(u_2, v_2)$ respectively. . . . .	56
3.10	Two possible segmentation results on the same 4-neighborhood graph, both satisfying the hard constraints (bigger white and black dots). (a) The optimum solution with minimum mean cut $\frac{(4 \times 5 + 8 \times 1)}{12} = 2.33$ . (b) The second solution has worse mean cut $\frac{(4 \times 5 + 4 \times 0)}{8} = 2.50$ . Since $\frac{4 \times 0}{4} < \frac{8 \times 1}{8}$ we may conclude that this graph-cut measure is not piecewise optimum. . . .	57

3.11 (a) A 4-neighborhood graph with one internal seed (white dot) and one external seed (bigger black dot). (b-c) The IFT-CT results from the internal seed with $\kappa = 4$ and $\kappa = 3$ , respectively. Note that, $\kappa = 4$ gives the IFT-CT result with lowest $E_3$ value satisfying all initial hard constraints ( $E_3(\hat{L}) = 3$ ). (d) The result by IFT-SC also has $E_3(\hat{L}) = 3$ . However, it is quite different from the IFT-CT due to the piecewise optimum property (Theorem 2). . . . .	59
3.12 (a) A gradient image of the ventricles with thick edges. (b) The result of IFT-CT by the selection of the best $\kappa$ . (c) An improved result is obtained by IFT-SC. (d) The XOR operation of the labels (in black) indicates a considerable divergence. . . . .	60
3.13 Two nested cut boundaries $\mathbf{C}_1$ and $\mathbf{C}_2$ , with the same optimum cut value $\tilde{E}_3$ (Eq. 3.22) are shown. The weights $\tilde{w}(a_1, b_1)$ and $\tilde{w}(a_2, b_2)$ are the maximum within their respective boundaries. The shaded area is a tie zone since $\tilde{w}(a_1, b_1) = \tilde{w}(a_2, b_2)$ . . . . .	63
3.14 (a) The segmentation by IFT-SC using Eq. 3.2. (b-d) Results of the min-cut/max-flow algorithm with increasing power values, 1,3 and 7 respectively. . . . .	64
3.15 (a) A synthetic gradient image with gaps. (b) The result of min-cut/max-flow algorithm. (c) The segmentation by IFT-SC using Eq. 3.2.	64
4.1 (a) The main elements of a spanning forest with two roots, where the thicker path indicates $P^*(q)$ . (b) An image graph with 4-adjacency, where the integers are the image values $I(p)$ and the bigger dots indicate two seeds. One is inside the brighter rectangle and one is in the darker background outside it. Note that, the background also contains brighter parts. (c) An optimum-path forest for $f_{\max}$ , with $\delta(p, q) =  I(q) - I(p) $ . The integers are the cost values and the rectangle is obtained as an optimum-path tree rooted at the internal seed. . . . .	70
4.2 An MR-T1 image of the brain with one seed $s$ inside the cortex. (a-b) The maximum influence zones of $s$ within the cortex for $f_{\max}$ with $\delta_3$ and with $\delta_6$ , respectively. The asymmetry of $\delta_6$ favors segmentation in anti-clockwise orientation, increasing the influence zone of $s$ . . . . .	73

4.3	A CT image of a knee where the patella can be segmented with two seed pixels, $s_1$ and $s_2$ , $f_{max}$ with $\delta_3$ , and without seed competition. (a) The result with a single value of $\kappa$ for both seeds. (b) The segmentation with a single value of $\kappa$ fails when we change the position of $s_1$ , because $s_1$ requires a higher value of $\kappa$ to get the brighter part of the bone and $B$ invades the background at this higher value of $\kappa$ . (c) The result can be corrected with distinct values of $\kappa$ for each seed. . . . .	75
4.4	(a) An image graph with 4-adjacency, where the integers are the image values $I(p)$ and the bigger dot is an arbitrary pixel. The object of interest is the brighter rectangle in the center. (b) A minimum-spanning tree computed from the arbitrary pixel, where the integers for each pixel $q$ are the arc weights $\delta(p, q) =  I(q) - I(p) $ , for $p = P(q)$ . (c) The minimum-spanning tree without arc orientation. A single seed $s_1$ can not extract the rectangle for any value of $\kappa$ . (d) The rectangle can be obtained with two seeds and distinct values of $\kappa$ , $s_1$ with $\kappa_1 = 3$ and $s_2$ with $\kappa_2 = 2$ . . . . .	76
4.5	A CT image of the orbital region where the eye ball is obtained by seed competition. (a) One internal seed and many external seeds are required for segmentation, using $f_{max}$ with $\delta_4$ . (b) The segmentation fails when some of the external seeds are removed. (c) A value of $\kappa$ is used to limit the influence zone of the internal seed in parts where the seed competition fails. . . . .	77
4.6	A CT image of the orbital region with one seed inside the eye ball. (a) A wavefront of cost $\kappa$ which represents the maximum extent of this seed inside the eye ball. (b) The wavefront of cost $\kappa + 1$ shows a considerable augment in size when it invades the background. (c) The pixel propagation order provides more continuous transitions of the wavefronts to select $\kappa$ , interactively. . . . .	79
4.7	Segmentation of a caudate nucleus with two internal seeds, $A$ and $B$ . (a) The leaking occurs before the object be filled. (b) The moment when $\kappa_A = 324$ is detected. (c) The instant when $\kappa_B$ is detected. (d) Final segmentation. . . . .	82
4.8	(a)-(g) Results of slice segmentation of the objects from 1 to 7, respectively, overlaid with the pre-processed images. . . . .	86
4.9	(a) A segmentation of the left caudate nucleus. (b) The result of dilating the binary image with a circular structuring element of radius 1. The similarity value between these two masks is 0.87. The differences in the experiments using distinct methods on this object (O2) are less significant than this small dilation. . . . .	88

5.1	(a) A coronal slice of the 3D cloud image of the cerebellum. (b) The shape-based weight image. (c) Example of an uncertainty region over a slice of a test image. . . . .	94
5.2	(a) An input image. (b) The recognition score for all positions. (c) The final segmentation at the best location. . . . .	95
5.3	Hierarchical search for the brain structures S1-S5: S1 is the brain without the brain stem, S2 represents the cerebrum, S3 is the cerebellum, and S4 and S5 are the right and left cerebral hemispheres. . . . .	95
5.4	(a) A cloud system for $c = 3$ objects shown in RGB color space. (b) The recognition score for all positions and (c) the final segmentation handling multiple objects simultaneously. . . . .	96
5.5	(a) The training instances ( $c = 1$ ) are mapped as nodes of a complete graph, (b) the groups are selected as maximal cliques. . . . .	97
5.6	IFT-SC example: (a) A 4-neighborhood graph, where the numbers indicate the arc weights. Three seeds are selected, one is internal (white dot) and two are external (black dots). (b) An optimum-path forest for the path-cost function $f_1$ . The numbers inside the nodes indicate the costs of the optimum paths, which are stored in a predecessor map $P$ . The label $L(q) = 0$ (black), or $L(q) = l$ (white) of each seed $q$ is propagated to all pixels within its respective optimum-path tree. . . . .	100
5.7	(a) The cerebellum is connected to the cerebrum through the brain stem. (b) The cerebral hemispheres are connected through the corpus callosum. .	101
5.8	(a) The image-based weight $W_i$ , (b) the object-based weight $W_o$ . . . . .	102
5.9	Segmentation results by <i>Cloud System Model</i> and the provided ground truth, shown as 3D renditions. . . . .	106
5.10	Sample slices from control subjects with segmentation results. . . . .	107
5.11	Sample slices from patients with segmentation results. . . . .	108

# Capítulo 1

## Introdução

### 1.1 Introdução à segmentação de imagens

Segmentar uma imagem consiste em particioná-la em regiões relevantes para uma dada aplicação. Comumente, objetos de interesse devem ser isolados dentro do domínio da imagem, a fim de permitir a obtenção de informações significativas para uma determinada aplicação. A segmentação de imagem é um dos problemas mais fundamentais e desafiadores em processamento de imagem e visão computacional. Como consequência, a segmentação da imagem tem muitas aplicações práticas, incluindo: percepção autônoma de máquina, reconhecimento de caracteres, inspeção industrial automática e sistemas de controle de tráfego.

Em publicidade, a segmentação de imagens naturais é usada na edição de fotos e vídeo, tal como para extrair objetos de primeiro plano de uma imagem e compô-los em um novo fundo [8, 122]. Em imagens médicas, a grande variedade de tecnologias de imagem (por exemplo, tomografia computadorizada, tomografia por emissão de pósitrons, angiografia por ressonância magnética, ressonância magnética, ressonância magnética funcional, SPECT, ultra-som) promoveu a prática da segmentação de imagem na medicina [45, 18, 63, 90]. Quando as imagens são adquiridas para uma região do corpo humano, geralmente há um sistema de objetos em estudo, que pode ser simplesmente um conjunto de órgãos dentro dessa região. Esses órgãos podem ser isolados por métodos de segmentação de imagem, a fim de cumprir os seguintes objetivos: medição de volume de tecidos e outras informações quantitativas; o diagnóstico de uma doença, bem como o planejamento do seu tratamento; a visualização de estruturas anatômicas; e educação médica.

A principal dificuldade da segmentação está na falta de informação sobre os objetos nas imagens. Isto é, dispomos de informações locais (e.g., brilho e cor) dos elementos da imagem, também conhecidos como *pixels*, para resolver um problema que usa informações

globais (e.g., forma, posição relativa e textura) e que portanto envolve uma interpretação de mais alto nível.

Em métodos de segmentação puramente baseados na imagem, o lapso de uma descrição global dos objetos na forma de um modelo matemático torna a identificação de objetos um problema intransponível, a menos que mediante assistência do usuário [59, 11]. Métodos de segmentação podem ser, portanto, divididos em abordagens baseadas em modelo e abordagens puramente baseadas na imagem.

Métodos baseados em modelo buscam suprir a carência de informações globais pelo emprego de modelos estatísticos criados por aprendizado supervisionado. Um conjunto de treinamento com instâncias do objeto é fornecido com a devida interação humana e esses dados são registrados em um espaço de referência comum para formar o modelo. Modelos de forma ativa [27] (ASM) e abordagens baseadas em atlas [51, 93, 89] são exemplos de métodos baseados em modelo que têm sido utilizados, por exemplo, para a segmentação de estruturas anatômicas do cérebro em imagens de RM [35, 52]. No entanto, o registro entre a imagem e o modelo durante a segmentação por vezes também ignora informações da imagem importantes, pelo ato de forçar os resultados a casarem com o modelo, enquanto métodos baseados na imagem teriam melhor precisão nesse aspecto.

Segmentação de imagens envolve o *reconhecimento* de objetos e o *delineamento* [45]. O reconhecimento é representado por tarefas cognitivas que determinam a localização aproximada de um objeto desejado em uma determinada imagem (detecção de objeto), verificam o quão correto é um resultado de segmentação, e identificam um objeto desejado de entre uma lista de objetos candidatos (classificação de objeto). Já o delineamento consiste em definir de forma precisa a extensão espacial do objeto. Os seres humanos superam geralmente computadores no reconhecimento de objetos, porém o oposto se verifica no caso do delineamento. Enquanto o usuário pode freqüentemente resolver o problema do reconhecimento através da simples seleção de pontos (sementes) ou por meio de uma ação inicializadora apropriada, delineamento perfeitamente repetível é um desafio devido à subjetividade do operador. Por outro lado, os computadores podem executar delineamento com muito mais precisão usando abordagens baseadas na imagem, mas a ausência de informação global torna o reconhecimento uma tarefa difícil. Isso explica por que algumas abordagens interativas bem sucedidas combinam reconhecimento pelo usuário com delineamento pelo computador de uma forma sinérgica, para uma segmentação mais eficaz e infalível [44, 36, 17, 94]. Seguindo esse mesmo raciocínio, temos que a sinergia entre as tarefas de reconhecimento e delineamento também se demonstra importante para a segmentação automática, onde o usuário é substituído por uma abordagem baseada no modelo do objeto. O emprego da sinergia entre reconhecimento e delineamento, na segmentação tanto interativa quanto automática, constitui o foco central deste trabalho.

## 1.2 Motivação

Este projeto tem por principal agente motivador as aplicações da segmentação no âmbito da neurologia envolvendo a análise quantitativa de estruturas cerebrais a partir de imagens de ressonância magnética (RM). Medidas de volume, textura e assimetria de forma/textura podem ser utilizadas para relacionar alterações morfológicas com diversas doenças, sendo útil nos seus diagnósticos e tratamentos [87, 12, 21, 104]. Essas técnicas de análise, porém, requerem uma definição precisa da extensão tridimensional das estruturas em estudo. Esta operação de segmentação de imagens consiste na atribuição de rótulos aos elementos da imagem de modo a associá-los à diferentes estruturas. No entanto, uma sequência espacial de fatias de RM forma uma imagem tridimensional (ou volume) tipicamente composta por cerca de dez milhões de elementos também conhecidos como *voxels*. Logo, em vista das dimensões do problema, a segmentação para a maioria das estruturas cerebrais requer assistência exaustiva do usuário e, como consequência, o estudo de doenças cerebrais por imagens de RM sofre de várias limitações. Neste sentido, torna-se necessária a pesquisa por métodos automáticos e eficientes que favoreçam a análise em larga escala de estruturas cerebrais, possibilitando o acompanhamento de alterações entre controles e pacientes, ou mesmo entre diferentes fases do tratamento de um indivíduo.

Os métodos desenvolvidos nesse trabalho têm por particular interesse a segmentação de imagens de ressonância magnética do cérebro humano de forma automática, ou com o mínimo de intervenção do usuário (FAPESP Proc. 05/59808-0). Este trabalho está também relacionado com o programa FAPESP-CInApCe que conta com os professores do Departamento de Neurologia da Faculdade de Ciências Médicas da UNICAMP e envolve o estudo de epilepsia e doenças degenerativas [37].

## 1.3 Objetivos

Neste trabalho, visamos dar continuidade aos trabalhos de segmentação de imagem usando grafos que exploram a Transformada Imagem-Floresta (IFT) [43]. Durante o mestrado, vários operadores de imagem para segmentação [38, 42] foram projetados com base no framework único da IFT, o que proporciona simplicidade e elegância. Um dos principais objetivos desse trabalho é reduzir as intervenções do usuário no contexto da segmentação de estruturas cerebrais em neurologia, estendendo as técnicas da IFT pelo emprego de modelos discretos dos objetos.

## 1.4 Visão Geral das Contribuições

Uma primeira contribuição desse trabalho, diz respeito a uma visão do problema da segmentação que considera o delineamento como o resultado de duas etapas: *realce* e *extração*. Do ponto de vista matemático, o problema da segmentação pode ser modelado com base em grafos, onde o realce está associado à estimativa dos pesos das arestas, enquanto a extração corresponde ao resultado de um algoritmo de delineamento para partição ótima no grafo.

Os resultados de extração por algoritmos de delineamento dependem fortemente da qualidade dos pesos das arestas considerados. No entanto, poucos trabalhos tratam especificamente o problema da estimativa dos pesos, lhe conferindo a sua real importância. Nos métodos tradicionais de segmentação [59, 11, 106, 124], a estimativa de peso dos arcos é geralmente tratada como um simples processo incorporado, desconsiderando, em muitos casos, as intervenções do usuário. Já em outros trabalhos [101, 100, 94] o realce e a extração são concebidos de forma integrada e indivisível elevando consideravelmente a complexidade dos métodos, com impactos no desempenho e/ou dificultando o seu entendimento.

Visando solucionar esses problemas, propomos um sistema para segmentação que trata separadamente a questão da formulação dos pesos do grafo. Neste trabalho consideramos essa separação de forma explícita enquanto a separação de modo implícito está sendo investigada (i.e., o usuário não toma ciência de que a separação está sendo efetuada). Logo, um módulo especializado no tratamento do problema da formulação dos pesos do grafo é considerado neste trabalho. Esse primeiro módulo consiste em um procedimento sinérgico no qual as ações do usuário são guiadas por uma resposta visual sobre a qualidade dos pesos calculados (bloco 1 do diagrama da Figura 1.1). No caso automático, uma abordagem customizada para a aplicação de interesse é adotada.

Tratando a estimativa de pesos separadamente temos as seguintes vantagens: a separação favorece a análise comparativa entre diferentes algoritmos de delineamento a partir do estudo teórico das suas propriedades no grafo, o que gera conclusões livres de parcialidade; ela também nos permite um melhor entendimento sobre a real contribuição que cada etapa exerce sobre os resultados finais da segmentação; com ela conseguimos também depurar melhor os programas evitando erros de implementação; e comparações empíricas mais justas entre os métodos são favorecidas, em vista do grafo base comum que proporciona iguais condições iniciais. Com a separação da estimativa de pesos, temos um segundo módulo para a extração interativa de objetos que explora a sinergia entre o reconhecimento pelo usuário e o delineamento (bloco 2 da Figura 1.1). Avanços teóricos e práticos são apresentados nessa área com base na Transformada Imagem-Floresta (IFT) [43].

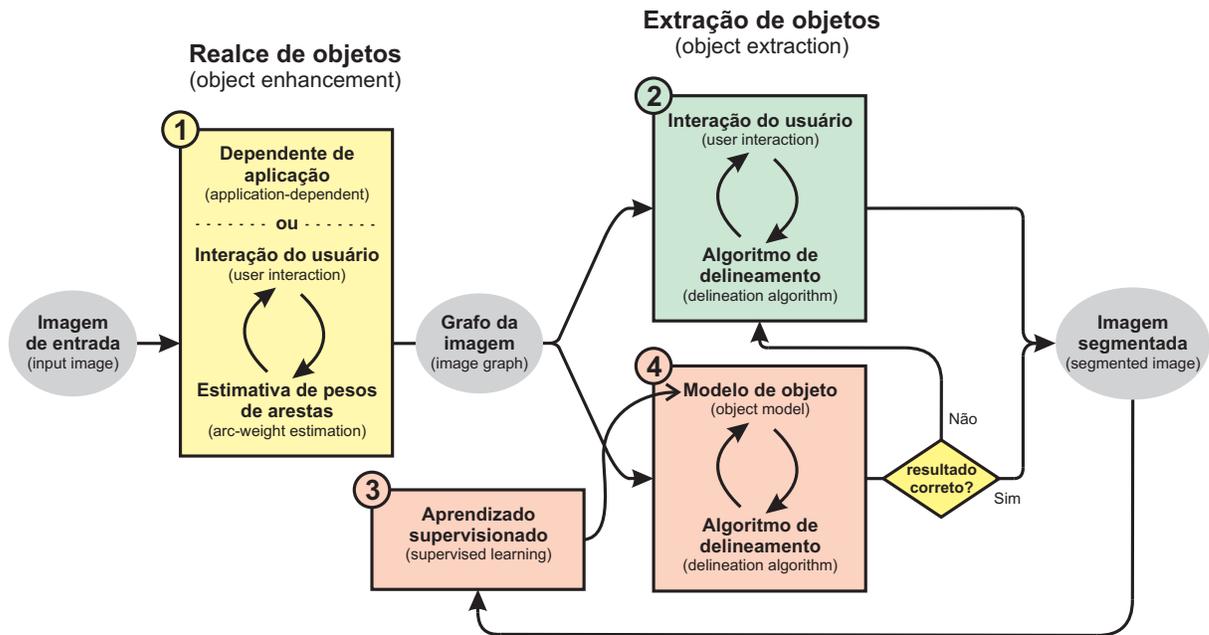


Figura 1.1: Diagrama do sistema de segmentação proposto.

Para um dado grafo da imagem, métodos de segmentação automática podem ser obtidos através da combinação sequencial de algum método específico para reconhecimento automático [121, 128], seguido por algum algoritmo de delineamento que completa a segmentação [64, 52]. Recentemente, no entanto, alguns métodos híbridos [56, 67, 22] têm adicionado mais dinamismo ao processo, de modo que o resultado do delineamento também influencia no reconhecimento e vice-versa (sinergia). Estes métodos híbridos propostos porém não são diretamente extensíveis para imagens multidimensionais, o que limita a sua aplicação.

Neste trabalho são propostos novos modelos híbridos para segmentação automática que são adequados aos problemas envolvendo imagens tridimensionais. Estes modelos compõem um terceiro módulo representado pelos blocos 3 e 4 no diagrama da Figura 1.1 que mostra o sistema proposto, bem como o seu fluxo de funcionamento. Os modelos são construídos por aprendizado supervisionado a partir de uma base de treinamento com imagens pré-segmentadas pelo especialista. Os resultados automáticos obtidos para novas imagens são então verificados pelo especialista para possíveis correções interativas em caso de falhas. Os três módulos propostos são apresentados a seguir em maiores detalhes.

**1. Sinergia na Estimativa de Peso das Arestas do Grafo.** Os métodos de segmentação normalmente fazem uso direto/indireto de algum conceito referente a grafos

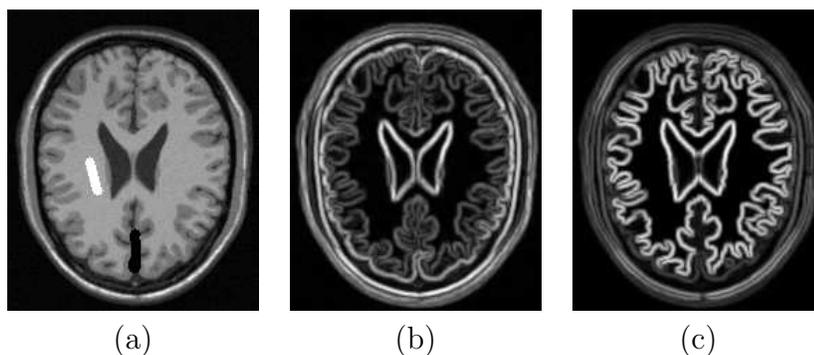


Figura 1.2: (a) Uma fatia de RM com marcadores para treinamento. (b) Visualização dos pesos das arestas inicial. (c) Um considerável realce é obtido para a substância branca através de aprendizado supervisionado dos marcadores.

da imagem, tais como o peso de aresta entre pixels. O peso pode representar diferentes atributos funcionais tais como similaridade, função de velocidade, afinidade, custo, distância, etc; dependendo dos frameworks usados, tais como *watershed*, *level sets*, *fuzzy connectedness*, *graph cuts*, etc. O delineamento preciso por estes métodos com mínima intervenção do usuário fortemente depende de uma estimativa adequada dos pesos das arestas, que em geral deve levar em conta as informações dos objetos obtidas a partir de marcadores selecionados pelo usuário durante a segmentação.

Informações de objeto extraídas dos marcadores (e.g., padrão de intensidade, cor) são cruciais para a melhoria da qualidade da estimativa dos pesos das arestas. No entanto, os marcadores utilizados pelo algoritmo de delineamento para a extração nunca devem ser utilizados na sua totalidade para recalculer os pesos, pois, muitas vezes, estes marcadores devem ser escolhidos em regiões onde o objeto e o fundo têm propriedades semelhantes. O recálculo dos pesos seguido por delineamento, com base nestes marcadores pode levar a uma estimativa de pesos baseada em exceções. Isso gera um efeito colateral, afetando outras partes da segmentação onde o usuário já estava satisfeito com os resultados, fazendo o usuário perder o controle sobre o processo.

Essas questões são tratadas no Capítulo 2 que apresenta um procedimento sinérgico para estimativa do peso das arestas, onde o usuário traça marcadores dentro de cada objeto (incluindo o fundo), os pesos das arestas são estimados a partir de atributos da imagem e das informações dos objetos (pixels sob marcadores), e uma resposta visual orienta a próxima ação do usuário (Figura 1.2). No caso automático, não dispomos de marcadores pelo usuário e portanto adotamos uma abordagem customizada para a aplicação de interesse que é descrita no Capítulo 5.

**2. Sinergia entre Reconhecimento pelo Usuário e Delineamento.** Em segmentação sinérgica assistida por operador, o usuário geralmente adiciona/remove marcadores (*seed pixels, anchor boundary points*) para o reconhecimento, enquanto o delineamento subsequente é executado pelo computador em tempo interativo. Acurácia torna-se um compromisso entre a paciência do usuário, para verificação e correção, e a qualidade do delineamento.

A pesquisa nessa área consiste em buscar a redução do número de intervenções do usuário necessárias durante o processo, conservando ou melhorando medidas de exatidão. Ela compreende duas vertentes: estudos teóricos (Figura 1.3) e práticos (avaliação empírica). Avanços nessas duas frentes de pesquisa são apresentados respectivamente nos Capítulos 3 e 4, que fundamentam a nossa opção preferencial por métodos de delineamento baseados na IFT [43].

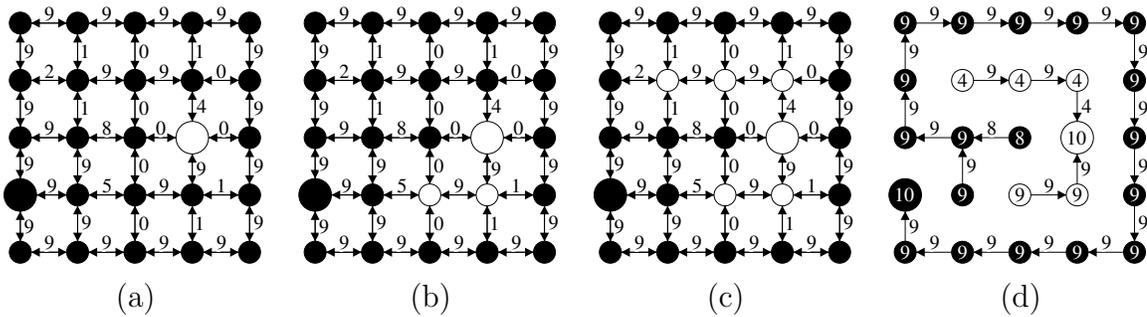


Figura 1.3: Avanços teóricos são obtidos pela análise das propriedades dos algoritmos nos grafos derivados das imagens. (a) Um grafo com vizinhança 4 a ser segmentado tendo duas restrições fortes (pontos maiores), onde o ponto branco deve pertencer ao objeto e o ponto maior preto deve pertencer ao fundo. (b-c) Dois possíveis resultados de segmentação. (d) A solução por floresta de caminhos ótimos conforme será discutido no Capítulo 3.

**3. Sinergia entre Reconhecimento por Modelos e Delineamento.** Esse terceiro módulo compreende a automação do segundo módulo pela substituição das intervenções humanas por modelos estatísticos dos objetos. A principal contribuição apresentada corresponde a proposta de um conjunto de três modelos baseados no conceito de *Clouds*. Esses modelos discretos são obtidos por aprendizado supervisionado e armazenam informações globais sobre a forma dos objetos, as quais permitem segmentá-los nas imagens com eficácia, explorando a sinergia entre reconhecimento e delineamento. Ou seja, várias interações entre as informações do modelo e um algoritmo de delineamento são usadas na definição do resultado final. Durante esse processo sinérgico, o delineamento é

favorecido pelas informações de forma a priori provenientes do modelo, enquanto o critério para reconhecimento é favorecido pelos objetos delineados (Figura 1.4).

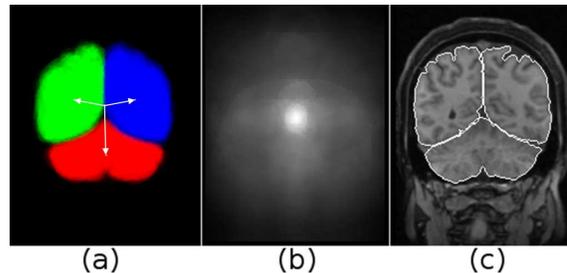


Figura 1.4: (a) Um sistema de *Clouds* para três objetos mostrado no espaço de cores RGB. (b-c) Valores obtidos pelo critério de reconhecimento para diversas posições do modelo sobre uma imagem de entrada e o resultado final de segmentação.

Os modelos são apresentados utilizando o algoritmo de delineamento IFT-SC descrito no Capítulo 3 ao invés do algoritmo apresentado no Capítulo 4, visto que esse último é mais adequado no ambiente interativo.

Os modelos propostos foram validados na segmentação de diferentes estruturas cerebrais. Com relação a tarefa de isolar o cérebro dos demais voxels em imagens de RM-T1, dentre os três modelos propostos o modelo *Cloud System Model* demonstrou melhores resultados tendo acurácias superiores as obtidas pela técnica padrão baseada em atlas, disponível no *software* SPM2, e também superiores as obtidas pela técnica automática *tree-pruning* [9]. Os modelos propostos também se aplicam a outras estruturas cerebrais e foram avaliados na segmentação automática 3D do cerebelo; e dos hemisférios cerebrais. Essas estruturas estão conectadas em diversas partes, gerando um grave desafio para estratégias simples de segmentação. No entanto, os métodos levam poucos segundos por estrutura, executando em PCs modernos, e proporcionam bons resultados que eliminam as intervenções do usuário ou, que ao menos as reduzem a simples correções locais pela técnica [36]. Esses modelos são tratados detalhadamente no Capítulo 5.

## 1.5 Organização da Tese

O texto desta tese está organizado agrupando os principais artigos publicados ou aceitos para publicação como resultado da pesquisa realizada.

### 1.5.1 Capítulo 2

Este capítulo (Synergistic Arc-Weight Estimation for Interactive Image Segmentation using Graphs) inclui os resultados publicados na revista CVIU [83]. Resumidamente, este método para realce de objetos pode ser descrito da seguinte forma. Para cada pixel sob os marcadores selecionados pelo usuário, existe um vetor de características associado (e.g., brilho, textura), bem como o rótulo de objeto fornecido pelo marcador. A partir dessas informações um valor de pertinência em relação a cada objeto é associado a cada pixel da imagem por aprendizado supervisionado seguido de classificação fuzzy. Os pixels sob marcadores formam portanto um conjunto de treinamento. No entanto, é razoável esperar que o tamanho máximo do conjunto de treinamento seja limitado por razões de eficiência. Logo devemos fazer o melhor uso possível desse conjunto. Para isso dividimos o conjunto de pixels sob marcadores em um conjunto de treinamento e outro de avaliação. O conjunto de treinamento é então modificado a partir dos seus erros no conjunto de avaliação usando o maior valor de pertinência como regra de classificação. A acurácia após cada iteração de retreinamento nos fornece uma curva de aprendizado. Nesse capítulo, a associação dos valores de pertinência descrita é baseada na comparação das diferentes concentrações de pontos de cada objeto no espaço de características, que estão inversamente relacionadas as distâncias aos  $k$  vizinhos mais próximos em cada objeto. Alguns experimentos apresentados neste capítulo mostram melhorias significativas na segmentação interativa da substância branca em fatias 2D de RM-T1.

### 1.5.2 Capítulo 3

Este capítulo (Links Between Image Segmentation based on Optimum-Path Forest and Minimum Cut in Graph) apresenta os resultados publicados no Journal of Mathematical Imaging and Vision em 2009 [79]. Este artigo corresponde a um trabalho teórico que relaciona métodos de segmentação baseados em floresta de caminhos ótimos [43] com métodos de corte em grafo, dando o embasamento teórico que explica em parte o grande sucesso do framework da IFT. Neste capítulo é demonstrado que diferentes métodos baseados na IFT oferecem resultados ótimos de segmentação sob dois pontos de vista: como uma floresta de caminhos ótimos e como um corte mínimo no grafo de acordo com uma medida de avaliação apropriada, que pode levar em conta também propriedades dos objetos seguindo os pesos de aresta descritos no Capítulo 2. Um outro importante resultado apresentado neste capítulo corresponde a prova da otimalidade por partes das bordas de corte geradas pela IFT com competição entre sementes internas e externas. Essa propriedade é essencial para conservar o controle do usuário durante correções interativas, que podem ser rapidamente efetuadas pelo algoritmo da DIFT [36]. É também mostrado que procedimentos adotados para contornar o viés existente no algoritmo de

*min-cut/max-flow* levam a aproximações da IFT com competição entre sementes internas e externas.

### 1.5.3 Capítulo 4

Este capítulo (Object delineation by  $\kappa$ -connected components) inclui os resultados publicados no Journal on Advances in Signal Processing em 2008 [80]. Neste capítulo são apresentados avanços práticos no delineamento de objetos através da avaliação empírica de operadores da IFT que combinam diferentes paradigmas. Os resultados mostram a redução do número de intervenções do usuário necessárias para a conclusão da segmentação interativa de diversas imagens médicas.

### 1.5.4 Capítulo 5

Este capítulo (Cloud Models: Their Construction and Employment in Automatic MRI Segmentation of the Brain) inclui os resultados publicados em dois artigos internacionais [81, 82], bem como uma extensão para o tratamento de múltiplos objetos simultaneamente que leva a ganhos nos resultados em função do melhor aproveitamento da informação de contexto no reconhecimento. Os modelos *Clouds* propostos nesse capítulo são criados por aprendizado supervisionado a partir de um conjunto de imagens de treinamento, que contém instâncias dos objetos de interesse (bloco 3 da Figura 1.1). Essencialmente, o modelo faz o papel do operador humano enquanto um algoritmo baseado no grafo da imagem realiza o delineamento, e ambos operam de forma sinérgica, até que um estado ótimo é alcançado (bloco 4 da Figura 1.1). Variações anatômicas acentuadas das estruturas são modeladas pelo emprego de múltiplas nuvens e as informações a priori da forma dos objetos são incorporadas na formulação dos pesos das arestas. Algumas das vantagens destes métodos são: eles descartam o uso de registro durante treinamento e segmentação; eles analisam toda a fronteira do objeto durante o delineamento e o reconhecimento ao invés de apenas alguns *pontos de controle* como em [27]; e eles podem ser facilmente estendidos para imagens multidimensionais. São apresentados experimentos de validação em um conjunto de dados tridimensionais de RM composto por 40 indivíduos normais de ambos os sexos, na faixa etária de 16 a 49 anos, e também em 40 imagens de pacientes.

# Capítulo 2

## Synergistic Arc-Weight Estimation for Interactive Image Segmentation using Graphs

### 2.1 Introduction

In image processing and computer vision, there are several situations in which user interaction becomes essential in obtaining effective image segmentation. The high-level, application-domain-specific knowledge of the user is often required in medical image analysis [45, 18, 63, 90] because of poorly defined structures, and in the digital matting of natural scenes [8, 122], because of their heterogeneous nature.

Image segmentation involves two tightly coupled tasks: *recognition* and *delineation* [45]. Recognition is the task of determining the approximate whereabouts of a desired object in the image, while delineation completes segmentation by precisely defining its spatial extent. Humans usually outperform computers in recognition, but the contrary can be observed in object delineation. While the user can reduce recognition to a simple click of the mouse inside the object, repeatable human delineation is challenging due to human subjectivity. On the other hand, computers can be very precise, even when they are not accurate, but often the absence of high-level object information (location, shape, appearance) makes object recognition a difficult task for computers. In order to overcome some of these shortcomings from both sides, some approaches have combined recognition by the user with delineation by the computer in a synergistic way [44, 36, 17, 94]. This topic forms the central focus of this paper.

In operator-assisted synergistic segmentation, the user usually adds/removes markers (seed pixels, anchor boundary points) for recognition, while subsequent delineation is performed by the computer in interactive time. Accuracy becomes a compromise between

the user’s patience for verification and correction, and the quality of delineation. The methods usually make direct/indirect use of some image-graph concept, such as arc weight between pixels. The weight may represent different attribute functionals such as similarity, speed function, affinity, cost, distance, etc; depending on different frameworks used, such as watershed, level sets, fuzzy connectedness, graph cuts, etc. The accurate delineation by these methods with minimum user intervention strongly depends on a suitable arc-weight estimation, which usually takes into account image attributes and/or object information often obtained from markers selected by the user during segmentation [17, 94]. Object information is very crucial for improving the quality of arc-weight estimation. However, the user’s actions need guidance from visual feedback about the quality of the arc weights. Further, the markers used for delineation should never be used to recompute weights. Very often, these markers need to be selected in regions where object and background have similar properties. Weight recomputation followed by delineation based on these markers may destroy other parts of the image where the user was already satisfied with the segmentation results, making the user to lose control over the process.

We propose a synergistic approach for arc-weight estimation, which is separated from the process of interactive image segmentation itself. As a training step, the user selects markers inside each object, where image background is also considered as an object, guided by a visual feedback about the quality of arc-weight estimation. The training markers may be used to start object delineation, but markers selected during segmentation are never allowed to modify arc-weight assignment. We use this approach as a basic step in several image segmentation methods, such as those based on the min-cut/max-flow algorithm [17] and approaches which can be easily implemented by the *image foresting transform* (IFT) [43]. Note that our aim is not to compare segmentation methods, but to show that several of them can benefit from a disciplined, systematic, objective, and effective procedure for arc-weight assignment, followed by some proper approach-specific adaptive procedure, such as the complement of the weights, owing to the nature of the meaning of weights in some methods; or by some tuning procedure (e.g., non-maximal suppression [79], increasing transformations [79], gradient orientation (Section 2.4.7)). The visualization of the arc weights also allows the user to choose the most appropriate method for a given image. For example, it is desirable in live wire that the arc weights be lower along the object’s boundary than in the neighborhood around it [45, 44]; the local affinities in relative-fuzzy connectedness [99] be higher inside and outside the object than on its boundary; the gradient values in watershed transforms be higher for pixels on the object’s boundary than in its interior and exterior [68, 11, 36]; the gradient values in tree pruning be higher on the object’s boundary than in its interior, and, at least, in a neighborhood in its exterior [9]; and the arc weights in graph-cut segmentation be lower across the object’s boundary than in its interior and exterior [17, 106, 124]. Additionally,

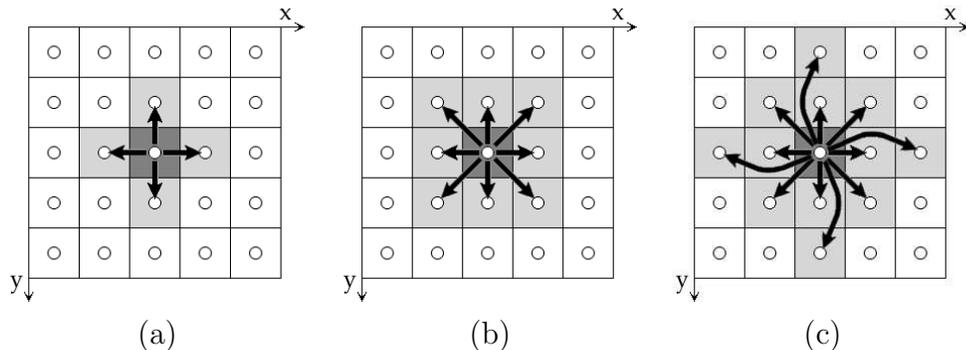


Figure 2.1: Euclidean adjacency relations for 2D images: (a) 4-neighborhood ( $\rho = 1$ ), (b) 8-neighborhood ( $\rho = \sqrt{2}$ ), and (c) extended adjacency to the 12 closest neighbors ( $\rho = 2$ ).

energy minimization in [17] using the min-cut/max-flow algorithm from source to sink nodes also requires higher arc weights between source and object pixels, lower arc weights between source and background pixels, lower arc weights between sink and object pixels, and higher arc weights between sink and background pixels. Clearly, the effectiveness of these approaches suffers when the above desirable conditions are not satisfied, and this explains why the visual feedback helps the user to choose the most appropriate method for a given segmentation task.

To outline this paper: Section 2.2 presents the basic concepts on image graphs and the terminology adopted in this paper. Arc-weight estimation is presented in Section 2.3 by showing how to exploit image attributes and object information provided by user-selected markers. Section 2.4 describes several interactive segmentation methods based on the arc-weight assignment of Section 2.3, and the main advantages of the synergistic approach are demonstrated in Section 2.5, including evaluation experiments with medical data from two imaging modalities (CT and MRI). Our conclusions are stated in Section 2.6.

## 2.2 Basic Concepts on Image Graphs

An image  $\hat{I}$  is a pair  $(D_{\hat{I}}, \vec{I})$  where  $D_{\hat{I}} \subset Z^n$  is the image domain and  $\vec{I}(s)$  assigns a set of  $m$  scalars  $I_i(s)$ ,  $i = 1, 2, \dots, m$ , to each pixel  $s \in D_{\hat{I}}$ . This definition applies to multi-dimensional and multi-parametric images. For example,  $\{I_1(s), I_2(s), I_3(s)\}$  may be the red, green and blue values of  $s$  in a color image  $\hat{I}$ . The subindex  $i$  is dropped for gray images since it becomes awkward when  $m = 1$ .

An irreflexive *adjacency relation*  $\mathcal{A}$  is a binary relation between distinct pixels. We use  $t \in \mathcal{A}(s)$  or  $(s, t) \in \mathcal{A}$  to indicate that  $t$  is adjacent to  $s$ . Once  $\mathcal{A}$  is fixed, the image  $\hat{I}$

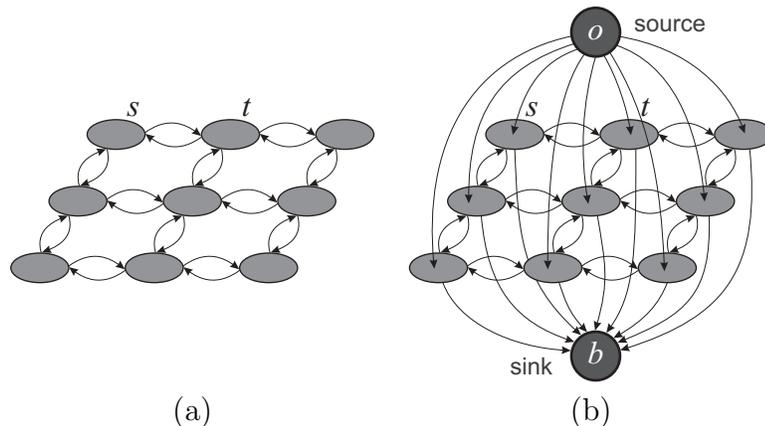


Figure 2.2: (a) A 2D image graph with 4-adjacent pixels  $s$  and  $t$ . (b) An extended graph obtained by adding two terminal nodes (*source*  $o$  and *sink*  $b$ ), which represent object and background respectively.

can be interpreted as a graph  $(D_{\hat{I}}, \mathcal{A})$ , whose nodes are the image pixels and whose arcs are the pairs  $(s, t)$  in  $\mathcal{A}$ . For example, one can take  $\mathcal{A}$  to consist of all pairs of pixels  $(s, t)$  in the Cartesian product  $D_{\hat{I}} \times D_{\hat{I}}$  such that  $d(s, t) \leq \rho$  and  $s \neq t$ , where  $d(s, t)$  denotes the Euclidean distance and  $\rho > 0$  is a specified constant (Figure 2.1).

A 2D image graph is illustrated in Figure 2.2a for  $\rho = 1$ . This graph topology can be the same for any segmentation method based on optimum paths. The approach based on the min-cut/max-flow algorithm can use this image graph extended by two virtual nodes, source  $o$  and sink  $b$ , with arcs  $(o, s)$  and  $(s, b)$  connecting them to each pixel  $s \in D_{\hat{I}}$  (Figure 2.2b). The arc weights  $w(s, t)$  of the image graph are estimated via training, as described next, and the extended arc weights,  $w(o, s)$  and  $w(s, b)$ , are estimated from intermediate results of the training step, as explained in Section 2.4.5.

## 2.3 Synergistic arc-weight estimation

Arc-weight estimation takes into account image attributes and object information in order to enhance the discontinuities between object and background. Let  $v$  be an algorithm which extracts attributes (color, gradient, texture) from any pixel  $s \in D_{\hat{I}}$  and returns a vector  $\vec{v}(s)$ . In the simplest case, we may take  $\vec{v}(s) = \vec{I}(s)$ . However, the best set of attributes depends on each given application<sup>1</sup>. In the segmentation of natural scenes, for example, one may exploit the *Lab* color space [127] and/or compute texture attributes

<sup>1</sup>The images used for arc-weight assignment in Figures 2.3, 2.4, 2.6, 2.10, and 2.14 are in the RGB color space.

around each pixel from the results of the image convolved with a bank of filters [66, 73, 84, 69, 85]. Other options are discussed in Section 2.3.2. For  $c$  objects  $l = 1, 2, \dots, c$ , including the background as object numbered  $c$  without loss of generality, the weight  $w(s, t)$  assigned to each arc  $(s, t) \in \mathcal{A}$  is a linear combination of an image-based weight  $0 \leq w_i(s, t) \leq K$  and an object-based weight  $0 \leq w_o(s, t) \leq K$ , which takes into account all  $c$  objects.

$$w(s, t) = \lambda w_o(s, t) + (1 - \lambda) w_i(s, t), \quad (2.1)$$

where  $0 \leq \lambda \leq 1$ . The weights  $w_i(s, t)$  exploit only image attributes to capture discontinuities that may exist between homogeneous regions. The weights  $w_o(s, t)$  take into account the image attributes for pixels under selected markers, drawn by the user inside each object  $l = 1, 2, \dots, c$ . They aim to characterize the discontinuities existing between each selected object and the rest of the image. The user can adjust the parameter  $\lambda$  and add/remove markers to recompute the arc weights. The quality of the arc weights is evaluated by visualizing a weight image  $\hat{W} = (D_{\hat{f}}, W)$ , where

$$W(s) = \max_{\forall t \in \mathcal{A}(s)} \{w(s, t)\} \quad (2.2)$$

for all  $s \in D_{\hat{f}}$ . Our aim is to make  $w(s, t)$  higher on the desired object boundaries than inside the objects, so  $\hat{W}$  must show a suitable boundary enhancement for a given image (see Figure 2.3). The complement of  $w(s, t)$  may be used depending on the segmentation method. The process of arc-weight assignment stops when the user is satisfied with the boundary enhancement. The image-based weights  $w_i(s, t)$  become more important ( $\lambda$  is lower), when nearby objects have similar image properties (Figures 2.7 and 2.14).

The following sections describe how to define object-based and image-based weights and discuss some implementation and customization issues.

### 2.3.1 Object-based weight assignment

Let  $\hat{d}(s, t) \geq 0$  be the distance between the corresponding attribute vectors,  $\vec{v}(s)$  and  $\vec{v}(t)$ , of two pixels  $s$  and  $t$ . One can use any distance function suitable for the defined attributes. The most common is the vector norm  $\|\vec{v}(t) - \vec{v}(s)\|$ , which is the one used in this paper, but some image attributes may require special distance algorithms [73, 85]. The pair  $(\vec{v}, \hat{d})$  then describes how the pixels of a dataset are distributed in the attribute space and we call it a *descriptor*.

Let  $\mathcal{S}_l \subset D_{\hat{f}}$  be the set of representative pixels (markers) selected by the user inside each object  $l = 1, 2, \dots, c$ . A suitable descriptor should group pixels of distinct objects in different regions of the attribute space, but the same object may be represented by

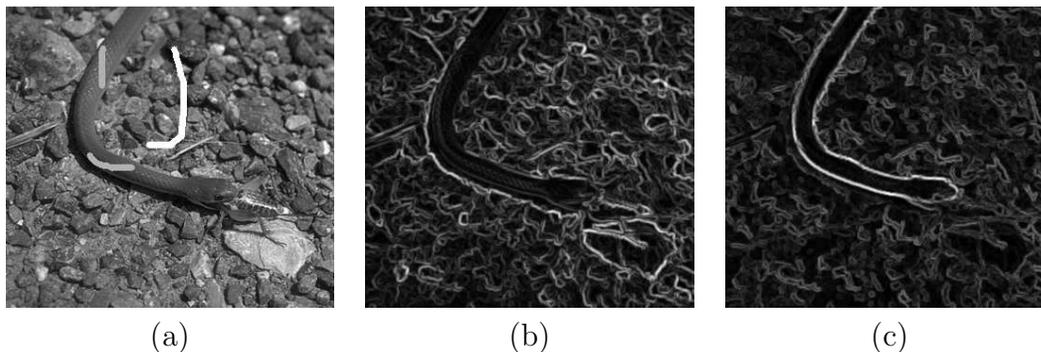


Figure 2.3: (a) An image with markers selected inside and outside the object. (b) The weight image  $\hat{W}$  considering only the image-based component ( $\lambda = 0$ ), and (c) by combining it with the object-based weight ( $\lambda = 0.8$ ),  $c = 2$ .

multiple clusters and pixels of distinct objects may fall in the same cluster. This explains the importance of pixel connectivity for the success of segmentation. We define  $\mathcal{A}_{k,l}$  as a special adjacency relation in the attribute space between any pair of pixels  $(s, t)$ , such that,  $s \in D_{\hat{f}}$ , and  $t \in S_l$  is a  $k$ -nearest neighbor of  $s$  in the attribute space.

$$t \in \mathcal{A}_{k,l}(s) \quad \text{if } t \in S_l \text{ is a } k\text{-nearest neighbor of } s \in D_{\hat{f}}. \quad (2.3)$$

We expect that the mean distance  $\bar{d}(s, \mathcal{A}_{k,l}(s))$  between  $s$  and its  $k$  neighbors in  $S_l$ ,  $l = 1, 2, \dots, c$ , be the smallest for pixels of the same object of  $s$ .

$$\bar{d}(s, \mathcal{A}_{k,l}(s)) = \frac{1}{k} \sum_{\forall t \in \mathcal{A}_{k,l}(s)} \hat{d}(s, t). \quad (2.4)$$

The fuzzy membership value  $\mu(l | \vec{v}(s))$  can then be conjectured to be proportional to the total mean distance  $\sum \bar{d}(s, \mathcal{A}_{k,i}(s))$  for  $i = 1, 2, \dots, c$ , and  $i \neq l$ .

$$\mu(l | \vec{v}(s)) \approx \left( \frac{K}{c-1} \right) \frac{\sum_{\forall i=1,2,\dots,c} \bar{d}(s, \mathcal{A}_{k,i}(s))}{\sum_{\forall i=1,2,\dots,c} \bar{d}(s, \mathcal{A}_{k,i}(s))}. \quad (2.5)$$

These membership values can be viewed through a set of images  $\hat{M}_i = (D_{\hat{f}}, M_i)$ ,  $i = 1, 2, \dots, c$  such that  $M_i(s) = \mu(l = i | \vec{v}(s))$  for all  $s \in D_{\hat{f}}$ . For multiple objects, the user should keep on drawing markers inside the dark regions of object  $i$  in each image  $\hat{M}_i$ ,  $i = 1, 2, \dots, c$ , until that object becomes brighter than the rest in this membership image. For the sake of simplicity, all examples in this paper use only “object and background”

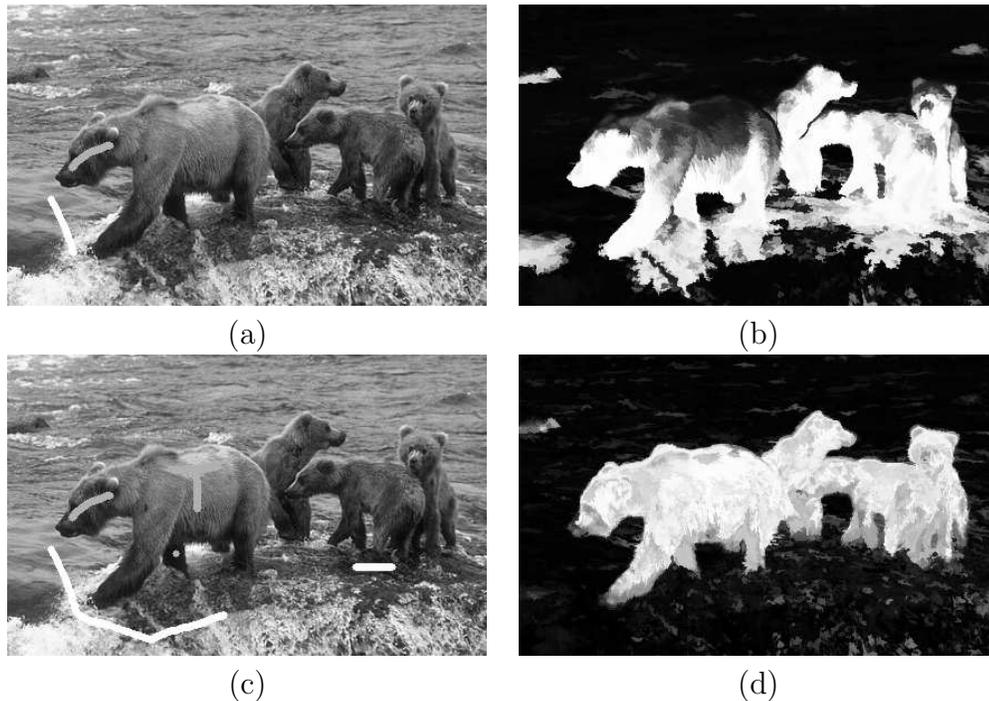


Figure 2.4: Image  $\hat{M}_1$ , where the desired object is a family of bears. (a-b) An initial marker selection and the corresponding membership image  $\hat{M}_1$ . (c-d) The estimation improves as the user adds internal markers on the dark regions of the object and external markers on the bright regions of the background in  $\hat{M}_1$ .

type of situation. In this case,  $\mu(l = 1 | \vec{v}(s)) = K - \mu(l = 2 | \vec{v}(s))$ , and then we can, and need to, show only  $\hat{M}_1$  with internal and external markers (Figure 2.4). As the user adds markers, the estimation improves and the object becomes increasingly distinguished (brighter) from the background (darker).

We could have estimated  $\mu(l | \vec{v}(s))$  by Baye's Theorem, by directly computing the posterior probability  $\mathcal{P}(l | \vec{v}(s))$  from the markers and the distances between  $s$  and its neighbors in  $\mathcal{A}_{k,l}(s)$  in the attribute space [34]. A similar approach to compute probability density functions is described in [95]. We compared with that approach and the results were equivalent to those obtained by Equation 2.5, which is simpler and more efficient. Note that, according to Equation 2.5, for any pixel  $s$  the sum of  $\mu(l | \vec{v}(s))$  for  $l = 1, 2, \dots, c$  is the maximum value  $K$ , as desired for a discrete surrogate of the probability quantized into  $K$  levels.

The discontinuities between each object  $l$  and the rest of the image can be captured

from a gradient vector  $\vec{G}_l(s)$ , defined for all  $s \in D_{\hat{f}}$  and computed as follows.

$$\vec{G}_l(s) = \sum_{\forall t \in \mathcal{A}(s)} [\mu(l | \vec{v}(t)) - \mu(l | \vec{v}(s))] \vec{st}, \quad (2.6)$$

where  $\vec{st}$  is the unit vector connecting  $s$  to  $t$  in the image domain. For a 2D Euclidean adjacency  $\mathcal{A}$  with  $\rho = \sqrt{2}$ , the gradient vector  $\vec{G}_l(s)$  is estimated from the vectorial sum of the first derivatives of  $\mu(l | \vec{v}(s))$  along the 8 directions, rather than from the  $x$  and  $y$  directions only.

For each arc  $(s, t) \in \mathcal{A}$ , we compute the magnitude of the mean gradient vector of its pixels and use it as the weight  $w_{o,l}(s, t)$  with respect to the object  $l$ . The final object-based weight  $w_o(s, t)$  is considered to be the maximum of  $w_{o,l}(s, t)$  among all objects.

$$w_{o,l}(s, t) = \left| \frac{\vec{G}_l(s) + \vec{G}_l(t)}{2} \right| \quad (2.7)$$

$$w_o(s, t) = \max_{l=1,2,\dots,c} \{w_{o,l}(s, t)\}. \quad (2.8)$$

The orientation of  $\vec{G}_l(s) + \vec{G}_l(t)$  can also be exploited to modify arc-weight assignment (Section 2.4.7).

As the user adds markers, the size of the union set  $\mathcal{Z} = \bigcup_{l=1,2,\dots,c} \mathcal{S}_l$  increases and Equation 2.5 becomes computationally more expensive to assign membership values to all pixels in  $D_{\hat{f}}$ . On the other hand, we do not need quantity, but quality, in choosing pixels for  $\mathcal{Z}$ . In order to choose the best representative pixels for each object from the drawn markers, and, at the same time, to estimate the best parameter  $k$ , we use supervised learning as described next.

### Supervised learning from markers

The main idea is to reduce the size of  $\mathcal{Z}$  by selecting a subset  $\mathcal{Z}_1 \subset \mathcal{Z}$  of the most representative pixels. These pixels are defined as those that maximize the classification accuracy of the remaining set of pixels  $\mathcal{Z}_2 = \mathcal{Z} \setminus \mathcal{Z}_1$ , using the maximum  $\mu(l | \vec{v}(s))$  as the decision rule for the pixels  $s \in \mathcal{Z}_2$  with respect to its neighbors  $\mathcal{A}_{k,l}(s) \subset \mathcal{Z}_1$  by Equation 2.5.

The set  $\mathcal{Z}$  is divided into two subsets,  $\mathcal{Z}_1$  and  $\mathcal{Z}_2$ , by randomly selecting the same percentage of pixels from each object. Set  $\mathcal{Z}_1$  has a maximum size (e.g., 100 pixels). When the number of seeds is less than that maximum size, we may divide  $\mathcal{Z}$  into 50% for  $\mathcal{Z}_1$  and 50% for  $\mathcal{Z}_2$ . The maximum  $\mu(l | \vec{v}(s))$  is used to classify the pixels in  $\mathcal{Z}_2$ . This process is repeated for each  $k$  from 1 to  $k_{max}$  (Equation 2.9) in order to obtain the best value of  $k$  for the given  $\mathcal{Z}_1$ . The misclassified pixels with the best  $k$  are randomly

replaced by pixels of the same object in  $\mathcal{Z}_1$ . The whole process is repeated over a few iterations  $T$  (e.g.,  $T = 5$ ) and the pair  $(\mathcal{Z}_1, k)$  of maximum accuracy is selected as output (see Algorithm 1).

$$k_{max} = \min_{\forall l=1,2,\dots,c} \left\{ \frac{|\mathcal{S}_l \cap \mathcal{Z}_1|}{2} \right\}. \quad (2.9)$$

**Algorithm 1** – LEARNING ALGORITHM

INPUT: Initial sets  $\mathcal{Z}_1$  and  $\mathcal{Z}_2$ , number  $T$  of iterations, and the descriptor  $(v, d)$ .  
 OUTPUT: The pair  $(\mathcal{Z}_1^*, k^*)$  of maximum accuracy.  
 AUXILIARY: Arrays  $FP$  and  $FN$  of sizes  $c$  for false positives and false negatives, list  $M$  of misclassified pixels and its auxiliary  $M^*$ , and array  $A$  of size  $T$  for the best accuracies.

1. Compute  $k_{max}$  by Equation 2.9.
2. For each iteration  $i = 1, 2, \dots, T$ , do
  3. Set  $Acc^* \leftarrow 0$ .
  4. For each  $k = 1, 2, \dots, k_{max}$ , do
    5.  $M \leftarrow \emptyset$ .
    6. For each class  $l = 1, 2, \dots, c$ , do
      7.  $FP(l) \leftarrow 0$  and  $FN(l) \leftarrow 0$ .
    8. For each sample  $s \in \mathcal{Z}_2$ , do
      9. Classify  $s$  with label  $1 \leq L(s) \leq c$ , as described above.
      10. If  $s \in \mathcal{S}_l$  and  $L(s) \neq l$ , then
        11.  $FP(L(s)) \leftarrow FP(L(s)) + 1$ .
        12.  $FN(l) \leftarrow FN(l) + 1$ .
      13.  $M \leftarrow M \cup t$ .
    14. Compute  $Acc$  by Equation 2.12.
    15. If  $Acc \geq Acc^*$ , then
      16.  $Acc^* \leftarrow Acc$ ,  $M^* \leftarrow M$ ,  $\mathcal{Z}_1^* \leftarrow \mathcal{Z}_1$  and  $k^* \leftarrow k$ .
  17. Save  $(\mathcal{Z}_1^*, k^*)$  and set  $A(i) \leftarrow Acc^*$ .
  18. While  $M^* \neq \emptyset$ 
    19.  $M^* \leftarrow M^* \setminus s$
    20. Replace  $s$  by a randomly selected pixel of the same
    21. class in  $\mathcal{Z}_1$ .
22. Select the instance of  $(\mathcal{Z}_1^*, k^*)$  with maximum accuracy  $A(i)$ ,  $i = 1, 2, \dots, T$ .

Accuracy is measured, as suggested in [92], by taking into account the fact that objects may have different sizes in  $\mathcal{Z}_2$ . If there are two objects, for example, with very different

sizes and the classifier always assigns the label of the largest object, its accuracy will fall drastically due to the high error rate on the smallest object. This accuracy is defined as follows. Let  $N_2(l)$  be the number of pixels of  $\mathcal{S}_l$  in  $\mathcal{Z}_2$ . We first define

$$e_{l,1} = \frac{FP(l)}{|\mathcal{Z}_2| - N_2(l)} \quad \text{and} \quad e_{l,2} = \frac{FN(l)}{N_2(l)}, \quad l = 1, \dots, c, \quad (2.10)$$

where  $FP(l)$  and  $FN(l)$  are the number of false positive and false negative pixels (Lines 11–12), respectively. That is,  $FP(l)$  is the number of pixels from other objects that were classified as being from the object  $l$  in  $\mathcal{Z}_2$ , and  $FN(l)$  is the number of pixels from the object  $l$  that were incorrectly classified as being from other objects in  $\mathcal{Z}_2$ . The errors  $e_{l,1}$  and  $e_{l,2}$  are used to define

$$E(l) = e_{l,1} + e_{l,2}, \quad (2.11)$$

where  $E(l)$  is the partial sum error of object  $l$ . Finally, the accuracy  $Acc$  of classification is expressed as

$$Acc = \frac{2c - \sum_{l=1}^c E(l)}{2c} = 1 - \frac{\sum_{l=1}^c E(l)}{2c}. \quad (2.12)$$

### 2.3.2 Image-based weight assignment

In general, one may use  $0 \leq \hat{d}(s, t) \leq K$  as the image-based weight  $w_i(s, t)$  in Equation 2.1. It is also possible to learn the posterior probability of a pixel (or arc) to be on a boundary from local image attributes [74]. We present another interesting option based on image smoothing at several scales.

Multiscale image smoothing can be accomplished by linear convolutions with Gaussians [66] and/or levelings [76, 119, 103, 102]. Except for Figure 2.15, the other examples in this paper use sequences of opening by reconstruction and closing by reconstruction, computed over each image band  $I_b$ ,  $b = 1, 2, \dots, m$ , for disks of radii  $r = 1, 2, \dots, S$  (e.g.,  $S = 4$  pixels). Gaussian filters provide smoother contours than morphological reconstructions, but the latter may be preferable to better conserve the natural shape indentations and profusions. In Figure 2.15, we illustrate the contour smoothness obtained by Gaussian filters with means equal to 0 and standard deviations  $\sigma = \frac{r}{3}$  for scales  $r = 1, 2, \dots, S = 6$  pixels.

Let  $\vec{v}_b(s) = (v_{b,1}(s), v_{b,2}(s), \dots, v_{b,S}(s))$  be the resulting pixel intensities  $v_{b,j}(s)$ ,  $j = 1, 2, \dots, S$ , of the multiscale smoothing on the image band  $I_b$ ,  $b = 1, 2, \dots, m$ . We compute a gradient vector  $\vec{G}_b(s)$  for each  $s \in D_{\hat{f}}$  and band  $b = 1, 2, \dots, m$ . The idea is the same

as in Equation 2.6, where  $\mathcal{A}$  may be Euclidean with  $\rho = \sqrt{2}$ .

$$\vec{G}_b(s) = \sum_{j=1}^S \sum_{\forall t \in \mathcal{A}(s)} [v_{b,j}(t) - v_{b,j}(s)] \vec{s}t \quad (2.13)$$

$$w_i(s, t) = \max_{b=1,2,\dots,m} \left\{ \left| \frac{\vec{G}_b(s) + \vec{G}_b(t)}{2} \right| \right\}. \quad (2.14)$$

Note that, the gradients  $\vec{G}_b(s)$  are filtered vectors, the gradient orientation of the mean vector of maximum magnitude may be used to modify arc-weight assignment (Section 2.4.7), and the best choice of attributes for a given image should be learned from the selected markers and a database of descriptors. One can select, for example, the descriptor which maximizes the accuracy in Algorithm 1.

## 2.4 Interactive segmentation methods

A segmentation result is represented by a label image  $\hat{L} = (D_{\hat{f}}, L)$ , in which each label  $1 \leq L(s) \leq c$  assigns a pixel  $s \in D_{\hat{f}}$  to one object out of  $c$  objects, including background. For the sake of simplicity, we have considered the case of  $c = 2$  in all examples of this paper. All methods presented in this section have been well published, so we will present only a short description with their graph parameters customized as a function of  $w(s, t)$  and  $\mu(l | \vec{v}(s))$ , although for other methods, different adaptive procedures may be required. The methods based on optimum paths are described by using the image foresting transform (IFT) [43]. We also describe the graph-cut approach based on the min-cut/max-flow algorithm of [17]. What is novel in this section is the way these methods are used in combination as tools in an interactive segmentation paradigm.

### 2.4.1 Image Foresting Transform

The *image foresting transform* (IFT) is a tool for the design, implementation, and evaluation of image processing operators based on connectivity values among pixels [43].

In a given image graph  $(D_{\hat{f}}, \mathcal{A})$ , a path  $\pi_t = \langle t_1, t_2, \dots, t \rangle$  is a sequence of two or more adjacent pixels with the terminus at a pixel  $t \in D_{\hat{f}}$ ,  $\pi_t = \langle t \rangle$  being considered a trivial path. A path  $\pi_t$  is *optimum* under a *path-value function*  $f(\pi_t)$ , when  $f(\pi_t) \leq f(\tau_t)$  for any other path  $\tau_t$ . The IFT computes an *optimum-path forest*  $P$  by minimizing (or maximizing) Equation 2.15 for every  $t \in D_{\hat{f}}$ .

$$V(t) = \min_{\forall \pi_t \text{ in } (D_{\hat{f}}, \mathcal{A})} \{f(\pi_t)\}, \quad (2.15)$$

where  $V(t)$  is the value of the optimum path with terminus  $t$ . The initial pixels of the optimum paths are called *roots* of the forest. By starting with trivial paths  $\pi_t = \langle t \rangle$  for all pixels  $t \in D_f$ , the IFT algorithm (a generalized Dijkstra's algorithm [28], which in practice, executes in linear time in most cases) first identifies the forest roots (minima/maxima of  $V$ ) and then propagates optimum paths to their adjacent pixels, continuing from these nodes to their neighbors, and following a non-decreasing (non-increasing) order of path values, according to the path-propagation rule below.

$$\text{if } f(\pi_s \cdot \langle s, t \rangle) < f(\pi_t) \text{ then } \pi_t \leftarrow \pi_s \cdot \langle s, t \rangle, \quad (2.16)$$

where  $\pi_s \cdot \langle s, t \rangle$  indicates the extension of a path  $\pi_s$  by an arc  $(s, t) \in \mathcal{A}$  (Figure 2.5a). These paths are represented in backwards, where  $P(t)$  indicates the predecessor node of  $t$  in the path  $\pi_t$  and  $R(t)$  is its root pixel for which  $P(R(t)) = \text{nil}$  (Figure 2.5b). An optimum-path forest  $P$  is a function which takes every pixel to *nil* in a finite number of iterations, such that all paths are optimum (Figure 2.5c).

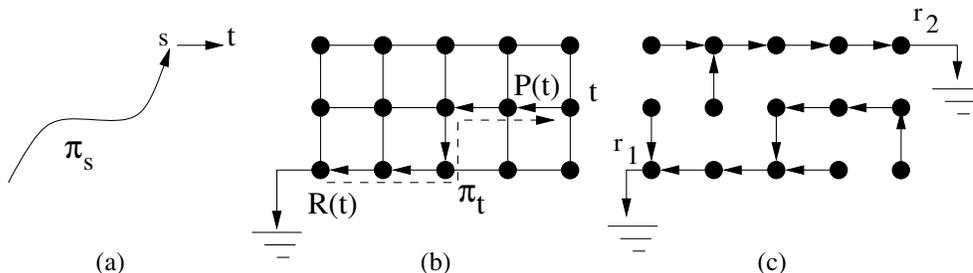


Figure 2.5: (a) Path  $\pi_t = \pi_s \cdot \langle s, t \rangle$  indicates the extension of path  $\pi_s$  by an arc  $(s, t) \in \mathcal{A}$ . (b) A 4-neighborhood graph showing a path  $\pi_t$  (dashed line) represented in backwards, where  $P(t)$  is the predecessor node of  $t$  and  $R(t)$  is the root pixel. (c) A forest  $P$  with two root nodes  $r_1$  and  $r_2$ .

The path-value functions define different IFT-based image operators, which are reduced to a local processing operation on one or more of the output maps  $V$ ,  $P$ , and  $R$  [40, 39, 95, 92, 9, 110]. The IFT algorithm is an optimum region (path) growing process from the roots of the forest (Figure 2.6). Variants can also gather on-the-fly other information, such as a root label for each pixel [68, 36], the propagation order of the pixels [80], the area of the wavefronts of same path value [80], and a graph-cut measure for the border of the growing regions [41].

Particularly, the image segmentation methods described in the following sections adopt a minimization of path-value functions  $f_1$  and  $f_2$ , and some of their variants, such that

the roots of the forest are constrained into the union set  $\mathcal{Z} = \bigcup_{l=1,2,\dots,c} \mathcal{S}_l$  of selected markers. We note that, these IFT algorithms run in linear time independently of  $|\mathcal{Z}|$ .

$$\begin{aligned}
 f_1(\langle t \rangle) &= \begin{cases} H(t) & \text{if } t \in \mathcal{Z} \\ +\infty & \text{otherwise.} \end{cases} \\
 f_1(\pi_s \cdot \langle s, t \rangle) &= \max\{f_1(\pi_s), w(s, t)\}
 \end{aligned} \tag{2.17}$$

$$\begin{aligned}
 f_2(\langle t \rangle) &= \begin{cases} H(t) & \text{if } t \in \mathcal{Z} \\ +\infty & \text{otherwise.} \end{cases} \\
 f_2(\pi_s \cdot \langle s, t \rangle) &= f_2(\pi_s) + w(s, t),
 \end{aligned} \tag{2.18}$$

where  $0 \leq H(t) < \infty$  is a handicap value and  $0 \leq w(s, t) \leq K$  is the fixed arc weight, as described in Section 2.3. Function  $f_1(\pi_t)$  computes the maximum arc weight along  $\pi_t$  and  $f_2(\pi_t)$  computes the sum of the arc weights along  $\pi_t$ .

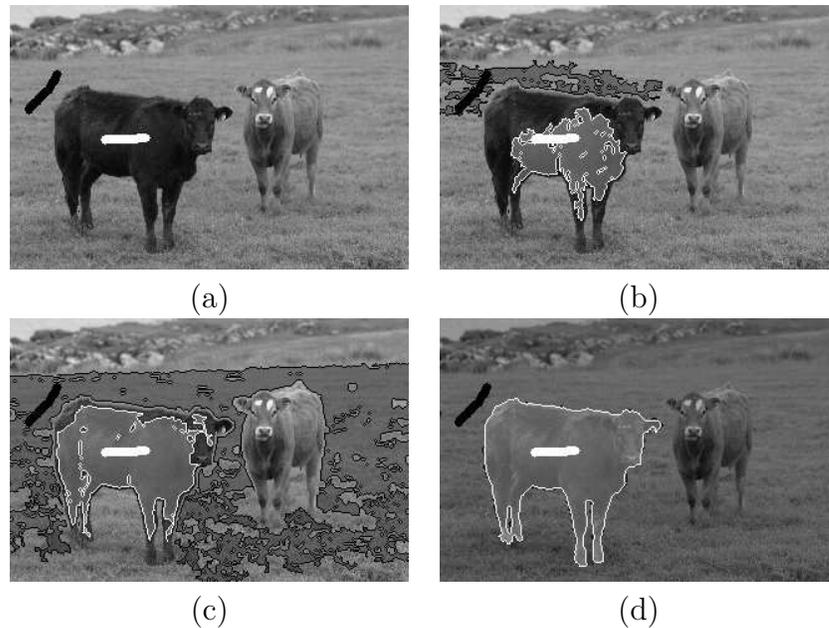


Figure 2.6: (a) An initial marker selection for segmentation. (b-d) The IFT region growing. (d) The regions meet each other at the object's boundary.

## 2.4.2 Segmentation by differential IFT (DIFT)

Multiple objects can be obtained by competition among markers in  $\mathcal{S}_l$ ,  $l = 1, 2, \dots, c$ . By assigning higher arc weights across the desired boundaries, the IFT with  $f_1$  (e.g.,  $H(t) = 0$ )

tends to propagate optimum paths inside the objects before they meet paths from seeds of other objects at the image boundaries (Figure 2.6). Additional seeds are required when this condition is not fully satisfied. Each seed  $r \in \mathcal{Z}$  defines an influence zone (optimum-path tree rooted at  $r$ ) composed of the pixels that are more strongly connected to  $r$  than to any other seed. Each object  $l$  is then defined by the union of the influence zones with that label in  $L$ . This essentially incorporates approaches, such as the watershed transform from markers [11, 68] and iterative relative-fuzzy connectedness [24]. The formal relation that exists between these approaches is studied in [79, 6]. The same strategy with  $f_2$  (e.g.,  $H(t) = 0$ ) would be a segmentation by weighted distance transform [8, 94].

In any case, the user may want to add/remove markers to correct the segmentation results (Figure 2.7). Instead of computing one IFT from the beginning for each new instance of seeds, the DIFT algorithm allows us to recompute the optimum-path forest in time proportional to the number of pixels in the modified regions [36] (sublinear time in practice).

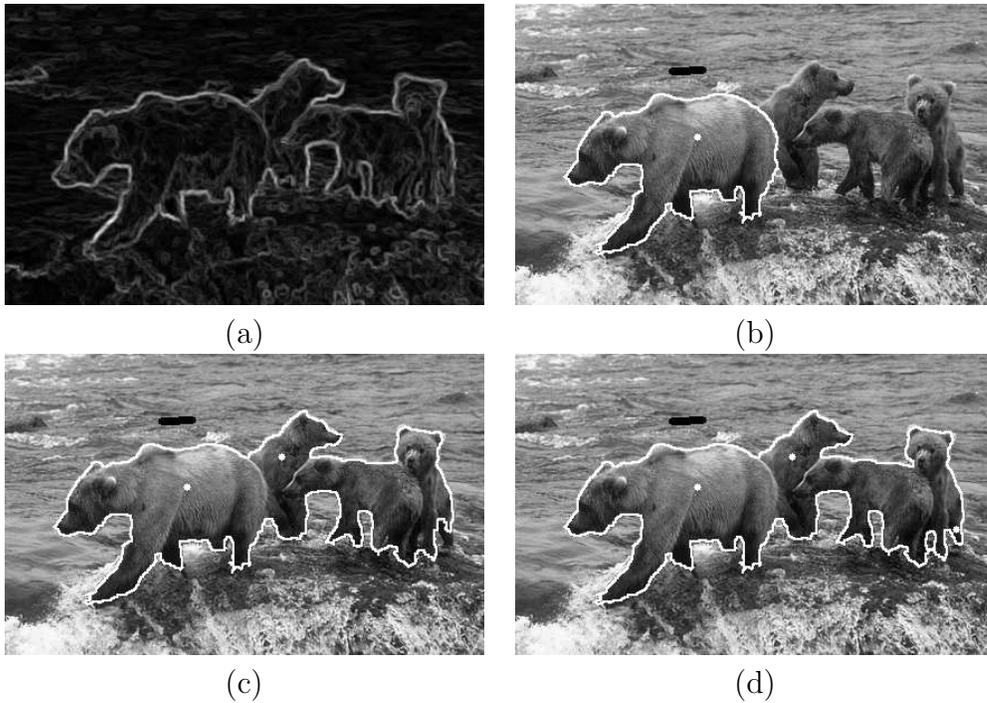


Figure 2.7: (a) A weight image  $\hat{W}$  obtained from Figure 2.4 using  $\lambda = 0.4$ . (b) An initial marker selection and segmentation. (c) An additional seed is inserted in order to include the other bears. (d) The final segmentation after some small corrections.

### 2.4.3 Segmentation by $\kappa$ -connected components

User involvement can be reduced when we exploit other properties of the optimum paths during the IFT algorithm [80]. The IFT with  $f_1$  (with  $H(t) = 0$ ) propagates wavefronts  $\mathcal{W}_u(s)$  of iso-optimum-path value  $u$  around each seed  $s$ , following an increasing order of values  $u = 0, 1, \dots, K$ . The maximal extent of a seed inside an object is defined by a  $\kappa_s$  value as  $\bigcup_{u=0,1,\dots,\kappa_s} \mathcal{W}_u(s)$  (Figure 2.8a). When the competition with external seeds fails (or there is no external seeds) and an optimum path from  $s$  invades the background, it usually crosses the boundary through its weakest link (arc with the lowest weight or *leaking arc*), ramifies and conquers a large region of surrounding pixels with the same path value  $\kappa_s + 1$  (Figure 2.8b). This background invasion is characterized by a considerable increase of  $|\mathcal{W}_u(s)|$ , which can be observed by displaying a curve of the total area  $\sum_{\forall s \in \mathcal{S}_l} |\mathcal{W}_u(s)|$  for  $u = 0, 1, \dots, K$  during propagation (Figure 2.8c). A single area threshold  $0\% < T < 100\%$  on the size  $|\mathcal{W}_u(s)|$  can be used to detect  $\kappa_s$  for all seeds  $s \in \mathcal{S}_l$  and forbid the leaking by stopping the region growing from  $s$ . The object is defined as the subset of pixels which are more strongly  $\kappa$ -connected to its internal seeds than to any other (Figure 2.8d). As discussed in Reference [79], a non-maximal suppression to make the weights in the object's boundary thinner is an adequate preprocessing that should be adopted in these  $\kappa$ -connected methods.

Note that, since the internal seeds also compete among themselves, with distinct  $\kappa_s$  values, the method can work even when leaking occurs before the object is fully segmented (Figure 2.8b). The method usually reduces the number of external seeds required to complete segmentation [80], and it is equivalent to the segmentation by differential IFT when we increase the number of external seeds. That is, it is more general than the previous approach.

### 2.4.4 Segmentation by tree pruning

Another idea to reduce/eliminate external seeds using the IFT with  $f_1$  (with  $H(t) = 0$ ) has been proposed in [38, 9]. In both approaches, the idea is to let the object and the background get connected through the leaking arcs by computing the IFT from internal seeds. The leaking arcs can be then detected interactively [38] or automatically [9]. By removing their subtrees from the forest  $P$ , the remaining forest defines the object. The first approach can handle multiple objects, but we will discuss here only the second approach.

In [9], there is no competition with external markers. They are called an external set  $\mathcal{B}$ , which is used to detect all leaking arcs automatically. In most cases the set  $\mathcal{B}$  is the image's border, but the user can also add external pixels to  $\mathcal{B}$  or internal seeds, if needed. The optimum paths that leak to the background are called *leaking paths*. They

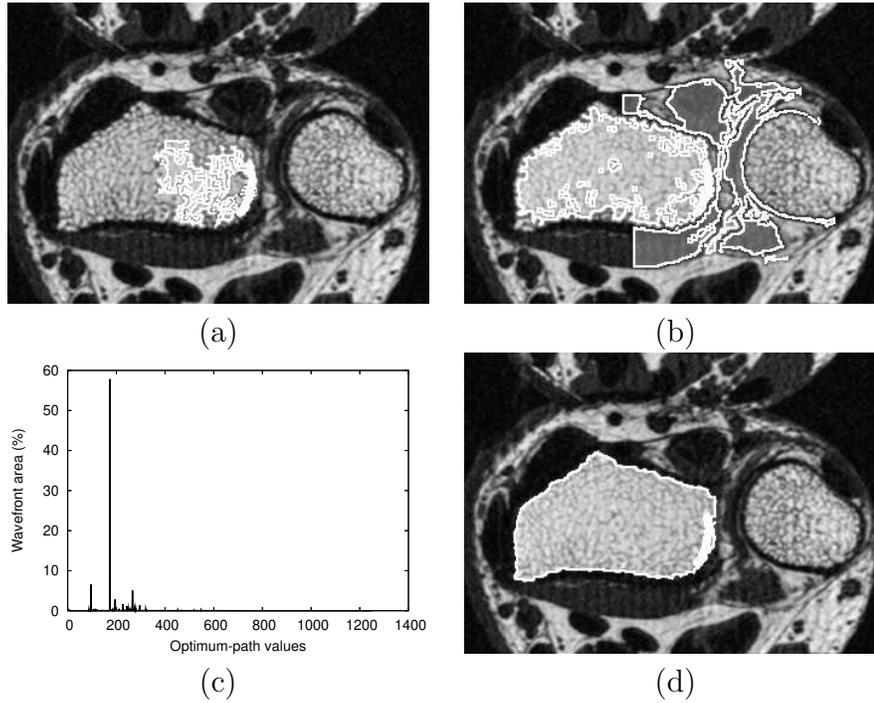


Figure 2.8: (a-b) The IFT region growing from internal seeds. There is a burst in the size of the wavefront when an optimum path reaches the background. (c) The total wavefront area for each optimum-path value  $u = 1, 2, \dots, K$  during propagation. (d) The resulting segmentation with  $\kappa$ -connected components.

can be enhanced by displaying the number of descendants that each forest node has in  $\mathcal{B}$  (Figure 2.9a). Since the leaking paths are ramified after leaking, there is a considerable decrease in the descendant number after the leaking arcs. The method can detect this variation, remove the leaking arcs and output the object (Figure 2.9b). It has been shown that segmentation by tree pruning is less sensitive to the heterogeneity of the background than the watershed transform from markers [9].

### 2.4.5 Segmentation by graph cut

Approaches for graph-cut segmentation are based on objective functions that measure some global property of the object's boundary using the arc weights. The idea is to assign weights to the arcs such that the minimum of this objective function corresponds to the desired segmentation (i.e., a *cut boundary* whose arcs connect the nodes between object and background).

Wu and Leahy [126] were the first to introduce a solution for graph cut using as



Figure 2.9: Example of license plate segmentation. (a) The original image overlaid by the number of descendants in the background set  $\mathcal{B}$ . (b) The resulting segmentation with tree pruning.

measure the sum of the arc weights in the cut boundary. Their cut measure had a bias toward small boundaries, and subsequently, other objective functions, such as average cut [30], mean cut [124], average association [105], normalized cut [106], ratio cut [123], and energy functions [17] have been proposed to circumvent this problem.

Interactive segmentation using the min-cut/max-flow algorithm [17, 63] uses extended image graphs (Figure 2.2), where two terminal nodes  $o$  and  $b$  (*source* and *sink*) represent object ( $l = 1$ ) and background ( $l = 2$ ), respectively, directly connected to all pixels  $s \in D_{\hat{l}}$  by arcs  $(o, s)$  and  $(s, b)$ . A variant of the min-cut/max-flow algorithm from source to sink [48, 15] is then used to speed up computation of the minimum-cut boundary according to the following equation:

$$\begin{aligned}
 E(\hat{L}) &= \sum_{\forall (s,t) \in \mathcal{A} \mid L(s)=1, L(t)=2} K - w(s, t) \\
 &+ \sum_{\forall s \in D_{\hat{l}} \mid L(s)=1} w(s, b) + \sum_{\forall s \in D_{\hat{l}} \mid L(s)=2} w(o, s), \quad (2.19)
 \end{aligned}$$

where  $w(s, b)$  and  $w(o, s)$  can be computed based on the membership values given in Equation 2.5:

$$w(o, s) = \alpha \cdot \mu(l = 1 \mid \vec{v}(s)) \quad (2.20)$$

$$w(s, b) = \alpha \cdot \mu(l = 2 \mid \vec{v}(s)). \quad (2.21)$$

Here,  $\alpha \geq 0$  specifies the relative importance of the arcs with the virtual nodes versus the arcs between pixels (Figure 2.10). As discussed in Reference [79], an interesting adaptive procedure to improve this method is to penalize arcs between pixels with high complemented weights by applying some increasing transformation (e.g., power functions, the exponential function [64]). This is especially important for low  $\alpha$  values since in this case Equation 2.19 becomes almost the same as in Wu and Leahy [126] and it helps to circumvent the undesirable bias.

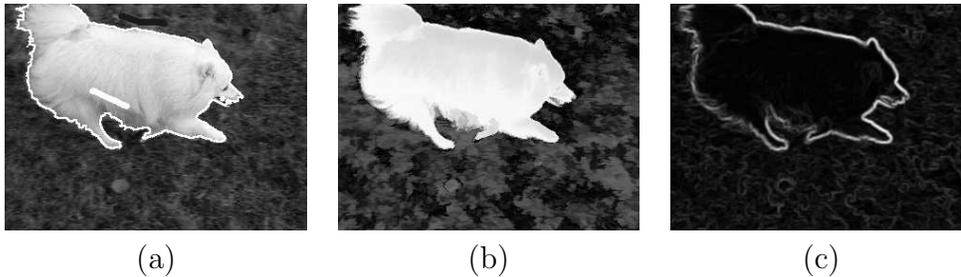


Figure 2.10: Graph-cut segmentation with  $\alpha = 20$  and  $\lambda = 0.5$ . (a) Marker selection for training and the result of segmentation. (b) The membership image  $\hat{M}_1$  used in  $w(o, s)$  and  $w(s, b)$ . (c) The weight image  $\hat{W}$  that reflects  $w(s, t)$ .

If the algorithm fails in delineating the desired boundary, the user forces arc weights with source and sink by adding markers inside and outside the object [17]. The problems related to the simultaneous segmentation of multiple objects are discussed in [16].

#### 2.4.6 Segmentation by IFT with graph cut

If the arc weights are higher on the desired boundary than inside the objects, then the borders of the growing regions from internal seeds must merge and fit to the desired boundary during the IFT propagation with  $f_1$  (with  $H(t) = 0$ ). Such borders work as cut boundaries and different cut measures may be computed on-the-fly for every instant (propagation order of each pixel) during region growing (Figure 2.11a). Within this considerably reduced search space, the minimum cut is expected to occur on the object's boundary (Figure 2.11b). The method has been evaluated for normalized cut, mean cut and energy functions [41]. When the weight condition is not fully satisfied, the desired cut is not a global minimum even within this reduced search space, but the user can add more internal seeds. The reduction of the search space represents a considerable efficiency gain with respect to some graph-cut approaches [106, 123].

We note that, a non-maximal suppression to tune the weights can help this method by allowing a better fit to the boundaries during the internal region growing [79].

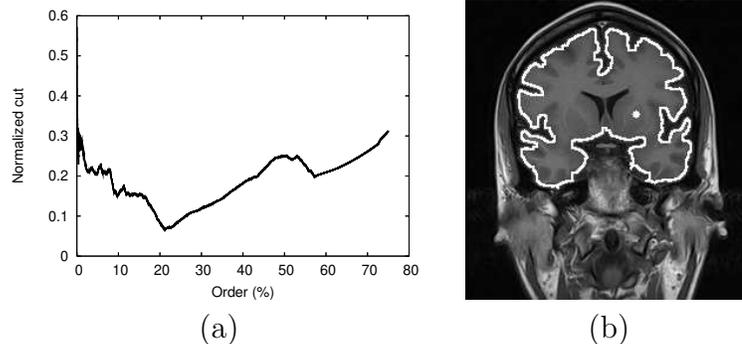


Figure 2.11: Segmentation example of a MR-brain image. (a) The normalized cut versus the pixel propagation order. (b) The respective segmentation.

### 2.4.7 Segmentation by live wire

In order to segment the object with live wire [44], the user selects a starting point on the object's boundary (point  $s_1$  in Figure 2.12a), and, for any subsequent position of the mouse cursor, the method computes an optimum path from  $s_1$  to that position in real time. As the user moves the cursor close to the boundary, the optimum segment snaps on to it. The user can quickly verify the longest segment, as the one with terminus at point  $s_2$  in Figure 2.12b, and deposit the mouse cursor at that position. The process is then repeated from  $s_2$  until the user decides to close the contour (Figures 2.12c-d).

The closed contour is an optimum curve that is constrained to pass through a sequence  $\langle \mathcal{S}^{(1)}, \mathcal{S}^{(2)}, \dots, \mathcal{S}^{(N)} \rangle$  of  $N$  anchor points (seeds) on the object's boundary, in that order, starting from  $\mathcal{S}^{(1)}$  and ending in  $\mathcal{S}^{(N)}$ , where each set  $\mathcal{S}^{(i)}$ ,  $i = 1, 2, \dots, N$ , has a single pixel  $s_i$  and  $s_1 = s_N$ . The optimum curve that satisfies those constraints consists of  $N - 1$  segments  $\pi_{s_2}, \pi_{s_3}, \dots, \pi_{s_N}$ , where each  $\pi_{s_i}$  is an optimum path connecting  $s_{i-1}$  to  $s_i$ . Therefore, we can solve this problem by  $N - 1$  executions of the IFT and the optimum contour can be obtained from the predecessor map  $P$  after the last execution. For  $i = 2, 3, \dots, N$ , the IFT is computed using the initial point  $s_{i-1} \in \mathcal{S}^{(i-1)}$  as seed, 8-adjacency relation and path-value function  $f_3$  (a variant of  $f_2$ ).

$$f_3(\langle t \rangle) = \begin{cases} V(t) & \text{if } t \in \pi_{s_2} \cup \dots \cup \pi_{s_{i-1}} \\ +\infty & \text{otherwise} \end{cases} \quad (2.22)$$

$$f_3(\pi_s \cdot \langle s, t \rangle) = f_3(\pi_s) + (K - \max\{\vec{G}(s, t) \cdot \vec{\eta}(s, t), 0\})^a, \quad (2.23)$$

where  $V(t)$  is the optimum path value of the previous executions,  $a > 0$  (e.g.,  $a =$

1.5),  $0 \leq |\vec{G}(s, t)| \leq K$  is the gradient vector estimated at the midpoint of arc  $(s, t)$ , and  $\vec{\eta}(s, t)$  is the unit vector  $\vec{st}$  rotated 90 degrees counter-clockwise. This formulation favors segmentation on a single orientation, but allows longer boundary segments. We use  $\vec{G}(s, t) = \frac{\vec{G}_l(s) + \vec{G}_l(t)}{2}$  obtained from the membership map, but the image gradient  $\vec{G}(s, t) = \frac{\vec{G}_b(s) + \vec{G}_b(t)}{2}$  of maximum magnitude for  $b = 1, 2, \dots, m$  is also an option, when there are no training markers. The initial path value  $V(s_1) = f_3(\langle s_1 \rangle) = 0$  and we make  $V(s_1 = s_i) = +\infty$  to compute the last segment.

We note that, other variants of live wire can also take advantage of the proposed arc-weight assignment [43, 58, 46, 57, 71, 54].

## 2.5 Experiments and Results

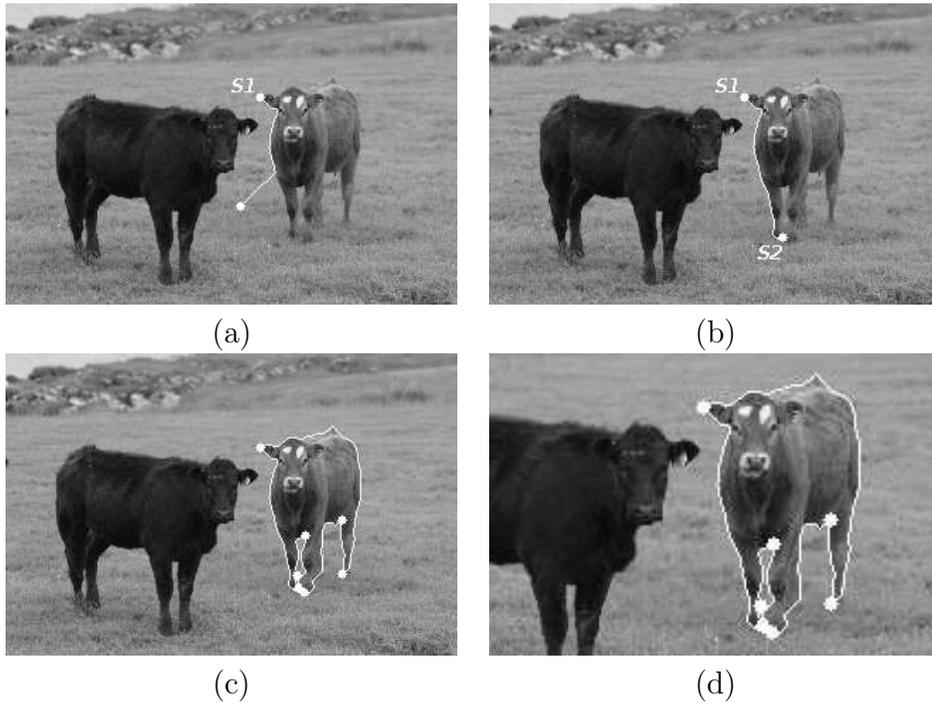


Figure 2.12: Contour tracking with live wire. (a) Initial point  $s_1$  is selected on the boundary and the user moves the mouse. (b) A second point  $s_2$  is selected on the boundary. (c-d) Final contour with 7 segments.

The examples in the previous sections have shown that the proposed process for arc-weight assignment is useful in several image segmentation methods. The synergism between the user and the computer offers some important advantages as well. Object

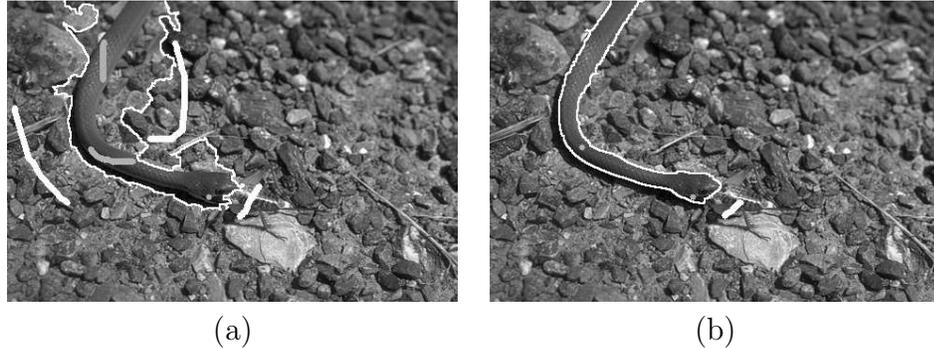


Figure 2.13: (a) Segmentation result of the watershed transform from markers considering only the traditional image-based component (Figure 2.3b). (b) A better result is obtained with less seeds, by the use of object-based weights (Figure 2.3c).

information is incorporated into arc-weight estimation under user supervision and control. In traditional segmentation methods [59, 11, 106, 124], arc-weight estimation is usually treated as a simple embedded process, disregarding, in many cases, the user interventions. For example, the watershed from markers over the weight image (Figure 2.13a) can be drastically improved by incorporating object information as presented in Section 2.3.1 (Figure 2.13b).

The arc-weight assignment process and computation without visual inspection by the user makes it difficult to understand what part is contributing more to the final segmentation result: arc-weight estimation or the segmentation algorithm. Hence, only a common base strategy for arc-weight assignment, with the proper adaptive procedures, allows fair comparisons among methods. For instance, it is easy to see that any approach based on the weights of Figure 2.3b will be at a clear disadvantage when compared with those based on the weights of Figure 2.3c.

Other approaches also incorporate object information into arc-weight estimation [17, 8, 94]. However, the absence of weight visualization and the use of segmentation markers for both arc-weight estimation and delineation make the user lose control over the segmentation process, when there exist ambiguities between object and background properties. Figure 2.14, in which the bigger horse is the object of interest, provides an illustration of this phenomenon. Figure 2.14a shows the DIFT segmentation from the same markers used for training and delineation. The corresponding membership image  $\hat{M}_1$  and weight image  $\hat{W}$  used for arc-weight assignment are shown in Figures 2.14b and 2.14c ( $\lambda = 0.5$ ). Note that the segmentation fails owing to the weak boundary between the bigger and the smaller horses, which have similar image properties. Additional markers can correct segmentation in a differential way (Figure 2.14d). However, arc weights

should never be recomputed from the new markers. If we do that, the membership image  $\hat{M}_1$  gets destroyed (Figure 2.14e) and the segmentation results would not be correct (Figure 2.14f). This explains the importance of having arc-weight estimation as a separated training step from image segmentation. During training, the user should select the most representative and distinguishable parts of the objects, and leave corrections to the interactive segmentation session, in order to avoid arc-weight estimation based on exceptions.

The visual feedback during training also assists the user in choosing the image segmentation method which is likely to require less markers. Figure 2.15a illustrates the DIFT segmentation of the left caudate nucleus in an MR-image, using the same markers for training and delineation. The corresponding membership image  $\hat{M}_1$  (Figure 2.15b) and weight image  $\hat{W}$  (Figure 2.15c) for  $\lambda = 0.5$  indicate that the arc weights on the object’s boundary are not strictly higher than inside and outside the boundary. However, when they indeed are greater, only one internal seed and one external seed are enough to complete segmentation by DIFT. Since this is not the case, segmentation would fail if we remove the external marker on the lateral ventricle (dark part) and additional markers are actually needed to refine the results shown in Figure 2.15a. However, arc weights seem to be higher on the object’s boundary than inside. This favors other methods such as segmentation by  $\kappa$ -connected components (Figure 2.15d), which provides the desired segmentation with only one internal seed. The nearby boundaries with similar properties would make more than two seeds required to complete segmentation with live wire (Figure 2.15e). Graph-cut segmentation fails because object and background have similar properties (Figure 2.15f). Correction in this case is impractical.

In order to validate the synergistic arc-weight estimation method, we need to show that it can really improve accuracy and efficiency (in terms of the amount of user help required) of a given segmentation method as compared to the direct approach based only on  $w_i$ , which does not require user assistance. We demonstrate this for one of the methods, namely DIFT, described in Section 2.4.2, instead of evaluating all methods in this manner since the latter is not likely to generate new insight. We used DIFT with ( $\lambda = 0.5$ ) and without  $w_o$  ( $\lambda = 0$ ) to segment different objects in 100 MRI and CT slice images. Among these, 40 slice images came from MRI-T1 acquisition simulations of phantoms (available at the BrainWeb site<sup>2</sup> [25]), 40 slice images were selected from CT cervical spine studies of 10 subjects, and 20 slice images came from CT thoracic studies of 10 subjects. This gave us a total of 100 2D-segmentations for each method as shown in Table 2.1. Sample objects are shown in Figure 2.16.

The ground-truth segmentations were available for each object and they were used to compute the following accuracy measures: the normalized number of false positive pixels

---

<sup>2</sup>URL: <http://www.bic.mni.mcgill.ca/brainweb/>

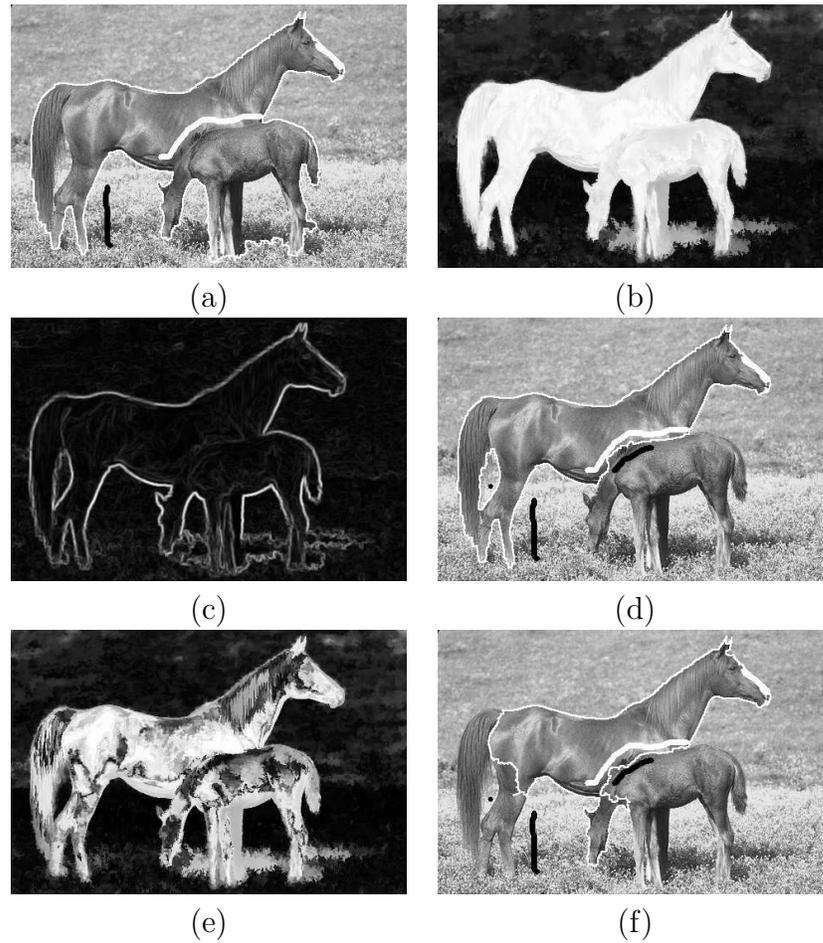


Figure 2.14: (a) DIFT segmentation using the same markers for training and delineation. (b-c) The respective membership image  $\hat{M}_1$  and weight image  $\hat{W}$  of the training ( $\lambda = 0.5$ ). (d) The correct segmentation is obtained with additional markers, which should never be used to recompute weights. (e) The membership image  $\hat{M}_1$  is destroyed if we recompute weights from the additional markers, affecting (f) the result of segmentation.

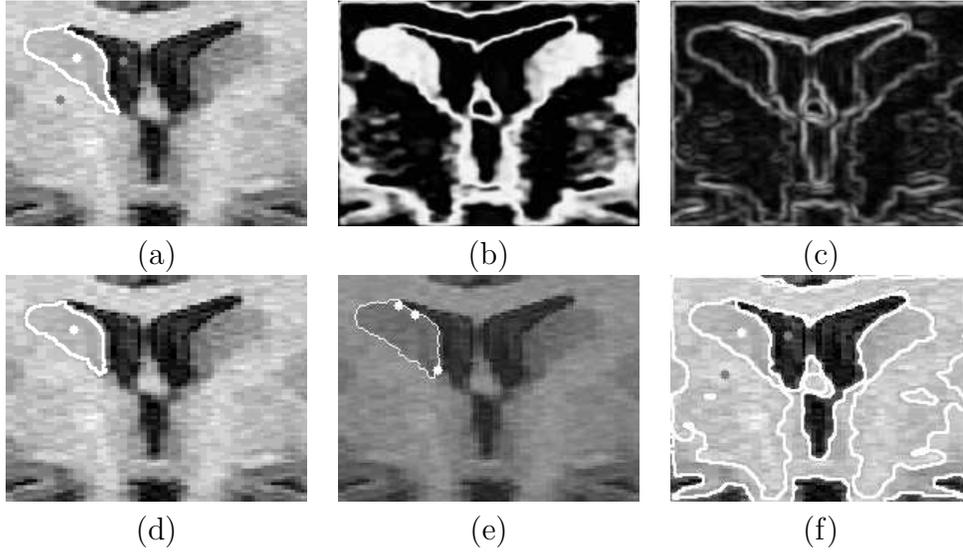


Figure 2.15: (a) The same training markers are used to delineate the left caudate nucleus by the DIFT algorithm. Additional markers are needed to refine segmentation. (b-c) The corresponding membership image  $\hat{M}_1$  and weight image  $\hat{W}$  for  $\lambda = 0.5$ . (d-f) Segmentations by  $\kappa$ -connected component, live wire and graph cut.

$m1 = FP/(FP+TN)$ , where  $FP+TN$  is the sum of the number of false positive and true negative pixels, the normalized number of false negative pixels  $m2 = FN/(FN + TP)$ , where  $FN + TP$  is the sum of the number of false negative and true positive pixels [113], and Dice similarity [32]. Efficiency can be measured by the number of markers (marking actions) required by the user to complete segmentation. The method with less markers requires less user involvement. Note that, in interactive segmentation, accuracy depends on the user's patience for verification and correction. In practice, the user tends to stop the corrective actions when the efforts needed to improve the results increase too much relative to the returned improvement in accuracy. Therefore, high accuracy with less number of markers is highly desirable. The mean and standard deviation of the accuracy and efficiency measures estimated in our experiments are presented for the DIFT with and without  $w_o$  in Tables 2.2 and 2.3, respectively.

In order to simulate a real environment, the similarity measure should be maximized as much as possible until the corrections become almost manual and impractical (i.e., until only small differences distributed along the boundary remain). In the case of O1 with  $w_o$ , about 5 – 10 markers are needed to achieve more than 90% of Dice similarity. This result is already equivalent to the best obtained without  $w_o$  (line 1 of Table 2.3), and can be further improved by adding more markers for small corrections (line 1 of Table 2.2). For

Object	Description	Modality	#Images
O1	Spinal-vertebra	CT	40
O2	White matter ( $N = 3\%$ and $INU = 20\%$ )	MRI-T1	20
O3	White matter ( $N = 5\%$ and $INU = 40\%$ )	MRI-T1	20
O4	Liver	CT	20

Table 2.1: Description, imaging modality and number of slice images for each object used in the experiments. Objects O2 and O3 use sample slices from phantoms with different degrees of noise ( $N$ ) and inhomogeneity ( $INU$ ).

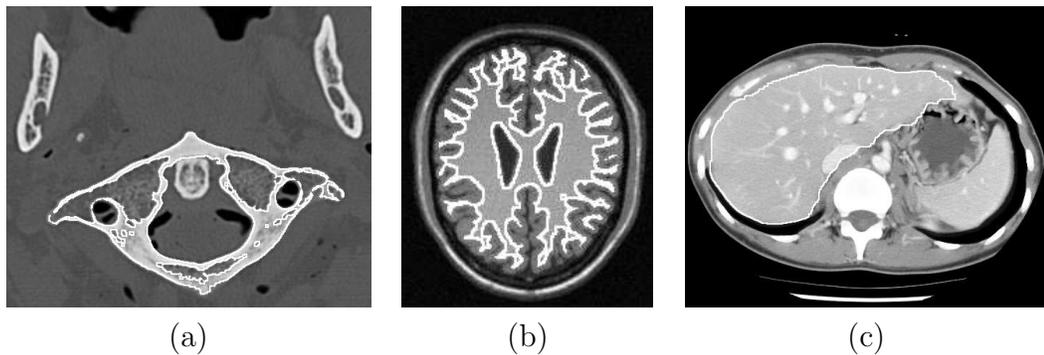


Figure 2.16: Sample slice images showing objects O1, O3 and O4. Object O2 is the same as object O3, but on images with less noise and inhomogeneity.

	Dice		m1		m2		#markers	
	Mean	Std dev	Mean	Std dev	Mean	Std dev	Mean	Std dev
O1	0.9329	0.0115	0.00277	0.00073	0.0830	0.0256	17.8	5.5
O2	0.9767	0.0020	0.00732	0.00096	0.0205	0.0038	13.6	4.6
O3	0.9663	0.0028	0.01059	0.00186	0.0297	0.0039	18.7	4.9
O4	0.9841	0.0027	0.00293	0.00118	0.0180	0.0045	11.1	3.0

Table 2.2: Segmentation results using DIFT with  $w_o$  ( $\lambda = 0.5$ ).

objects O2 and O3, in general, only 4 – 5 markers are sufficient to obtain more than 95% of Dice when using  $w_o$ , while three times more interactions are required without  $w_o$  to reach 90% similarity with the true segmentation. As we keep adding more markers, the results converge to the values listed in Tables 2.2 and 2.3. In the case of O4, the results with and without  $w_o$  are very similar. We already have more than 91% of Dice similarity with 3 markers and about 96% with 6 markers.

The results indicate that the use of  $w_o$  usually provides higher accuracy and higher efficiency. By considering the standard deviation, the DIFT with  $w_o$  presented better performance than the DIFT without  $w_o$  in all cases, except in the case of object O4, where both were equivalent. In all cases, it was enough to consider only two markers to compute  $w_o$ , and these markers were also used to start delineation (i.e., they are counted in Table 2.2). The use of  $w_o$  enhances the desired boundaries and suppresses unwanted borders (see Figure 2.17). This explains the reduction of markers required to complete segmentation, as verified in all cases (Tables 2.2 and 2.3). Since the remaining errors are distributed along the boundary, the similarity measure for each object will vary according with the perimeter/area ratio. Note that objects with complex shapes (O1) produce lower similarity values as compared to simple shapes (O4).

Although we used the DIFT for the experiments, there may be more adequate methods depending on the application. For example, the visual feedback during the training of object O1 (Figures 2.18a-b) indicates that the graph-cut approach is likely to require less markers. Note that object O1 contains several background parts (holes) inside it, and at least one background seed at each hole will be required by the DIFT. Indeed, results similar to those of line 1 in Table 2.2 can be obtained with the graph-cut approach by using only two markers for training (Figure 2.18a) and four markers to remove the background bones during delineation (Figure 2.18c). Note, however, that the use of  $w_o$  is imperative in the graph-cut approach. By choosing  $\alpha = 40$ , the method becomes practically a threshold on the membership map (Figure 2.18b) followed by corrections. This gives another strong indication of the importance of  $w_o$ .

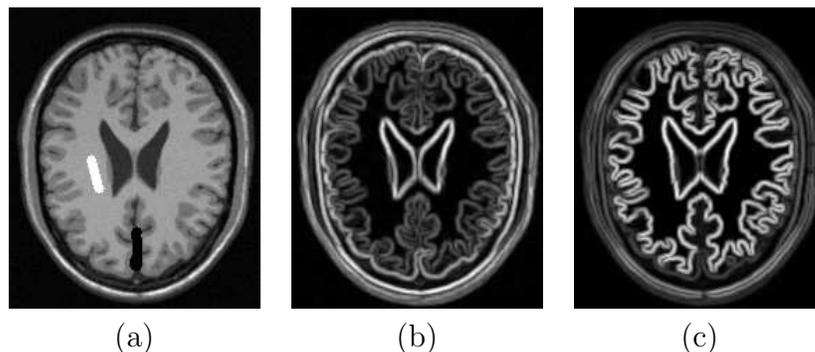


Figure 2.17: (a) Sample slice of object O2 with training markers. (b-c) The corresponding weight images  $\hat{W}$  for  $\lambda = 0$  and  $\lambda = 0.5$ , respectively. By using  $w_o$  a suitable boundary enhancement is obtained for the white matter.

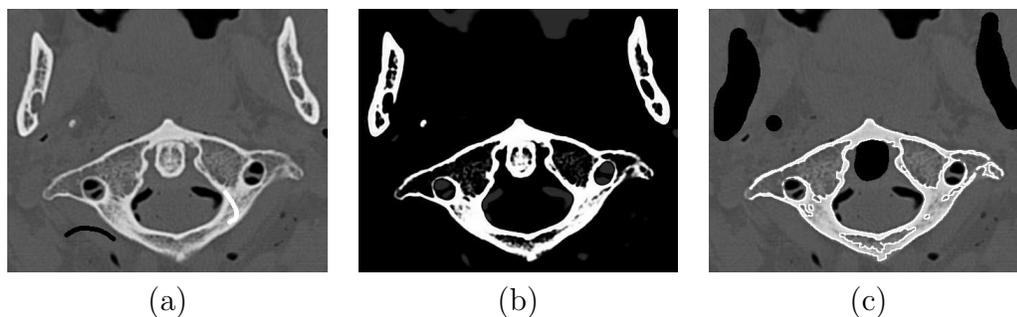


Figure 2.18: (a) Sample slice of object O1 with training markers. (b) The corresponding membership image  $\hat{M}_1$ . (c) Graph-cut segmentation with  $\alpha = 40$  and  $\lambda = 0.5$ . The background bones are easily removed using 4 markers with a larger brush.

	Dice		m1		m2		#markers	
	Mean	Std dev	Mean	Std dev	Mean	Std dev	Mean	Std dev
O1	0.9058	0.0109	0.00347	0.00054	0.1219	0.0196	31.6	7.4
O2	0.9621	0.0039	0.01836	0.00316	0.0128	0.0030	26.0	9.5
O3	0.9562	0.0041	0.02064	0.00295	0.0168	0.0034	28.9	9.8
O4	0.9862	0.0024	0.00261	0.00081	0.0152	0.0034	13.4	3.5

Table 2.3: Segmentation results using DIFT without  $w_o$  ( $\lambda = 0$ ).

## 2.6 Conclusion

We have presented an interactive method for arc-weight estimation, which can be employed effectively by several graph-based segmentation approaches as we demonstrated. Our method exploits in a synergistic way the human abilities for recognition and the computer abilities for delineation. While the user draws markers inside each object (including background), arc weights are estimated from image attributes and object information (pixels under the markers), and a visual feedback guides the user's next action toward improving accuracy. Markers should be drawn on the most representative and distinguishable parts of the objects in order to make arc-weight estimation effective. The training markers can be used to start delineation and additional markers selected on similar parts of the objects can correct segmentation, but they should never be used to recompute weights that are estimated in the training step.

We validated the method by showing that the combination of object-based weights with image-based weights usually improves accuracy and efficiency in interactive segmentation, as compared to the same method with only image-based weights. More recently, this approach for arc-weight estimation together with object delineation by DIFT was also successfully used to improve object tracking in video, handling partial occlusion, camera motion, and deformable objects [77]. In this approach, after interactive image segmentation in a first frame, the method combines motion estimation with automatic segmentation of the remaining frames using the proposed framework. In interactive segmentation, the importance of weight visualization to choose the most suitable segmentation approach was also evident from the examples presented in the paper. The selection of the best image attributes, however, requires further investigation. These attributes can be learned from the drawn markers. Another area that requires further work, which is a current limitation of the method, is the user action of how to draw the markers in an effective manner. This is at present somewhat of an art. Our future work will focus on these directions.

## Acknowledgments

The authors thank FAPESP and CNPq for the financial support. The images of Figures 2.3, 2.4, 2.6, 2.10, and 2.14 have been obtained from <http://www.eecs.berkeley.edu/Research/Projects/CS/vision/grouping>

## Capítulo 3

# Links Between Image Segmentation based on Optimum-Path Forest and Minimum Cut in Graph

### 3.1 Introduction

Discrete Mathematics provides an elegant framework for image processing, rich of efficient algorithms with proofs of correctness. As a consequence, many image segmentation methods have been modeled as graph-search problems.

Two popular approaches exploit undirected and weighted image graphs, where the pixels are the nodes, the arcs are defined by an adjacency relation, and the arc weights are similarity values computed based on image properties. The first approach minimizes a functional of the arc weights leading to a cut in the graph that separates object and background [106, 17, 15, 123]. The second approach [43], called image foresting transform (IFT), can reduce segmentation to the computation of an *optimum-path forest* according to a *connectivity function*, which assigns a value to any path in the graph, including *trivial* paths formed by a single node. That is, considering the maximum value among all possible paths with terminus at each node, the optimum path is trivial for some nodes, called *roots*, and the remaining nodes will have an optimum path coming from their most strongly connected root, partitioning the graph into an optimum-path forest (disjoint sets of optimum-path trees). Two distinct region-based segmentation paradigms, with internal and external seeds (i.e., roots by imposition) and with only internal seeds, can be solved by IFT. In the first paradigm, internal and external seeds compete with each other for their most strongly connected pixels, such that the image is partitioned into two optimum-path forests — one rooted at the internal seeds, defining the object, and the other rooted at the external seeds, representing the background [68, 36]. The second

paradigm solves segmentation by computing one optimum-path forest from only internal seeds and applying some other criterion to cut optimum paths such that the remaining forest defines the object [80, 9, 38, 41]. Indeed, both paradigms can be easily extended to multiple objects, but we will focus on binary image segmentation (object/background) with interactive seed selection. Some variants have also shown the importance of a hybrid paradigm [80].

These approaches based on optimum-path forest and minimum cut in graph are usually regarded to as unrelated. Recently, some links between them were clarified for a particular case, in which an increasing transformation is applied to all arc weights [1]. The present work advances the state of the art in graph-based image segmentation by better clarifying the relation between these approaches. We theoretically prove that some IFT-based methods from both paradigms, with internal and external seeds and with only internal seeds, indeed minimize some graph-cut measures. These IFT-based methods are closely related to popular segmentation approaches, such as *absolute-fuzzy connectedness* (AFC) [116], *relative-fuzzy connectedness* (RFC) [115, 99], *iterative relative-fuzzy connectedness* (IRFC) [24] and *watershed transforms from markers* (WT) [11]. We clarify their differences and the advantages of the IFT-based approach. These methods have been successfully used in many applications [62, 86, 114, 88, 52, 36, 112], which validates the importance of these theoretical results. In view of that, we also extend the theorem stated in [1] by establishing the necessary conditions to its converse. The results provide better understanding of the methods and help the selection of the best algorithm for a given application.

Section 3.2 presents the basic notions on image graphs and Section 3.3 presents the concepts about the IFT, which will be used for image segmentation based on optimum-path forest in Section 3.4. The approach based on minimum cut in graph is briefly described in Section 3.5. Sections 3.6, 3.7 and 3.8 present the theorems and their corresponding proofs. Taking into account the theoretical results, some comparative analysis involving methods from both approaches is carried out along the text, but in Section 3.9 the analysis takes into account only IFT with internal and external seed competition and the min-cut/max-flow segmentation. Our conclusions are stated in Section 3.10.

## 3.2 Basic Concepts on Image Graphs

A multi-dimensional and multi-spectral image  $\hat{I}$  is a pair  $(\mathcal{I}, \vec{I})$  where  $\mathcal{I} \subset Z^n$  is the image domain and  $\vec{I}(t)$  assigns a set of  $m$  scalars  $I_i(t)$ ,  $i = 1, 2, \dots, m$ , to each pixel  $t \in \mathcal{I}$ . The subindex  $i$  is removed when  $m = 1$ .

An *adjacency relation*  $\mathcal{A}$  is a binary relation on  $\mathcal{I}$ . We use  $t \in \mathcal{A}(s)$  and  $(s, t) \in \mathcal{A}$  to

indicate that  $t$  is adjacent to  $s$ . Once the adjacency relation  $\mathcal{A}$  has been fixed, the image  $\hat{I}$  can be interpreted as a graph  $(\mathcal{I}, \mathcal{A})$  whose nodes (or vertices) are the image pixels in  $\mathcal{I}$  and whose arcs are the pixel pairs  $(s, t)$  in  $\mathcal{A}$ . We are interested in irreflexive and symmetric relations. For example, one can take  $\mathcal{A}$  to consist of all pairs of pixels  $(s, t)$  in the Cartesian product  $\mathcal{I} \times \mathcal{I}$  such that  $d(s, t) \leq \rho$  and  $s \neq t$ , where  $d(s, t)$  denotes the Euclidean distance and  $\rho$  is a specified constant (e.g., 4-neighborhood, when  $\rho = 1$ , and 8-neighborhood, when  $\rho = \sqrt{2}$ , in case of 2D images).

Each arc  $(s, t) \in \mathcal{A}$  has a fixed weight  $w(s, t) \geq 0$  which may be computed from local image and object properties extracted from  $\vec{I}$  and some global information (e.g., markers [83]). Graph-cut segmentation methods [106, 123, 17] usually assign lower weights to arcs across the object's boundary (i.e., an affinity measure between pixels  $s$  and  $t$ ) while some methods based on optimum-path forest [68] work with higher arc weights across the object's boundary (i.e., a dissimilarity measure). However, the latter may be easily adapted to the first scheme [6] and, therefore, lower arc weights across the object's boundary will be considered without loss of generality. For example, one may use the complement of a gradient magnitude (i.e.,  $I_{max} - \frac{I(s)+I(t)}{2}$  for a gradient image  $\hat{I}$  with maximum value  $I_{max}$ ). In this work we consider only undirected and weighted graphs. That is, the adjacency relation is symmetric and  $w(s, t) = w(t, s)$  for all  $(s, t) \in \mathcal{A}$ .

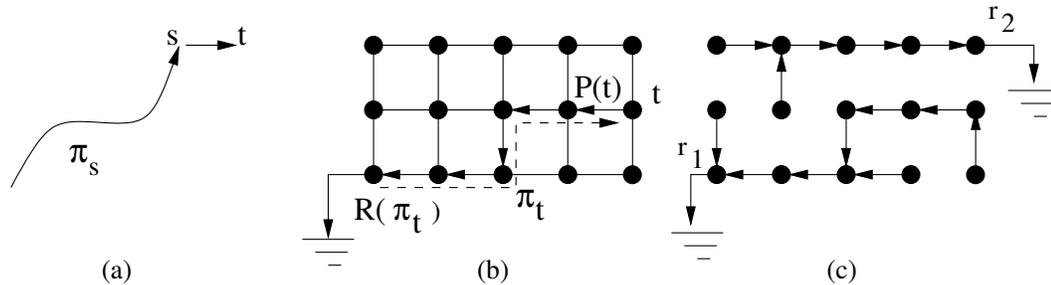


Figure 3.1: (a) Path  $\pi_t = \pi_s \cdot \langle s, t \rangle$  indicates the extension of path  $\pi_s$  by an arc  $(s, t)$ . (b) A 4-neighborhood graph showing a path  $\pi_t$  (dashed line) represented in backwards, where  $P(t)$  is the predecessor node of  $t$  and  $R(\pi_t)$  is the root pixel. (c) A spanning forest  $P$  with two root nodes,  $r_1$  and  $r_2$ .

For a given image graph  $(\mathcal{I}, \mathcal{A})$ , a path  $\pi_t = \langle t_1, t_2, \dots, t \rangle$  is a sequence of adjacent pixels with terminus at a pixel  $t$ . A path is *trivial* when  $\pi_t = \langle t \rangle$ . A path  $\pi_t = \pi_s \cdot \langle s, t \rangle$  indicates the extension of a path  $\pi_s$  by an arc  $(s, t)$  (Figure 3.1a). All paths considered in this work are simple paths, that is, paths with no repeated vertices (pixels).

A *predecessor map* is a function  $P$  that assigns to each pixel  $t$  in  $\mathcal{I}$  either some other adjacent pixel in  $\mathcal{I}$ , or a distinctive marker *nil* not in  $\mathcal{I}$  — in which case  $t$  is said to be

a *root* of the map. A *spanning forest* is a predecessor map which contains no cycles — i.e., one which takes every pixel to *nil* in a finite number of iterations (Figures 3.1b and 3.1c, where  $R(\pi_t)$  is a root node and  $P(t)$  is the predecessor node of  $t$  in the path  $\pi_t$ ). For any pixel  $t \in \mathcal{I}$ , a spanning forest  $P$  defines a path  $\pi_t$  recursively as  $\langle t \rangle$  if  $P(t) = \text{nil}$ , and  $\pi_s \cdot \langle s, t \rangle$  if  $P(t) = s \neq \text{nil}$ .

### 3.3 Image Foresting Transform (IFT)

A *connectivity function* computes a value  $f(\pi_t)$  for any path  $\pi_t$ , usually based on arc weights. Let  $\Pi(\mathcal{I}, \mathcal{A}, t)$  be the set of all paths in the graph  $(\mathcal{I}, \mathcal{A})$  with terminus at  $t$ . In this work, a path is *optimum* according to the following definition.

**Definition 1 (Optimum path)** *A path  $\pi_t$  is optimum if  $f(\pi_t) \geq f(\tau_t)$  for any other path  $\tau_t \in \Pi(\mathcal{I}, \mathcal{A}, t)$ .*

By taking to each pixel  $t \in \mathcal{I}$  one optimum path with terminus  $t$ , we obtain the optimum-path value  $V(t)$ , which is uniquely defined by

$$V(t) = \max_{\forall \pi_t \in \Pi(\mathcal{I}, \mathcal{A}, t)} \{f(\pi_t)\}. \quad (3.1)$$

The *image foresting transform* (IFT) algorithm solves the above optimization problem by dynamic programming [43]. The IFT takes an image  $\hat{I}$ , a path-value function  $f$  and an adjacency relation  $\mathcal{A}$ ; and assigns one optimum path  $\pi_t$  to every pixel  $t \in \mathcal{I}$  such that an *optimum-path forest*  $P$  is obtained — i.e., a spanning forest where all paths are optimum. However,  $f$  must be *smooth*, that is, satisfy Definition 2, as demonstrated in [43]. The attributes of the forest include the map  $V$ , the roots  $R(\pi_t)$ , root labels  $L(t)$ , and the predecessor  $P(t)$  of  $t$  in the optimum path. The image operators are then reduced to a local processing of these attributes [43]. If we consider  $\leq$  instead of  $\geq$  in Definition 1, and minimize  $V(t)$  in Equation 3.1, we obtain a dual definition of optimality. To convert a problem to its dual form, we simply have to invert the sign of the path-value function (i.e.,  $g(\pi_t) = -f(\pi_t)$ ). It is important to state that the IFT was originally presented in this equivalent dual form [43]. In this paper we consider the first schema (Definition 1 and Equation 3.1).

**Definition 2 (Smooth path-value function)** *A path-value function  $f$  is smooth if for any pixel  $t \in \mathcal{I}$ , there is an optimum path  $\pi_t$  which either is trivial, or has the form  $\tau_s \cdot \langle s, t \rangle$  where*

(C1)  $f(\tau_s) \geq f(\pi_t)$ ,

(C2)  $\tau_s$  is optimum,

(C3) for any optimum path  $\tau'_s$ ,  $f(\tau'_s \cdot \langle s, t \rangle) = f(\pi_t)$ .

An interesting property of an optimum-path forest is that any path starting in a root node is also a complete optimum path (path-value function must be *smooth*), according to the following definition.

**Definition 3 (Complete optimum path)** *A path  $\pi_{t_n} = \langle t_1, t_2, \dots, t_n \rangle$  is complete optimum if all paths  $\pi_{t_i} = \langle t_1, t_2, \dots, t_i \rangle$ ,  $i = 1, 2, \dots, n$  are optimum paths.*

Note that, the applications for the image foresting transform (IFT) go beyond region-based image segmentation [44, 40, 39]. In this work, however, we are only interested in region-based image segmentation by IFT. It is also important to state that although we are restricting our analysis to undirected graphs, the IFT can also handle segmentation using directed graphs [43].

### 3.4 Region-based segmentation using IFT

The segmentation of an image is represented by a labeled image  $\hat{L} = (\mathcal{I}, L)$ , where  $L(t) = 1$  for object pixels and  $L(t) = 0$  for background pixels. Each segmentation defines an *induced cut boundary*  $\mathcal{C}$  in the graph.

**Definition 4 (Induced cut boundary)** *The segmentation given by a labeled image  $\hat{L}$  defines an induced cut boundary in the graph, which is the set  $\mathcal{C}$  of arcs  $(s, t)$  such that  $L(s) = 1$  and  $L(t) = 0$ .*

We consider image segmentation from two seed sets,  $\mathcal{S}_1$  and  $\mathcal{S}_0$  ( $\mathcal{S}_1 \cap \mathcal{S}_0 = \emptyset$ ), containing pixels interactively selected inside and outside the object, respectively. A feasible segmentation must satisfy these sets of hard constraints.

**Definition 5 (Feasible segmentation)** *The segmentation given by a labeled image  $\hat{L}$  is feasible if  $L(t) = 1$  for all  $t \in \mathcal{S}_1$  and  $L(t) = 0$  for all  $t \in \mathcal{S}_0$ .*

We are interested in the particular case of smooth path-value functions, the monotonically decremental path-value function  $f_{\min}$ . This function basically assigns to

any path  $\pi_t$  the minimum arc-weight along  $\pi_t$ . Equation 3.2 presents it in the recursive form.

$$\begin{aligned} f_{\min}(\langle t \rangle) &= \begin{cases} w_{\max} + 1 & \text{if } t \in \mathcal{S}_1 \cup \mathcal{S}_0 \\ -\infty & \text{otherwise} \end{cases} \\ f_{\min}(\pi_s \cdot \langle s, t \rangle) &= \min\{f_{\min}(\pi_s), w(s, t)\}, \end{aligned} \quad (3.2)$$

where  $w_{\max}$  represents the maximum arc-weight in the graph and the search for optimum paths is constrained to start in  $\mathcal{S}_1 \cup \mathcal{S}_0$  (roots by imposition).

There are basically two distinct region-based segmentation paradigms by optimum-path forest, with internal  $\mathcal{S}_1$  and external  $\mathcal{S}_0$  seeds and with only internal seeds  $\mathcal{S}_1$  (i.e.,  $\mathcal{S}_0 = \emptyset$ ).

### 3.4.1 IFT with only internal seeds

This region-based segmentation paradigm solves segmentation by computing one optimum-path forest from only internal seeds and applying some pruning criterion to cut optimum paths such that the remaining forest defines the object [80, 9, 38, 41]. We focus on a particular method of this paradigm denoted by IFT-CT (*IFT segmentation by Connectivity Threshold*), which is related to the classical method called *absolute-fuzzy connectedness* (AFC) [116].

In IFT-CT, seeds are specified inside the object and the optimum-path value from the seed set is computed to each pixel, such that the object is obtained by thresholding the resulting connectivity map  $V(t)$  (Equation 3.1). The resulting segmentation is defined as the maximal subset of  $\mathcal{I}$ , leading to a feasible segmentation (Definition 5 with  $\mathcal{S}_0 = \emptyset$ ), wherein all pixels  $t$  are reached by optimum paths whose values  $V(t)$  are greater than or equal to a given threshold  $\kappa$  (see Figure 3.2).

In AFC [116], the graph is implicitly defined by arcs with fixed weights computed by a fuzzy affinity relation, which encodes both the adjacency relation  $\mathcal{A}$  and the arc weights  $w(s, t)$ . While in IFT-CT, the adjacency relation and arc weights are defined separately. For multiple seeds, a single linear-time execution of the IFT-CT is enough to compute the connectivity map  $V(t)$ . However, given that AFC does not allow connectivity through zero-weighted arcs, their results may differ. Now, if we allow different arc weights for each seed, function  $f_{\min}$  is no longer *smooth* (Definition 2) [43]. In this case, the object must be defined as the union of all individual IFT-CT segmentation results, computed for each seed separately [80].

Several other methods exist [80, 9, 38, 41] with different pruning criteria, aiming to improve IFT-CT. In [80], a variant is proposed with multiple  $\kappa$  automatic thresholds leading to greater flexibility. Given that, the IFT algorithm computes optimum paths

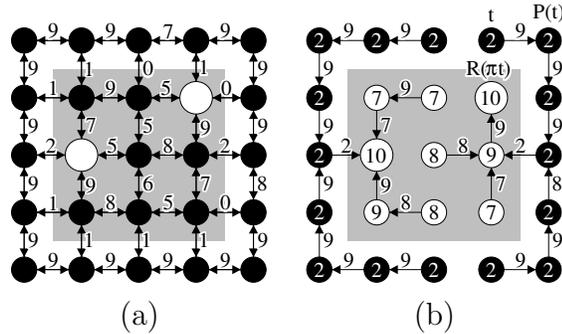


Figure 3.2: (a) A 4-neighborhood graph, where the numbers indicate the arc weights and the object is the *shaded square*. Two seeds are selected inside the object (bigger white dots). (b) An optimum-path forest for the path-value function defined via Eq. 3.2. The numbers inside the nodes indicate the values of the optimum paths (Eq. 3.1). The object is obtained as the white dots after using a threshold  $\kappa = 3$ .

progressively from the seeds by extending optimum paths already computed [43], we have an ordered region growing from the internal seeds. The analysis of the induced cut boundaries along this ordered region growing is exploited as pruning criterion in [41], with the advantage of being independent of parameters. However, this region growing of the IFT from internal seeds may invade the background (i.e., the leaking problem) before filling the entire object, causing the methods [41] and IFT-CT to fail and requiring more seeds for correction. With regard to this issue, the methods [9, 38] are more robust. In these latter methods, a combinatorial property of the forest is exploited to automatically identify the *leaking pixels* (boundary parts which are crossed by optimum paths) and eliminate their subtrees, such that the remaining forest defines the object [9, 38].

### 3.4.2 IFT with internal and external seeds

The method IFT-SC (*IFT segmentation by Seed Competition*) consists of the computation of an optimum-path forest  $P$  using internal and external seeds,  $\mathcal{S}_1$  and  $\mathcal{S}_0$ . Its segmentation  $\hat{L}$  is defined as follows, where  $\pi_t$  is the optimum path with terminus  $t$  obtained from  $P$ .

$$L(t) = \begin{cases} 1 & \text{if } R(\pi_t) \in \mathcal{S}_1, \\ 0 & \text{otherwise.} \end{cases} \quad (3.3)$$

From an *optimum-path forest* we obtain an image partition, where each seed is root of

an *optimum-path tree* composed by pixels more strongly connected to that seed than to any other, and ties are broken by some tie-breaking policy as discussed next. For a given optimum-path forest, the object is defined as the union of all optimum-path trees rooted at the internal seeds (see Figure 3.3).

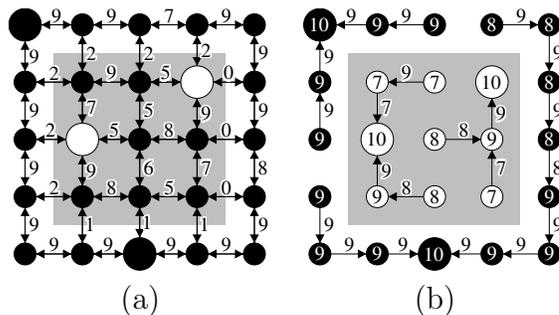


Figure 3.3: (a) A 4-neighborhood graph, where the numbers indicate the arc weights and the object is the *shaded square*. Four seeds are selected, two are inside the object (white dots) and two are in the background (bigger black dots). (b) An optimum-path forest for the path-value function defined by Eq. 3.2. The numbers inside the nodes indicate the values of the optimum paths (Eq. 3.1). The label  $L(s) = 0$  (black), or  $L(s) = 1$  (white) of each seed  $s$  is propagated to all pixels within its respective optimum-path tree.

As observed in reference [43], the optimum-path forest may not be unique. For example, if all paths have the same value, then any spanning forest will be optimum. Since path values are usually discrete, multiple solutions are common when the arc weights are defined in a short range (Figures 3.4a-c). Ties between paths  $\pi_t$  and  $\tau_t$  from seeds  $s_1 = R(\pi_t)$  and  $s_2 = R(\tau_t)$  with the same label ( $\{s_1, s_2\} \subset \mathcal{S}_0$  or  $\{s_1, s_2\} \subset \mathcal{S}_1$ ) are never a problem, since they lead to exactly the same final segmentation result  $\hat{L}$ . Hence, any solution in this case is satisfactory. However, a special care has to be taken in the case of seeds with different labels, which constitute the basis of the real *tie zones* as follows.

**Definition 6 (Tie-zone pixel)** A pixel  $t$  is a tie-zone pixel if there exist two complete optimum paths  $\pi_t$  and  $\tau_t$  such that  $R(\pi_t) \in \mathcal{S}_0$  and  $R(\tau_t) \in \mathcal{S}_1$ .

The *tie zones* are informally defined as maximal connected components of *tie-zone pixels*, while some works just consider the union of all *tie-zone pixels* [5]. The following more precise and formal definition of *tie zone* for path-value function  $f_{\min}$  will be adopted through the rest of this paper.

**Definition 7 (Tie zone)** A tie zone is a maximal set  $\mathcal{T}$  of tie-zone pixels, which forms a subtree in some optimum-path forest.

Figure 3.4d shows an example of *tie zones*, where the *tie zone pixels* are highlighted in gray. Every node  $t$  in an optimum-path forest can be seen as the root of a subtree, which is composed by all nodes that are reached by optimum paths that pass through  $t$  in this optimum-path forest. Consider the upper left node as coordinate (0,0) in Figure 3.4a such that x-coordinates increase from left to right; and y-coordinates increase from top to bottom. In Figure 3.4b, the nodes (2,1) and (3,3) are roots of optimum-path subtrees composed by *tie-zone pixels*. Therefore, we have two *tie zones* in Figure 3.4d. In this work we assume optimum-path forests such that each tie zone, that may be reached from distinct labels (in different optimum-path forests), will receive just one of the labels. This may be accomplished either by using a *LIFO* tie-breaking policy [43], or simply by assigning a fixed label (1 or 0) to all tie zones (Figures 3.4b-c). Thus, when the partition of ambiguous regions is important, this should be treated by means of a better arc-weight estimation.

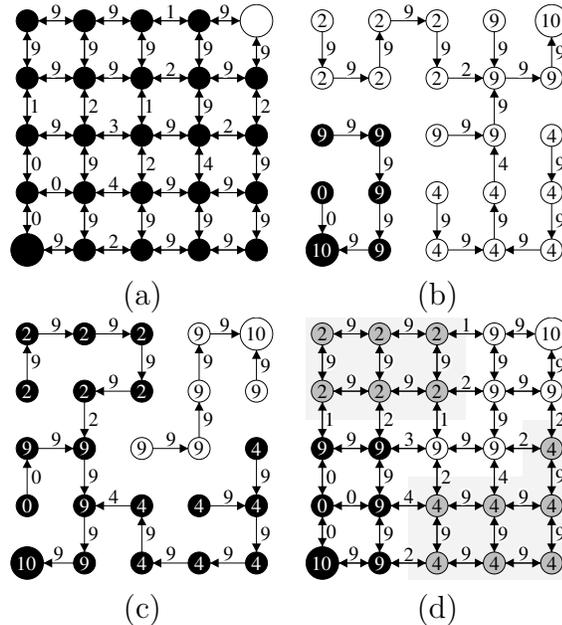


Figure 3.4: (a) A 4-neighborhood graph with one object seed (white dot) and one background seed (bigger black dot). (b-c) Two possible optimum-path forests using Eq. 3.2. The numbers inside the nodes indicate the values of the optimum paths (Eq. 3.1). (d) Two tie zones having values 2 and 4 are shown in gray. The pixel (0, 3) with optimum value 0 is not a tie-zone pixel, since it is unambiguously surrounded by a black influence zone. Note that, any path leading to this node from the white dot also has value 0, but these paths are disregarded since they are not complete optimum.

The IFT-SC formulation [68, 36] captures the essential features of the *watershed*

*transform from markers* (WT) [11], although there is no unique and precise definition for a watershed transform in the literature [96]. Indeed, it was proven that the tie zones of the IFT-SC include all solutions predicted by many discrete definitions of WT [7]. An extensive discussion of the theoretical relations among several different types of watersheds, and also their relation to minimum spanning forests can be found in [29].

The method IFT-SC is also closely related to relative-fuzzy connectedness (RFC) [115, 99]. However, in RFC the tie zones are left unassigned to either background or foreground. Apart from this way to handle ties in RFC, the main difference between these approaches resides in the fact that in RFC the optimum-path values (defined as *strength of connectedness* in [115]) from the internal and external seeds are computed independently for each seed. Hence, it is not possible to guarantee that the computed paths from one seed are complete optimum (Definition 3) in relation to the other seeds (i.e., in RFC an optimum path from a seed  $s_1$  to a target pixel  $t$  may pass through a region having pixels more strongly connected to some other seed  $s_2$ ). This significantly increases the number of ties, causing holes within the objects, and the iterative relative-fuzzy connectedness (IRFC) [24] was proposed to circumvent this problem. It is basically an iterative refinement strategy that imposes additional constraints based on the results from previous iterations. What it essentially does is to penalize paths that are not complete optimum in relation to the other seeds. Thus, disregarding the divergences in the treatment of ties, IFT-SC and IRFC should give similar segmentation results, but being the IFT-SC simpler and faster due to the simultaneous label propagation from all seeds.

In fact, the main motivation for IFT-SC was to eliminate the choice of  $\kappa$  in IFT-CT, favoring the simultaneous segmentation of multiple objects. Actually RFC can be viewed as AFC with automatically calculated thresholds, one for each object [23]. On the other hand, some variants have shown the importance of a hybrid IFT-based method which includes competition among internal and external seeds, and simultaneous automatic computation of multiple  $\kappa$  values per object [80]. The IFT can also extend these methods, by using a more general definition for strength of connectedness based on *smooth* path-value functions [43] (e.g., additive function [8, 94]) or even based on *non-smooth* path-value functions (e.g., non-fixed arc weights [55]). However, in the last case, the IFT results into a spanning forest which may not be optimal [43].

### 3.5 Graph-cut segmentation

Approaches for graph-cut segmentation are based on objective functions that measure some global property of the object's boundary from the arc-weight assignment (Section 3.2). The idea is to assign weights to the arcs such that the minimum of this

objective function (a graph-cut measure) corresponds to the desired segmentation (i.e., a *cut boundary* whose arcs connect the nodes between object and background).

Wu and Leahy [126] were the first to introduce a solution for graph cut using as measure the sum of the arc weights in the cut boundary (Equation 3.4). Their cut measure has the bias toward small boundaries and other objective functions, such as *average cut* [30], *mean cut* [124], *average association* [105], *normalized cut* [106], *ratio cut* [123], and *energy functions* [17, 15, 60] have been proposed to circumvent this problem.

$$E_1(\hat{L}) = \sum_{\forall (s,t) \in \mathcal{A} | L(s)=1, L(t)=0} w(s,t) \quad (3.4)$$

Unfortunately, the problem of segmenting a desired object in a given image cannot be simply reduced to finding a minimum of an objective function in the entire search space, since false-cut boundaries due to similarities between object and background are very common in practice. Indeed it was verified that even in a reduced search space that includes the desired cut from the user's point of view, it does not always correspond to the minimum cut [41].

In view of this, two different strategies have been proposed in the literature. One produces a hierarchical partition tree by recursively applying an unsupervised graph-cut algorithm inside each partition obtained from the previous iteration and then delegating the object recognition for a high-level method that analyzes this tree in a second step [106, 123]. The second strategy incorporates hard constraints (seed pixels) to reduce the search space [17, 15]. Although the second strategy is mostly used in interactive segmentation, where the hard constraints are provided by mouse clicks and drags, the hard constraints may be implemented by probabilistic models (e.g., brain atlases in MRI [64, 52]) in order to achieve automatic segmentation. Hence, the difference between these strategies, when used for single object segmentation, is basically when the recognition is made: *a posteriori* or *a priori*.

One of the main problems in the first strategy is that the minimum cut in a generic graph is NP-hard, when we consider the entire search space [106, 123]. Heuristic solutions still present poor computational performance [49] and their results are sometimes far from the desired one [19]. The second strategy can benefit from the rich set of fast recognition approaches available in the literature [121, 128, 82], which give the object's approximate whereabouts and location in the image, providing the hard constraints, and avoiding this expansive tree computation. But on the other hand, the results may be heavily affected by premature errors during this automatic setting of hard constraints. In fact, Sections 3.6 and 3.7 prove that IFT-SC and IFT-CT (Section 3.4) are indeed graph-cut approaches under the second strategy.

In the second strategy, we have graph-cut methods [17, 15] that extend the work of Wu and Leahy [126] by adding two terminal nodes (*source* and *sink*) to the image graph, which represent object and background, respectively. These nodes are directly connected to all pixels by arcs whose weights reflect penalties for assigning a pixel to object and background based on region properties (*probability maps*). A min-cut/max-flow algorithm from source to sink [48] is then used to compute the minimum-cut boundary according to the following equation:

$$E_2(\hat{L}) = E_1(\hat{L}) + \lambda \left( \sum_{\forall s \in \mathcal{I} | L(s)=1} \overline{\mathcal{P}}_o(s) + \sum_{\forall t \in \mathcal{I} | L(t)=0} \overline{\mathcal{P}}_b(t) \right) \quad (3.5)$$

where  $\overline{\mathcal{P}}_o(s)$  and  $\overline{\mathcal{P}}_b(t)$  are the complement of the probability maps  $\mathcal{P}_o(s)$  and  $\mathcal{P}_b(t)$  that measure how well the intensities of pixels  $s$  and  $t$  fit into a known intensity model (i.e., histogram) of the object and background, respectively.

If the method fails in detecting the desired boundary, the user can impose the arc weights with source and sink by selecting seed pixels inside and outside the object [17]. The running time of these algorithms is still polynomial [15] (i.e., typically  $O(mn^2)$  where  $m$  is the number of arcs and  $n$  is the number of *nodes*). It also has an ad-hoc parameter  $\lambda \geq 0$  which specifies the relative importance of the region properties term versus the boundary properties term. If  $\lambda$  is low their cut becomes the same as in Wu and Leahy [126], which has a bias toward small boundaries, and if  $\lambda$  is high the method becomes very dependent on the probability maps (Figures 3.5b-c). Moreover, the obtained segmentation (Figure 3.5d) is not guaranteed to be connected with the seed sets in the image space, whenever object and background present regions with similar features (Figure 3.5a). Therefore, this approach seems to be more useful in applications where reliable probability maps are available [97].

Other possible solution is to raise the arc weights to the power of  $n$  (Equation 3.6), considering  $\tilde{w}(s, t) = [w(s, t)]^n$  instead of  $w(s, t)$  in Equation 3.4. As we increase the power value  $n$ , larger boundaries are favored (Figures 3.5e-f), avoiding the need to compute the probability maps.

$$\begin{aligned} \tilde{E}_1(\hat{L}) &= \sum_{\forall (s,t) \in \mathcal{A} | L(s)=1, L(t)=0} [w(s, t)]^n \\ &= \sum_{\forall (s,t) \in \mathcal{A} | L(s)=1, L(t)=0} \tilde{w}(s, t) \end{aligned} \quad (3.6)$$

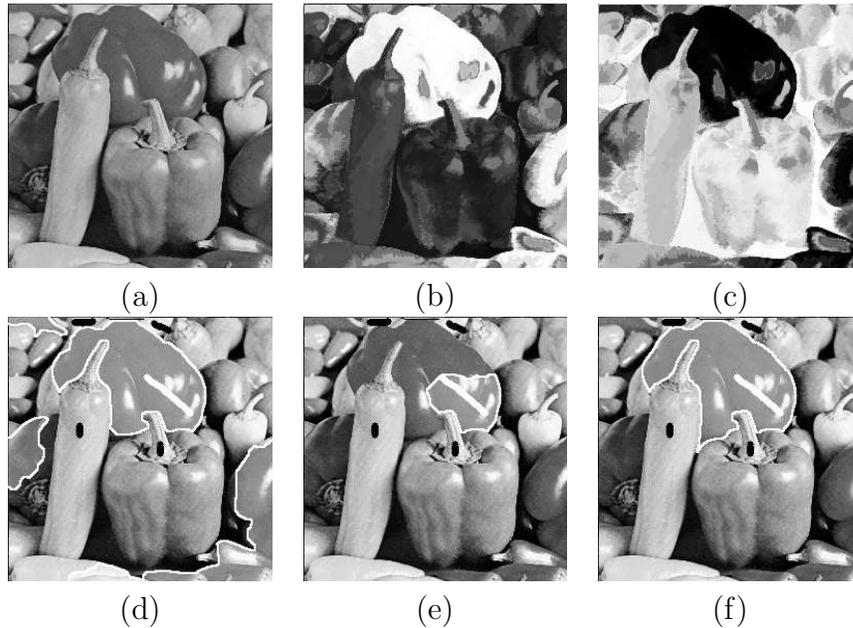


Figure 3.5: (a) An image of peppers. (b-c) The probability maps  $\mathcal{P}_o$  and  $\mathcal{P}_b$  for the bigger pepper at the top. (d) The segmentation by graph-cut using Eq. 3.5 with  $\lambda = 2$ . (e-f) Graph-cut results using Eq. 3.6 with increasing power values, 2 and 5, respectively.

Therefore, from the discussion above, we have two possible solutions to fix the undesirable bias of  $E_1$ . One is to consider  $E_2$  by changing the graph topology at the price of losing the connectivity notion in the original graph  $(\mathcal{I}, \mathcal{A})$ , which makes the method to behave like a threshold depending on  $\lambda$  [14]; and the second is to penalize arcs with high weights  $w(s, t)$  by applying some increasing transformation, as the power of  $n$  in  $\tilde{E}_1$  or as the exponential used in [64], but conserving the topology of the graph  $(\mathcal{I}, \mathcal{A})$ . In fact, it will be proven in Section 3.8 the formal conditions under which optimum-path forest by IFT-SC and graph-cut segmentation by Equation 3.6 produce the same results. Next, Sections 3.6 and 3.7 show that the methods IFT-SC and IFT-CT based on optimum-path forest are indeed graph-cut approaches, each with its own graph-cut measure.

### 3.6 IFT-SC as a graph-cut approach

The theorems assume an undirected graph, with fixed arc weights (Section 3.2) and will be proven for object/background segmentation.

From an optimum-path forest we obtain an image partition in two disjoint sets with distinct labels. Let  $\mathcal{C}_{sc}$  be any cut boundary induced by an IFT-SC segmentation  $\hat{L}$  with

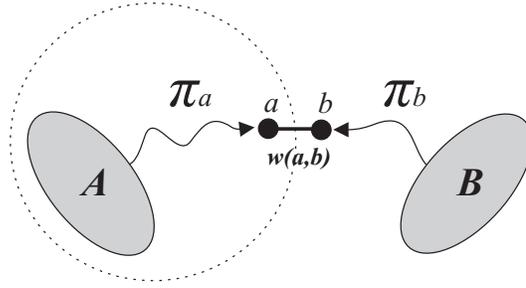


Figure 3.6: The dashed line represents a segmentation by an optimum-path forest with the arc  $(a, b)$  in its cut boundary. The sets of nodes  $A$  and  $B$  represents internal and external seeds respectively.

a single label for each tie zone. For any arc  $(a, b) \in \mathcal{C}_{sc}$ , at least one of the following inequalities is true with the left-hand side being strictly lower than the right-hand side (Figure 3.6).

$$f_{\min}(\pi_a \cdot \langle a, b \rangle) < f_{\min}(\pi_b) \quad (3.7)$$

$$f_{\min}(\pi_b \cdot \langle b, a \rangle) < f_{\min}(\pi_a) \quad (3.8)$$

This is a consequence of path optimality (Equation 3.1) for  $f_{\min}$  (Equation 3.2) when a single label is assigned to each tie zone. If all tie zones are assigned to the object then Equation 3.7 holds. If all tie zones are assigned to the background then Equation 3.8 holds. In the case of *LIFO* tie-breaking policy, at least one of these inequalities will be true for all arcs in the cut.

For example, assume that all tie zones were labeled to the background. If the extension of path  $\pi_b$  by the arc  $(b, a)$  has higher value than path  $\pi_a$ , then this path  $\pi_a$  is not optimum. Otherwise if the extension of  $\pi_b$  has the same value of path  $\pi_a$ , then pixel  $a$  is in a tie zone and should not be part of the object. Therefore, Equation 3.8 is the only valid configuration left.

In fact we may conclude even more and consider the equations below instead of Equations 3.7-3.8:

$$w(a, b) < f_{\min}(\pi_b) \quad \text{and} \quad w(a, b) \leq f_{\min}(\pi_a) \quad (3.9)$$

$$w(a, b) < f_{\min}(\pi_a) \quad \text{and} \quad w(a, b) \leq f_{\min}(\pi_b) \quad (3.10)$$

When all tie zones are labeled to the background, the first inequality in Equation 3.10 can be proved by contradiction as follows. If  $w(a, b) \geq f_{\min}(\pi_a)$  then  $f_{\min}(\pi_a \cdot \langle a, b \rangle) = f_{\min}(\pi_a)$ . There are two possibilities, the extension of path  $\pi_a$  by arc  $(a, b)$  has value  $f_{\min}(\pi_a) > f_{\min}(\pi_b)$ , or it has value  $f_{\min}(\pi_a) \leq f_{\min}(\pi_b)$ . If the first case is true, then  $\pi_b$  is not optimum leading to a contradiction. In the second case, we have that  $f_{\min}(\pi_b \cdot \langle b, a \rangle) = \min\{f_{\min}(\pi_b), w(b, a)\} \geq f_{\min}(\pi_a)$ , since  $w(a, b) \geq f_{\min}(\pi_a)$  and  $f_{\min}(\pi_a) \leq f_{\min}(\pi_b)$ , but this is invalid according to Equation 3.8. Therefore,  $w(a, b) < f_{\min}(\pi_a)$  is the only valid configuration. The second inequality in Equation 3.10 may also be proved by contradiction. If  $w(a, b) > f_{\min}(\pi_b)$  then  $f_{\min}(\pi_b \cdot \langle b, a \rangle) = f_{\min}(\pi_b)$ , and  $a$  should belong to the background, otherwise  $f_{\min}(\pi_a) > f_{\min}(\pi_b)$  and  $b$  could not belong to the background since  $f_{\min}(\pi_a \cdot \langle a, b \rangle) > f_{\min}(\pi_b)$ . But  $a$  in the background leads to a contradiction proving the second inequality given in Equation 3.10. See Figure 3.7 for an example in 1D.

Equation 3.9 can be proven in a similar way when all tie zones are labeled to the object.

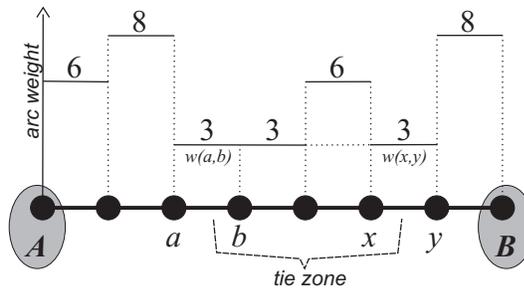


Figure 3.7: A 1D example. All pixels in the tie zone are reached by paths from A and B with the same optimum value 3. If this tie zone is labeled to the background (set B) then  $a$  is reached from A with  $f_{\min}(\pi_a) = 6$  and  $b$  is reached from B with  $f_{\min}(\pi_b) = 3$ . Note that  $w(a, b) = 3$  and the Equation 3.10 is valid ( $3 < 6$  and  $3 \leq 3$ ).

**Theorem 1 (Optimum-path forest cut in IFT-SC)** Any segmentation  $\hat{L}$  defined by an optimum-path forest with path-value function  $f_{\min}$  and with a single label value for each tie zone (Definition 7) minimizes the graph-cut measure  $E_3$  defined by Equation 3.11 among all possible segmentation results (Definition 5).

$$E_3(\hat{L}) = \max_{\forall (s,t) \in \mathcal{A} \mid L(s)=1, L(t)=0} w(s, t) \quad (3.11)$$

**Proof:** We will prove the theorem in the case when all tie zones are labeled to the background, the other case having essentially identical proof. Let  $E_{\min}$  be the minimum value of  $E_3$  defined by Equation 3.11 among all segmentation results satisfying the sets of hard constraints  $\mathcal{S}_1$  and  $\mathcal{S}_0$  (Definition 5). Let  $\mathcal{C}_{\text{sc}}$  be the cut boundary induced (Definition 4) by a segmentation  $\hat{L}$  obtained through IFT-SC with path-value function  $f_{\min}$  and let  $(a, b)$  be the arc with maximum weight  $w(a, b)$  in  $\mathcal{C}_{\text{sc}}$  (that is,  $E_3(\hat{L}) = w(a, b)$ ). For any path  $\pi_t$ , let  $\text{Arc}(\pi_t)$  be the set of all arcs within  $\pi_t$ , or the empty set if  $\pi_t$  is a trivial path.

Any optimum cut  $\mathcal{C}_{\min}$ , which minimizes the graph-cut measure  $E_3$  defined by Equation 3.11 among all possible feasible segmentations, must contain at least one arc from the set  $\text{Arc}(\pi_a) \cup \{(a, b)\} \cup \text{Arc}(\pi_b)$  in order to disconnect object and background seeds (Figure 3.6). There are three possibilities:

1.  $\mathcal{C}_{\min} \cap \text{Arc}(\pi_a)$  is not empty.
2.  $\mathcal{C}_{\min} \cap \text{Arc}(\pi_b)$  is not empty.
3.  $(a, b) \in \mathcal{C}_{\min}$ .

In case 1, the path  $\pi_a$  will have one arc  $(x, y) \in \mathcal{C}_{\min}$ . From Equation 3.11 we may conclude that  $w(x, y) \leq E_{\min}$  and also that  $f_{\min}(\pi_a) \leq E_{\min}$  by the definition of  $f_{\min}$  (Equation 3.2). Putting everything together with Equation 3.10 we have  $w(a, b) < f_{\min}(\pi_a) \leq E_{\min}$ , which implies that  $E_3(\hat{L}) < E_{\min}$ . But this is not possible according to the definition of  $E_{\min}$ .

In case 2, the path  $\pi_b$  will have one arc in  $\mathcal{C}_{\min}$  and therefore we may conclude, in a way analogous to what was done in case 1, that  $f_{\min}(\pi_b) \leq E_{\min}$ .

$$f_{\min}(\pi_b) \leq E_{\min} \tag{3.12}$$

Assuming that Theorem 1 is false we have:

$$w(a, b) > E_{\min} \tag{3.13}$$

By combining the above hypothesis with Equation 3.10, we have  $f_{\min}(\pi_a) > E_{\min}$ . Since  $w(a, b) > E_{\min}$  and  $f_{\min}(\pi_a) > E_{\min}$ , from the definition of  $f_{\min}$  (Equation 3.2), it is also easy to see that:

$$f_{\min}(\pi_a \cdot \langle a, b \rangle) = \min\{f_{\min}(\pi_a), w(a, b)\} > E_{\min} \tag{3.14}$$

From Equation 3.12 we finally obtain:

$$f_{\min}(\pi_a \cdot \langle a, b \rangle) > E_{\min} \geq f_{\min}(\pi_b) \tag{3.15}$$

But from these results, we reach a contradiction because  $\pi_b$  will not be an optimum path if Equation 3.15 holds. Therefore, hypothesis 3.13 with case 2 is false.

In case 3, from  $(a, b) \in \mathcal{C}_{\min}$  we have that  $w(a, b) \leq E_{\min}$  (Equation 3.11). But  $w(a, b) < E_{\min}$  is not possible according to the definition of  $E_{\min}$ . Hence, we have that  $w(a, b) = E_{\min}$  as we wanted to prove (Theorem 1).

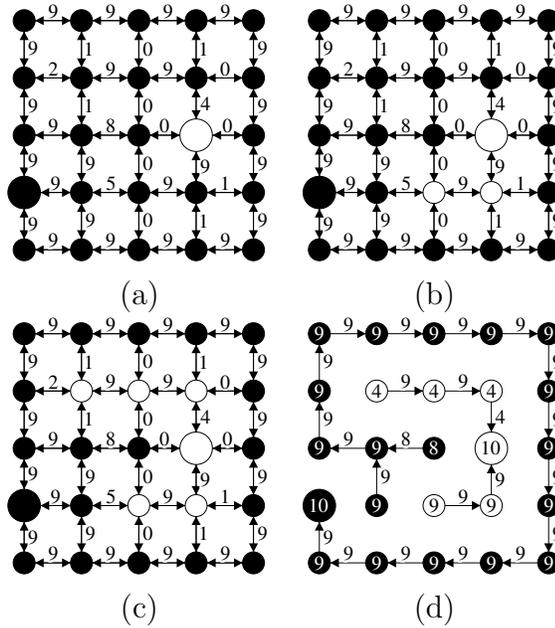


Figure 3.8: (a) A 4-neighborhood graph with one internal seed (white dot) and one external seed (bigger black dot). (b-c) Two optimum segmentation candidates for  $E_3$  (Eq. 3.11) satisfying the hard constraints. (d) The chosen solution by optimum-path forest using Eq. 3.2, where the numbers inside the nodes indicate the values of the optimum paths (Eq. 3.1).

For  $\hat{L}$  to be a segmentation by IFT-SC under the stated conditions, Theorem 1 is necessary but not sufficient. Figures 3.8a-d illustrate this aspect. Figure 3.8a shows an image graph with two seed pixels. Figures 3.8b and 3.8c show two possible segmentation results, satisfying these hard constraints. Both have the same optimum cut  $E_3(\hat{L}) = 5$  (Equation 3.11), but only Figure 3.8c corresponds to a segmentation by IFT-SC (Figure 3.8d). Fortunately, there is another cut property that gives a stronger characterization of an optimum-path forest segmentation by IFT-SC.

**Theorem 2 (Piecewise optimum property in IFT-SC)** *Let  $\mathcal{C}_{sc}$  be any cut boundary induced by a segmentation  $\hat{L}$  defined by an optimum-path forest with path-value function*

$f_{\min}$  and with a single label for each tie zone (Definition 7). Let  $E_3$  be also defined for a set of arcs  $\mathcal{X}$  as  $E_3(\mathcal{X}) = \max_{\forall (s,t) \in \mathcal{X}} \{w(s,t)\}$ . For any subset  $\mathcal{C}_{\text{sub}} \subset \mathcal{C}_{\text{sc}}$ , let  $\overline{\mathcal{C}}_{\text{sub}}$  be defined as  $\mathcal{C}_{\text{sc}} \setminus \mathcal{C}_{\text{sub}}$ . Any non-empty subset  $\mathcal{C}_{\text{sub}} \subset \mathcal{C}_{\text{sc}}$  minimizes  $E_3(\mathcal{X})$  among all possible set of arcs  $\mathcal{X}$  whose union  $\mathcal{X} \cup \overline{\mathcal{C}}_{\text{sub}}$  defines an induced cut boundary of a feasible segmentation (Definition 5).

For example in Figure 3.8, if we consider all arcs in the cut except the arc with weight 5 as being the subset, then the next greatest arc has value 4 in Figure 3.8b and value 2 in Figure 3.8c. A valid IFT-SC segmentation will be one with the lower value (Figure 3.8d).

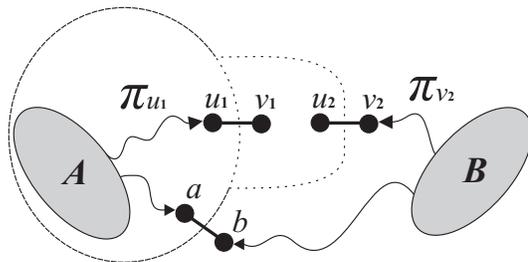


Figure 3.9: Two possible optimum cut boundaries are shown, both with optimum value in Equation 3.11 given by  $w(a,b)$ . The dashed lines represent the boundary pieces at issue, with greatest weights given by  $w(u_1, v_1)$  and  $w(u_2, v_2)$  respectively.

**Proof:** We will prove the theorem in the case when all tie zones are labeled to the background, the other case having essentially identical proof. Let  $\overline{\mathcal{C}}_{\text{sub}}$  be a subset of the arcs of  $\mathcal{C}_{\text{sc}}$ . Let's consider two arbitrary sets of arcs  $\mathcal{X}_1$  and  $\mathcal{X}_2$  such that  $\mathcal{X}_i \cup \overline{\mathcal{C}}_{\text{sub}}$  defines a feasible segmentation (Definition 5), for  $i = 1, 2$ . Let the arcs  $(u_1, v_1) \in \mathcal{X}_1$  and  $(u_2, v_2) \in \mathcal{X}_2$  be arcs with maximum weight within these sets (Figure 3.9). Theorem 2 states that a set  $\mathcal{X}_i$  with lower maximum-weight should be selected. There are two cases,  $w(u_1, v_1) < w(u_2, v_2)$  or  $w(u_1, v_1) > w(u_2, v_2)$ .

In the first case we have that  $w(u_1, v_1) < w(u_2, v_2)$  and according to the theorem  $\mathcal{X}_1$  should be picked. Let's prove by contradiction and assume that  $\mathcal{X}_2$  is the optimum-path forest result by IFT-SC. From Equation 3.10 we know that the optimum path to pixel  $u_2$  from the internal seeds satisfies  $w(u_2, v_2) < f_{\min}(\pi_{u_2})$  and from the hypothesis  $w(u_1, v_1) < w(u_2, v_2)$ , we may conclude that  $w(u_1, v_1) < f_{\min}(\pi_{u_2})$ . But  $\pi_{u_2}$  must pass through  $\mathcal{X}_1$  which implies a contradiction, since  $f_{\min}(\pi_{u_2}) \leq w(u_1, v_1)$  by Equation 3.2.

Similarly, in the second case we have that  $w(u_1, v_1) > w(u_2, v_2)$  and according to the theorem  $\mathcal{X}_2$  should be picked. By assuming its logical negation we get a contradiction like before by similar arguments. Hence, Theorem 2 holds as we wanted to prove.

It is important to state that some graph-cut measures, such as the mean cut [124] (Equation 3.16), are not piecewise optimum (see Figure 3.10). In these measures, changes in one part of the boundary have a global effect, changing the understanding of what is best in other parts. This instability is undesirable in interactive segmentation, because the user loses control over segmentation when local interventions for correction may modify other parts where the user was already satisfied with the segmentation results. Therefore, the piecewise optimum property is indeed very important.

$$E_4(\hat{L}) = \frac{E_1(\hat{L})}{\sum_{\forall(s,t) \in \mathcal{A}} \mathbb{1}_{L(s)=1, L(t)=0}} \quad (3.16)$$

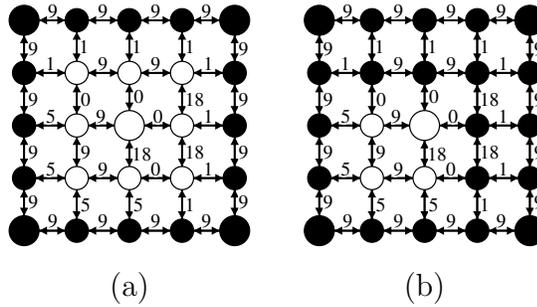


Figure 3.10: Two possible segmentation results on the same 4-neighborhood graph, both satisfying the hard constraints (bigger white and black dots). (a) The optimum solution with minimum mean cut  $\frac{(4 \times 5 + 8 \times 1)}{12} = 2.33$ . (b) The second solution has worse mean cut  $\frac{(4 \times 5 + 4 \times 0)}{8} = 2.50$ . Since  $\frac{4 \times 0}{4} < \frac{8 \times 1}{8}$  we may conclude that this graph-cut measure is not piecewise optimum.

### 3.7 IFT-CT as a graph-cut approach

For a given image graph  $(\mathcal{I}, \mathcal{A})$ , the segmentation  $\hat{L}$  obtained by the IFT-CT with a connectivity threshold  $\kappa$ , is feasible (Definition 5) with respect to the internal seeds in  $\mathcal{S}_1$ , and its induced cut  $\mathcal{C}_{ct}$  has the following property:

$$E_3(\hat{L}) < \kappa \quad (3.17)$$

Equation 3.17 states that all arcs in the cut  $\mathcal{C}_{ct}$  have values lower than  $\kappa$ . From the

definition of IFT-CT, we have that  $f_{\min}(\pi_a) \geq \kappa$  for any interior pixel  $a$  and optimum path  $\pi_a$ . If we assume that the above property (Equation 3.17) is false, then we have at least one arc  $(a, b) \in \mathcal{C}_{\text{ct}}$  such that  $w(a, b) \geq \kappa$ . From the definition of  $f_{\min}$  (Equation 3.2) we have that:

$$f_{\min}(\pi_a \cdot \langle a, b \rangle) = \min\{f_{\min}(\pi_a), w(a, b)\} \quad (3.18)$$

Since  $f_{\min}(\pi_a) \geq \kappa$  and  $w(a, b) \geq \kappa$ , the above equation implies that  $f_{\min}(\pi_a \cdot \langle a, b \rangle) \geq \kappa$ . But from the definition of IFT-CT, this also implies that pixel  $b$  must belong to the object leading to a contradiction. Therefore, the property given in Equation 3.17 is true.

Figure 3.11 shows that there may be more than one feasible segmentation (Definition 5) with induced cut boundary satisfying this property. Both results in Figures 3.11b and 3.11d have  $E_3(\hat{L}) = 3$  which is lower than  $\kappa = 4$ . In fact, the IFT-CT segmentation  $\hat{L}$  is always given by the smallest region possessing the property given in Equation 3.17 (Figure 3.11b). To understand this, note that, the IFT-CT segmentation may be obtained by removing from the original image graph  $(\mathcal{I}, \mathcal{A})$  all arcs whose weights are lower than  $\kappa$ , and by taking then the maximal connected components in the resulting graph, such that they contain at least one seed pixel. This can be accomplished by a simple breadth-first search from the internal seeds in the remaining graph. Since any cut satisfying Property 3.17 will have all of its arcs removed, the breadth-first search will stop when the first cut with this property is found. In fact, the implementation which thresholds the connectivity map  $V(t)$  is more suitable for interactive segmentation, because the user may change the value of  $\kappa$  with real-time response. That is, the map  $V(t)$  encodes all possible IFT-CT results for any value of  $\kappa$  (Figures 3.11b-c).

Therefore, the cut  $\mathcal{C}_{\text{ct}}$  can be finally defined using the characterization from Theorem 3.

**Theorem 3 (Optimum-path forest cut in IFT-CT)** *The segmentation  $\hat{L}$  defined by the IFT-CT in a graph with fixed arc weights and using path-value function  $f_{\min}$  minimizes the graph-cut measure  $E_5$  defined by Equation 3.19 among all segmentation results satisfying the internal hard constraints in  $\mathcal{S}_1$  (Definition 5 with  $\mathcal{S}_0 = \emptyset$ ).*

$$E_5(\hat{L}) = \sum_{\forall s \in \mathcal{I} | L(s)=1} 1 + \mathcal{U}(E_3(\hat{L}) - \kappa) \cdot N \quad (3.19)$$

where  $\mathcal{U}$  is the *unit step function* ( $\mathcal{U}(x) = 1$  if  $x \geq 0$ , and 0 otherwise) and  $N$  is the total number of nodes/pixels. The second term in Equation 3.19 acts as a penalty which makes impracticable cuts that do not satisfy Property 3.17. Among all possible results satisfying Property 3.17, the first term in Equation 3.19 guarantees the selection of the one with smallest area.

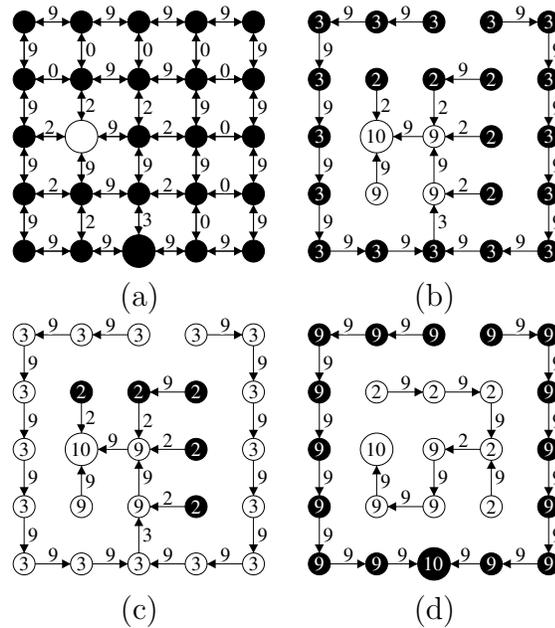


Figure 3.11: (a) A 4-neighborhood graph with one internal seed (white dot) and one external seed (bigger black dot). (b-c) The IFT-CT results from the internal seed with  $\kappa = 4$  and  $\kappa = 3$ , respectively. Note that,  $\kappa = 4$  gives the IFT-CT result with lowest  $E_3$  value satisfying all initial hard constraints ( $E_3(\hat{L}) = 3$ ). (d) The result by IFT-SC also has  $E_3(\hat{L}) = 3$ . However, it is quite different from the IFT-CT due to the piecewise optimum property (Theorem 2).

From Figure 3.11 it is clear that, this cut  $\mathcal{C}_{ct}$  does not have the piecewise optimum property with respect to  $E_3$  (Theorem 2), since it always choose the smallest region. In real applications, if the arc weights are computed based on a thick gradient (Figure 3.12a), this may result in the lost of a narrow band around the object (Figures 3.12b-d). In this sense, it seems that a non-maximal suppression to thin the edges is an adequate preprocessing that should be adopted in IFT-CT. Therefore, the results of IFT-SC are in general different from the ones obtained by IFT-CT for any threshold  $\kappa$  (Figures 3.11b-d and Figures 3.12b-c). Another clear difference is regarding to the presence of holes inside the objects, which may appear in IFT-CT.

### 3.8 The link with the min-cut/max-flow algorithm

A min-cut/max-flow algorithm from source to sink [48], with the internal seeds in  $\mathcal{S}_1$  connected to the source and the external seeds in  $\mathcal{S}_0$  connected to the sink, computes a

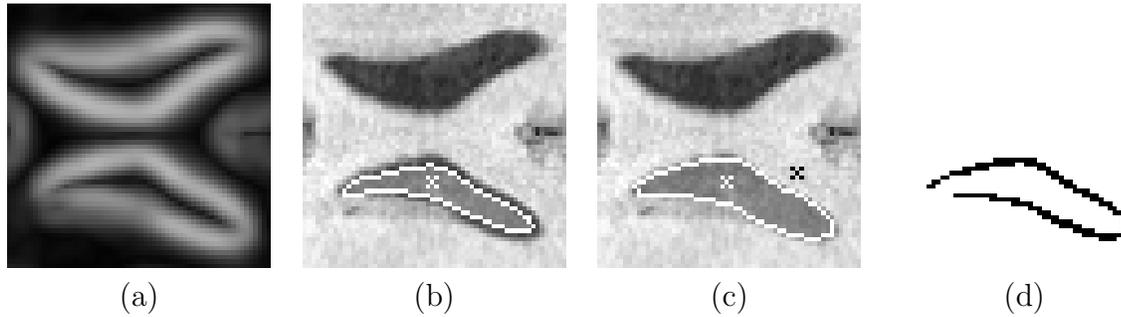


Figure 3.12: (a) A gradient image of the ventricles with thick edges. (b) The result of IFT-CT by the selection of the best  $\kappa$ . (c) An improved result is obtained by IFT-SC. (d) The XOR operation of the labels (in black) indicates a considerable divergence.

cut boundary which minimizes the graph-cut measure  $E_1$  defined by Equation 3.4 among all possible segmentation results (i.e., method [17] with parameter  $\lambda = 0$ ). The same algorithm can be used to minimize  $\tilde{E}_1$  (Equation 3.6).

The graph-cut measures  $E_1$  (Equation 3.4) and  $\tilde{E}_1$  (Equation 3.6) naturally lead to optimum solutions, that are also piecewise optimum. That is, for a given feasible segmentation  $\hat{L}$ , its induced cut boundary  $\mathcal{C}$  is optimum, only if any of its subsets is also optimum. For example, for a cut  $\mathcal{C}$ , let  $\mathcal{Y} \subset \mathcal{C}$  be a fixed subset that is known to be part of the optimum solution. Let  $K$  be the sum of all arc weights within this fixed set  $\mathcal{Y}$ . The cut  $\mathcal{C}$  will be optimum only if  $\mathcal{X} = \mathcal{C} \setminus \mathcal{Y}$  is also optimum:

$$\begin{aligned}
 E_{\min} &= \min_{\forall \mathcal{C}} \left\{ \sum_{\forall (s,t) \in \mathcal{C}} w(s,t) \right\} \\
 &= \min_{\forall \mathcal{X}} \left[ \sum_{\forall (s,t) \in \mathcal{X}} w(s,t) + \sum_{\forall (s,t) \in \mathcal{Y}} w(s,t) \right] \\
 &= \min_{\forall \mathcal{X}} \left\{ \sum_{\forall (s,t) \in \mathcal{X}} w(s,t) \right\} + K
 \end{aligned}$$

Finally, assuming Theorems 1 and 2, the next theorem completes the link between IFT-SC and graph-cut segmentation based on the min-cut/max-flow algorithm, by establishing the necessary conditions under which they produce exactly the same result (i.e., the converse of a theorem presented in [1]). As shown in Section 3.5, the variant  $\tilde{E}_1$  can circumvent the undesirable bias of the graph-cut measure  $E_1$  toward small

boundaries and avoid the need to compute probability maps (Figures 3.5e-f). Indeed, this and other similar variants (increasing transformations) have been used in practice to improve graph-cut segmentation by emphasizing the differences between low and high arc weights [64]. However, the next theorem indicates that such transformations make the graph-cut segmentation to behave like IFT-SC, with more or less intensity, depending on which transformation is used.

**Theorem 4 (Equivalence for increasing transformation)** *Let  $\hat{L}_{sc}$  be a segmentation by IFT-SC with path-value function  $f_{\min}$  and let  $\hat{L}_{flow}$  be a segmentation computed by a min-cut/max-flow algorithm, using the same sets of hard constraints  $\mathcal{S}_1$  and  $\mathcal{S}_0$  in the graph  $(\mathcal{I}, \mathcal{A})$ . In the absence of tie zones (Definition 7), there exists a real finite number  $m$  such that, for any  $n \geq m$ , the segmentation results  $\hat{L}_{sc}$  and  $\hat{L}_{flow}$  are exactly the same when  $\hat{L}_{flow}$  is computed by using  $\tilde{w}(s, t) = [w(s, t)]^n$  as arc weights.*

**Proof:** By raising arc weights  $w(s, t)$  to the power of  $n$  ( $n > 1$ ), we obtain new arc weights  $\tilde{w}(s, t)$ , preserving the order (Equation 3.20) and emphasizing their differences (Equation 3.21).

$$w(s, t) > w(u, v) \Leftrightarrow \tilde{w}(s, t) > \tilde{w}(u, v) \quad (3.20)$$

$$\frac{w(s, t)}{w(u, v)} = K \Leftrightarrow \frac{\tilde{w}(s, t)}{\tilde{w}(u, v)} = K^n \quad (3.21)$$

Let  $\tilde{E}_3$  be the graph-cut measure  $E_3$  (Equation 3.11) obtained from Theorem 1 but with  $\tilde{w}(s, t)$  in the place of  $w(s, t)$ :

$$\begin{aligned} \tilde{E}_3(\hat{L}) &= \max_{\forall (s,t) \in \mathcal{A} | L(s)=1, L(t)=0} [w(s, t)]^n \\ &= \max_{\forall (s,t) \in \mathcal{A} | L(s)=1, L(t)=0} \tilde{w}(s, t) \end{aligned} \quad (3.22)$$

For any two distinct segmentation results  $\hat{L}_1$  and  $\hat{L}_2$ , it is easy to see that:

$$E_3(\hat{L}_1) < E_3(\hat{L}_2) \Leftrightarrow \tilde{E}_3(\hat{L}_1) < \tilde{E}_3(\hat{L}_2) \quad (3.23)$$

This is a direct consequence of the order being preserved (Equation 3.20) and, therefore, the optimum-path forest segmentation results with path-value function  $f_{\min}$  are not affected by this increasing transformation [1].

As we increase the value of  $n$ , the arc-weight ratio changes exponentially (Equation 3.21). Clearly there will be a point ( $n \geq m$ ) where any arc weight  $\tilde{w}(s, t)$  will be greater than the sum of all other arc weights  $\tilde{w}(u, v)$  having lower values (Equations 3.24 and 3.25).

$$\begin{aligned} & \frac{\sum_{\forall(u,v) | \tilde{w}(u,v) < \tilde{w}(s,t)} \tilde{w}(u, v)}{\tilde{w}(s, t)} \\ &= \sum_{\forall(u,v) | \tilde{w}(u,v) < \tilde{w}(s,t)} \left[ \frac{w(u, v)}{w(s, t)} \right]^n \end{aligned} \quad (3.24)$$

Since  $w(u, v)/w(s, t) < 1$  it is easy to see that:

$$\lim_{n \rightarrow \infty} \sum_{\forall(u,v) | \tilde{w}(u,v) < \tilde{w}(s,t)} \left[ \frac{w(u, v)}{w(s, t)} \right]^n = 0 \quad (3.25)$$

Therefore, for any two distinct segmentation results  $\hat{L}_1$  and  $\hat{L}_2$ , we may conclude an important result when  $n \geq m$ :

$$\tilde{E}_3(\hat{L}_1) < \tilde{E}_3(\hat{L}_2) \Rightarrow \tilde{E}_1(\hat{L}_1) < \tilde{E}_1(\hat{L}_2) \quad (3.26)$$

If the left-hand side of Equation 3.26 is true, then there is an arc in the cut boundary of  $\hat{L}_2$  with higher weight than all arcs in the cut of  $\hat{L}_1$ . This arc alone has greater value than the sum given in  $\tilde{E}_1(\hat{L}_1)$  (Equation 3.6). Therefore, disregarding the cases when  $\tilde{E}_3(\hat{L}_1) = \tilde{E}_3(\hat{L}_2)$ , we may conclude that, under the declared conditions, minimizing  $\tilde{E}_3$  or  $\tilde{E}_1$  should lead to the same results.

The case  $\tilde{E}_3(\hat{L}_1) = \tilde{E}_3(\hat{L}_2)$  always happens in the presence of tie zones. In Figure 3.13, suppose that  $\mathbf{C}_1$  has  $\tilde{E}_3 = \tilde{w}(a_1, b_1)$  and  $\mathbf{C}_2$  has  $\tilde{E}_3 = \tilde{w}(a_2, b_2)$ . If both boundaries have optimum  $\tilde{E}_3$  values, then  $\tilde{w}(a_1, b_1) = \tilde{w}(a_2, b_2)$  and the entire region comprised in between these two nested boundaries,  $\mathbf{C}_1$  and  $\mathbf{C}_2$ , could be a tie zone. Although, both boundaries are optimum according to  $\tilde{E}_3$  (Equation 3.22), in general only one is with respect to  $\tilde{E}_1$  (Equation 3.6), depending on the values of the second greatest arcs  $\tilde{w}(u_1, v_1)$  and  $\tilde{w}(u_2, v_2)$ . We cannot guarantee that the best regarding to  $\tilde{E}_1$  (Equation 3.6) will be chosen by the optimum-path forest algorithm. For example, if all tie zones are assigned to the background, then  $\mathbf{C}_1$  will be always chosen. Therefore, the equivalence between these methods cannot be verified in the presence of tie zones.

Even in the absence of tie zones, the case  $\tilde{E}_3(\hat{L}_1) = \tilde{E}_3(\hat{L}_2)$  may still happen as

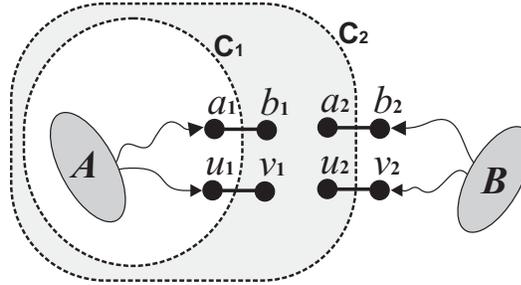


Figure 3.13: Two nested cut boundaries  $C_1$  and  $C_2$ , with the same optimum cut value  $\tilde{E}_3$  (Eq. 3.22) are shown. The weights  $\tilde{w}(a_1, b_1)$  and  $\tilde{w}(a_2, b_2)$  are the maximum within their respective boundaries. The shaded area is a tie zone since  $\tilde{w}(a_1, b_1) = \tilde{w}(a_2, b_2)$ .

occurred in Figure 3.8. However, in this case, the optimum  $\tilde{E}_1$  solution is guaranteed by the piecewise optimum property (Theorem 2). For instance, for  $n = 2$ , Figure 3.8c has  $\tilde{E}_1 = 34$  and Figure 3.8b has  $\tilde{E}_1 = 43$ . In practice, tie zones are usually represented by a few pixels for suitable arc-weight assignment, which makes this result really relevant.

### 3.9 Comparative analysis between the main paradigms

Several comparisons among methods were already made along the previous sections. In this section, we restrict our attention to IFT-SC and the min-cut/max-flow segmentation. From the theoretical point of view, Theorem 4 states that, in the absence of tie zones, the IFT-SC (Equation 3.11) is a particular case of the min-cut by max-flow (Equation 3.6). Therefore, we have here an important intersection between the frameworks of the IFT and the max-flow approach.

As pointed by Allène et al. [1], the value of power  $n$  in Equation 3.6 acts as a smoothing term. However, in practice, if we decrease  $n$ , many important saliences are lost, such as the cow's paws in Figure 3.14. Note also that if smooth boundary is an important issue, it always can be done on a second step by a shape filtering [39]. Therefore, the main advantage of  $\tilde{E}_1$  over  $E_3$  concerns the  $E_3$  worst case, when the desired cut has some arcs with maximum weight (i.e., gradient with perfect gaps (Fig. 3.15a)). In  $\tilde{E}_1$  these arcs are avoided as much as possible by the selection of shortest cuts within the areas of low image contrast (Fig. 3.15b), while  $E_3$  may fail if the seeds are provided closely inside and outside the gaps (Fig. 3.15c).

On the other hand, Theorem 4 indicates that, in the absence of tie zones, the tricks used

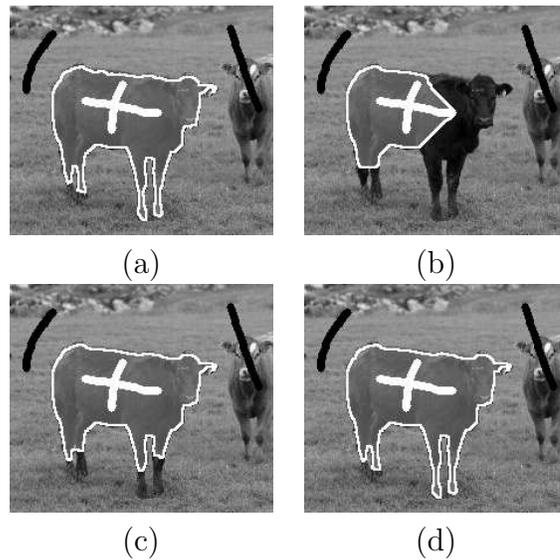


Figure 3.14: (a) The segmentation by IFT-SC using Eq. 3.2. (b-d) Results of the min-cut/max-flow algorithm with increasing power values, 1,3 and 7 respectively.

to improve the results of the min-cut/max-flow algorithm by employing transformations which emphasize the weight differences [64], lead in fact to approximations of the IFT-SC segmentation. The minimum cut by the max-flow algorithm has also its own implementation drawbacks. First, efficient implementations usually consider discrete arc weights and, therefore, the method may suffer from integer overflow as we increase the value of power  $n$ . For example, note that the best result of max-flow given in Figure 3.14d required  $n = 7$ . A second issue is related to its efficiency, while IFT-based methods have linear-time implementations [43], the max-flow algorithm is still polynomial [15] and

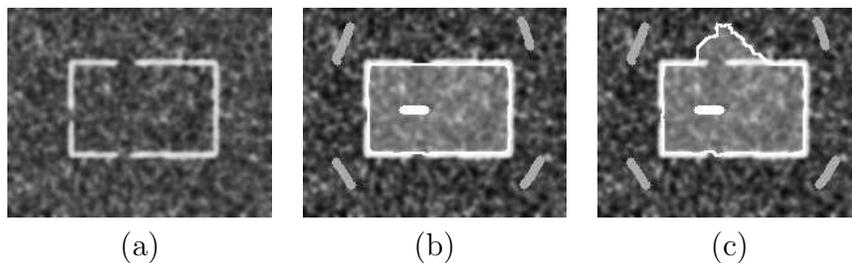


Figure 3.15: (a) A synthetic gradient image with gaps. (b) The result of min-cut/max-flow algorithm. (c) The segmentation by IFT-SC using Eq. 3.2.

is not extensive to simultaneous segmentation of multiple objects [1]. Therefore, from the opposite point of view, the max-flow segmentation under the equivalence conditions (Theorem 4) is, in fact, a limited implementation of IFT-SC, which does not output neither the forest  $P$  nor the map  $V(t)$ ; it only gives the label image  $\hat{L}$ .

Although the absence of tie zones (Definition 7) seems to be a very strong assumption, we have noted for several real images that the tie zones either do not appear or are represented by a few pixels, when we use a suitable arc-weight estimation [83]. In fact, an upper bound for the error of assuming Theorem 4 always true for a sufficiently high value of  $n$  is given by the tie zones, which can be computed as well. Note that a local optimization of  $\tilde{E}_1$  restricted to these tie zones can further solve the problem.

### 3.10 Conclusions

We presented a self-contained paper with theoretical proofs of important existing connections between relevant segmentation methods of the literature. It was proven that some existing IFT-based methods indeed minimize the graph-cut measures  $E_3$  and  $E_5$  (Theorems 1, 2 and 3). It was also shown that the absence of tie zones is the necessary condition to the converse of the theorem stated in [1] (Theorem 4). However, it is important to note that the relations presented here only apply to the path-value function  $f_{\min}$  and that the applications of the IFT go much beyond this particular class of operator [43].

Theorem 1 shows that IFT-SC provides optimum segmentation results from two points of view: as an optimum-path forest and as a minimum cut in the graph according to measure  $E_3$ , which can take into account both image and object properties by using the arc weights as described in [83]. Theorem 2 shows that this optimality is even stronger, given that it is also piecewise optimum. These results give the theoretical foundations to explain the great success and popularity of IFT-SC [68, 36] and its related methods (e.g., WT [11], RFC [115, 99], IRFC [24]). The connectivity between pixels is a key concept, which is naturally exploited by IFT-based methods. More recently, it has also been used to overcome graph-cut shortcomings in a combined approach, called DijkstraGC [117]. Apart from such hybrid approaches, it was shown that, to fix the undesirable bias of the min-cut algorithm toward small boundaries, we must change the graph topology or change the arc weights by employing some increasing transformation that penalizes arcs with high weights. The first solution comes with its own drawbacks, as discussed in Sections 3.5 and 3.9, while the second solves the bias problem nicely and can be used even when it is hard to obtain good probability maps for object and background [64]. On the other hand, Theorem 4 indicates that these transformations are in fact making the min-cut segmentation to behave like IFT-SC, with more or less intensity, depending on

which increasing function is used. This indicates that segmentation methods based on the min-cut algorithm should compare their approaches with IFT-SC in order to justify the use of a more computationally expensive algorithm, especially for 3D applications, or at least mention IFT-SC as a related method. Our paper clarifies many of these aspects, presenting the definitions and theorems in a natural flow of evolution.

As future work, we intend to investigate ways to minimize tie zones by suitable arc-weight assignment and to incorporate shape and appearance model for automatic image segmentation.

## **Acknowledgments**

The authors thank FAPESP (Procs. 05/59808-0, 07/52015-0), CNPq (Procs. 302617/2007-8 and 472402/2007-2), and CAPES for the financial support.

# Capítulo 4

## Object delineation by $\kappa$ -connected components

### 4.1 Introduction

Image segmentation has been a challenge which involves object *recognition* and *delineation*. Recognition is represented by cognitive tasks that determine the approximate location of a desired object in a given image (object detection), verify the correctness of a segmentation result, and identify a desired object among candidate ones (object classification). Delineation is the task that completes segmentation by defining the precise spatial extent of the desired object in the image. Effective recognition requires object properties while accurate delineation usually depends on image properties to distinguish object and background.

In the context of interactive segmentation, a human operator performs the recognition tasks and the computer performs delineation. In order to make these approaches automatic, we must substitute the human operator by a mathematical model. Model-based approaches have used object properties to build numerical, geometrical and statistical models for segmentation [59, 27, 26], and for simple object detection [121]. Since that a mathematical model usually acts worse than a human expert in the recognition task, it is important to develop interactive methods, which minimize the user's time and involvement in the delineation process, such that their automation becomes feasible. For example, we are interested in reducing the user intervention to simple selection of a few pixels in the image.

Delineation methods are usually based on a functional of the arc-weights such as graph-cut approaches [106, 17, 60, 15, 124] or based on a connectivity functional in the form of a path-cost function [114, 99, 120, 11]. This work advances the state of the art of delineation methods based on connectivity functional, being the recognition tasks

performed by human operators.

Fuzzy connectedness/watersheds are image segmentation approaches based on seed pixels, which have been successfully used in many applications [62, 86, 114, 88, 52, 112]. The relation between *relative-fuzzy connectedness* [99, 55, 115] and watershed transform by markers [120, 11] has been pointed out in [43] and formally proved in [6]. They are essentially the same method (one is the dual of the other), where the seeds are specified inside and outside the object, each seed defines an influence zone composed by pixels more strongly connected to that seed than to any other, and the object is defined by the union of the influence zones of its internal seeds. In *absolute-fuzzy connectedness* [116], a seed is specified inside the object and the *strength of connectedness* of each pixel with respect to that seed is computed, such that the object is obtained by thresholding the resulting connectivity image. Clearly, these approaches represent two distinct region-based segmentation paradigms, with and without competition among seeds.

We present extensions to these works, using the framework of the *image foresting transform* (IFT) [43]— a general tool for the design, implementation, and evaluation of image processing operators based on connectivity. In the IFT, the image is interpreted as a graph, whose nodes are the image pixels and whose arcs are defined by an *adjacency relation* between pixels. The cost of a path in this graph is determined by an application-specific *path-cost function*, which usually depends on local image properties along the path— such as color, gradient, and pixel position. For suitable path-cost functions and a set of seed pixels, one can obtain an image partition as an *optimum-path forest* rooted at the seed set. That is, each seed is root of a *minimum-cost path tree* whose pixels are reached from that seed by a path of minimum cost, as compared to the cost of any other path starting in the seed set. The IFT essentially reduces image operators to a simple local processing of attributes of the forest [68, 39, 111, 36, 38].

The *strength of connectedness* of a pixel with respect to a seed is inversely related to the cost of the optimum path connecting the seed to that pixel in the graph. In absolute-fuzzy connectedness, the object can be obtained by selecting pixels reached from an internal seed by an optimum path whose cost is less than or equal to a number  $\kappa$ . In this case, the object is said a single  $\kappa$ -connected component (a minimum-cost path tree). The object can also be defined as the union of all  $\kappa$ -connected components created from each seed separately, which requires one IFT for each seed. In relative-fuzzy connectedness, seeds selected inside and outside the object compete among themselves, partitioning the image into an optimum-path forest, and the object is defined by the union of the optimum-path trees rooted at its internal seeds. The initial appeal for relative-fuzzy connectedness was the possibility to delineate multiple objects simultaneously, without depending on thresholds. However, the use of thresholding together with seed competition provides a

hybrid approach which turns out to be more efficient than the previous ones in many situations. While the previous approaches either assume no competition with a single value of  $\kappa$  for all seeds or eliminate  $\kappa$  for seed competition, we show that seeds with different values of  $\kappa$  can considerably improve segmentation in both paradigms. Of course, this comes with the problem of finding the values of  $\kappa$  for each seed, but we provide automatic and user-friendly interactive ways to determining them.

Section 4.2 describes some definitions related to the IFT, making them more specific for region-based image segmentation. For sake of simplicity, we will describe the methods for gray-scale and two-dimensional images, but they are extensive to multi-parametric and multi-dimensional data sets. The proposed variants and their algorithms are presented in Sections 4.3 and 4.4. Section 4.5 demonstrates the improvements with respect to the previous approaches. Conclusion and future work are presented in Section 4.6.

## 4.2 Background

An *image*  $\hat{I}$  is a pair  $(D_I, I)$  consisting of a finite set  $D_I$  of *pixels* (points in  $\mathbb{Z}^2$ ) and a mapping  $I$  that assigns to each pixel  $p$  in  $D_I$  a *pixel value*  $I(p)$  in some arbitrary value space.

An *adjacency relation*  $A$  is a binary relation between pixels  $p$  and  $q$  of  $D_I$ . We use  $q \in A(p)$  and  $(p, q) \in A$  to indicate that  $q$  is adjacent to  $p$ . Once the adjacency relation  $A$  has been fixed, the image  $\hat{I}$  can be interpreted as a directed graph  $(D_I, A)$  whose nodes are the image pixels in  $D_I$  and whose arcs are the pixel pairs  $(p, q)$  in  $A$ . We are interested in irreflexive, symmetric, and translation-invariant relations. For example, one can take  $A$  to consist of all pairs of pixels  $(p, q)$  in the Cartesian product  $D_I \times D_I$  such that  $d(p, q) \leq \rho$  and  $p \neq q$ , where  $d(p, q)$  denotes the Euclidean distance and  $\rho$  is a specified constant (i.e. 4-adjacency, when  $\rho = 1$ , and 8-adjacency, when  $\rho = \sqrt{2}$ ).

A *path* is a sequence  $\pi = \langle p_1, p_2, \dots, p_n \rangle$  of pixels where  $(p_i, p_{i+1}) \in A$ , for  $1 \leq i \leq n-1$ . The path is *trivial* if  $n = 1$ . Let  $org(\pi) = p_1$  and  $dst(\pi) = p_n$  be the origin and destination of a path  $\pi$ . If  $\pi$  and  $\tau$  are paths such that  $dst(\pi) = org(\tau) = p$ , we denote by  $\pi \cdot \tau$  the concatenation of the two paths, with the two joining instances of  $p$  merged into one. In particular,  $\pi \cdot \langle p, q \rangle$  is a path resulting from the concatenation of its longest prefix  $\pi$  and the last arc  $(p, q) \in A$ .

A *predecessor map* is a function  $P$  that assigns to each pixel  $q \in D_I$  either some other pixel in  $D_I$ , or a distinctive marker *nil* not in  $D_I$  — in which case  $q$  is said to be a *root* of the map. A *spanning forest* is a predecessor map which contains no cycles — in other words, one which takes every pixel to *nil* in a finite number of iterations. For any pixel  $q \in D_I$ , a spanning forest  $P$  defines a path  $P^*(q)$  recursively as  $\langle q \rangle$ , if  $P(q) = nil$ , or  $P^*(p) \cdot \langle p, q \rangle$  if  $P(q) = p \neq nil$  (see Figure 4.1a).

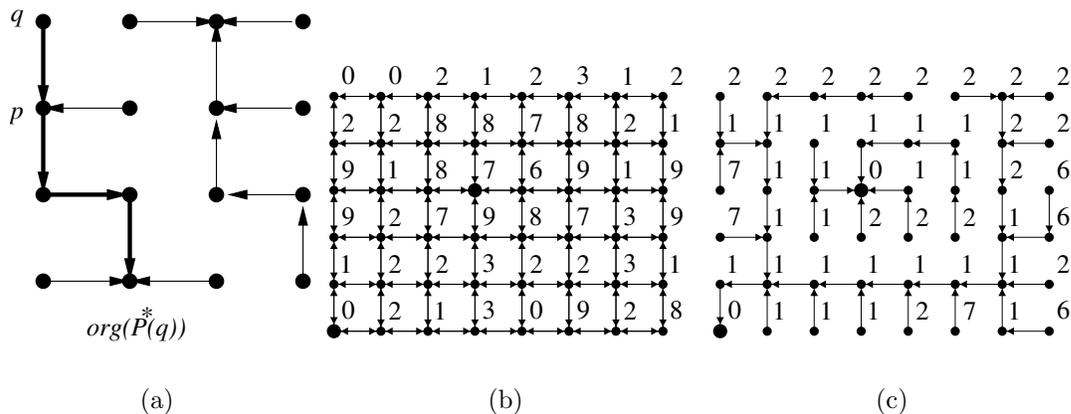


Figure 4.1: (a) The main elements of a spanning forest with two roots, where the thicker path indicates  $P^*(q)$ . (b) An image graph with 4-adjacency, where the integers are the image values  $I(p)$  and the bigger dots indicate two seeds. One is inside the brighter rectangle and one is in the darker background outside it. Note that, the background also contains brighter parts. (c) An optimum-path forest for  $f_{\max}$ , with  $\delta(p, q) = |I(q) - I(p)|$ . The integers are the cost values and the rectangle is obtained as an optimum-path tree rooted at the internal seed.

A pixel  $q$  is *connected* to a pixel  $p$  if there exists a path in the graph from  $p$  to  $q$ . In this sense, every pixel is connected to itself by its trivial path. Since  $A$  is symmetric, we can also say that  $p$  is connected to  $q$ , or simply  $p$  and  $q$  are connected. Therefore, a *connected component* is a subset of  $D_I$  wherein all pairs of pixels are connected.

A *path-cost function*  $f$  assigns to each path  $\pi$  a *path cost*  $f(\pi)$ , in some totally ordered set  $V$  of cost values, whose maximum element is denoted by  $+\infty$ . A path  $\pi$  is *optimum* if  $f(\pi) \leq f(\tau)$  for any other path  $\tau$  with  $dst(\tau) = dst(\pi)$ , irrespective of its starting point. The IFT establishes some conditions applied to optimum paths, which are satisfied by only *smooth* path-cost functions. That is, for any pixel  $q \in D_I$ , there must exist an optimum path  $\pi$  ending at  $q$  which either is trivial, or has the form  $\tau \cdot \langle p, q \rangle$  where

$$(C1) \quad f(\tau) \leq f(\pi),$$

$$(C2) \quad \tau \text{ is optimum,}$$

$$(C3) \quad \text{for any optimum path } \tau' \text{ ending at } p, f(\tau' \cdot \langle p, q \rangle) = f(\pi).$$

The IFT takes an image  $\hat{I}$ , a smooth path-cost function  $f$  and an adjacency relation  $A$ ; and returns an *optimum-path forest*— a spanning forest  $P$  such that  $P^*(q)$  is optimum

for every pixel  $q \in D_I$ . In the forest, there are three important attributes for each pixel: its predecessor in the optimum path, the cost of that path, and the corresponding root (or some label associated with it). The IFT-based image operators result from simple local processing of one or more of these attributes.

For given seed set  $S \subset D_I$ , the concept of *strength of connectedness* [116, 98] of a pixel  $q \in D_I$  with respect to a seed  $s \in S$  can be interpreted as an image property inversely related to the cost of the optimum path from  $s$  to  $q$  according to the *max-arc* path-cost function  $f_{\max}$ :

$$\begin{aligned} f_{\max}(\langle q \rangle) &= \begin{cases} 0, & \text{if } q \in S, \\ +\infty, & \text{otherwise.} \end{cases} \\ f_{\max}(\pi \cdot \langle p, q \rangle) &= \max\{f_{\max}(\pi), \delta(p, q)\}, \end{aligned} \quad (4.1)$$

where  $(p, q) \in A$ ,  $\pi$  is any path ending at  $p$  and starting in  $S$ , and  $\delta(p, q)$  is a non-negative *dissimilarity function* between  $p$  and  $q$  which depends on image properties, such as brightness and gradient (see Figures 4.1b and 4.1c).

One may think of smoothness as a more general definition for strength of connectedness. In this work, we discuss only  $f_{\max}$  because the comparison with previous approaches and our practical experience in region-based segmentation, which shows that  $f_{\max}$  often leads to better results than other commonly known smooth cost functions.

### 4.3 Image segmentation by $\kappa$ -connectivity

We assume given a seed set  $S$  either interactively, by simple mouse clicks, or automatically, based on some *a priori* knowledge about the approximate location of the object. The adjacency relation  $A$  is usually a simple 8-neighborhood, but sometimes it is important to allow farther pixels be adjacent. This may reduce the number of seeds required to label nearby components of a same object, such as letters of a word in the image of a text. Some examples of  $\delta$  functions for  $f_{\max}$  are given below:

$$\delta_1(p, q) = K \left( 1 - \exp \left( -\frac{1}{2\sigma^2} (I(p) - I(q))^2 \right) \right) \quad (4.2)$$

$$\delta_2(p, q) = G(q) \quad (4.3)$$

$$\delta_3(p, q) = K \left( 1 - \exp \left( -\frac{1}{2\sigma^2} \left( \frac{I(p) + I(q)}{2} - I(s) \right)^2 \right) \right) \quad (4.4)$$

$$\delta_4(p, q) = \min_{\forall s \in S} \{ \delta_3(p, q) \} \quad (4.5)$$

$$\delta_5(p, q) = a\delta_1(p, q) + b\delta_3(p, q) \quad (4.6)$$

$$\delta_6(p, q) = \begin{cases} \delta_3(p, q)(1 + \vec{g}(p, q) \cdot \vec{\eta}(p, q)), & \text{if } E_r(p, q) > D_r(p, q), \\ K, & \text{otherwise} \end{cases} \quad (4.7)$$

where  $K$  is a positive integer (e.g. the maximum image intensity),  $\sigma$  is an allowed intensity variation,  $G(q)$  is a gradient magnitude computed at  $q$ , and  $I(s)$  is the intensity of a seed  $s \in S$ , such that  $s = \text{org}(P^*(p))$  in  $\delta_3$  and  $\delta_4$  considers all seeds in  $S$ . The parameters  $a$  and  $b$  are constants such that  $a + b = 1$ , and  $\vec{g}(p, q)$  is a normalized gradient vector computed at arc  $(p, q)$ ,  $\vec{\eta}(p, q)$  is the unit vector of the arc  $(p, q)$ ,  $E_r(p, q)$  and  $D_r(p, q)$  are the pixel intensities at a distance  $r$  to the left and right sides of the arc  $(p, q)$ , respectively.

The dissimilarity functions aim to penalize arcs that cross borders, by assigning higher arc weights to them. We are interested in using the above functions under two possible segmentation paradigms: with and without seed competition. Functions  $\delta_1$  and  $\delta_2$  assume low inhomogeneity within the object. They represent gradient magnitudes with different image resolutions and lead to smooth functions in both paradigms. In fact,  $f_{\max}$  is smooth whenever  $\delta(p, q)$  is fixed for any  $(p, q) \in A$ . Function  $\delta_3$  exploits the dissimilarity between object and pixel intensities, being the object represented by its seed pixels. Although  $f_{\max}$  is smooth for  $\delta_3$  with no seed competition, it may not be smooth in the case of competition among seeds [43] (i.e. the IFT results a spanning forest, but it may be non-optimal). This problem was the main motivation for  $\delta_4$  [99]. However, sometimes  $\delta_3$  with seed competition provides better segmentation results than  $\delta_4$  (see Section 4.5). Function  $\delta_3$  may also limit the influence zones of the seeds, when the intensities inside the object vary linearly toward the background. Function  $\delta_5$  reduces this problem, and in the case of seed competition, one can also replace  $\delta_3$  by  $\delta_4$  in Equation 4.6. The basic idea in function  $\delta_6$  stemmed from [45], where the intensities on the left and right sides of each arc are used to compute its weight, such that longer boundary segments are favored in only one orientation (either clockwise or anti-clockwise). We are extending this idea to provide oriented region growing. Function  $\delta_6$  is suitable to objects, such as the cortex, composed by intermediary intensities with respect to the intensities on both of its sides. For MR-T1 images of the brain, the GM intensities in the cortex are expected to be higher than the

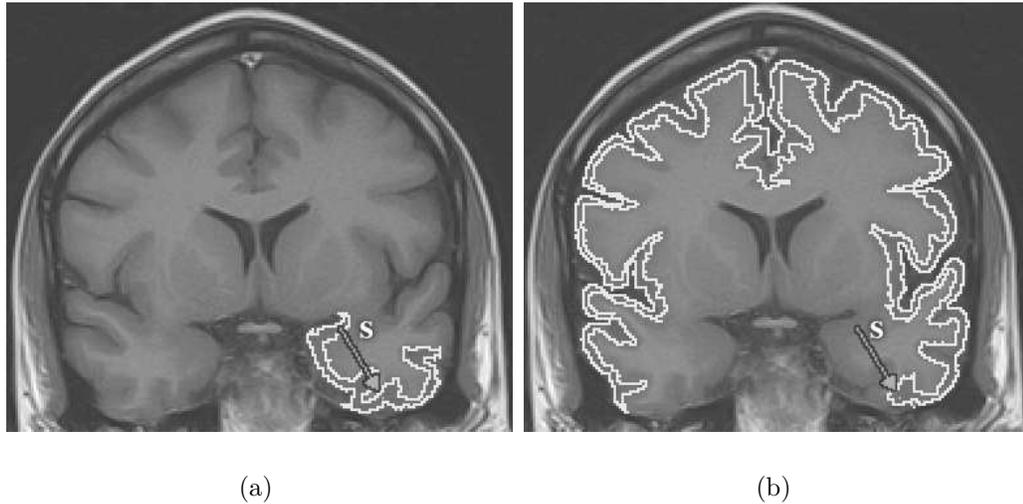


Figure 4.2: An MR-T1 image of the brain with one seed  $s$  inside the cortex. (a-b) The maximum influence zones of  $s$  within the cortex for  $f_{\max}$  with  $\delta_3$  and with  $\delta_6$ , respectively. The asymmetry of  $\delta_6$  favors segmentation in anti-clockwise orientation, increasing the influence zone of  $s$ .

intensities in one side (CSF) and lower than the intensities in the other side (WM). To grow regions in anti-clockwise, we expect that the intensity  $E_r(p, q)$  at a distance  $r$  to the left (WM) of an arc  $(p, q)$  be higher than the intensity  $D_r(p, q)$  at the same distance  $r$  to the right (CSF) of the arc. We favor or penalize the arc dissimilarities based on this rule in  $\delta_6$ . The term  $\vec{g}(p, q) \cdot \vec{\eta}(p, q)$  also penalizes arcs which cross boundaries. The result is that a same seed  $s$  allows to delineate more pixels in the cortex with  $\delta_6$  (Figure 4.2b), following the anti-clockwise orientation, than with  $\delta_3$  (Figure 4.2a). Other interesting ideas of dissimilarity functions for  $f_{\max}$  are presented in [116, 55, 101, 99, 100].

The basic differences between the formulations proposed in [55] and [99, 115] are that (i) the former assumes  $\delta(p, q) = \delta(q, p)$  for all  $(p, q) \in A$ , and requires smooth path-cost functions, and (ii) the later allows asymmetric dissimilarity relations (e.g.  $\delta_2$ ), and non-smooth cost functions (e.g.  $f_{\max}$  with  $\delta_3$  and seed competition). The strength of connectedness between image pixels in (i) is a symmetric relation, while it may be asymmetric in (ii). The main theoretical differences between our formulation and these ones are presented next.

### 4.3.1 Object definition without seed competition

We say that a pixel  $p$  is  $\kappa$ -connected to a seed  $s \in S$ , if there exists an optimum path  $\pi$  from  $s$  to  $p$  such that  $f(\pi) \leq \kappa$ . This  $\kappa$ -connectivity relation will be asymmetric whenever the dissimilarity  $\delta(p, q)$  is asymmetric.

An *object* is a maximal subset of  $D_I$  wherein all pixels  $p$  are at least  $\kappa$ -connected to one pixel  $s \in S$ . Similarly to the method presented in [116], the object is the union of all  $\kappa$ -connected components with respect to each seed  $s \in S$ , which must be computed separately. This makes  $f_{\max}$  smooth for all dissimilarity functions described in Equations 4.2- 4.7.

The algorithm described in [116] assumes that the object can be defined by a single value of  $\kappa$  for all seeds in  $S$ . Figure 4.3a illustrates an example where this assumption works. However, a simple change in the position of a seed can fail segmentation (Figure 4.3b), because the influence zone of each seed inside the object is actually limited by a distinct value of cost  $\kappa$  (Figure 4.3c). Moreover, the choice of seeds with distinct values of  $\kappa$  usually reduces the number of seeds required to complete segmentation. This situation is better understood when we relate the concepts of minimum-spanning tree and minimum-cost path tree for  $f_{\max}$  and symmetric  $\kappa$ -connectivity relations [1].

A *minimum-spanning tree* is a spanning forest  $P$  with a single arbitrary root, where the sum of the arc weights  $\delta(p, q)$  for all pairs  $(p, q) \in A$ , such that  $P(q) = p$ , is minimum, as compared to any other minimum-spanning tree obtained from the original graph  $(D_I, A)$  (Figures 4.4a and 4.4b). If we remove the orientation of the arcs in Figure 4.4b, every pair of pixels in  $P$  is connected by a path which is also optimum according to  $f_{\max}$  (Figure 4.4c). That is, the minimum-spanning tree encodes all possible minimum-cost path trees for  $f_{\max}$ . A  $\kappa$ -connected object with respect to a seed  $s$  can be obtained by taking the component connected to  $s$ , after removing all arcs from  $P$  whose  $\delta(p, q) > \kappa$ . Suppose, for example, that the object is the brighter rectangle in the center of Figure 4.4a. Figure 4.4c shows that only the left side of the rectangle is obtained with  $s_1$  and  $\kappa_1 = 3$ . If  $\kappa_1 = 4$ ,  $s_1$  reaches the right side of the rectangle but invades the background. The rectangle can be obtained with three seeds and  $\kappa = 2$ . However, different values of  $\kappa$  reduce the number of seeds to two,  $s_1$  with  $\kappa_1 = 3$  and  $s_2$  with  $\kappa_2 = 2$  (Figure 4.4d).

There may be many arcs connecting object and background in a minimum-spanning tree. The choice of a single value of  $\kappa$  is equivalent to remove the arcs whose weight  $\delta(p, q)$  is minimum among those connecting object and background. This usually divides the object into several  $\kappa$ -connected components (minimum-spanning trees) and the segmentation will require one seed for each component. When we allow different values of  $\kappa$ , the object components become larger, and consequently, the number of seeds is reduced.

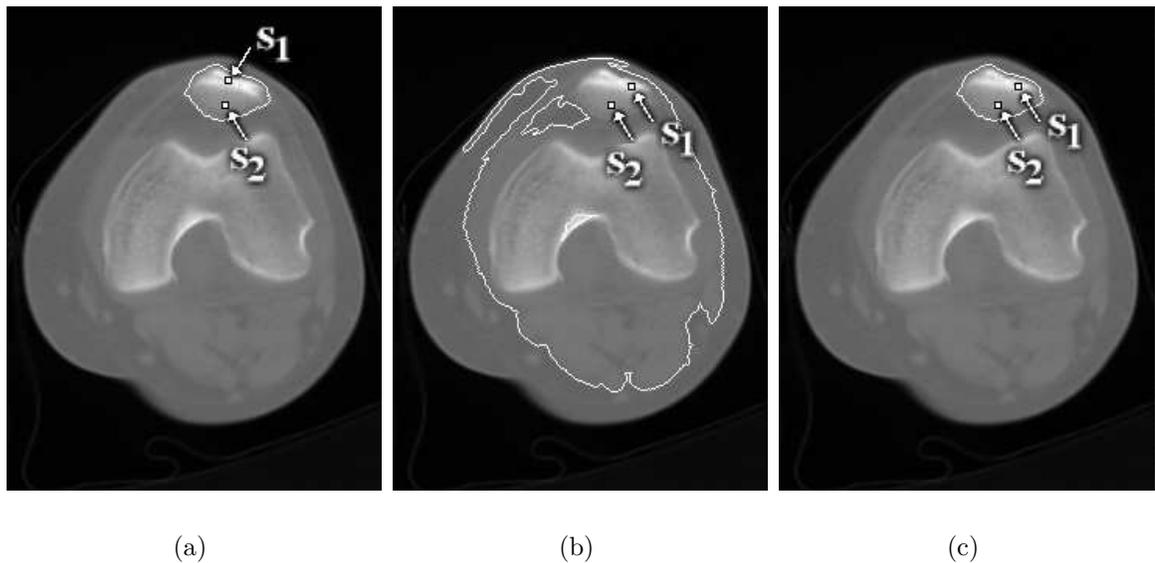


Figure 4.3: A CT image of a knee where the patella can be segmented with two seed pixels,  $s_1$  and  $s_2$ ,  $f_{max}$  with  $\delta_3$ , and without seed competition. (a) The result with a single value of  $\kappa$  for both seeds. (b) The segmentation with a single value of  $\kappa$  fails when we change the position of  $s_1$ , because  $s_1$  requires a higher value of  $\kappa$  to get the brighter part of the bone and  $B$  invades the background at this higher value of  $\kappa$ . (c) The result can be corrected with distinct values of  $\kappa$  for each seed.

### 4.3.2 Object definition with seed competition

In [99, 55], seeds are selected inside and outside the object, and the *object* is defined by the subset of pixels which are more strongly connected to its internal seeds than to any other. This is the same as removing the arcs of maximum weight from the paths that connect object and background in the minimum-spanning tree. For example, the rectangle in Figure 4.4c is obtained by changing the position of  $s_1$  to any pixel in the background and selecting  $s_2$  as shown in Figure 4.4d. The main motivation for this paradigm was to eliminate the choice of  $\kappa$ , favoring the simultaneous segmentation of multiple objects.

We define the *object* as the subset of pixels which are more strongly  $\kappa$ -connected to its internal seeds than to any other. That is, the seeds will compete among themselves for pixels reached from more than one seed by paths whose costs are less than or equal to  $\kappa$ . In which case, the pixel is conquered by the seed whose path cost is minimum. Note that, even the internal seeds compete among themselves, and a distinct value of  $\kappa$  may be required for each seed. When the seed competition fails, these thresholds should limit the influence zones of the seeds avoiding connection between object and background, and the

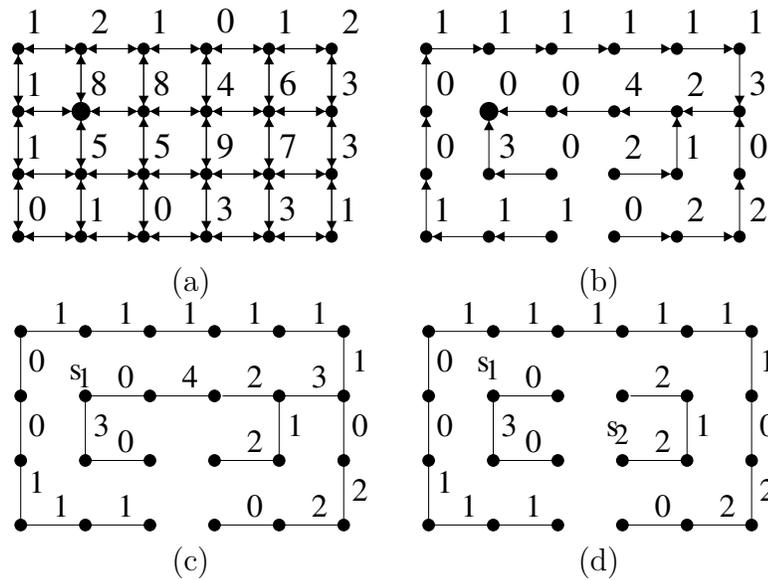


Figure 4.4: (a) An image graph with 4-adjacency, where the integers are the image values  $I(p)$  and the bigger dot is an arbitrary pixel. The object of interest is the brighter rectangle in the center. (b) A minimum-spanning tree computed from the arbitrary pixel, where the integers for each pixel  $q$  are the arc weights  $\delta(p, q) = |I(q) - I(p)|$ , for  $p = P(q)$ . (c) The minimum-spanning tree without arc orientation. A single seed  $s_1$  can not extract the rectangle for any value of  $\kappa$ . (d) The rectangle can be obtained with two seeds and distinct values of  $\kappa$ ,  $s_1$  with  $\kappa_1 = 3$  and  $s_2$  with  $\kappa_2 = 2$ .

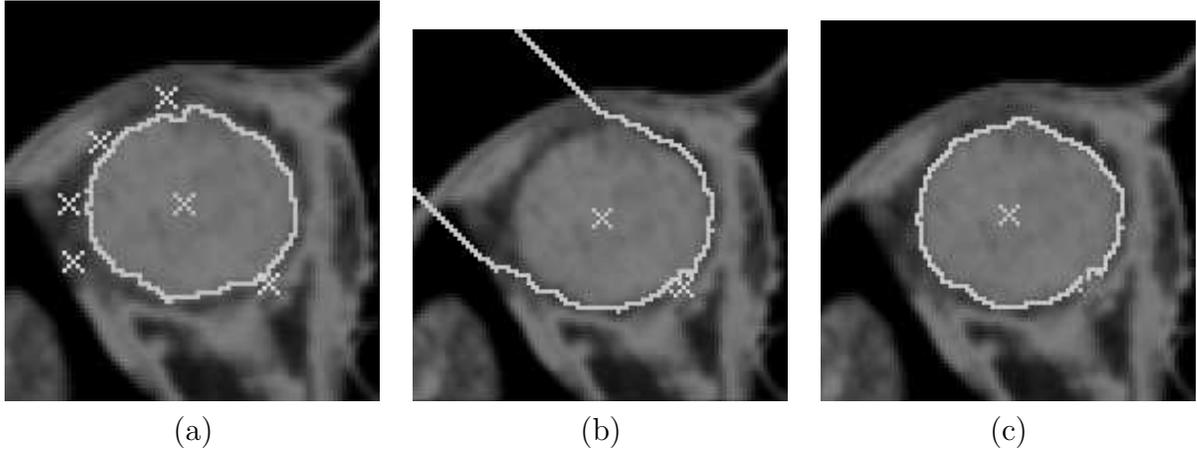


Figure 4.5: A CT image of the orbital region where the eye ball is obtained by seed competition. (a) One internal seed and many external seeds are required for segmentation, using  $f_{\max}$  with  $\delta_4$ . (b) The segmentation fails when some of the external seeds are removed. (c) A value of  $\kappa$  is used to limit the influence zone of the internal seed in parts where the seed competition fails.

pixels do not conquered by any seed should be considered as belonging to the background.

In general, the use of distinct values of  $\kappa$  together with seed competition reduces the number of seeds required to complete segmentation. Figure 4.5a shows an example where many seeds have to be carefully selected in the background to delineate the object. The segmentation fails when some of these seeds are removed (Figure 4.5b), but it works when we limit the extent of the internal seed to some value of  $\kappa$  (Figure 4.5c).

The algorithms and the problem of determining these thresholds for the internal seeds are addressed next.

## 4.4 Algorithms

The IFT uses a variant of Dijkstra’s algorithm [33] to compute three attributes for each pixel  $p \in D_I$  [43]: its predecessor  $P(p)$  in the optimum path, the cost  $C(p)$  of that path, and the corresponding root  $R(p)$ . In the algorithms presented in this section, we do not need to create the predecessor map  $P$  and the root map  $R$  is only used in the case of seed competition.

The IFT with  $f_{\max}$  propagates wavefronts  $W_{cst}$  of same cost  $cst$  around each seed, following the order of the costs  $cst = 0, 1, \dots, K$ . By assigning higher values of  $\delta(p, q)$  to arcs that cross the object’s boundary, the wavefronts fill first the object and, when they

leak to the background, a considerable increase in their areas can be observed (Figures 4.6a and 4.6b). That is, many pixels in the background are reached by optimum paths whose cost is the lowest value  $\delta(p, q)$  among the dissimilarities of the arcs  $(p, q)$  that cross the boundary. This ordered region growing process is exploited to compute the values  $\kappa_s$  of each seed  $s \in S$  automatically and interactively.

#### 4.4.1 Automatic computation of $\kappa_s$

First consider the wavefronts around a seed  $s$  selected inside a given object. All pixels  $p$  in the wavefront  $W_{cst}$  around  $s$  have optimum cost  $C(p) = cst$ ,  $0 \leq cst \leq K$ . If the object is a single  $\kappa$ -connected component with respect to  $s$ , then there exists a threshold  $\kappa_s$ ,  $0 \leq \kappa_s \leq K$ , such that the object can be defined by the union of all wavefronts  $W_{cst}$ , for  $cst = 0, 1, \dots, \kappa_s$ . We can specify a fixed  $\kappa_s$  for this particular application, but this is susceptible to intensity variations. Another alternative is to search for matchings between the shape of the object and the shape of the wavefronts. One drawback is the speed of segmentation, but this may be justified in some applications. A more complex situation occurs when the object definition requires more than one seed pixel. Each seed defines its own maximal extent inside the object and we need to match the shape of the object with the shape of the union of their influence zones.

The approach presented here is much simpler and yet effective. It stems from the previously mentioned observation about the areas of the wavefronts, when they invade the background. The ordered region growing process of a seed  $s$  must stop when the size of its wavefront of cost  $cst$  is greater than an area threshold  $0\% < T < 100\%$ , computed over the image size, and the value of  $\kappa_s$  is determined as  $\max\{cst - 1, 0\}$ . The choice of one value  $\kappa_s$  for each seed  $s \in S$  is then substituted by the choice of  $T$ , which limits the maximum sizes of the wavefronts. This threshold can be verified by selecting internal seeds and setting  $T = 99\%$ . The total area of the wavefronts during propagation can be displayed as a curve. A peak on this curve indicates the maximum possible value for  $T$  at the instant of leaking. Some animations of this ordered region growing process are provided in <http://www.liv.ic.unicamp.br/demo/miranda-kconnected.avi>.

The algorithms are presented for single object delineation without seed competition and multiple object definition with seed competition.

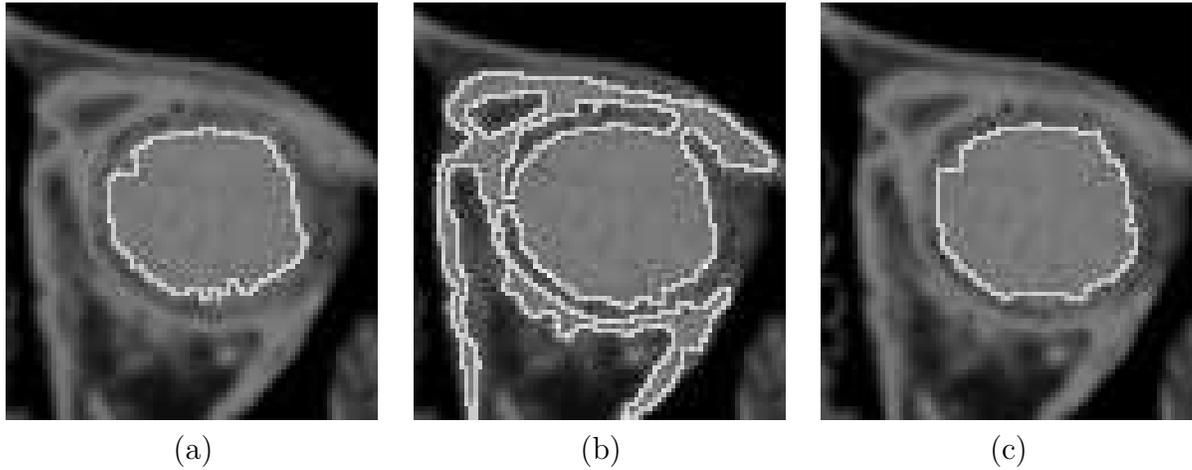


Figure 4.6: A CT image of the orbital region with one seed inside the eye ball. (a) A wavefront of cost  $\kappa$  which represents the maximum extent of this seed inside the eye ball. (b) The wavefront of cost  $\kappa + 1$  shows a considerable augment in size when it invades the background. (c) The pixel propagation order provides more continuous transitions of the wavefronts to select  $\kappa$ , interactively.

**Algorithm 2** – SINGLE OBJECT DEFINITION WITHOUT SEED COMPETITION

INPUT: Image  $\hat{I} = (D_I, I)$ , adjacency  $A$ , internal seeds  $S$ , path-cost function  $f_{\max}$ , and the size threshold  $T$ .

OUTPUT: Binary image  $\hat{L} = (D_I, L)$ , where  $L(p) = 1$ , if  $p$  belongs to the object, and  $L(p) = 0$  otherwise.

AUXILIARY: A priority queue  $Q$ , variables  $tmp$ ,  $\kappa$ ,  $cst$  and  $size$ , and cost map  $C$  defined in  $D_I$ .

1. For every pixel  $p \in D_I$ , set  $L(p) \leftarrow 0$ .
2. While  $S \neq \emptyset$ , do
  3. For every pixel  $p \in D_I$ , set  $C(p) \leftarrow +\infty$ .
  4. Remove a seed  $s$  from  $S$ .
  5. Set  $C(s) \leftarrow 0$ ,  $size \leftarrow 0$ ,  $cst \leftarrow 0$ ,  $\kappa \leftarrow +\infty$ , and insert  $s$  in  $Q$ .
  6. While  $Q \neq \emptyset$  and  $\kappa = +\infty$ , do
    7. Remove a pixel  $p$  from  $Q$  such that  $C(p)$  is minimum.
    8. For every  $q \in A(p)$ , such that  $C(q) > C(p)$ , do
      9. Set  $tmp \leftarrow \max\{C(p), \delta(p, q)\}$ .
      10. If  $tmp < C(q)$ , then
        11. If  $C(q) \neq +\infty$ , then remove  $q$  from  $Q$ .
        12. Set  $C(q) \leftarrow tmp$  and insert  $q$  in  $Q$ .



case of seed competition, the root map is used to find in constant time the root of each pixel in  $S$ . The influence zone of a seed  $s \in S$  is limited either when it meets the influence zone of other seed at the same minimum cost or when the value  $\kappa_s$  of  $s$  is found.

One advantage of the presented algorithms as compared to classical segmentation methods based on seed competition occurs when the object contains several background parts (holes) inside it. In this case, the use of  $\kappa_s$  usually eliminates the need for at least one background seed at each hole. On the other hand, some small noisy parts of the object may not be conquered by the internal seeds due to the use of  $\kappa_s$ . The labeled image can be post-processed, such that holes with area below a threshold are closed [75, 118]. The area closing operator has shown to be a very effective complement for the presented algorithms. In many situations, the objects do not have holes and high area thresholds can be used to reduce the number of internal seeds.

The animations in <http://www.liv.ic.unicamp.br/demo/miranda-kconnected.avi> were created by using Algorithm 3. It is usually preferable with respect to Algorithm 2, because it allows faster multiple object segmentation. Note that, a wavefront of one seed can leak to the background before the object be fully filled by the wavefronts of other seeds. Figure 4.7a illustrates an example where the leaking occurs for seed  $A$  before the object be filled. The moment when  $\kappa_A = 324$  is detected is shown in Figure 4.7b and Figure 4.7c shows the instant when  $\kappa_B = 770$  is detected. The figures show only a region of interest of the original image, where the segmentation was done with  $T = 1\%$ . The final segmentation is shown in Figure 4.7d. Even when the dissimilarities are not higher for arcs that cross the object's boundary, Algorithm 3 can work either due to the seed competition among internal seeds (parts of the object can be filled without leaking) or due to the automatic  $\kappa_s$  computation, as shown in the example of Figure 4.7.

#### 4.4.2 Interactive computation of $\kappa_s$

A first approach is to compute the IFT for every pixel  $p \in D_I$ , such that the cost  $C(p)$  of the optimum path that reaches  $p$  from  $S$  is found. In the case of seed competition, the corresponding root  $R(p) \in S$  is also propagated to each pixel  $p \in D_I$ . Then, the user moves the cursor of the mouse over the image, and for each position  $q$  of the cursor, the program displays the influence zone of the corresponding root  $s = R(q) \in S$  defined by pixels  $p \in D_I$ , such that  $C(p) \leq C(q)$  and  $R(p) = R(q)$ . This interactive process can be repeated until the user selects a pixel  $q$  to confirm the influence zone of  $s$  (i.e.  $\kappa_s = C(q)$ ). The user can repeat this interactive process for each seed  $s \in S$ , in both paradigms.

One drawback of the method above is the abrupt size variations of the wavefronts (Figures 4.6a and 4.6b) which makes the selection of pixel  $q$  sometimes difficult. We circumvent the problem by exploiting the *propagation order*  $O(p)$  (a number from 1 to

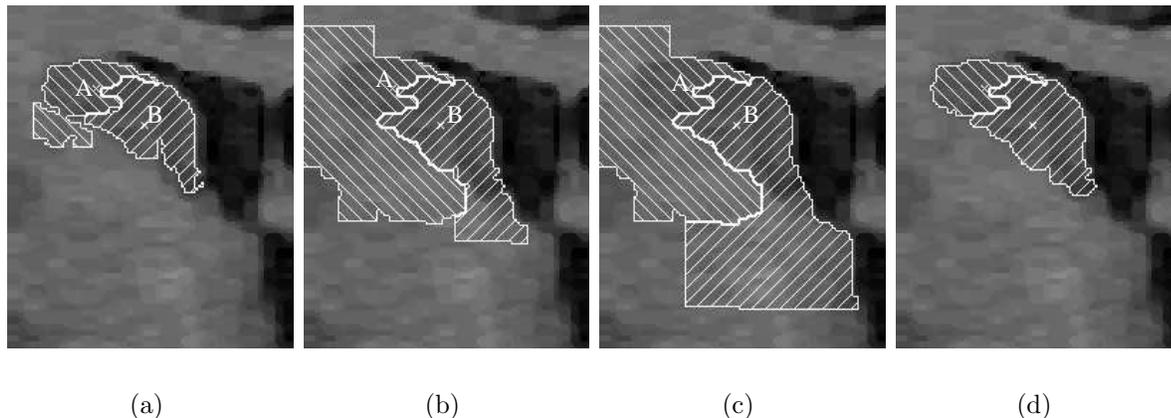


Figure 4.7: Segmentation of a caudate nucleus with two internal seeds,  $A$  and  $B$ . (a) The leaking occurs before the object be filled. (b) The moment when  $\kappa_A = 324$  is detected. (c) The instant when  $\kappa_B$  is detected. (d) Final segmentation.

$|D_I|$ ) of each pixel  $p$  removed from  $Q$  during execution of the IFT. Note that, a pixel  $p$  propagates before a pixel  $q$  (i.e.  $O(p) < O(q)$ ) when it is reached by an optimum path from  $S$ , whose cost  $C(p)$  is less than the cost  $C(q)$  of the optimum path that reaches  $q$ . When  $C(p) = C(q)$ , we assume a *first-in-first-out* (FIFO) tie-breaking policy for  $Q$ . That is, among all pixels with the same minimum cost in  $Q$ , the one first reached by an optimum path from  $S$  is removed for propagation. Therefore, we also compute the propagation order  $O(p)$  of each pixel  $p \in D_I$ . When the user moves the cursor to a position  $q$ , the program displays the influence zone of the corresponding root  $s = R(q) \in S$  defined by pixels  $p \in D_I$ , such that  $O(p) \leq O(q)$  and  $R(p) = R(q)$ . The rest of the process is the same. Note that, although  $\kappa_s = C(q)$ , only the pixels  $p$  in the wavefront  $W_{C(q)}$  which have  $O(p) \leq O(q)$  are selected as belonging to the influence zone of  $s$ . This provides smoother transitions between consecutive wavefronts (Figure 4.6c) as compared to the first idea. The algorithms are presented below.

#### Algorithm 4 – SINGLE OBJECT DEFINITION WITHOUT SEED COMPETITION

INPUT: Image  $\hat{I} = (D_I, I)$ , adjacency  $A$ , internal seeds  $S$ , and path-cost function  $f_{\max}$ .  
 OUTPUT: Binary image  $\hat{L} = (D_I, L)$ , where  $L(p) = 1$ , if  $p$  belongs to the object, and  $L(p) = 0$  otherwise.  
 AUXILIARY: Priority queue  $Q$ , variables  $tmp$ ,  $ord$ , and cost map  $C$  and propagation order map  $O$  defined in  $D_I$ .

1. For every pixel  $p \in D_I$ , set  $L(p) \leftarrow 0$ .

2. While  $S \neq \emptyset$ , do
3.     For every pixel  $p \in D_I$ , set  $C(p) \leftarrow +\infty$ .
4.     Remove a seed  $s$  from  $S$ .
5.     Set  $C(s) \leftarrow 0$ ,  $ord \leftarrow 0$ , and insert  $s$  in  $Q$ .
6.     While  $Q \neq \emptyset$ , do
7.         Remove a pixel  $p$  from  $Q$  such that  $C(p)$  is minimum.
8.         Set  $O(p) \leftarrow ord + 1$  and  $ord \leftarrow ord + 1$ .
9.         For every  $q \in A(p)$ , such that  $C(q) > C(p)$ , do
10.             Set  $tmp \leftarrow \max\{C(p), \delta(p, q)\}$ .
11.             If  $tmp < C(q)$ , then
12.                 If  $C(q) \neq +\infty$ , then remove  $q$  from  $Q$ .
13.                 Set  $C(q) \leftarrow tmp$  and insert  $q$  in  $Q$ .
14. The user selects a pixel  $q$  on the image.
15. For every pixel  $p \in D_I$ , do
16.     If  $O(p) \leq O(q)$ , then set  $L(p) \leftarrow 1$ .

**Algorithm 5** – MULTIPLE OBJECT DEFINITION WITH SEED COMPETITION

- INPUT: Image  $\hat{I} = (D_I, I)$ , adjacency  $A$ , path-cost function  $f_{\max}$ , and a labeled image  $\hat{L} = (D_I, L)$ , where  $L(p) = i$ ,  $0 \leq i \leq k$ , if  $p$  is a seed pixel selected inside object  $i > 0$  among  $k$  objects, being  $i = 0$  reserved for seeds in the background, and  $L(p) = -1$  otherwise.
- OUTPUT: A labeled image  $\hat{L} = (D_I, L)$ , where  $L(p) = i$ ,  $0 \leq i \leq k$ .
- AUXILIARY: Priority queue  $Q$ , variables  $tmp$  and  $ord$ , and  $C$ ,  $R$ ,  $O$  are maps defined in  $D_I$  to store cost, root and propagation order of each pixel, respectively.

1. Set  $ord \leftarrow 0$ .
2. For every pixel  $p \in D_I$ , do
3.     Set  $R(p) \leftarrow p$ .
4.     If  $L(p) = -1$ , then set  $C(p) \leftarrow +\infty$  and  $L(p) \leftarrow 0$ .
5.     Else, set  $C(p) \leftarrow 0$  and insert  $p$  in  $Q$ .
6. While  $Q \neq \emptyset$ , do
7.     Remove a pixel  $p$  from  $Q$  such that  $C(p)$  is minimum.
8.     Set  $O(p) \leftarrow ord + 1$  and  $ord \leftarrow ord + 1$ .
9.     For every  $q \in A(p)$ , such that  $C(q) > C(p)$ , do
10.         Set  $tmp \leftarrow \max\{C(p), \delta(p, q)\}$ .
11.         If  $tmp < C(q)$ , then
12.             If  $C(q) \neq +\infty$ , then remove  $q$  from  $Q$ .
13.             Set  $C(q) \leftarrow tmp$ ,  $R(q) \leftarrow R(p)$ , and insert  $q$  in  $Q$ .
14. While the user is not satisfied.
15.     The user can select a pixel  $q$  on the image.

16.  $\left[ \begin{array}{l} \text{For every pixel } p \in D_I, \text{ do} \\ 17. \quad \left[ \begin{array}{l} \text{If } O(p) \leq O(q) \text{ and } R(p) = R(q), \text{ then set } L(p) \leftarrow L(R(p)). \end{array} \right. \end{array} \right.$

## 4.5 Evaluation

We have selected 100 images from Magnetic Resonance (MR) and Computerized Tomography (CT) data sets of 7 objects for evaluation (see Table 4.1 and Figure 4.8). Each object consists of some slices that represent different degrees of challenge for segmentation. The original images have been pre-processed to increase the similarities between pixels inside the objects and the contrast between object and background. Each of four users have performed segmentation over the 100 images using each of three methods, M1, M2, and M3, with interactive seed selection (mouse clicks).

**M1** Object delineation without seed competition and automatic/interactive computation of  $\kappa_s$ . This method uses Algorithms 2 and 4. When M1 requires a single  $\kappa_s$  for all seeds, it indicates that absolute-fuzzy connectedness (AFC) would work.

**M2** Object delineation with seed competition and automatic  $\kappa_s$  computation. This method uses only Algorithm 3. We did not evaluate Algorithm 5, because preliminary tests indicated that user intervention to add external seeds in Algorithm 3 is simpler and more effective than to indicate  $\kappa_s$  in Algorithm 5.

**M3** Object delineation with seed competition and without  $\kappa_s$  computation. As mentioned in Section 4.1, relative-fuzzy connectedness (RFC) and watershed transform by markers (WT) are the same method [6] (one is the dual of the other), represented here by M3.

Therefore, the user can correct segmentation by adding/removing seeds in M1, M2 and M3, and in the case of M1, by pointing the mouse to the pixel, whose propagation order indicates the correct  $\kappa_s$  implicitly (Section 4.4.2).

M1 aims to show two aspects about AFC: (i) a single  $\kappa$  for all seeds is not sufficient in most cases and (ii) the problem of computing multiple  $\kappa_s$  thresholds can be easily solved by a wavefront area threshold  $0\% < T < 100\%$ , computed over the image size. Note that, being M1 an extension of AFC, there is no situation where AFC works and M1 would fail. M2 aims to reduce the number of required seeds with respect to M3 by automatic  $\kappa_s$  computation. When this automatic procedure fails, M2 becomes M3. Therefore, in the worst case, the efficiency of M2 should be the same of M3.

Object	Description	Imaging Modality	Number of Slices
O1	left eye ball	CT-orbit	15
O2	left caudate nucleus	MR-brain	15
O3	lateral ventricles	MR-brain	15
O4	corpus callosum	MR-brain	10
O5	patella	CT-knee	15
O6	femur	CT-knee	15
O7	white matter	MR-brain	15

Table 4.1: Description, imaging modality and number of slices for each object used in the experiments.

Given that M1 and M2 are extensions of AFC and RFC/WT, we expect that they do not affect the accuracy of the original approaches, which is assumed to be good from the results of several other works [62, 86, 114, 88, 52, 112]. The experiments then aimed to show that M1 works in situations where AFC would fail, M1 and M2 require less user interaction than M3, and the methods produce similar results.

The choice of parameters took a couple of minutes per object, by trying the methods in a first slice. Then, the parameters were fixed to the rest of the slices. Note that, this can be done only once for any given application (object of interest and imaging protocol). We have chosen the best dissimilarity function for each object and method (Table 4.2). We used the 8-neighborhood as adjacency relation  $A$  and set the wavefront area threshold  $T$  to 1% of the image size (except for O2 where  $T = 0.5\%$  in M1 and  $T = 0.2\%$  in M2). Since objects from O1 to O6 do not have holes, we set the area closing threshold to some arbitrary high value (e.g. 500 pixels). The only exception was O7, whose the area closing threshold could not be higher than 3 pixels due to its holes. In function  $\delta_2$ , we used the magnitude of the Sobel's gradient. The value of  $\sigma$  was 20 for all cases involving  $\delta_3$  and  $\delta_4$ . Note also that  $\delta_3$  has been chosen in some situations involving seed competition, despite  $f_{\max}$  is not smooth.

Each object was represented by a set of  $l$  binary slices  $\hat{L}_i = (D_I, L_i)$ ,  $i = 1, 2, \dots, l$ , where  $L_i(p) = 1$  for object pixels and 0 otherwise. Let  $\hat{L}_i$  and  $\hat{L}'_i$  be the binary images resulting from the segmentation of a same object slice using different methods. The similarity between these results was measured by:

$$1.0 - \frac{\sum_{i=1}^{i=l} \sum_{\forall p \in D_I} L_i(p) \oplus L'_i(p)}{\sum_{i=1}^{i=l} \sum_{\forall p \in D_I} L_i(p) + \sum_{i=1}^{i=l} \sum_{\forall p \in D_I} L'_i(p)}, \quad (4.8)$$

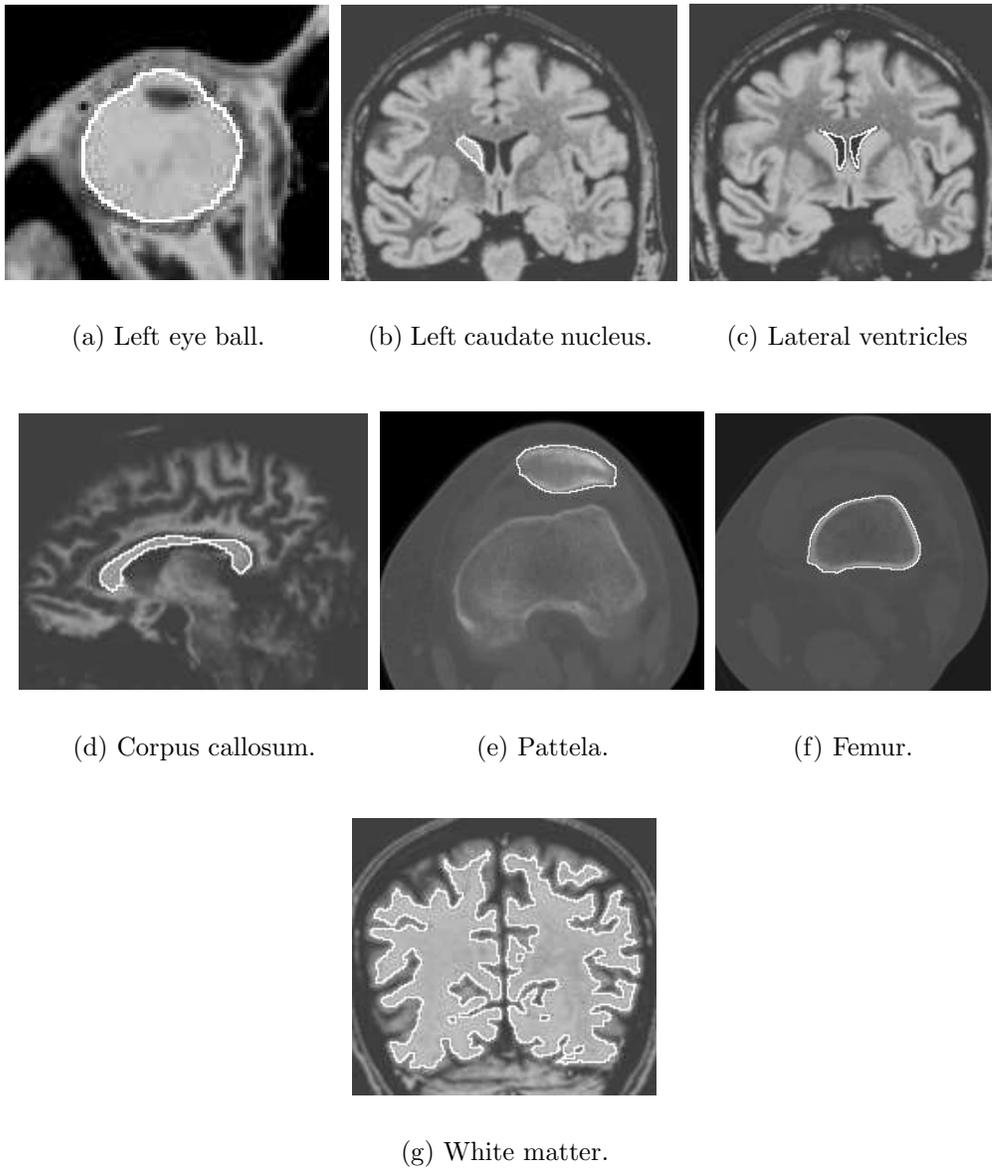


Figure 4.8: (a)-(g) Results of slice segmentation of the objects from 1 to 7, respectively, overlaid with the pre-processed images.

Object	M1	M2	M3
O1	$\delta_3$	$\delta_2$	$\delta_2$
O2	$\delta_3$	$\delta_4$	$\delta_2$
O3	$\delta_3$	$\delta_3$	$\delta_3$
O4	$\delta_3$	$\delta_4$	$\delta_2$
O5	$\delta_3$	$\delta_4$	$\delta_4$
O6	$\delta_3$	$\delta_3$	$\delta_2$
O7	$\delta_3$	$\delta_3$	$\delta_3$

Table 4.2: The dissimilarity functions used for each combination of object and method.

where  $\oplus$  is the “exclusive or” operation (i.e.  $L_i(p) \oplus L'_i(p) = 1$ , if  $L_i(p) \neq L'_i(p)$ , and 0 otherwise). Given that we have four different users using three distinct methods, we may assume that similarity values around 0.90 represent good agreement in delineation (Figure 4.9).

The number of user interactions in M3 is the total number of seeds selected inside (NIS - *Number of Internal Seeds*) and outside (NES - *Number of External Seeds*) the object. In M1, the amount of user interaction is represented by the total number of *interactive  $\kappa_s$  detections* (IKD) and NIS. The *automatic  $\kappa_s$  detections* (AKD) are chosen as much as possible in order to reduce the number of user interactions. In M2, the number of user interactions is computed as in M3, but the number of seeds is expected to be much less due to automatic  $\kappa_s$  detection.

Instead of quantifying the number of user interactions for a fixed value of  $\kappa$ , we decided to quantify the number of different  $\kappa_s$  values found in M1 for cases of multiple seeds. The *percentages of different  $\kappa_s$  values* (PDK) are presented in Table 4.3, together with the average number of interactions and similarity values among all users. Note that, O3 was detected with a same value of  $\kappa$ , but the other objects required from 6.8% to 92.4% of different  $\kappa_s$  values. O5 did not count because it was segmented with only one seed per slice. Therefore, AFC would work only for O3 and O5. On average, M3 and M1 required 2.8 and 1.1 more user interactions than M2, respectively. The advantages of M1 and M2 over M3 increase in more complex situations, such as in the delineation of O7, where M3 required 6 times more user interactions than M1 and M2. Although the performances of M1 and M2 have been equivalent, M2 is preferable because its extension to 3D does not suffer from interactive  $\kappa_s$  indication and it can provide simultaneous segmentation of multiple objects.

Table 4.4 shows in detail the average values of NIS, NES, IKD, and AKD for each object and method. Note that, AKD varied from 59% to 100% of NIS, being on average 90% of NIS in M1 and 88% of NIS in M2. This demonstrates the effectiveness of

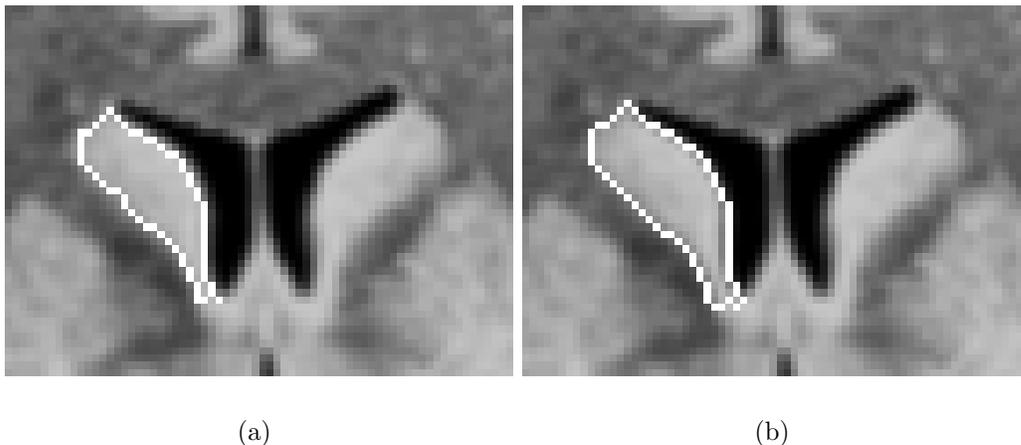


Figure 4.9: (a) A segmentation of the left caudate nucleus. (b) The result of dilating the binary image with a circular structuring element of radius 1. The similarity value between these two masks is 0.87. The differences in the experiments using distinct methods on this object (O2) are less significant than this small dilation.

the proposed approach for automatic  $\kappa_s$  detection and explains the reduction of user interactions in M1 and M2 with respect to M3. Note also that, the number of external seeds was considerably reduced in comparison to M3. This is an important result for future automation, because seed competition is sensitive to the location of the external seeds due to the heterogeneity of the background.

## 4.6 Conclusions

We have presented four IFT-based algorithms for object delineation based on  $\kappa$ -connected components with and without seed competition. They differ from the previous approaches in the following aspects: computation of different values of  $\kappa$  for each seed, effective automatic  $\kappa_s$  detection, and user-friendly  $\kappa_s$  computation, where the user moves the cursor of the mouse to indicate the pixel whose propagation order defines the object. The use of propagation order rather than the pixel cost is important to create smoother transitions between possible objects, facilitating the user's work. The new methods have considerably reduced the number of user interactions in medical image segmentation with respect to the previous approaches. We believe that these results are extensive to other image types by suitable choice of pre-processing and dissimilarity function.

The interactive  $\kappa_s$  detection counted with real time response for every position of the

	<b>M1</b>	<b>PDK</b>	<b>M2</b>	<b>M3</b>	<b>M1,M2</b>	<b>M1,M3</b>	<b>M2,M3</b>
O1	35.0	83.4%	29.5	77.6	0.965	0.969	0.962
O2	38.5	92.4%	29.3	38.8	0.904	0.890	0.915
O3	31.8	0.0%	31.3	61.3	0.992	0.932	0.935
O4	29.7	57.4%	27.5	46.8	0.922	0.914	0.918
O5	15.0	—	15.0	61.0	0.973	0.954	0.946
O6	26.3	27.1%	26.3	37.8	0.992	0.982	0.981
O7	47.5	6.8%	46.3	284.8	0.973	0.931	0.930

Table 4.3: The percentages of different  $\kappa_s$  values (PDK) in M1, the average numbers of user interactions for each object and method, and the average similarity values between different methods for a same object.

	<b>M1</b>			<b>M2</b>			<b>M3</b>	
	NIS	IKD	AKD	NIS	NES	AKD	NIS	NES
O1	30.7	4.3	26.4	18.0	11.5	13.0	26.8	50.8
O2	30.0	8.5	21.5	25.3	4.0	24.3	18.8	20.0
O3	30.0	1.8	28.2	30.3	1.0	30.3	30.3	31.0
O4	24.5	5.2	19.3	22.3	5.2	19.8	22.3	24.5
O5	15.0	0.0	15.0	15.0	0.0	15.0	44.0	17.0
O6	26.3	0.0	26.3	26.3	0.0	15.5	22.8	15.0
O7	47.5	0.0	47.5	46.0	0.3	46.0	66.0	218.8

Table 4.4: Average numbers of internal seeds (NIS), interactive  $\kappa_s$  detections (IKD), external seeds (NES), and automatic  $\kappa_s$  detections (AKD).

cursor, but this may not be feasible in 3D segmentation involving several slices. In this sense, the algorithms based on interactive  $\kappa_s$  detection are more adequate for 2D/3D segmentation in a slice by slice fashion, where seeds may be automatically propagated along the slices. In such a case, the interactive  $\kappa_s$  detection can be used to correct segmentation when the automatic detection of  $\kappa_s$  fails.

Seed competition with automatic  $\kappa_s$  detection (Algorithm 3) seems to be the most promising approach. We are currently investigating two approaches for 3D segmentation of medical images: (i) automatic segmentation with only internal seeds and automatic  $\kappa_s$  detection, and (ii) interactive segmentation with automatic  $\kappa_s$  detection, where the user can add/remove internal and external seeds, and subsequent IFTs are executed in a differential way [36].

## Acknowledgments

The authors thank FAPESP (Procs. 05/59808-0, 05/58103-3, 05/56578-4, and 03/13424-1), CNPq (Procs. 302617/2007-8 and 472402/2007-2), and CAPES for the financial support. Furthermore, the authors thank the reviewers and editors for their contributions.

# Capítulo 5

## Cloud Models: Their Construction and Employment in Automatic MRI Segmentation of the Brain

### 5.1 Introduction

Image segmentation involves object *recognition* and *delineation* [83]. Recognition is the task of determining an object's approximate location in the image. Delineation completes segmentation by defining the exact spatial extent of the object. Humans usually outperform computers in object recognition, but the reverse is true for delineation. While the user can often solve the recognition problem by simple point (seed) selection or by an effective initialization action, precise delineation is challenging due to the intra and inter operator subjectivity. On the other hand, computers can be very precise, even when they are not accurate, but the absence of global information (e.g., an object model) makes object recognition a difficult task for them. This explains why some successful interactive approaches combine recognition by the user with delineation by the computer in a synergistic way, for effective and foolproof segmentation [44, 36, 17, 94].

Segmentation methods can be roughly divided into *model-based* and *image-based* approaches. Model-based methods create statistical models by employing supervised learning. A training set of object's instances is provided with appropriate human interaction and these data are registered into a common reference space to form the model. Active shape models [27] (ASM) and atlas-based approaches [51, 93, 89] are examples of model-based methods that have been used for MR-image segmentation of anatomic structures in the brain [35, 52]. Accurate registration is a separate problem in these methods which is also required during segmentation. In ASM, landmarks have to be selected on the surface of the training objects and their correspondence provides a

statistical model of possible variations in shape. The registration between the image and the model during segmentation also sometimes ignores important image information, by the act of forcing the results to fit with the model. Brain atlases are usually created by registration of training images based on certain landmarks (e.g., anterior and posterior commissures) and deformation fields. In the reference space, image structures suffer from different degrees of distortion, making the matching among corresponding voxels inexact. Image-based methods in turn exploit image properties for more effective delineation, but their lapses in global information makes object recognition an insurmountable problem.

In view of these dilemmas, some recent methods have addressed automatic segmentation by combining model-based approaches for recognition with image-based approaches for delineation [67, 22, 81, 82]. Essentially, the model plays the role of the human operator while an image-based algorithm performs delineation, and both operate in a synergistic way until an optimum state is reached. In this paper, we pursue our previous work on *object cloud models* [81, 82] which present the following advantages: the cloud models dismiss registration during training and segmentation; they take into account the entire object's boundary during delineation and recognition rather than only some *control points*, as in [27]; and they can be easily extended to multidimensional images.

An *object cloud model* (OCM) was introduced in [81] as a triple comprising a fuzzy object, a delineation algorithm, and a criterion function. It captures shape variations of a given object to form an uncertainty region for its boundary. For any image position, delineation is executed in the uncertainty region to obtain a candidate object and the criterion function assigns a score to it. Image segmentation is defined by the candidate with the highest score. In order to capture more shape differences, we proposed the *cloud bank model* (CBM) in [82], which uses multiple clouds per object. In some applications, however, multiple objects may define a *cloud system* by adding their relative position into the model. This is the case of MR-T1 images of the brain and several other medical imaging applications. Therefore, we propose in this paper the *cloud system model* (CSM) to handle multiple objects simultaneously. The methods are compared for the automatic MR-T1 image segmentation of: (S1) the brain (without stem), (S2) the cerebral hemispheres, (S3) the cerebellum, and (S4 and S5) the right and left hemispheres.

In the literature, most approaches to segment S1-S5 are based on atlas registration or surface-extraction techniques. CSM does not require registration and segments S1-S5 simultaneously, precisely and accurately. Besides, a quick evaluation of several other tools revealed that CSM is much simpler and faster. For example, it is about 5 times faster than BrainVisa [72], with no need for manual landmark specification. As compared with the recent work in [65], which takes about 3 hours using deformable registration and 17 atlases, the proposed approach is much simpler, faster and produces similar Dice measures for S3. Other examples are SurfRelax [61], which takes about 6 minutes and

requires image warping onto a template, CLASP<sup>1</sup> and FreeSurfer [31], which take hours to complete segmentation of S1-S5.

We present the general definitions of all object cloud models - OCM, CBM, and CSM - in Section 5.2; instantiate them for brain MR-T1 image segmentation in Section 5.3; discuss experimental results in Section 5.4; and state conclusions in Section 5.5. The experiments show the advantages of CSM and improvements over our previous works [81, 82] for this particular application.

## 5.2 Object Cloud Models

For a given object of interest (e.g., brain, cerebellum), a set of training images with the object's instances must be provided. These instances should capture among them shape variations of that object in order to teach the computer how to recognize it in the image. In atlas-based approaches [51], the images are registered to a chosen reference image by finding a geometric transformation that best matches them according to a criterion of similarity. During the registration process, the reference image remains fixed while the others are deformed into its geometric space. Following this, the model is obtained as the averaged image template together with the tissue distribution maps obtained by averaging segmentations over all subjects. Suitable reference selection is itself a problem, since the atlas can be biased towards the anatomy of the chosen image. Hence, this image should be the one that best represents the anatomy of a population under study [93].

Suppose, instead of registering the training instances, we only translate their binary segmentations on to a common reference point (geometric center) and compute their average. This results in an image with a fuzzy appearance that resembles a *cloud* (Figure 5.1a). From it we may obtain relevant shape informations such as prior boundary knowledge (see Figure 5.1b and Section 5.2.3). Each *cloud* image also defines (i) an interior region consisting of voxels that belong to all training instances, (ii) an exterior region with voxels that do not belong to any instance, and (iii) an uncertainty region composed of voxels that belong to some but not to all instances (Figure 5.1c).

The *cloud* model (OCM) is a triple that consists of a fuzzy object (*cloud* image), a delineation algorithm  $A$  (this may be any algorithm, but, for the reasons mentioned earlier, preferably an image-based approach), and a functional  $F$ . To segment a new image, the *cloud* moves over the image and, for each position, algorithm  $A$  is executed inside the uncertainty region to obtain a candidate segmentation. The functional is evaluated on this segmentation to obtain a matching score for recognition, by taking into account local and global object properties (e.g., shape and texture). The desired segmentation is expected

---

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

to be the one with maximum score [81] (Figures 5.2a-c). However, when the variability in the training set is too high the shape information within the *cloud* image tends to degenerate and the method loses precision. To circumvent this problem, we may separate the training instances into *groups* (clusters) of high similarity in shape, rotation and size (texture may be used as well). Each group defines its own *cloud* image resulting in a bank of clouds, and the desired segmentation is expected to be the one with maximum score among segmentations obtained from all individual *clouds* from this *cloud bank model* (CBM) [82]. In this manner, the need to define a reference image is completely obviated. As new images are added to the training set, they start new groups or are inserted in some existing group (the groups may have overlap). In medical imaging and other applications (e.g., license plate recognition), it is possible to acquire images as per a disciplined regimen so that a small number of groups in the bank will suffice. Preprocessing, that depends only on the image being processed, can also help in reducing the number of groups.

Note that delineation is constrained in the uncertainty region, and it also exploits prior shape information (see Section 5.2.3) which are defined by the model. Recognition is based on the functional, but it is applied to the delineated objects. Thus, the model employs recognition and delineation in a tightly coupled manner [67].

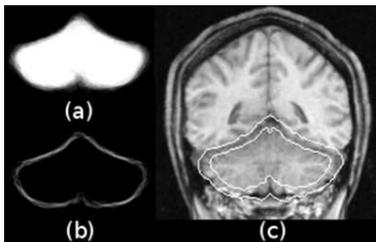


Figure 5.1: (a) A coronal slice of the 3D cloud image of the cerebellum. (b) The shape-based weight image. (c) Example of an uncertainty region over a slice of a test image.

In the case of multiple objects, each object has its own OCM (or CBM), separately, and the segmentation can follow independently, or in a hierarchical search. In this last case, the search spaces for the internal objects are constrained by the larger objects that subsume them (e.g., Figure 5.3, used in [81, 82]). However, in some applications (such as medical), the arrangement among objects does not change, so it is possible to make better use of contextual information, by computing their mean relative positions with respect to a common reference point (centroid of all objects). As a result, we have a *cloud system* (Figure 5.4a). To segment a new image, we consider the prior displacement knowledge in order to fix the position of each object cloud relative to the moving reference point. For

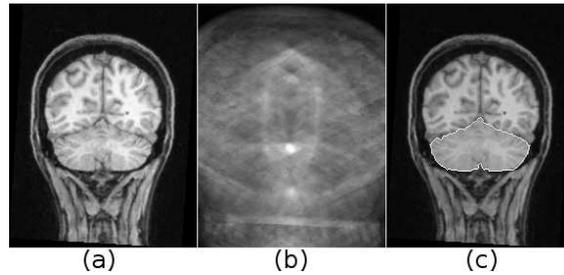


Figure 5.2: (a) An input image. (b) The recognition score for all positions. (c) The final segmentation at the best location.

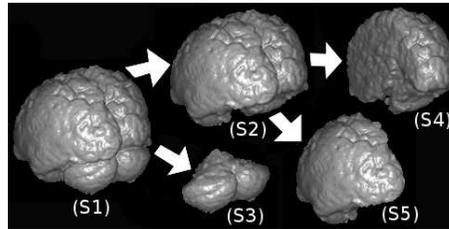


Figure 5.3: Hierarchical search for the brain structures S1-S5: S1 is the brain without the brain stem, S2 represents the cerebrum, S3 is the cerebellum, and S4 and S5 are the right and left cerebral hemispheres.

each search position, delineation is done inside the uncertainty regions of all object clouds, and a combined score for recognition is obtained by the functional  $F$ . This simultaneous treatment of multiple objects makes it less likely to miss the right position, due to the better use of contextual information (Figures 5.4b-c) [22].

Again, when the variability of the training instances is too large, we can separate the instances into *groups* of high similarity, but now we must also consider possible correlations among the objects in terms of relative position, rotation, and size. This leads to a *cloud system model* (CSM), which is formally presented below. The previous cloud models (OCM,CBM) are subsumed by CSM as particular cases.

An image  $\hat{I}$  is a pair  $(\mathcal{I}, \vec{I})$  where  $\mathcal{I} \subset Z^n$  is the image domain and  $\vec{I}(p)$  assigns a set of  $b$  scalars  $I_i(p)$ ,  $i = 1, 2, \dots, b$ , to each voxel  $p \in \mathcal{I}$ . This definition applies to multi-dimensional and multi-parametric images. We are interested in  $n = 3$  and  $b \geq 1$ . The index  $i$  is removed when  $b = 1$ . For  $c$  objects  $l = 1, 2, \dots, c$ , a segmentation instance is represented by a labeled image  $\hat{L} = (\mathcal{I}, L)$ , in which each label  $1 \leq L(p) \leq c$  assigns a voxel  $p \in \mathcal{I}$  to one object out of  $c$  objects, and  $L(p) = 0$  is used to designate background

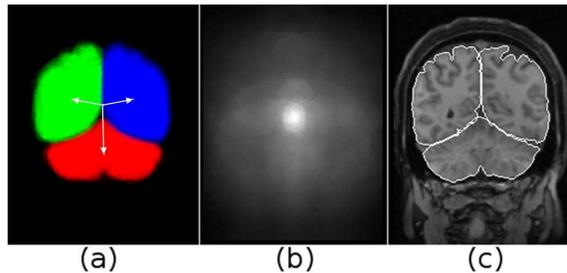


Figure 5.4: (a) A cloud system for  $c = 3$  objects shown in RGB color space. (b) The recognition score for all positions and (c) the final segmentation handling multiple objects simultaneously.

voxels. A binary image  $\hat{B} = (\mathcal{I}, B)$  may be used to represent each object such that  $B(p) = 1$  for object voxels and  $B(p) = 0$  for background voxels.

Training instances are given as a set of labeled images  $\hat{L}_i = (\mathcal{I}_i, L_i)$ ,  $i = 1, 2, \dots, N$ . Suppose these instances are separated into  $m$  groups of high similarity, as to be discussed in Section 5.2.1. Each group is represented by a set  $\mathcal{G}_g$  of images,  $g = 1, 2, \dots, m$ , such that  $\hat{L}_i \in \mathcal{G}_g$  if the image  $\hat{L}_i$  is in the  $g$ th group. Let  $\hat{B}_{i,l} = (\mathcal{I}_i, B_{i,l})$  be the binary image of the  $l$ th object in the  $i$ th image (i.e.,  $B_{i,l}(p) = 1$  if  $L_i(p) = l$ , and  $B_{i,l}(p) = 0$  otherwise). For any given object label  $l$  and group  $g$ , the average of the binary images  $\hat{B}_{i,l}$  for all  $i$  in  $\mathcal{G}_g$ , after translating them to a fixed reference point, creates a *cloud image*  $\hat{C}_{g,l} = (\mathcal{C}, C_{g,l})$ , where  $C_{g,l}(p) \in [0, 1]$ . For any cloud,  $C_{g,l}(p) = 1$  in its interior,  $0 < C_{g,l}(p) < 1$  in its uncertainty region, and  $C_{g,l}(p) = 0$  in its exterior. In the single object case (i.e.,  $c = 1$ ), the method becomes the same as in CBM [82], and it becomes a single OCM [81] when  $c = 1$  and  $m = 1$ . But in the case of multiple objects (i.e.,  $c > 1$ ), it also exploits the relative positions among the objects within each group. For a given group  $g$ , the arrangement of the object clouds is captured by the displacement vectors  $\vec{D}_{g,l}$ ,  $l = 1, 2, \dots, c$ , which store the average positions of the object's centroids in relation to their joint centroid for all images in  $\mathcal{G}_g$ .

$$\vec{D}_{g,l} = \frac{1}{|\mathcal{G}_g|} \sum_{\forall i | \hat{L}_i \in \mathcal{G}_g} (\vec{P}_{i,l} - \vec{P}_i), \quad (5.1)$$

where  $|\mathcal{G}_g|$  is the cardinality of the set  $\mathcal{G}_g$ ,  $\vec{P}_{i,l}$  is the centroid's coordinates of the object  $l$  in the binary image  $\hat{B}_{i,l}$ ,  $l = 1, 2, \dots, c$ , and  $\vec{P}_i$  is the centroid's coordinates of their union in the label image  $\hat{L}_i$ . Note that this formulation also includes the single object case since  $\vec{D}_{g,l}$  becomes the null vector when  $c = 1$ .

The following subsections provide more details about all relevant parts of the CSM,

such as the grouping strategy, the object search, the graph assembly, the delineation algorithm  $A$ , and the functional  $F$ . The model components and parameters are then customized for the specific application of MR-T1 image segmentation of the brain as described in Section 5.3.

### 5.2.1 Grouping

The grouping can be done by representing the training instances  $\hat{L}_i$  as nodes of a complete graph; the arcs between  $\hat{L}_i$  and  $\hat{L}_j$  are weighted by a metric that valuates their similarity. In the single object case ( $c = 1$ ), we may consider the Dice similarity as this metric after centralizing the training instances by their centroid vectors  $\vec{P}_{i,l}$  (Figure 5.5a). In the case of multiple objects ( $c > 1$ ), we centralize the label images  $\hat{L}_i$ ,  $i = 1, 2, \dots, N$ , by  $\vec{P}_i$  and consider a combined similarity value (e.g., the mean Dice similarity among corresponding object's instances). The groups are then selected as maximal cliques [109], wherein all pairs of training instances have similarity above a threshold. Thresholding helps us to assure that only compatible data are used within the same cloud system, and the size of the uncertainty regions can be controlled by the threshold (Figure 5.5b). On the other hand, we must keep the number of groups as low as possible in order to make the method computationally efficient. We start with  $N$  maximal cliques, such that each contains one training image. Since these cliques may have overlap, we select only a minimal subset of them that comprises all the images. In most cases this can be accomplished by successively applying three simple rules: eliminate repeated cliques (or cliques having all unselected nodes contained in some other clique), select all cliques with at least one exclusive element, and ignore remaining cliques that have all elements already selected. When this fails, there is a cycle of dependencies that may be broken by arbitrarily selecting one of the remaining cliques, and then resuming the three rules.

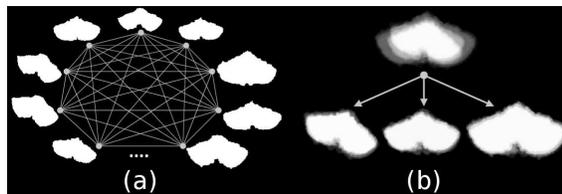


Figure 5.5: (a) The training instances ( $c = 1$ ) are mapped as nodes of a complete graph, (b) the groups are selected as maximal cliques.

If there are only a few training images available, they may be complemented with images created by random transformations (within acceptable limits) of the given images

which capture variations that are likely to happen in the application.

### 5.2.2 Locating objects

Let  $\hat{I} = (\mathcal{I}, I)$  be an image to be segmented. For each search position with coordinates given by  $\vec{p}$ , each cloud image  $\hat{C}_{g,l}$  is positioned with its center at  $\vec{q} = \vec{p} + \vec{D}_{g,l}$ , and its uncertainty region is projected over a set of voxels  $\mathcal{U} \subset \mathcal{I}$ . That is, the displacement vectors  $\vec{D}_{g,l}$  are used to fix the cloud's position for each object relative to the moving search point  $\vec{p}$ . Then, for each position  $\vec{p}$  of a search region, a score  $F_l$  is obtained for each label  $l = 1, 2, \dots, c$ , by analyzing the candidate segmentations computed by the delineation algorithm for each cloud in a given cloud system. A combined score for recognition is obtained by the functional  $F$ , such as the mean value  $(\sum_{l=1}^c F_l)/c$ . This process is repeated for all groups, and the best location among all groups is selected as the final result.

A multiscale search can be used to speed up this recognition task, by starting the search at the lowest resolution and refining the best detected locations in the higher resolutions. During this refinement in the higher resolutions, we may also allow each cloud of a cloud system to move independently from the others in order to have more flexibility.

### 5.2.3 Arc-weight estimation for delineation

Image-based delineation algorithms usually make direct/indirect use of some image-graph concept, and their success strongly depends on a suitable arc-weight estimation, which usually takes into account image attributes and/or object information obtained by supervised learning [83]. For the given image  $\hat{I} = (\mathcal{I}, I)$  to be segmented, we associate a weight image  $\hat{W} = (\mathcal{I}, W)$  which is in turn associated with a graph  $(\mathcal{I}, \mathcal{A})$ . The graph's nodes are the voxels  $p \in \mathcal{I}$  and arcs  $(p, q) \in \mathcal{A}$  are defined between 6-neighbors. For convenience, we store the weights in voxel resolution and use their interpolated values during execution for the graph. That is, each arc  $(p, q) \in \mathcal{A}$  is weighted by the mean value  $w(p, q) = \frac{W(p)+W(q)}{2}$ . The weight  $W(p)$  assigned to each voxel  $p \in \mathcal{I}$  is a linear combination of an image-based weight  $W_i(p)$ , an object-based weight  $W_o(p)$ , and a shape-based weight  $W_s(p)$  provided by the cloud model.

$$W(p) = \lambda_i W_i(p) + \lambda_o W_o(p) + \lambda_s W_s(p), \quad (5.2)$$

where  $\lambda_i + \lambda_o + \lambda_s = 1$ . The weight  $W_i(p)$  aims at capturing discontinuities that may exist between homogeneous regions and is taken as the magnitude of an image gradient. The weights  $W_o(p)$  take into account prior knowledge about the intensities of the objects under consideration in order to characterize the discontinuities that exist between them

and the rest of the image. This weight is usually application-dependent (see Section 5.3.2), although some general techniques exist to estimate them [83]. The weight  $W_s(p)$  encodes prior shape information obtained from each *cloud* image  $\hat{C}_{g,l}$  (Figure 5.1a) by computing its gradient magnitude for all groups and labels (Figure 5.1b). It is combined with the other weights as each *cloud* moves over the image (i.e., each cloud has its own  $W_s(p)$ ), usually with a low  $\lambda_s$  value since its major role is only to discriminate regions with poorly defined borders.

## 5.2.4 Delineation algorithm

For delineation, we use an algorithm called IFT-SC (*IFT segmentation by Seed Competition*) which is based on the image foresting transform [43] (IFT) and is supported by the theoretical foundations given in [79].

For each search position, each cloud image  $\hat{C}_{g,l}$  defines an uncertainty region as a set  $\mathcal{U} \subset \mathcal{I}$  of voxels (Figure 5.1c). The interior and exterior regions contain boundary voxels, which have at least one voxel in  $\mathcal{U}$  as a 6-neighbor. These boundary voxels form one internal set  $\mathcal{S}_i$  and one external set  $\mathcal{S}_e$  of seeds for the IFT-SC. A *path*  $\pi$  in the image graph  $(\mathcal{I}, \mathcal{A})$  is a sequence of adjacent voxels  $\langle p_1, p_2, \dots, p_n \rangle$ . For the given set of seeds  $\mathcal{S} = \mathcal{S}_i \cup \mathcal{S}_e$ , the cost of a path  $\pi$  is defined by a *path-cost function*. The considered cost functions are:

$$f_1(\pi) = \begin{cases} \max_{i=1,2,\dots,n-1} w(p_i, p_{i+1}) & \text{if } p_1 \in \mathcal{S} \\ +\infty & \text{otherwise} \end{cases} \quad (5.3)$$

$$f_2(\pi) = \begin{cases} \sum_{i=1,2,\dots,n-1} [w(p_i, p_{i+1})]^\eta & \text{if } p_1 \in \mathcal{S} \\ +\infty & \text{otherwise} \end{cases} \quad (5.4)$$

When  $\mathcal{U}$  contains the object's boundary (Figure 5.1c), it is expected that the arc weights within  $\mathcal{U}$  are higher on the object's boundary than inside and outside it. The seed sets  $\mathcal{S}_i$  and  $\mathcal{S}_e$  compete for voxels in  $\mathcal{U}$ , such that a voxel receives label  $L(p) = 0$  if the minimum-cost path comes from  $\mathcal{S}_e$ , and label  $L(p) = l$  otherwise. The object is then defined as the union between the interior of the *cloud* and the voxels with labels  $L(p) = l$  in  $\mathcal{U}$ .

The IFT solves this minimization problem by computing an *optimum-path forest* — a function  $P$  which contains no cycles and assigns to each node  $q \in \mathcal{I}$  either its predecessor node  $P(q) \in \mathcal{I}$  in the optimum path with terminus  $q$  or a distinctive marker  $P(q) = \text{nil} \notin \mathcal{I}$ , when  $\langle q \rangle$  is optimum (i.e.,  $q$  is said to be a *root* of the forest). The cost functions by Eq. 5.3 and 5.4 force the roots to be in  $\mathcal{S}$  (Figure 5.6). As we change the parameter  $\eta > 0$  in  $f_2$ , we obtain a whole family of solutions that includes  $f_1$  as a special case in the limit when  $\eta \rightarrow \infty$ . This is noticeable from the empirical results shown in [82]. We

note that,  $f_2$  usually imposes more regularization to the object's boundary than  $f_1$ , but on the other hand,  $f_1$  fits to the protrusions and indentations of the boundary better than  $f_2$ . The IFT-SC is especially important, because our model is being projected to allow user's intervention for small corrections if necessary, and it was shown in [79] that the boundaries obtained by the IFT-SC with path-cost function  $f_1$  are also piecewise optimal. This property is essential to conserve user control during interactive corrections as discussed in [79], and these corrections can be quickly performed by using the DIFT algorithm [36].

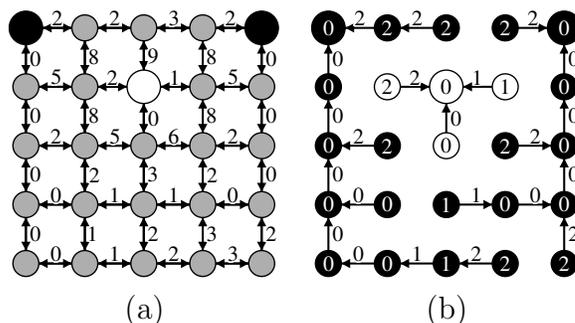


Figure 5.6: IFT-SC example: (a) A 4-neighborhood graph, where the numbers indicate the arc weights. Three seeds are selected, one is internal (white dot) and two are external (black dots). (b) An optimum-path forest for the path-cost function  $f_1$ . The numbers inside the nodes indicate the costs of the optimum paths, which are stored in a predecessor map  $P$ . The label  $L(q) = 0$  (black), or  $L(q) = l$  (white) of each seed  $q$  is propagated to all pixels within its respective optimum-path tree.

### 5.2.5 Functional for recognition

The functional  $F$  should be the one that best discriminates the objects under consideration, being consequently application-dependent. Several delineation algorithms are already based on some sort of functional, like for example a graph-cut measure [17]. Indeed, the IFT-SC with path-cost function  $f_1$  also optimizes a graph-cut measure according to [79]. A question may be raised concerning the use of the same functional for both delineation and recognition score. This is usually not the best option, because the functionals for delineation are in general designed to best match boundary properties, while the recognition functional could be far more complex, taking into account more global information, such as the shape details and internal information. The functional used for brain segmentation is discussed in Section 5.3.3.

### 5.3 Application to brain segmentation

In this work, we consider the segmentation of the brain structures S1-S5 as summarized in Figure 5.3. The segmentation of the left and right cerebral hemispheres enables the investigation of asymmetries in shape and texture which may be related to several degenerative diseases. The separation of the cerebellum makes its analysis possible independently of the rest of the brain, which is important in understanding the relationships between aging and the decline in cognitive functioning [65]. The hemispheres are connected through the corpus callosum. The cerebellum is connected to the rest of the brain through the brain stem and through its top due to partial volume. The absence of a clear boundary between these structures poses a challenge for segmentation (Figure 5.7).

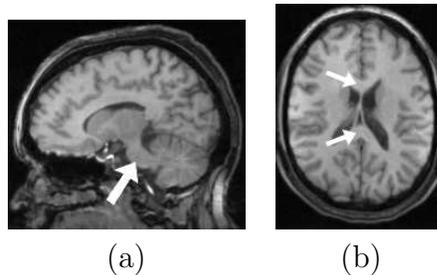


Figure 5.7: (a) The cerebellum is connected to the cerebrum through the brain stem. (b) The cerebral hemispheres are connected through the corpus callosum.

This application of MR brain image segmentation perfectly fits the requirements for the CSM model. The images are acquired in a well controlled environment, where the patient remains lying in a horizontal position without moving. In extreme cases (e.g., Claustrophobia) even mild sedation may be adopted. Hence, only a small number of clouds will suffice as desired.

#### 5.3.1 Preprocessing & grouping

As stated in Section 5.2, preprocessing that depends only on the image being processed can help in reducing the total number of groups required. The MR-T1 images are interpolated to the same cubic dimensions ( $0.98mm^3$ ) and aligned by the mid-sagittal plane [10] (MSP). This approach is fast (a few seconds), free of parameters, and independent of templates. Interpolation and alignment reduces the number of groups by reducing the data variability, and the MSP also reduces the search region. Note that, the MSP alignment operation does not depend on any reference image. For grouping, we used the methodology of Section 5.2.1, with the combined similarity threshold of 0.8.

### 5.3.2 Arc-weight estimation

In this application, we may take advantage of the fact that most of the brain structures are surrounded by CSF in order to emphasize their boundaries. However, owing to the intensity non-standardness among images, and also the inhomogeneity, the partial volume effects, and their interplay [70], it is difficult to characterize the exact intensity profile of the CSF in advance. But we may still enforce a more specific intensity interval which is more likely to contain the transition from CSF to GM. Consider  $t_1$  as a lower bound of the CSF intensity, and  $t_2$  as a lower bound for the white matter. We can accentuate the intensity slope in the interval  $[t_1, t_2]$  by considering the filtered image  $\hat{I}' = (\mathcal{I}, I')$  as follows.

$$I'(p) = \begin{cases} I(p) & \text{if } I(p) < t_1 \\ (1 - \gamma)t_1 + \gamma \cdot I(p) & \text{if } I(p) \in [t_1, t_2] \\ (t_2 - t_1)(\gamma - 1) + I(p) & \text{otherwise} \end{cases} \quad (5.5)$$

where  $\gamma > 1$  defines the new slope value (e.g.,  $\gamma = 5$ ), such that differences within  $[t_1, t_2]$  in the original image  $\hat{I}$  become higher in  $\hat{I}'$ . Subsequently, we consider a simple application-specific weight  $W_o(p)$  that emphasizes dark voxels close to bright areas in the filtered image  $\hat{I}'$ . For all voxels  $p \in \mathcal{I}$ ,  $W_o(p)$  is computed as the sum of differences  $I'(q) - I'(p)$  for all brighter voxels  $q$  (i.e.,  $I'(q) > I'(p)$ ) in a small neighbourhood around  $p$ . Thus, a suitable emphasis is obtained in the superior part of the cerebellum, and the transitions from GM to WM become weakened (Figure 5.8). In the experiments we considered  $t_1$  as the Otsu's optimal threshold (which gave a good lower bound for the CSF in all cases), and we considered  $t_2$  as the mean intensity in  $\hat{I}$  considering only values above  $t_1$ . The other weights  $W_i(p)$  and  $W_s(p)$  are computed as discussed in Section 5.2.3,  $W_s(p)$  being especially important to complete poorly defined borders (Figure 5.7).

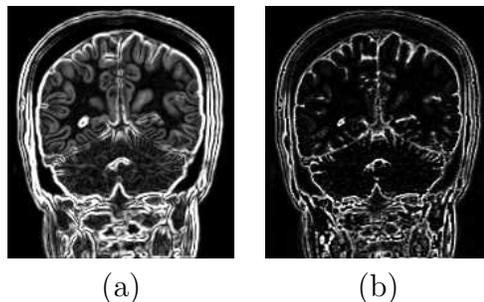


Figure 5.8: (a) The image-based weight  $W_i$ , (b) the object-based weight  $W_o$ .

### 5.3.3 Recognition functional & delineation algorithm

The previous versions [81, 82] used the mean-cut measure [124] as the functional  $F_l$ , since it provides a good summary of the object’s boundaries, being free of any undesirable bias. In this work, we improve on this functional by including a penalty factor based on our application-specific knowledge. The likelihood of having image intensities below  $t_1$  inside the brain structures is supposed to be very low. Therefore, if the proportion of voxels below  $t_1$  is high, among all voxels achieved for the object from  $\mathcal{U}$ , then this is a strong indication that the cloud is poorly positioned.

With respect to the delineation algorithm, some works [64] propose the use of a graph-cut algorithm with an increasing transformation (i.e., the exponential function). But since we have a potent arc-weight estimation strategy which also includes prior shape information (filling any gaps on the object’s boundary), a graph-cut algorithm under this scenario will theoretically lead to an approximation of the IFT-SC according to [79]. Hence, it is preferable to use the faster IFT-SC algorithm.

Algorithm 6 below performs the IFT-SC delineation and functional computation  $F_l$  simultaneously for any given image location. It takes time proportional to the number of voxels in  $\mathcal{U}$  (sublinear), when the priority queue  $Q$  is implemented properly [44]. For multiple objects, the scores  $F_l$  are combined as  $F = (\sum_{l=1}^c F_l) / c$ , as discussed in Section 5.2.2.

#### Algorithm 6 – DELINEATION ALGORITHM FOR $f_1$

INPUT: Image  $\hat{I}$ , weight image  $\hat{W}$ , adjacency  $\mathcal{A}$ , seed sets  $\mathcal{S}_i$  and  $\mathcal{S}_e$ , and uncertainty region  $\mathcal{U}$ .

OUTPUT: Label map  $L$  initially zeroed, an optimum-path forest stored in a predecessor map  $P$  initially with *nil* and functional  $F_l = 0$  initially.

AUXILIARY: Cost map  $c$  initially zeroed, variables  $cst$ , object acquisitions  $oa = 0$ , penalty acquisitions  $pa = 0$  and cut size  $sz = 0$  initially, priority queue  $Q$  initially empty, and status map  $s$  to indicate when a voxel has been inserted in  $Q$  (1), has never been inserted in  $Q$  (0), or has been removed from  $Q$  (2).

1. **For** all  $p \in \mathcal{U}$ , set  $c(p) \leftarrow +\infty$  and  $s(p) \leftarrow 0$ .
2. **For** all  $p \in \mathcal{S}_i$ , set  $L(p) \leftarrow l$ ,  $s(p) \leftarrow 1$ , and insert  $p$  in  $Q$ .
3. **For** all  $p \in \mathcal{S}_e$ , set  $L(p) \leftarrow 0$ ,  $s(p) \leftarrow 1$ , and insert  $p$  in  $Q$ .
4. **While**  $Q$  is not empty, **do**
5.     Remove from  $Q$  a voxel  $p$  such that  $c(p)$  is minimum.
6.     Set  $s(p) \leftarrow 2$ .
7.     **For each**  $q$  such that  $(p, q) \in \mathcal{A}$ , **do**
8.         **If**  $c(q) > c(p)$ , **then**

```

9.   |   |   |   Compute  $cst \leftarrow \max\{c(p), \frac{W(p)+W(q)}{2}\}$ .
10.  |   |   |   If  $cst < c(q)$ , then
11.  |   |   |   |   If  $s(q) = 1$ , remove  $q$  from  $Q$ .
12.  |   |   |   |   Set  $c(q) \leftarrow cst$ ,  $L(q) \leftarrow L(p)$ .
13.  |   |   |   |   Set  $P(q) \leftarrow p$ .
14.  |   |   |   |   Insert  $q$  in  $Q$  and  $s(q) \leftarrow 1$ .
15.  |   |   |   Else
16.  |   |   |   |   If  $s(q) = 2$  and  $L(q) \neq L(p)$ , then
17.  |   |   |   |   |   Set  $F_l \leftarrow F_l + \frac{W(p)+W(q)}{2}$ .
18.  |   |   |   |   |   Set  $sz \leftarrow sz + 1$ .
19.  |   |   |   If  $L(p) = l$ , then
20.  |   |   |   |   Set  $oa \leftarrow oa + 1$ .
21.  |   |   |   |   If  $I(p) < t_1$ , then set  $pa \leftarrow pa + 1$ .
22.  |   |   |   Set  $F_l \leftarrow (F_l/sz) \times (1 - pa/oa)$ .

```

Lines 1–3 initialize maps and insert seed voxels in  $Q$ . Lines 4–14 compute the maps  $L$ ,  $c$  and  $s$  during the IFT. The main loop computes an optimum path cost from the seeds to every node  $p$  in a non-decreasing order of values (Lines 4–14). At each iteration, a path of minimum cost  $c(p)$  is obtained in  $P$  when we remove its last voxel  $p$  from  $Q$  (Line 5). Lines 7–14 evaluate if the path that reaches an adjacent voxel  $q$  through  $p$  is cheaper than the current path with terminus  $q$  and update  $Q$ ,  $c(q)$ ,  $s(q)$ ,  $L(q)$  and  $P(q)$  accordingly. The remaining lines compute the functional  $F_l$  on-the-fly.

Note that, before changing position of  $\mathcal{U}$ , the maps and auxiliary variables can be reinitialized in sublinear time, such that the search for a desired object can be done more efficiently. The above algorithm can be easily modified for  $f_2$  if we substitute line 9 by  $cst \leftarrow c(p) + \left[ \frac{W(p)+W(q)}{2} \right]^\eta$ .

### 5.3.4 Multiple object search

Instead of making a hierarchical search (i.e., the pipeline of Figure 5.3) as in [81, 82], we made the search simultaneously for the three objects S3, S4 and S5, as described in Section 5.2.2. Later, object S2 is obtained as  $S4 \cup S5$ , and S1 as  $S3 \cup S4 \cup S5$ . The search was constrained only inside the MSP, and we also used a multiscale strategy to speed up the recognition task by using a three-level Gaussian pyramid.

## 5.4 Results

We have first evaluated the method on the MRI datasets of 40 normal subjects from both genders, in the age range from 16 to 49 years. The images were acquired with a 2T Elscint scanner and voxel size of  $0.98 \times 0.98 \times 1.00 \text{ mm}^3$ . We used the leave-one-out approach to compute the mean and standard deviation of the Dice similarity measure between the ground truth and the automatic segmentation (Figures 5.10 and 5.9). Table 5.1 shows the results for different methods. The CSM model was executed with four groups, using path-cost function  $f_1$ , and with arc weights computed using  $\lambda_i = 0.15$ ,  $\lambda_o = 0.75$ , and  $\lambda_s = 0.10$ . The obtained results are considerably superior to those of OCM [81], especially for the cerebellum; and are also much more precise (i.e., lower standard deviation) than the previous CBM [82]. The simultaneous search of multiple objects is also faster than the hierarchical search used in [81, 82]. The mean execution time per group using a 3GHz Pentium IV PC was 27.5 sec, while the hierarchical search took 41 sec. Therefore, the total mean execution time per data set to segment all objects is 110 sec. We also note that our results for brain segmentation S1 are considerably superior to those reported in [9] for *tree pruning* [9] and SPM2 [50] (mean errors of 9.4% and 14.1%, respectively), the latter being a widely used template-based approach for medical research. This is especially notable concerning the removal of the brain stem.

We also conducted experiments with 40 patient images which were acquired post-surgery, some of which had strong morphological changes (Figure 5.11). In this case, some special care has been taken. After training on the 40 controls, we considered a greater degree of contingency in the size of  $\mathcal{U}$ , and we also applied a post-processing operation to clear peripheral voxels below  $t_1$ . For 38 images the results were similar to those for controls, and their worst 5 images had the mean and standard deviation of Dice similarity for S3, S4 and S5, respectively of  $94.52\% \pm 0.68\%$ ,  $96.84\% \pm 0.38\%$ , and  $96.21\% \pm 0.87\%$ . The other two remaining images had larger errors and required interactive repairing by the DIFT algorithm [36].

	OCM [81]		CBM [82]		CSM	
Obj.	Mean	S.dev	Mean	S.dev	Mean	S.dev
S1	96.49	0.51	97.06	1.06	97.37	0.46
S2	96.60	0.51	97.20	1.21	97.59	0.48
S3	92.31	1.22	94.39	0.95	94.97	0.84
S4	95.47	0.44	96.66	1.36	97.16	0.48
S5	95.37	0.60	96.43	1.16	97.05	0.55

Table 5.1: Mean (%) and standard deviation (%) of the Dice similarity.

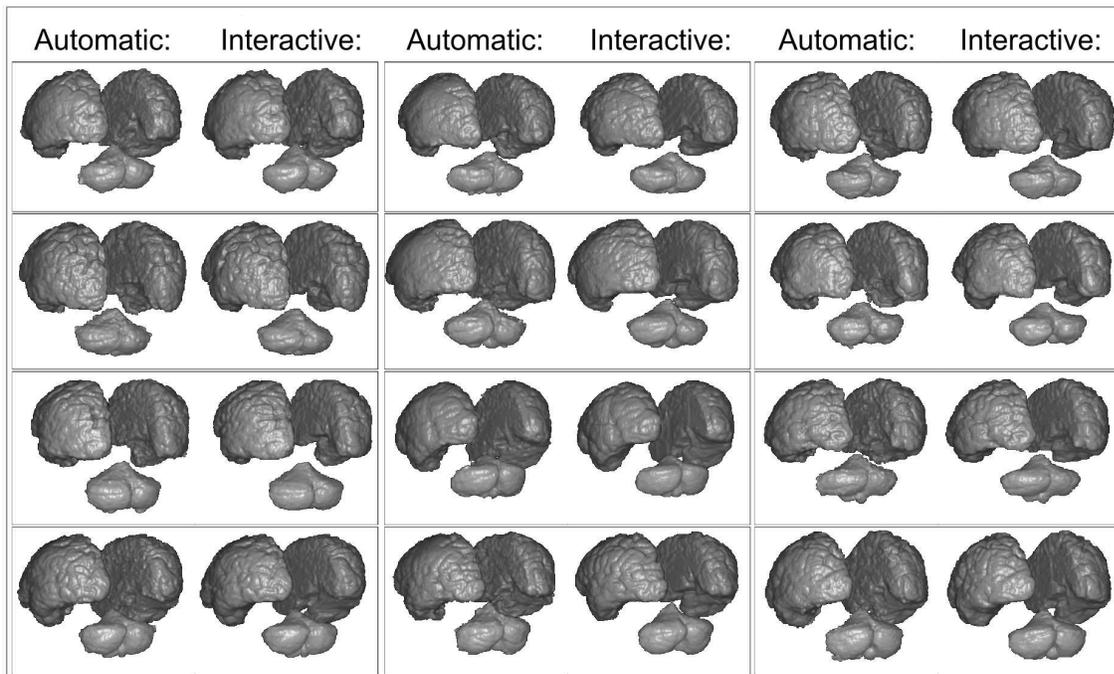


Figure 5.9: Segmentation results by *Cloud System Model* and the provided ground truth, shown as 3D renditions.

## 5.5 Conclusion

We have presented model-based approaches for automatic image segmentation, which employ recognition and delineation in a tightly coupled manner. The previous approaches, called OCM [81] and CBM [82], are particular cases of the proposed method, named Cloud System Model (CSM). The three methods were compared for automatic MR-T1 image segmentation of several brain structures. The results indicated that CSM is more precise and more accurate than OCM and CBM for this particular application. CSM is simpler and faster than other approaches commonly used in the literature in brain research. It can also be easily implemented in parallel, taking advantage of machines with multiple processors and cores.

Our future work will include evaluation of CSM using other medical imaging modalities, employing better quality images from a 3T MRI scanner, segmenting sub-cortical brain structures, and other image analysis applications.

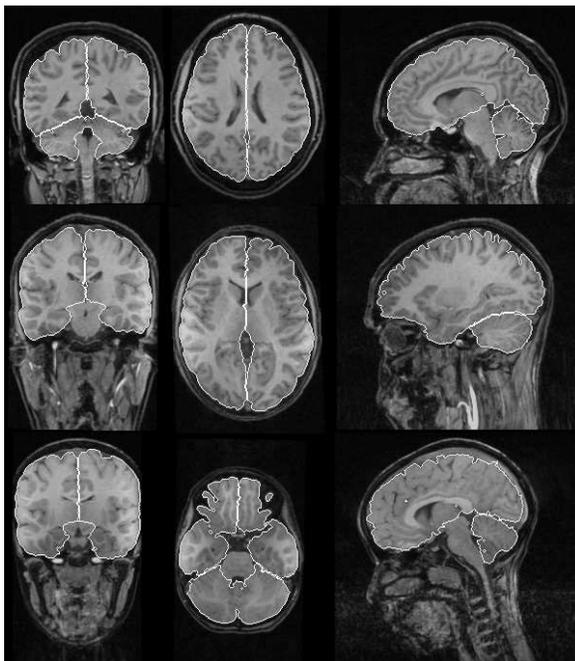


Figure 5.10: Sample slices from control subjects with segmentation results.

## Acknowledgment

The authors thank FAPESP (Proc. 05/59808-0 and Proc. 07/52015-0), and CNPq (Proc. 302617/2007-8) for the financial support, and Dr. Fernando Cendes (FCM-UNICAMP) for the images.

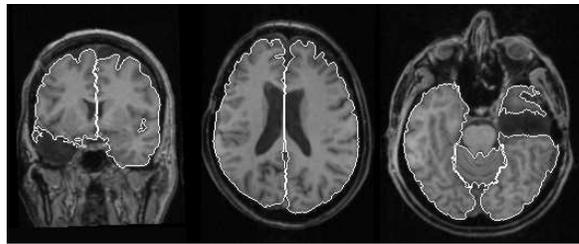


Figure 5.11: Sample slices from patients with segmentation results.

# Capítulo 6

## Conclusões e Trabalhos Futuros

### 6.1 Contribuições & Desafios

Nesta tese foram apresentadas várias contribuições com publicações que tratam os diferentes subproblemas que compõem o problema maior da segmentação. Os métodos desenvolvidos foram demonstrados no âmbito da medicina com aplicações importantes na segmentação de estruturas cerebrais em imagens volumétricas de ressonância magnética T1. Relembramos, porém, que os métodos apresentados também se aplicam a outros domínios de aplicação, tal como demonstrado pelos vários exemplos envolvendo imagens naturais que foram apresentados ao longo dos capítulos. O sistema proposto apresenta três módulos que englobam as diferentes etapas da segmentação, desde a concepção do grafo da imagem, passando por análises teóricas dos principais algoritmos em grafo, chegando até ao delineamento automático por modelos de objetos. As principais contribuições e dificuldades encontradas em cada uma dessas três áreas são sumarizadas abaixo.

**1. Estimativa Sinérgica de Peso das Arestas do Grafo.** No Capítulo 2, nós apresentamos um método interativo para estimar o peso das arestas, que realça as dissimilaridades entre objeto e fundo, tornando o processo de extração de objetos mais eficaz.

O método pode ser utilizado efetivamente por várias abordagens de segmentação baseadas em grafos como demonstramos. Nosso método explora de forma sinérgica as habilidades humanas de reconhecimento e as capacidades do computador para o delineamento. Enquanto o usuário desenha marcadores dentro de cada objeto (incluindo o fundo), pesos de arco são estimados a partir de atributos de imagens e informações dos objetos (pixels sob os marcadores), e uma resposta visual guia a próxima ação do usuário visando melhorar a acurácia.

Os marcadores devem ser traçados sobre as partes mais representativas e distintas

dos objetos, a fim de tornar a estimativa de pesos eficaz. Os marcadores de treinamento podem ser utilizados para iniciar o delineamento e marcadores adicionais selecionadas nas partes semelhantes dos objetos podem ser usados para corrigir a segmentação, mas estes últimos nunca devem ser utilizados para recalcular os pesos que foram estimados na etapa de treinamento.

Um desafio importante nessa área de pesquisa trata-se de como incorporar a informação espacial da localização dos marcadores no processo de realce de modo a ampliar o poder de discriminação entre objeto e fundo. Isto é, muitas vezes objeto e fundo possuem sobreposições no espaço de características que poderiam ser resolvidas através da informação espacial no domínio da imagem. Porém, por outro lado é desejável que o realce seja o menos sensível possível em relação ao posicionamento dos marcadores. Conciliar esses aspectos divergentes é um desafio. Outro ponto diz respeito aos meios pelos quais a separação entre os marcadores usados nas etapas de realce e extração possa ser feita de maneira automática e transparente ao usuário. O emprego de estruturas de indexação para acelerar o cálculo da componente supervisionada dos pesos é outro aspecto a ser considerado.

**2. Avanços Teóricos em Algoritmos de Delineamento.** No Capítulo 3 foram apresentados um conjunto de teoremas que fundamentam a nossa opção preferencial por métodos de delineamento baseados na IFT [43]. Teorema 1 mostra que o algoritmo IFT-SC oferece resultados ótimos de segmentação sob dois pontos de vista: como uma floresta de caminhos ótimos e como um corte mínimo no grafo de acordo com a medida  $E_3$ , que pode levar em conta tanto propriedades da imagem quanto do objeto segundo os pesos de aresta descritos no Capítulo 2. Teorema 2 mostra que essa otimalidade é ainda mais forte, dado que ela é também ótima por partes (*piecewise optimum*). Já o Teorema 4 mostra que procedimentos para contornar o viés existente no algoritmo de *min-cut/max-flow* levam a aproximações do IFT-SC. Estes resultados dão a base teórica para explicar o grande sucesso e popularidade do IFT-SC [68, 36] e seus métodos relacionados (e.g., WT [11], RFC [115, 99], IRFC [24]).

No entanto, o método IFT-SC carece de restrições de suavidade e regularização das bordas na sua formulação. Nos experimentos do artigo [82] testamos uma variante com função de custo de caminho aditiva que proporcionou bordas mais regulares. Outra idéia é tratar a regularização das bordas através de um pós-processamento específico que leva em conta conhecimentos da aplicação. Esse é um campo importante de pesquisa a ser explorado, embora o próprio modelo dos objetos proporciona em parte regularização das bordas.

**3. Sinergia entre Modelo dos Objetos e Delineamento.** No Capítulo 5, foi apresentado um conjunto de modelos baseados no conceito de *Clouds*, bem como uma extensão para múltiplos objetos chamada *Cloud System Model*, que melhorou os resultados em uma tarefa difícil, envolvendo a segmentação de estruturas cerebrais em imagens de MR-T1. Os métodos também podem ser facilmente implementados em paralelo, aproveitando-se das máquinas com múltiplos processadores. Os modelos baseados em *Clouds* parecem ser estratégias híbridas simples mas poderosas que atingem automação no reconhecimento e no delineamento, por tecê-las como tarefas fortemente acopladas.

Durante o desenvolvimento desse modelo, surgiram algumas dificuldades no caso de múltiplos objetos. Nossa primeira implementação considerava a execução simultânea do algoritmo de delineamento para todos objetos, tirando proveito desse recurso suportado pelo algoritmo da IFT-SC. Isto é, os voxels sementes de todos objetos competiam ao mesmo tempo durante a propagação dos caminhos ótimos. Isso gerou vários problemas. Primeiro, as regiões de incerteza tiveram que ser classificadas de acordo com os objetos envolvidos de modo a impedir que sementes de outros objetos as conquistassem. Por exemplo, se uma região do modelo era do objeto 1 para alguns dados do treinamento e dos objetos 2 ou 3 para os demais dados, então essa região era considerada como incerteza do tipo 1-2-3. Logo, sementes do objeto 4 eram impedidas de disputar essa região. Isso, por sua vez, gerou alguns problemas topológicos de regiões ilhadas que não recebiam o rótulo correto. Portanto, a solução final adotada foi executar o delineamento para cada objeto separadamente, levando em conta as suas disposições espaciais relativas, e depois calcular uma pontuação de reconhecimento conjunta. Essa solução é também mais flexível permitindo que grafos com pesos personalizados para cada objeto sejam empregados.

## 6.2 Outras Contribuições

Vários outros artigos foram publicados ao longo da tese. A técnica de poda de árvores [38], desenvolvida durante o mestrado, foi aperfeiçoada com um algoritmo de detecção de pontos de poda mais robusto e independente de parâmetros *ad-hoc*. Esse resultado foi apresentado no SIBGRAPI de 2006 [78], e uma versão estendida foi publicada no Journal of Mathematical Imaging and Vision em 2007 [9]. Essa técnica foi avaliada na segmentação automática do cérebro, mas não elimina tronco e medula, funcionando como técnica de *skull stripping* apenas. Logo, esses resultados são inferiores aos apresentados nessa tese. Outro artigo de segmentação publicado no período combinava medidas de corte em grafo diversas com a IFT [41]. Este trabalho foi um primeiro passo importante no entendimento prático das relações existentes entre diferentes métodos em grafo, o que motivou a escrita dos teoremas do Capítulo 3.

Também nesse período, a primeira versão de um classificador de padrões por floresta

de caminhos ótimos (OPF) foi publicada em conferência internacional [91]. Em vista dos bons resultados que foram obtidos por esse classificador OPF, em comparação com outros classificadores populares tais como redes neurais e SVM, dois artigos [107, 108] foram publicados em conferências internacionais avaliando o uso do OPF para a estimativa de pesos do grafo seguindo as idéias chaves descritas no Capítulo 2. Foram publicados também artigos sobre descritores e análise de formas [3, 4] e uma versão estendida em revista desses trabalhos foi aceita para publicação na Pattern Recognition [2].

### 6.3 Extensões Futuras

Há diversas oportunidades para aperfeiçoar o sistema proposto de segmentação, dentre as quais:

- Avaliar os modelos *Clouds* em um número maior de pacientes e controles, em imagens de melhor qualidade a partir de um scanner de 3T.
- Estender esses modelos para imagens multimodais (por exemplo, T2, PD).
- Ampliar o alcance desses modelos, para segmentar outras estruturas do cérebro, tais como o núcleo caudado e o hipocampo. As medições volumétricas do hipocampo segmentado, podem então ser utilizadas na detecção e quantificação de atrofia do hipocampo [20, 125, 53], que está relacionada com a epilepsia do lobo temporal [13].
- Estender os teoremas do Capítulo 3 para grafos direcionados.
- Incorporar a informação da orientação das bordas dos objetos nos modelos *Clouds*.
- Paralelizar os métodos de modo a tirar proveito das arquiteturas novas contendo múltiplos processadores.
- Demonstrar os métodos em outros domínios de aplicação de modo a popularizar as técnicas empregadas.

# Referências Bibliográficas

- [1] C. Allène, J.Y. Audibert, M. Couprie, J. Cousty, and R. Keriven. Some links between min-cuts, optimal spanning forests and watersheds. In *Proceedings of the 8th Int. Symposium on Mathematical Morphology (ISMM)*, pages 253–264, 2007.
- [2] F.A. Andaló, P.A.V. Miranda, R.S. Torres, and A.X. Falcão. Shape feature extraction and description based on tensor scale. *Pattern Recognition*, 43(1):26–36, Jan.
- [3] F.A. Andaló, P.A.V. Miranda, R.S. Torres, and A.X. Falcão. Detecting contour saliences using tensor scale. In *Proceedings of the 14th IEEE International Conference on Image Processing (ICIP)*, volume VI, pages 349–352, San Antonio, Texas, Sep 2007. IEEE SPS.
- [4] F.A. Andaló, P.A.V. Miranda, R.S. Torres, and A.X. Falcão. A new shape descriptor based on tensor scale. In *Proceedings of the Intl. Symposium on Mathematical Morphology and its Applications to Signal and Image Processing (ISMM)*, pages 141–152, Rio de Janeiro, Brazil, Oct 2007. MCT/INPE.
- [5] R. Audigier and R.A. Lotufo. The tie-zone watershed: Definition, algorithm and applications. In *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, pages 654–657, 2005.
- [6] R. Audigier and R.A. Lotufo. Seed-relative segmentation robustness of watershed and fuzzy connectedness approaches. In *Proceedings of the XX Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI)*, pages 61–68, Belo Horizonte, MG, Oct 2007. IEEE CPS.
- [7] R. Audigier and R.A. Lotufo. Watershed by image foresting transform, tie-zone, and theoretical relationship with other watershed definitions. In *Proceedings of the 8th Intl. Symposium on Mathematical Morphology and its Applications to Signal and Image Processing (ISMM)*, pages 277–288, Rio de Janeiro, Brazil, Oct 2007. MCT/INPE.

- [8] X. Bai and G. Sapiro. Distance cut: Interactive segmentation and matting of images and videos. In *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, volume 2, pages II – 249–II – 252, San Antonio, Texas, 2007.
- [9] F.P.G. Bergo, A.X. Falcão, P.A.V. Miranda, and L.M. Rocha. Automatic image segmentation by tree pruning. *Journal of Mathematical Imaging and Vision*, 29(2-3):141–162, Nov 2007. doi: 10.1007/s10851-007-0035-4.
- [10] F.P.G. Bergo, G.C.S. Ruppert, L.F. Pinto, and A.X. Falcão. Fast and robust mid-sagittal plane location in 3D MR images of the brain. In *Intl. Conf. on Bio-inspired Systems and Signal Proc (BIOSIGNALS)*, pages 92–99, Funchal, Portugal, Jan 2008.
- [11] S. Beucher and F. Meyer. The morphological approach to segmentation: The watershed transformation. In *Mathematical Morphology in Image Processing*, chapter 12, pages 433–481. Marcel Dekker, 1993.
- [12] 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.
- [13] L. Bonilha, C. Rorden, S. Appenzeller, A. C. Coan, F. Cendes, and L. M. Li. Gray matter atrophy associated with duration of temporal lobe epilepsy. *Neuroimage*, 32(3):1070–1079, Sep 2006.
- [14] Y. Boykov and G. Funka-Lea. Graph cuts and efficient N-D image segmentation. *International Journal of Computer Vision*, 70(2):109–131, 2006.
- [15] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(9):1124–1137, Sep 2004.
- [16] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11):1222–1239, Nov 2001.
- [17] Y.Y. Boykov and M.-P. Jolly. Interactive graph cuts for optimal boundary & region segmentation of objects in N-D images. In *International Conference on Computer Vision (ICCV)*, volume 1, pages 105–112, 2001.

- [18] 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.
- [19] J. Carballido-Gamio, S.J. Belongie, and S. Majumdar. Normalized cuts in 3D for spinal MRI segmentation. *IEEE Transactions on Medical Imaging*, 23(1):36–44, Jan 2004.
- [20] G. D. Cascino. Clinical correlations with hippocampal atrophy. *Magnetic Resonance Imaging*, 13(8):1133–1136, 1995.
- [21] G. Castellano, L. Bonilha, L.M. Li, and F. Cendes. Texture analysis in medical images. *Clinical Radiology, in print*, 2004.
- [22] X. Chen, J.K. Udupa, X. Zheng, D. Torigian, and A. Alavi. Automatic anatomy recognition via multi object oriented active shape models. In *SPIE on Medical Imaging*, volume 7259, Orlando, Florida, USA, Feb 2009.
- [23] K.C. Ciesielski and J.K. Udupa. Affinity functions: Recognizing essential parameters in fuzzy connectedness based image segmentation. In *Proceedings of SPIE on Medical Imaging: Image Processing*, volume 7259, 2009.
- [24] K.C. Ciesielski, J.K. Udupa, P.K. Saha, and Y. Zhuge. Iterative relative fuzzy connectedness for multiple objects with multiple seeds. *Computer Vision and Image Understanding*, 107(3):160–182, 2007. doi: 10.1016/j.cviu.2006.10.005.
- [25] 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 Transactions on Medical Imaging*, 17(3):463–468, June 1998.
- [26] T. Cootes, G. Edwards, and C.J. Taylor. Active appearance models. In *European Conference on Computer Vision (ECCV)*, volume 2, pages 484–498, 1998.
- [27] 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.
- [28] T. Cormen, C. Leiserson, and R. Rivest. *Introduction to Algorithms*. MIT, 1990.
- [29] J. Cousty, G. Bertrand, L. Najman, and M. Couprie. Watersheds, Minimum Spanning Forests, and the Drop of Water Principle. Technical Report IGM2007-01, Université de Marne-la-Vallée, 2007.

- [30] I.J. Cox, S.B. Rao, and Y. Zhong. Ratio regions: A technique for image segmentation. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 557–564, 1996.
- [31] A.M. Dale, B. Fischl, and M.I. Sereno. Cortical surface- based analysis: I. segmentation and surface reconstruction. *NeuroImage*, 9:179–194, 1999.
- [32] L.R. Dice. Measures of the amount of ecologic association between species. *Ecology*, 26:297–302, 1945.
- [33] E.W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1:269–271, 1959.
- [34] R.O. Duda, P.E. Hart, and D.G. Stork. *Pattern Classification*. Wiley-Interscience, 2 edition, 2001.
- [35] N. Duta and M. Sonka. Segmentation and interpretation of MR brain images using an improved knowledge-based active shape model. In *Proceedings of Information Processing in Medical Imaging (IPMI)*, pages 375–380, 1997. doi: 10.1007/3-540-63046-5.
- [36] A.X. Falcão and F.P.G. Bergo. Interactive volume segmentation with differential image foresting transforms. *IEEE Transactions on Medical Imaging*, 23(9):1100–1108, 2004.
- [37] A.X. Falcão, F.P.G. Bergo, F.O. Favretto, G.C.S. Ruppert, P.A.V. Miranda, and F.A.M. Cappabianco. *Processamento, Visualização e Análise de Imagens Anatômicas do Cérebro Humano, in: Neurociências e Epilepsia*. Ed. Plêiade, São Paulo, 2008. Eds: L.M. Li, P.T. Fernandes, R.J.M. Covolan, F. Cendes.
- [38] A.X. Falcão, F.P.G. Bergo, and P.A.V. Miranda. Image segmentation by tree pruning. In *Proceedings of the XVII Brazillian Symposium on Computer Graphics and Image Processing (SIBGRAPI)*, pages 65–71. IEEE, Oct 2004.
- [39] 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.
- [40] 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, Feb 2001.

- [41] A.X. Falcão, P.A.V. Miranda, and A. Rocha. A linear-time approach for image segmentation using graph-cut measures. In *8th Intl. Conf. on Advanced Concepts for Intelligent Vision Systems (ACIVS)*, volume LNCS 4179, pages 138–149, Antwerp, Belgium, 2006. Springer.
- [42] A.X. Falcão, P.A.V. Miranda, A. Rocha, and F.P.G. Bergo. Object detection by  $\kappa$ -connected seed competition. In *Proceedings of the XVIII Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI)*, pages 97–104, Natal, RN, Oct 2005. IEEE CPS.
- [43] A.X. Falcão, J. Stolfi, and R.A. Lotufo. The image foresting transform: Theory, algorithms, and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(1):19–29, 2004.
- [44] 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.
- [45] 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.
- [46] M. Farber, J. Ehrhardt, and H. Handels. Live-wire-based segmentation using similarities between corresponding image structures. *Computerized Medical Imaging and Graphics*, 31(7):549–560, Oct 2007.
- [47] P. Felkel, M. Bruckschwaiger, and R. Wegenkittl. Implementation and complexity of the watershed-from-markers algorithm computed as a minimal cost forest. *Computer Graphics Forum (EUROGRAPHICS)*, 20(3):(C) 26–35, 2001.
- [48] L. Ford and D. Fulkerson. *Flows in Networks*. Princeton University Press, 1962.
- [49] C. Fowlkes, S. Belongie, and J. Malik. Efficient spatiotemporal grouping using the nystrom method. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 231–238, 2001.
- [50] 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.
- [51] K.A. Ganser, H. Dickhaus, R. Metzner, and C.R. Wirtz. A deformable digital brain atlas system according to Talairach and Tournoux. *Medical Image Analysis*, 8(1):3–22, Mar 2004.

- [52] 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 Transactions on Medical Imaging*, 23(4):447–458, Apr 2004.
- [53] A. Hammers, R. Heckemann, M.J. Koeppe, J.S. Duncan, J.V. Hajnal, D. Rueckert, and P. Aljabar. Automatic detection and quantification of hippocampal atrophy on mri in temporal lobe epilepsy: a proof-of-principle study. *Neuroimage*, 36(1):38–47, May 2007.
- [54] H.G. He, J. Tian, Y. Lin, and K. Lu. A new interactive segmentation scheme based on fuzzy affinity and live-wire. In *Fuzzy Systems and Knowledge Discovery*, volume 3613, pages 436–443, 2005.
- [55] G.T. Herman and B.M. Carvalho. Multiseeded segmentation using fuzzy connectedness. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23:460–474, May 2001.
- [56] J.Liu and J.K. Udupa. Oriented active shape models. In *SPIE on Medical Imaging*, volume 6144, pages 30–40, 2006.
- [57] H.W. Kang. G-wire: A livewire segmentation algorithm based on a generalized graph formulation. *Pattern Recognition Letters*, 26(13):2042–2051, Oct 2005.
- [58] H.W. Kang and S.Y. Shin. Enhanced lane: Interactive image segmentation by incremental path map construction. *Graphical Models*, 64(5):282–303, Sep 2002.
- [59] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1:321–331, 1987.
- [60] V. Kolmogorov and R. Zabih. What energy functions can be minimized via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(2):147–159, Feb 2004.
- [61] J. Larsson. *Imaging Vision: Functional Mapping of Intermediate Visual Processes in Man*. PhD thesis, Karolinska Institutet, 2001. ISBN 91-7349-090-3.
- [62] T. Lei, J.K. Udupa, P.K. Saha, and D. Odhner. Artery-vein separation via MRA - An image processing approach. *IEEE Transactions on Medical Imaging*, 20(8), 2001.
- [63] K. Li, X. Wu, D.Z. Chen, and M. Sonka. Optimal surface segmentation in volumetric images: A graph-theoretic approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(1):119–134, 2006.

- [64] L. Liang, K. Rehm, R.P. Woods, and D.A. Rottenberg. Automatic segmentation of left and right cerebral hemispheres from MRI brain volumes using the graph cuts algorithm. *NeuroImage*, 34(3):1160–1170, 2007.
- [65] F. Lijn, M. Bruijne, Y.Y. Hoogendam, S. Klein, R. Hameeteman, M. Breteler, and W.J. Niessen. Cerebellum segmentation in MRI using atlas registration and local multi-scale image descriptors. In *Proceedings of the IEEE International Symposium on Biomedical Imaging: From Nano to Macro (ISBI)*, Boston, MA, 2009. to appear.
- [66] T. Lindeberg. Scale-space theory: A basic tool for analysing structures at different scales. *Journal of Applied Statistics*, 21:224–270, 1994.
- [67] J. Liu and J.K. Udupa. Oriented active shape models. *IEEE Transactions on Medical Imaging*, 28(4):571–584, 2009.
- [68] R.A. Lotufo and A.X. Falcão. The ordered queue and the optimality of the watershed approaches. In *Proceedings of the Intl. Symposium on Mathematical Morphology and its Applications to Signal and Image Processing (ISMM)*, volume 18, pages 341–350. Kluwer, Jun 2000.
- [69] D. Lowe. Distinctive image features from scale-invariant keypoints. In *Proc. of the International Journal of Computer Vision*, volume 20, pages 91–110, 2003.
- [70] A. Madabhushi and J.K. Udupa. Interplay between intensity standardization and inhomogeneity correction in MR image processing. *IEEE Transactions on Medical Imaging*, 24(5):561–576, 2005.
- [71] F. Malmberg, E. Vidholm, and I. Nystrom. A 3D live-wire segmentation method for volume images using haptic interaction. In *Discrete Geometry for Computer Imagery*, volume 4245, pages 663–673, 2006.
- [72] J. Mangin, J. Regis, and V. Frouin. Shape bottlenecks and conservative flow systems. In *Math. Method in Biomed. Image Analysis (MMBIA)*, pages 319–328, San Francisco, CA, 1996. IEEE Computer Society Press.
- [73] B.S. Manjunath and W.Y. Ma. Texture features for browsing and retrieval of image data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(8):837–842, 1996.
- [74] D.R. Martin, C.C. Fowlkes, and J. Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(5):530–549, May 2004.

- [75] A. Meijster and M.H.F. Wilkinson. A comparison of algorithms for connected set openings and closings. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(4):484–494, Apr 2002.
- [76] F. Meyer. Levelings, image simplification filters for segmentation. *Journal of Mathematical Imaging and Vision*, 20(1-2):59–72, 2004.
- [77] R. Minetto, J.P. Papa, T.V. Spina, A.X. Falcão, N.J. Leite, and J. Stolfi. Fast and robust object tracking using image foresting transform. In *Proc. of the 16th Intl. Conf. on Systems, Signals, and Image Processing*, Chalkida, Greece, 2009. IEEE. to appear.
- [78] P. A. V. Miranda, F. P. G. Bergo, L. M. Rocha, and A. X. Falcão. Tree-pruning: A new algorithm and its comparative analysis with the watershed transform for automatic image segmentation. In *Proceedings of the XIX Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI)*, pages 37–44. IEEE, Oct 2006.
- [79] P.A.V. Miranda and A.X. Falcão. Links between image segmentation based on optimum-path forest and minimum cut in graph. *Journal of Mathematical Imaging and Vision*, 2009. doi: 10.1007/s10851-009-0159-9.
- [80] P.A.V. Miranda, A.X. Falcão, A. Rocha, and F.P.G. Bergo. Object delineation by  $\kappa$ -connected components. *EURASIP Journal on Advances in Signal Processing*, pages 1–14, 2008. doi: 10.1155/2008/467928.
- [81] P.A.V. Miranda, A.X. Falcão, and J.K. Udupa. Clouds: A model for synergistic image segmentation. In *Proceedings of the IEEE International Symposium on Biomedical Imaging (ISBI)*, pages 209–212, Paris, France, May 2008.
- [82] P.A.V. Miranda, A.X. Falcão, and J.K. Udupa. Cloud bank: A multiple clouds model and its use in MR brain image segmentation. In *Proceedings of the IEEE International Symposium on Biomedical Imaging: From Nano to Macro (ISBI)*, pages 506–509, Boston, MA, 2009. doi: 10.1109/ISBI.2009.5193095.
- [83] P.A.V. Miranda, A.X. Falcão, and J.K. Udupa. Synergistic arc-weight estimation for interactive image segmentation using graphs. *Computer Vision and Image Understanding*, 2009. doi: 10.1016/j.cviu.2009.08.001.
- [84] N. Mittal, D.P. Mital, and Kap Luk Chan. Features for texture segmentation using gabor filters. In *Seventh International Conference on Image Processing And Its Applications (Conf. Publ. No. 465)*, volume 1, pages 353–357, 1999.

- [85] J.A. Montoya-Zegarra, J.P. Papa, N.J. Leite, R.S. Torres, and A.X. Falcão. Learning how to extract rotation-invariant and scale-invariant features from texture images. *EURASIP Journal on Advances in Signal Processing*, pages 1–15, 2008. doi: 10.1155/2008/691924.
- [86] 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 Neuroradiology*, 23:356–363, Mar 2002.
- [87] J. Natsume and et al. MRI volumetry of the thalamus in temporal, extratemporal, and idiopathic generalized epilepsy. *Neurology*, 60:1296–1300, 2003.
- [88] 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.
- [89] W.L. Nowinski. Anatomical targeting in functional neurosurgery by the simultaneous use of multiple schaltenbrand-warhen brain atlas microseries. *Medical Image Analysis*, (71):103–116, 1998.
- [90] S.D. Olabarriaga and A.W.M. Smeulders. Interaction in the segmentation of medical images: A survey. *Medical Image Analysis*, 5(2):127–142, Jun 2001.
- [91] J.P. Papa, A.X. Falcão, P.A.V. Miranda, C.T.N. Suzuki, and N.D.A. Mascarenhas. Design of robust pattern classifiers based on optimum-path forests. In *Proceedings of the Intl. Symposium on Mathematical Morphology and its Applications to Signal and Image Processing (ISMM)*, pages 337–348. MCT/INPE, 2007.
- [92] J.P. Papa, A.X. Falcão, C.T.N. Suzuki, and N.D.A. Mascarenhas. A discrete approach for supervised pattern recognition. In *Proceedings of the 12th International Workshop on Combinatorial Image Analysis*, volume LNCS 4958, pages 136–147, Buffalo, NY, USA, Apr 7th-9th 2008. Springer.
- [93] A. Parraga, A. Susin, J. Pettersson, B. Macq, and M. De Craene. 3D atlas building in the context of head and neck radiotherapy based on dense deformation fields. In *Proceedings of the XX Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI)*, pages 321–328. IEEE, Oct 2007. doi: 10.1109/SIBGRAPI.2007.23.
- [94] A. Protiere and G. Sapiro. Interactive image segmentation via adaptive weighted distances. *IEEE Transactions on Image Processing*, 16(4):1046–1057, Apr 2007.

- [95] L.M. Rocha, A.X. Falcão, and L.G.P. Meloni. A robust extension of the mean shift algorithm using optimum path forest. In *Proceedings of the 12th International Workshop on Combinatorial Image Analysis*, pages 29–38, Buffalo, NY, USA, Apr 7th-9th 2008. RPS.
- [96] J.B.T.M. Roerdink and A. Meijster. The watershed transform: Definitions, algorithms and parallelization strategies. *Fundamenta Informaticae*, 41:187–228, 2000.
- [97] A.M. Sá, M.B. Vieira, A.A. Montenegro, P.C.P. Carvalho, and L. Velho. Actively illuminated objects using graph-cuts. In *Proceedings of the XIX Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI)*, pages 45–52. IEEE, Oct 2006.
- [98] P.K. Saha and J.K. Udupa. Fuzzy connected object delineation: Axiomatic path strength definition and the case of multiple seeds. *Computer Vision and Image Understanding*, 83:275–295, 2001.
- [99] 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(1):42–56, 2001.
- [100] P.K. Saha and J.K. Udupa. Tensor scale-based fuzzy connectedness image segmentation. In *Proceedings of SPIE on Medical Imaging*, volume 5032, pages 1580–1590, Feb 2003.
- [101] P.K. Saha, J.K. Udupa, and D. Odhner. Scale-based fuzzy connected image segmentation: Theory, algorithms, and validation. *Computer Vision and Image Understanding*, 77(2):145–174, 2000.
- [102] P. Salembier, A. Oliveras, and L. Guarrido. Antiextensive connected operators for image and sequence processing. *IEEE Transactions on Image Processing*, 7(4):555–570, Apr 1998.
- [103] P. Salembier and J. Serra. Flat zones filtering, connected operators, and filters by reconstruction. *IEEE Transactions on Image Processing*, 4(8):1153–1160, Aug 1995.
- [104] E.K. Sandock and et al. Significance of cerebellar atrophy in intractable temporal lobe epilepsy: A quantitative MRI study. *Epilepsia*, 41:1315–1320, 2000.
- [105] S. Sarkar and K.L. Boyer. Quantitative measures of change based on feature organization: Eigenvalues and eigenvectors. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 478–483, 1996.

- [106] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, Aug 2000.
- [107] T.V. Spina, J.A. Montoya-Zegarra, A.X. Falcão, and P.A.V. Miranda. Fast interactive segmentation of natural images using the image foresting transform. In *16th International Conference on Digital Signal Processing (DSP)*, Santorini, Greece, 2009. to appear.
- [108] T.V. Spina, J.A. Montoya-Zegarra, Paulo A.V. Miranda, and A.X. Falcão. Improving user control with minimum involvement in user-guided segmentation by image foresting transform. In *Intl. Conf. on Computer Analysis of Images and Patterns*, Münster, 2009.
- [109] V. Stix. Finding all maximal cliques in dynamic graphs. *Comput. Optim. Appl.*, 27(2):173–186, 2004.
- [110] R.S. Torres and A.X. Falcão. Contour salience descriptors for effective image retrieval and analysis. *Image and Vision Computing*, 25(1):3–13, Jan 2007.
- [111] 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.
- [112] Y.P. Tsai, C.C. Lai, Y.P. Hung, and Z.C. Shih. A bayesian approach to video object segmentation via merging 3-D watershed volumes. *IEEE Transactions on Circuits and Systems for Video Technology*, 15(1):175–180, Jan 2005.
- [113] J.K. Udupa, V.R. LeBlanc, Y. Zhuge, C. Imielinska, H. Schmidt, L.M. Currie, B.E. Hirsch, and J. Woodburn. A framework for evaluating image segmentation algorithms. *Computerized Medical Imaging and Graphics*, 30(2):75–87, Mar 2006.
- [114] J.K. Udupa and P.K. Saha. Fuzzy connectedness and image segmentation. *Proceedings of the IEEE*, 91, Oct 2003.
- [115] J.K. Udupa, P.K. Saha, and R.A. Lotufo. Relative fuzzy connectedness and object definition: Theory, algorithms, and applications in image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24:1485–1500, 2002.
- [116] 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.

- [117] S. Vicente, V. Kolmogorov, and C. Rother. Graph cut based image segmentation with connectivity priors. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, Anchorage, Alaska, Jun 2008.
- [118] L. Vincent. Morphological area opening and closings for greyscale images. In *Shape in Picture'92 - NATO Workshop*. Springer, Sep 1992.
- [119] L. Vincent. Morphological grayscale reconstruction in image analysis. *IEEE Transactions on Image Processing*, 2(2):176–201, Apr 1993.
- [120] L. Vincent and P. Soille. Watersheds in digital spaces: An efficient algorithm based on immersion simulations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(6), Jun 1991.
- [121] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages I-511–I-518, 2001.
- [122] J. Wang and M.F. Cohen. An iterative optimization approach for unified image segmentation and matting. In *Proceedings of the Tenth International Conference on Computer Vision (ICCV)*, pages 936–943, Washington, DC, USA, 2005. IEEE Computer Society.
- [123] S. Wang and J.M. Siskind. Image segmentation with ratio cut. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(6):675–690, Jun 2003.
- [124] S. Wang and J.M. Siskind. Image segmentation with minimum mean cut. In *International Conference on Computer Vision (ICCV)*, volume 1, pages 517–525, 2001.
- [125] J. Webb, A. Guimond, P. Eldridge, D. Chadwick, J. Meunier, J. P. Thirion, and N. Roberts. Automatic detection of hippocampal atrophy on magnetic resonance images. *Magnetic Resonance Imaging*, 17(8):1149–1161, Oct 1999.
- [126] Z. Wu and R. Leahy. An optimal graph theoretic approach to data clustering: Theory and its applications to image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(11):1101–1113, Nov 1993.
- [127] G. Wyszecki and W. Stiles. *Color Science: Concepts and Methods, Quantitative Data and Formulas*. J. Wiley and Sons, 1982.
- [128] D. Zheng, Y. Zhao, and J. Wang. An efficient method of license plate location. *Pattern Recognition Letters*, 26(15):2431–2438, Nov 2005.