

Franklin César Flores

***Watershed* com Marcadores Propagados
para Segmentação Interativa
de Objetos em Sequências de Imagens**

Tese de Doutorado apresentada à Faculdade de Engenharia Elétrica e de Computação como parte dos requisitos para obtenção do título de Doutor em Engenharia Elétrica. Área de concentração: Engenharia de Computação.

Aprovação em 04/12/2009

Banca Examinadora:

Prof. Dr. Roberto de Alencar Lotufo - FEEC-UNICAMP
(Orientador)

Prof. Dr. Clésio Tozzi - FEEC-UNICAMP

Prof. Dr. José Eduardo Cogo Castanho - FEB-UNESP

Prof. Dr. José Mario De Martino - FEEC-UNICAMP

Prof. Dr. Roberto Hirata Jr. - IME-USP

Campinas, SP
2009

FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DA ÁREA DE ENGENHARIA - BAE - UNICAMP

F663w	Flores, Franklin César Watershed com Marcadores Propagados para Segmentação Interativa de Objetos em Sequências de Imagens. Franklin César Flores. – Campinas, SP: [s.n.], 2009. Orientador: Roberto de Alencar Lotufo. Tese de Doutorado - Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação. 1. Morfologia Matemática. 2. Processamento de imagens. 3. Visão por computador. I. Lotufo, Roberto de Alencar. II. Universidade Estadual de Campinas. Faculdade de Engenharia Elétrica e de Computação. III. Título
-------	---

Título em Inglês:	Watershed from Propagated Markers to Interactive Segmentation of Objects in Image Sequences
Palavras-chave em Inglês:	Mathematical Morphology, Image processing, Computer vision
Área de concentração:	Engenharia de Computação
Titulação:	Doutor em Engenharia Elétrica
Banca Examinadora:	Clésio Tozzi, José Eduardo Cogo Castanho, José Mario De Martino, Roberto Hirata Jr.
Data da defesa:	04/12/2009
Programa de Pós-Graduação:	Engenharia Elétrica

COMISSÃO JULGADORA - TESE DE DOUTORADO

Candidato: Franklin César Flores

Data da Defesa: 4 de dezembro de 2009

Título da Tese: "Watershed com Marcadores Propagados para Segmentação Interativa de Objetos em Sequências de Imagens"

Prof. Dr. Roberto de Alencar Lotufo (Presidente): _____

Prof. Dr. José Eduardo Cogo Castanho: _____

Prof. Dr. Roberto Hirata Júnior: _____

Prof. Dr. José Mario De Martino: _____

Prof. Dr. Clésio Luis Tozzi: _____

Resumo

Esta tese de doutorado apresenta um método interativo para segmentação de objetos em sequências de imagens - o *watershed* com marcadores propagados. Este método, uma combinação de segmentação morfológica clássica com estimação de movimento, possui quatro características importantes: i) interatividade, ii) generalidade, iii) resposta rápida e iv) edição manual progressiva. *Watershed* com marcadores propagados consiste em segmentar interativamente os objetos de interesse no primeiro quadro e, subsequentemente, computar e propagar marcadores para segmentar os mesmos objetos nos quadros seguintes. Além da proposta do paradigma do *watershed* com marcadores propagados, esta tese também apresenta variações para o paradigma citado e um novo *benchmark* para avaliação quantitativa de métodos interativos para segmentação de objetos em sequências de imagens.

Palavras-chave: *Watershed* com Marcadores Propagados, Segmentação de Objetos em Sequências de Imagens, Segmentação Interativa, Morfologia Matemática, Segmentação Morfológica, *Watershed* com Marcadores.

Abstract

This doctorate thesis introduces an assisted method to object segmentation in image sequences - the watershed from propagated markers. This method, a combination of classical morphological segmentation with motion estimation, has four important characteristics: i) interactivity, ii) generality, iii) rapid response and iv) progressive manual edition. Watershed from propagated markers consists in to segment interactively the objects of interest in the first frame and, subsequently, to compute and propagate markers in order to segment the same objects in the next frames. Besides the proposal of the watershed from propagated markers paradigm, this thesis also presents variations to the cited paradigm and a new benchmark to quantitative evaluation of interactive object segmentation methods applied to image sequences.

Keywords: Watershed from Propagated Markers, Object Segmentation in Image Sequences, Interactive Segmentation, Mathematical Morphology, Morphological Segmentation, Watershed from Markers.

*À minha mãe (in memoriam),
ao meu pai
e ao meu irmão.*

Agradecimentos

A Deus.

À minha mãe Irene (*in memoriam*) e ao meu pai Geraldo.

Ao meu irmão Wallace.

Ao meu orientador, prof. Lotufo.

À FEEC-UNICAMP. Em especial, ao DCA e à CPG. Agradecimentos também ao prof. Michel Yacoub.

À Banca Examinadora.

À Universidade Estadual de Maringá. Em especial, ao Departamento de Informática.

Aos colegas do LCA que me apoiaram.

Aos colegas orientados e orientandos do prof. Lotufo. Em especial, à Letícia Rittner pelo trabalho conjunto.

Aos amigos de Mandaguari-PR, Maringá-PR e Campinas-SP.

Aos membros da República AoBerg.

Ao amigo Airton Marco Polidório.

Sumário

Lista de Figuras	xv
Lista de Tabelas	xix
Trabalhos Publicados Pelo Autor	xxi
1 Introdução	1
1.1 Objetivos	2
1.2 Revisão Bibliográfica	4
1.3 Metodologia	5
1.3.1 <i>Watershed</i> com marcadores propagados	6
1.3.2 Variações do <i>watershed</i> com marcadores propagados	6
1.3.3 Aprimoramento pelo uso de gradiente espaço-temporal	8
1.3.4 <i>Benchmark</i> para avaliação quantitativa de métodos assistidos de segmentação de objetos em sequências de imagens	10
1.4 Contribuições da tese	10
1.5 Organização da Tese	11
2 Object Segmentation in Image Sequences by Watershed from Markers: A Generic Approach	13
2.1 Abstract	14
2.2 Introduction	14
2.3 Related Works	16
2.4 Watershed from Markers	16
2.5 Motion Estimation by Marker Propagation	17
2.6 The Proposed Method	17
2.7 Prontovideo	20
2.8 Conclusion	22
3 Watershed from Propagated Markers: An Interactive Method to Morphological Object Segmentation in Image Sequences	23
3.1 Abstract	24
3.2 Introduction	24
3.3 Related Works	26

3.4	Watershed from Markers	26
3.5	The Proposed Method	27
3.6	Motion Estimation Applied to Marker Propagation	27
3.6.1	Region-based method	28
3.6.2	Second-order differential method	29
3.6.3	Horn-Schunck method	29
3.6.4	Lucas-Kanade method	29
3.7	Marker Propagation Improvements	30
3.7.1	Binding of markers	30
3.7.2	Adjustment by local averaging of the displacement vectors	33
3.8	Benchmark for Quantitative Evaluation of Assisted Object Segmentation Methods in Image Sequences	34
3.8.1	Motion information	35
3.8.2	Number of user interferences in each frame	36
3.8.3	Time spent in the edition of each frame	36
3.8.4	Segmentation error in each frame (Compared to the Ground Truth)	36
3.8.5	Example: <i>Akiyo</i> sequence	37
3.9	Experimental Results	38
3.9.1	Comparison among optical flow estimators	38
3.9.2	Comparison among the proposed heuristics	40
3.9.3	User intervention	45
3.9.4	Parameters	49
3.10	Conclusion	62
3.11	Acknowledgements	63
4	Watershed from Propagated Markers Improved by the Combination of Spatio-Temporal Gradient and Binding of Markers Heuristics	65
4.1	Abstract	65
4.2	Introduction	66
4.3	Preliminary Review	67
4.3.1	3-D structuring elements	67
4.3.2	Band sequence extraction	68
4.3.3	Spatio-temporal gradient	68
4.3.4	Sequence sampling	68
4.4	Watershed From Propagated Markers	69
4.4.1	Binding of markers	69
4.4.2	Spatio-temporal gradient improvement	70
4.5	Experimental Results	71
4.5.1	Foreman sequence	72
4.5.2	Carphone sequence	73
4.6	Conclusion	75

5	A Decomposição Uniforme: Uma Nova Decomposição Hierárquica Aplicada à Segmentação Morfológica de Objetos em Sequências de Imagens	77
5.1	Resumo	78
5.2	Introdução	78
5.3	Revisão Preliminar	80
5.3.1	Funções de extinção	80
5.3.2	Segmentação hierárquica	82
5.4	A Decomposição Uniforme	86
5.4.1	Conceitos preliminares	86
5.4.2	O algoritmo	88
5.4.3	A propriedade hierárquica da decomposição uniforme	91
5.4.4	Outras propriedades da decomposição uniforme	92
5.4.5	Resultados experimentais	92
5.5	A Decomposição Uniforme e o <i>Watershed</i> com Marcadores Propagados	98
5.5.1	Resultados experimentais	100
5.6	Conclusão	104
6	Conclusão	113
6.1	Principais resultados	114
6.1.1	Variações do paradigma de segmentação proposto	114
6.1.2	Uso do gradiente espaço-temporal	115
6.1.3	Um <i>benchmark</i> para avaliação quantitativa de métodos de segmentação assistida de objetos em sequências de imagens	115
6.2	Considerações finais	115
6.3	Trabalhos futuros	116
A	Sobre a Amarração de Marcadores	117
A.1	Variação	117
A.2	Criação de “Pares Completos” de Marcadores	117
	Referências bibliográficas	124

Lista de Figuras

2.1	The Proposed Method: <i>Left Column</i> - Image sequence (frames 0 to 3). <i>Right Column</i> - Object segmented for each frame.	19
2.2	Prontovideo (a) Main Window (b) Mask Window	21
3.1	Binding of markers. (a) The internal (red) and external (green) markers delimit a region crossed by the object border. (b) Markers are propagated by the same displacement vector, computing in function of the delimited region.	31
3.2	Creation of the marker pairs and the regions delimited by them. (a) Mask of the segmented object. (b) The “crown” of the mask that must be sliced in order to obtain the delimited regions. (c) The seeds to be used to slice the “crown” of the mask. (d) Watershed lines. (e) Delimited regions (labeled). (f) Pair of markers (labeled) wrapping the borders of the mask.	32
3.3	Wrapping of markers (a) The mask does not touch the borders of the frame. (b) The mask touches the frame borders.	33
3.4	Marker neighbourhood $N_i(\mathcal{M})$	34
3.5	<i>Akiyo</i> sequence (frames 1 to 150).	39
3.6	Motion information from <i>Akiyo</i> sequence.	40
3.7	Manual segmentation (Ground Truth) of <i>Akiyo</i> sequence (frames 1 to 150).	41
3.8	Analysis of the Ground Truth segmentation (manual one) of the <i>Akiyo</i> sequence. (a) Number of user interferences in each frame. (b) Time spent in the edition of each frame (in seconds).	42
3.9	Segmentation of <i>Akiyo</i> sequence by Watershed from Propagated Markers (frames 1 to 150).	43
3.10	Analysis of the segmentation of <i>Akiyo</i> sequence by Watershed from Propagated Markers. (a) Number of user interferences in each frame. (b) Time spent in the edition of each frame (in seconds).	44
3.11	Analysis of the segmentation of <i>Akiyo</i> sequence by Watershed from Propagated Markers: Segmentation Error in each frame (Compared to the Ground Truth.)	45
3.12	Marker propagation comparison - Applied to <i>Foreman</i> sequence. (a) By region-based estimation. (b) Segmentation result. (c) By second-order differential estimation. (d) Segmentation result. (e) By Horn-Schunck estimation. (f) Segmentation result. (g) By Lucas-Kanade estimation. (h) Segmentation result.	46

3.13	Heuristics comparison - Applied to <i>Foreman</i> sequence. (a) No heuristics. (b) Segmentation result. (c) Local averaging of the displacement vectors. (d) Segmentation result. (e) Binding of markers. (f) Segmentation result.	47
3.14	<i>Foreman</i> sequence (frames 1 to 150).	50
3.15	Motion information from <i>Foreman</i> sequence.	51
3.16	Manual segmentation (Ground Truth) of <i>Foreman</i> sequence (frames 1 to 150).	52
3.17	Analysis of the Ground Truth segmentation (manual one) of the <i>Foreman</i> sequence. (a) Number of user interferences in each frame. (b) Time spent in the edition of each frame (in seconds).	53
3.18	Segmentation of <i>Foreman</i> sequence by Watershed from Propagated Markers (No propagation . Frames 1 to 150).	54
3.19	Analysis of the segmentation of <i>Foreman</i> sequence by Watershed from Propagated Markers (No propagation). (a) Number of user interferences in each frame. (b) Time spent in the edition of each frame (in seconds).	55
3.20	Analysis of the segmentation of <i>Foreman</i> sequence by Watershed from Propagated Markers (No propagation): Segmentation Error in each frame (Compared to the Ground Truth.)	56
3.21	Segmentation of <i>Foreman</i> sequence by Watershed from Propagated Markers (Lucas-Kanade . Frames 1 to 150).	57
3.22	Analysis of the segmentation of <i>Foreman</i> sequence by Watershed from Propagated Markers (Lucas-Kanade). (a) Number of user interferences in each frame. (b) Time spent in the edition of each frame (in seconds).	58
3.23	Analysis of the segmentation of <i>Foreman</i> sequence by Watershed from Propagated Markers (Lucas-Kanade): Segmentation Error in each frame (Compared to the Ground Truth.)	59
3.24	Parameters: assuming a constant length marker ($\mathbf{m} = 5$). (a) $\mathbf{w} = 5$. (b) Segmentation result. (c) $\mathbf{w} = 10$. (d) Segmentation result. (e) $\mathbf{w} = 15$. (f) Segmentation result. (g) $\mathbf{w} = 20$. (h) Segmentation result.	60
3.25	Parameters: assuming a constant marker distance ($\mathbf{w} = 5$). (a) $\mathbf{m} = 5$. (b) Segmentation result. (c) $\mathbf{m} = 10$. (d) Segmentation result. (e) $\mathbf{m} = 15$. (f) Segmentation result. (g) $\mathbf{m} = 20$. (h) Segmentation result.	61
4.1	Quantitative evaluation for <i>Foreman</i> sequence. (a) Motion information from <i>Foreman</i> sequence. (b) Number of interferences for each frame.	72
4.2	Quantitative evaluation for <i>Foreman</i> sequence. (a) Time spent in each frame (secs.). (b) Segmentation error (Percentile. Compared to the ground-truth).	73
4.3	Segmentation of <i>Foreman</i> sequence (frames 1 to 150). Using ∇_{B_5} and $\mathcal{W}_{B_4}$	74
4.4	Segmentation of <i>Carphone</i> sequence (frames 1 to 150). Using $\nabla_{B_{17}}$ and $\mathcal{W}_{B_8}$	76
5.1	Exemplo da limiarização por valores de extinção por área. (a) Função inicial e legenda com os valores de extinção. (b) Aplicação de limiar 40. (c) Aplicação de limiar 59.	83
5.2	Exemplo da limiarização por valores de extinção por área. (a) Aplicação de limiar 79. (b) Aplicação de limiar 93. (c) Aplicação de limiar 127.	84

5.3	Exemplo da limiarização por valores de extinção por área. (a) Aplicação de limiar 131.	85
5.4	Exemplo de função com marcadores rotulados.	85
5.5	Segmentação hierárquica. <i>Primeira linha : 13 regiões.</i> (a) Critério de área. (b) Critério de disco. (c) Clássica (limiarização dos valores de extinção). <i>Segunda linha : 34 regiões.</i> (d) Critério de área. (e) Critério de disco. (f) Clássica (limiarização dos valores de extinção). <i>Terceira linha : 101 regiões.</i> (g) Critério de área. (h) Critério de disco. (i) Clássica (limiarização dos valores de extinção).	95
5.6	Pré-processamento ao registro de imagens. (a) Deslocamento da imagem. (b) Marcadores aplicados para segmentar a Imagem Dois. (c) Decomposição uniforme da Imagem Um. (d) Imagem Um (com o resultado da segmentação sobreposto). (e) Segmentação da Imagem Dois. (f) Imagem Dois (com o resultado da segmentação sobreposto).	97
5.7	Segmentação interativa. Decomposições e chaveamentos da (a) região superior esquerda; (b) região superior direita; (c) região inferior direita; (d) região inferior esquerda (composta com a imagem original).	99
5.8	Aplicação da decomposição uniforme no <i>framework</i> do <i>watershed</i> com marcadores propagados. (a) Quadro corrente. (b) Composição com a máscara corrente de segmentação D_f . (c) Pré-marcadores internos computados pela decomposição uniforme do quadro corrente de acordo com D_f . (d) Pré-marcadores externos. (e) Conjunto de marcadores internos M_f . (f) Conjunto de marcadores externos M_b	101
5.9	Aplicação da decomposição uniforme no <i>framework</i> do <i>watershed</i> com marcadores propagados. (a) Erosão dos marcadores M_f . (b) Erosão dos marcadores M_b . (c) Marcadores internos e externos propagados e sobrepostos ao quadro seguinte. (d) Segmentação do quadro seguinte.	102
5.10	Sequência <i>Foreman</i> (quadros 1 a 150).	105
5.11	Informação de movimento da sequência <i>Foreman</i>	106
5.12	Segmentação manual (Padrão-Ouro) da sequência <i>Foreman</i> (quadros 1 a 150).	107
5.13	Análise da segmentação Padrão-Ouro (manual) da sequência <i>Foreman</i> . (a) Número de interferências do usuário em cada quadro. (b) Tempo gasto na segmentação de cada quadro (em segundos).	108
5.14	Segmentação da sequência <i>Foreman</i> por <i>watershed</i> com marcadores propagados - com suporte da decomposição uniforme e da estimação de movimentos por Lucas-Kanade (quadros 1 a 150).	109
5.15	Análise da segmentação da sequência <i>Foreman</i> por <i>watershed</i> com marcadores propagados - com suporte da decomposição uniforme e da estimação de movimentos por Lucas-Kanade. (a) Número de interferências do usuário em cada quadro. (b) Tempo gasto na segmentação de cada quadro (em segundos).	110
5.16	Análise da segmentação da sequência <i>Foreman</i> por <i>watershed</i> com marcadores propagados - com suporte da decomposição uniforme e da estimação de movimentos por Lucas-Kanade. Erro de segmentação em cada quadro (comparado com o Padrão-Ouro.)	111

A.1	Varição da amarração de marcadores. (a) Máscara de segmentação tocando a borda da imagem em dois lugares. (b) Distância geodésica de um dos pedaços do contorno da máscara. (c) Sementes atribuídas a um dos pedaços do contorno. O outro pedaço não recebeu sementes. (d) Sementes atribuídas por todo o contorno (que não toca a borda da imagem).	118
A.2	Varição da amarração de marcadores. (a) Distância geodésica calculada ao redor do contorno da máscara. (b) Sementes atribuídas em função da distância geodésica ilustrada na Fig A.2 (a).	118
A.3	Criação de “Pares Completos” de Marcadores. (a) Máscara de segmentação. (b) Sementes.	119
A.4	Criação de “Pares Completos” de Marcadores. (a) Máscara de segmentação expandida. (b) Imagem expandida das sementes. (c) “Coroa” da máscara expandida. (d) Linhas de <i>watershed</i> que fatiam a “coroa” da máscara expandida. (e) “Coroa” fatiada.	120
A.5	Seleção das sementes. (a) Remoção das regiões que não possuem marcador interno. (b) Remoção das regiões que não possuem marcador externo. (c) Interseção das imagens Fig. A.5 (a) e Fig. A.5 (b) contendo as regiões que possuem marcador interno e externo. (d) Sementes selecionadas (menores) e (em destaque) as sementes removidas.	122
A.6	Marcadores amarrados. (a) Região associada a um par de marcadores. (b) Pares de marcadores envolvendo o contorno da máscara de segmentação. (c) Marcadores internos. (d) Marcadores externos.	123

Lista de Tabelas

3.1	Quantitative Evaluation of the Choice of Parameters: Amount of Intervention.	56
3.2	Quantitative Evaluation of the Choice of Parameters: Time Spent to Accomplish the Segmentation Task.	59
3.3	Quantitative Evaluation of the Choice of Parameters: Mean Time Spent for Frame (in seconds) and Mean Segmentation Error	62
4.1	Quantitative assessment of binding of markers improvements. Applications to <i>Foreman</i> and <i>Carphone</i> sequences (150 frames each sequence) (* : Mean / Frame).	70
5.1	Valores de extinção para os mínimos regionais da função na Fig. 5.4.	82
5.2	Erros de segmentação avaliados pela função $F'(I)$ de Borsotti et al.	94
5.3	Erros de segmentação avaliados pela função $Q(I)$ de Borsotti et al.	94

Trabalhos Publicados Pelo Autor

1. F.C. Flores, R.A. Lotufo. “Watershed from Propagated Markers: An Interactive Method to Morphological Object Segmentation in Image Sequences”. *Aceito para publicação na revista Image and Vision Computing*, 2010.
2. F.C. Flores, R.A. Lotufo. “Watershed from Propagated Markers Improved by the Combination of Spatio-Temporal Gradient and Binding of Markers Heuristics”. *Aceito para publicação nos Proceedings of VISAPP’2010*, Angers, França, Maio de 2010.
3. F.C. Flores, R.A. Lotufo. “Benchmark for Quantitative Evaluation of Assisted Object Segmentation Methods to Image Sequences”. *Em IEEE Proceedings of SIBGRAP’2008*, Campo Grande, Brasil, pg. 95-102, Outubro de 2008.
4. F.C. Flores, R.A. Lotufo. “Watershed from Propagated Markers Improved by a Marker Binding Heuristic”. *Em Mathematical Morphology and its Applications to Image and Signal Processing* (em ISMM’ 2007), Rio de Janeiro, Brasil, pg. 313-323, Outubro de 2007.
5. F.C. Flores, R.A. Lotufo. “Object Segmentation in Image Sequences by Watershed from Markers: A Generic Approach”. *Em IEEE Proceedings of SIBGRAP’2003*, São Carlos, Brasil, pg. 347-352, Outubro de 2003.
6. R.A. Lotufo, R. Machado, F.C. Flores, A.X. Falcão, R. Koo, G.S. Mazzela and R.M. Costa. “Pron-tovideo: An Image Sequence Segmentation Tool Applied to Video Edition”. *Em IEEE Proceedings of SIBGRAP’2001*, Florianópolis, Brasil, pg. 404, Outubro de 2001.

Capítulo 1

Introdução

Segmentação de objetos em sequências de imagens é a segmentação quadro-a-quadro de um objeto cuja semântica permanece inalterada. Técnicas de segmentação de objetos têm sido aplicadas em diversos *frameworks*, tais como edição de vídeo (*video masking*), codificação de vídeo (MPEG-4), monitoramento por vídeo e imageamento biomédico.

Há duas categorias de segmentação de objetos em sequências de imagens: automáticas (ou não-supervisionadas) e assistidas (ou supervisionadas ou, ainda, interativas). Na segmentação automática, os objetos são detectados automaticamente no quadro inicial e, então, perseguidos nos quadros seguintes, através da aplicação de técnicas de estimação de movimento. Na segmentação automática, não há intervenção do usuário nos resultados obtidos.

Na segmentação assistida, é permitida ao usuário a intervenção no processo de segmentação. O usuário pode escolher, por exemplo, os objetos a serem segmentados e como eles serão perseguidos. Além disso, o usuário tem a opção de corrigir e alterar os resultados de segmentação.

A segmentação automática tem sobre a segmentação assistida a vantagem de resolver sistematicamente o problema sem requerer esforço do usuário. Porém, é usual que o método cometa erros durante o processo, propagando-os para os próximos quadros e, assim, comprometendo os resultados.

A segmentação assistida tem, como uma de suas vantagens, a possibilidade de se supervisionar e corrigir os resultados da segmentação, quando necessário. Outra vantagem é a de não restringir tarefas de segmentação somente ao desenvolvedor de um método: com o conhecimento específico do problema, um usuário que não entende de processamento de imagens pode guiar o processo de segmentação do objeto de interesse.

Existem métodos assistidos com aplicação mais restrita, como alguns desenvolvidos para processamento de imagens médicas [1, 2]. Estamos interessados, porém, em um método de natureza genérica, i.e., que possa ser aplicado para solucionar problemas em diversos *frameworks*.

Muitos trabalhos sobre segmentação automática de objetos em sequências de imagens podem ser encontrados na literatura [3]. Porém, a literatura sobre métodos assistidos não é tão rica. Um dos motivos é que poucas técnicas de segmentação são apropriadas para interatividade. Em particular, o *watershed* com marcadores é muito apropriado para segmentação assistida.

Apesar do grande número de trabalhos relacionados à segmentação de objetos em sequências de imagens, na prática, a ferramenta computacional mais comumente usada para fazer esta segmentação ainda é a manual; o usuário delinea os contornos dos objetos usando ferramentas de desenho. Isto é justificado, uma vez que o método manual é o mais genérico aplicado a tais segmentações, que são

feitas de acordo com a percepção visual do usuário.

1.1 Objetivos

O principal objetivo desta tese é propor um método genérico para segmentação assistida de objetos em sequências de imagens [4]. Este método consiste em uma combinação de uma eficiente técnica de segmentação de imagens estáticas com técnicas de estimação de movimento [5, 6]. A técnica de segmentação está amarrada à técnica de estimação de movimento, uma vez que os marcadores para os objetos a serem segmentados são também propagados para os quadros seguintes objetivando a perseguição dos referidos objetos. Tais marcadores nada mais são do que um parâmetro que indica genericamente uma amostra do objeto a ser segmentado.

O *watershed* com marcadores propagados, como é chamado o método proposto nesta tese, depende da combinação dos bons funcionamentos dos métodos de segmentação e de propagação. Quanto melhor a segmentação, melhores os marcadores que serão propagados para o quadro seguinte. E quanto melhor a propagação, melhor será a máscara de segmentação computada para o quadro seguinte.

Para um método de segmentação assistida de objetos em sequências de imagens, é desejável a apresentação das seguintes propriedades:

1. **Interatividade:** o usuário pode intervir nos resultados de segmentação: é permitido ao usuário adicionar ou remover marcadores, corrigir segmentações imprecisas e escolher como os marcadores serão propagados. Por ser *interativo*, o método para permitir a escolha dos objetos de interesse no primeiro quadro bem como a correção de erros pontuais ao longo do processo de edição.
2. **Generalidade:** a técnica pode ser aplicada para qualquer sequência de imagens. Por ser *genérico*, não é necessário nenhum conhecimento *a priori* sobre a sequência. O usuário define os objetos a serem segmentados.
3. **Resposta Rápida:** com a implementação eficiente do método de segmentação, o método deve *responder rapidamente* após a inserção/remoção de cada marcador. Uma vez que o usuário interage basicamente através da inserção/remoção de um marcador, a resposta deve ser rápida para que o usuário não espere muito tempo para avaliar o resultado da segmentação. Além disso, com a boa escolha de um método de estimação de movimento, a propagação também deve ter uma rápida resposta.
4. **Edição Manual Progressiva:** o usuário não precisa “olhar para trás” e checar as segmentações nos quadros anteriores; elas são consideradas prontas. Também não é necessário apagar todos os marcadores impostos a um quadro quando ocorre uma segmentação ruim; tal segmentação é localmente corrigida através da adição ou remoção de marcadores a esta região. A *edição progressiva* permite encerrar a edição de um quadro após a sua segmentação ser positivamente avaliada, de modo que este quadro e os demais já processados não precisem mais serem visitados. A verificação da sequência toda após cada interação do usuário é usualmente proibitiva. Para isto, as futuras intervenções do usuário não devem intervir nos quadros passados.

A formulação do *watershed* com marcadores propagados levou à elaboração de um método com propriedades supracitadas. A interatividade permite a escolha do problema a ser resolvido e dos objetos de interesse a serem segmentados a partir do quadro inicial. A generalidade permite que o método seja aplicado para se resolver problemas em diversos *frameworks* onde se trabalhe com sequências de imagens. A implementação eficiente dos métodos que dão suporte ao *watershed* com marcadores propagados permitem que cada interferência do usuário resulte em uma resposta rápida do método. E a edição progressiva dispensa a checagem dos quadros anteriores durante o processo de segmentação, uma vez que o método proposto não altera os quadros segmentados e deixados prontos para trás.

Dada a sua generalidade, o método pode ser aplicado na solução de diversos problemas. Por exemplo, para segmentar objetos em sequências de imagens, o usuário pode impor os marcadores aos objetos de interesse e, uma vez que estes objetos são segmentados, os marcadores devem perseguir seus respectivos objetos no quadro seguinte. Imageamento médico é uma outra área onde o método pode ser aplicado, por exemplo, na segmentação de imagens 3D de ressonância magnética: o usuário pode impor os marcadores à primeira imagem no *deck* de imagens e ativar a propagação dos marcadores objetivando a segmentação volumétrica de um órgão.

Escolhemos o método de *watershed* com marcadores para a tarefa de segmentar os objetos em cada quadro. Este método, originado na Morfologia Matemática, reduz o problema de segmentação ao problema de se encontrar marcadores para os objetos que devem ser segmentados. Consiste na aplicação do gradiente morfológico seguido do operador *watershed*. O *watershed* com marcadores apresenta diversas vantagens: i) implementação que permite rápida execução do método; ii) o marcador é uma amostra do objeto a ser segmentado; iii) robustez e; iv) bastante estudado pela comunidade de Morfologia Matemática.

O *watershed* com marcadores é um método muito adequado para segmentação interativa, haja visto que os marcadores para os objetos de interesse podem ser adicionados ou removidos manualmente. Tal edição de marcadores pode ser facilmente realizada, por exemplo, com o *mouse*. Além disso, por ser o *watershed* um método de rápida execução - tipicamente algumas frações de segundo quando executado em imagens de 0,5 Mpixel, ele responde rapidamente cada vez que um marcador é inserido ou removido, o que possibilita com que seja executado a cada edição de marcadores.

A pré-segmentação dos objetos de interesse em um dado quadro também é facilitada pelas propriedades do método de segmentação. Uma vez que os marcadores, gerados em função dos objetos de interesse segmentados no quadro anterior, são propagados adequadamente, os mesmos objetos serão segmentados no quadro corrente. Em outras palavras, os objetos segmentados no quadro anterior definem, através dos marcadores propagados, os objetos a serem segmentados no quadro seguinte.

O método escolhido, nesta tese, para ser aplicado para a propagação de marcadores como estimador foi o método de estimação de movimento proposto por Lucas e Kanade. Este método é aplicado sob a suposição de que o movimento da imagem dentro de uma determinada região é constante. Isto entra em consonância com a formalização do método de segmentação proposto nesta tese. Como uma marcador não se deforma, cada ponto que o compõe deve ser propagado pelo mesmo vetor-deslocamento. Além disso, este método gerou os melhores resultados experimentais, comparado com outros métodos de propagação estudados.

Um modo trivial de se gerar e propagar marcadores consiste basicamente em propagar para o quadro seguinte os marcadores que o próprio usuário impôs manualmente no quadro corrente. Esta, porém, não é uma boa opção. Tais marcadores não são tão consistentes, além de usualmente não

serem suficientes para uma boa pré-segmentação no quadro seguinte. Isto justificou o desenvolvimento de métodos para a criação de marcadores mais apropriados para a segmentação de objetos em sequências de imagens.

Esta tese apresenta duas variações de implementação do *watershed* com marcadores propagados, propostas objetivando melhores resultados da aplicação do paradigma de segmentação em questão. Uma das variações é baseada na quebra do contorno da máscara de segmentação em pequenos segmentos de fácil edição. A outra variação baseia-se na decomposição hierárquica da máscara de segmentação e do fundo.

Além das variações, esta tese também apresenta a utilização do gradiente espaço-temporal (3-D) da sequência a ser segmentada como entrada para a aplicação do *watershed* bidimensional em cada quadro. O uso deste gradiente mostrou significativa melhoria dos resultados de segmentação da sequência, bem como a redução do número de interações necessárias para se obter as segmentações desejadas.

Este trabalho ainda apresenta um *benchmark* para avaliação quantitativa de métodos assistidos para segmentação de objetos em sequências de imagens. Esta avaliação é feita de acordo com vários critérios tais como robustez de segmentação e facilidade para segmentar objetos ao longo da sequência. Esta é uma proposta inicial para o *benchmark* de avaliação que, futuramente, apresentará medidas adicionais.

O *benchmark* foi aplicado em várias avaliações do *watershed* com marcadores propagados ao longo desta tese. As avaliações apresentadas nesta tese utilizando o *benchmark* de avaliação foram do tipo intraoperador, comumente utilizado para realização de experimentos de avaliação [7, 8].

1.2 Revisão Bibliográfica

A literatura apresenta, basicamente, dois tipos de métodos de segmentação assistida de objetos: os baseados em região e os baseados em contorno. O primeiro tipo segmenta o objeto pela classificação de todos os pontos que o compõem. O *watershed* é um exemplo de método baseado em região. O método baseado em contorno segmenta o objeto através do delineamento de sua borda. Como exemplo, citamos o paradigma do *Live Wire* [7]. Os cinco primeiros métodos citados abaixo são baseados em região. Os demais métodos, baseados em contorno.

Goldman *et al.* [9] propõem um método de perseguição de movimento 2-D aplicável em diversas aplicações interativas de vídeo, tais como anotação de objetos em movimento, navegação de vídeo e composição de vídeo. O método de perseguição consiste de uma etapa *offline* de pré-processamento seguida pela interface interativa que permite seleção de vídeo ao usuário. O pré-processamento consiste em calcular núvens de partículas que representam o movimento de alguns pontos na cena e na classificação destes pontos em K grupos. A etapa de interação permite ao usuário marcar os grupos relacionados aos objetos de interesse.

Bai *et al.* [10] apresentam um método para segmentação de objetos em sequências de imagens baseado em classificações locais da borda do objeto. A máscara de segmentação é inicialmente obtida interativamente. Em seguida, classificadores locais são aplicados ao longo da borda do objeto. Tais classificadores são então propagados para o quadro seguinte para segmentar o mesmo objeto. Estes classificadores locais integram medidas como cor, textura, forma e movimento. A propagação é realizada pelo cálculo de fluxo ótico. Após a propagação, os classificadores são atualizados. A

correção da segmentação é feita em cada janela sobreposta sobre a borda, aplicando segmentação em corte de grafo até convergir localmente ao resultado desejado.

Armstrong *et al.* [11] apresenta uma interface 3-D para segmentação volumétrica, apresentando o volume de imagens como um paralelepípedo. Marcadores são adicionados para demarcar objeto e fundo e a segmentação é realizada pela aplicação de um algoritmo de corte em grafo em cascata (CGC). Finalmente, o volume segmentado é renderizado usando OpenGL.

Ahn e Kim [12] propõem um método semi-automático para segmentação de objetos em sequências de imagens. Nesta proposta, o objeto de interesse e o fundo são segmentados manualmente e funções de densidade de probabilidade de cores são construídas para ambos. Em seguida, correlação espaço-temporal é utilizada para estimar uma região de coerência ao longo do contorno do objeto. O objeto é, então, segmentado através da minimização de uma função energia que recebe como entrada informações de cor, coerência e suavização.

Bai e Sapiro [13] propõem um método para segmentação interativa de imagens estáticas e dinâmicas. O método de segmentação para imagens estáticas recebe, além da imagem de entrada, marcadores (que os autores chamam de *scribbles*) para o objeto e para o fundo. A segmentação é realizada através do cálculo de distâncias geodésicas ponderadas entre os *pixels* da imagem e cada marcador. Os pesos são dados em função de um gradiente espaço-temporal. O método estático é estendido para o caso dinâmico através de restrições temporais do cálculo das distâncias geodésicas e por adição de marcadores nos quadros errôneos.

O próximo método apresentado aqui faz uso de um método de segmentação baseado no contorno do objeto. No método em questão, conhecido como *Live Wire* [7], a segmentação do contorno é dada sempre pelo caminho ótimo calculado entre pares consecutivos de pontos-âncora, colocados sequencialmente e interativamente ao longo da borda do objeto. A extensão do *Live Wire* para o caso 3-D [14] se dá pela aplicação do *Live Wire* 2-D no primeiro quadro do volume de imagem e na estimação de pontos-âncora para o quadro seguinte, em função dos pontos-âncoras impostos no quadro atual. Liu *et al.* [15] propõem um método em dois passos para análise de movimento de juntas em imagens médicas. Esta análise de movimento consiste na segmentação de uma juntas em diferentes *scans* do mesmo osso. No primeiro passo do método proposto, os autores fazem uso da extensão do *Live Wire* para o caso 3-D.

Um outro método, baseado no *Live Wire* 3-D, é proposto por Poon, Hamarneh e Abugharbieh [16]. O usuário segmenta manualmente algumas fatias do volume e, em seguida, o método automaticamente determina os pontos-âncora em todas as fatias em uma determinada direção ortogonal, para que o *Live Wire* 2-D possa ser aplicado em cada uma destas fatias. Os autores também apresentam uma interface 3-D intuitiva que implementa o método.

A comparação do método desenvolvido neste trabalho com os da literatura é de difícil realização pois usualmente os métodos são bastante complexos para sua implementação e os resultados presentes nos artigos são de difícil comparação. Foi neste sentido que apresentamos no capítulo 3 uma proposta de metodologia de *benchmarking* para segmentação assistida de sequência de imagens que visa comparar resultados de forma quantitativa.

1.3 Metodologia

Para uma visão geral e inicial da tese, um resumo da mesma é apresentado a seguir:

1.3.1 *Watershed* com marcadores propagados

O método proposto neste projeto de tese para segmentação de objetos em sequências de imagens - o ***watershed* com marcadores propagados** [4, 17] - consiste, basicamente, nos seguintes passos:

1. Os objetos de interesse são segmentados manualmente no primeiro quadro, através da aplicação do *watershed* com marcadores impostos interativamente.
2. Dada a máscara de segmentação dos objetos de interesse obtida no Passo anterior, marcadores internos (para os objetos de interesse) e externos (para o fundo) são criados em função de tal máscara.
3. Cada um destes marcadores é propagado para o próximo quadro por estimativa de movimento. Este novo conjunto de marcadores internos e externos é usado na aplicação do *watershed* com marcadores no quadro seguinte.
4. Se necessário, o usuário interage com os marcadores propagados, fazendo correções através da adição ou remoção de marcadores.
5. Vá para o Passo 2, até a sequência ser inteiramente processada.

Se a qualidade da segmentação não é aprovada em algum quadro, o usuário pode facilmente lidar com a edição dos marcadores problemáticos. Os marcadores gerados no Passo 2 devem ser apropriados para tratar deformações dos objetos, permitirem sua edição manual e individual, e serem propagados com rapidez e robustez para o próximo quadro. O objeto a ser segmentado é processado até o final da sequência ou até deixar a cena ou ser totalmente ocluso. Se é parcialmente ocluso, o método possibilita ao usuário intervir a fim de regularizar o processo.

O *watershed* com marcadores propagados é apresentado com maiores detalhes nos capítulos 2 e 3.

1.3.2 Variações do *watershed* com marcadores propagados

Diversos trabalhos foram desenvolvidos a partir do que foi reportado em [17]. Alguns destes trabalhos consistiram, principalmente, em apresentar aprimoramentos do paradigma proposto para segmentação de objetos em sequências de imagens. Estes aprimoramentos foram idealizados diante das dificuldades apresentadas pelo método como foi inicialmente proposto. Uma destas dificuldades é a propagação do marcador. Como a informação passada ao estimador de movimento é coletada da região sob o marcador, inicialmente pouca informação era provida ao estimador. Uma das motivações para o estudo dos aprimoramentos era prover o estimador com informação adicional para melhorar os resultados da propagação.

Outra motivação é a criação de marcadores que sejam de fácil edição para o usuário sem que se comprometa o resultado da segmentação como um todo. Por exemplo, o marcador poderia ser criado grande para ser de fácil edição. No entanto, a sua remoção poderia ocasionar um erro de segmentação que exigiria um grande número de interferências para ser corrigido. A idéia é resolver esta situação com a criação de marcadores localmente consistentes.

Esta tese apresenta dois tipos de variação:

- Baseada no **contorno** da máscara de segmentação: o contorno da erosão da máscara e o contorno da erosão do fundo são obtidos. Ambos os contornos são quebrados em pequenos segmentos que formam, respectivamente, o conjunto de marcadores internos e externos para cada objeto.
 - Os contornos são quebrados de modo a criar pequenos marcadores fáceis de serem editados. Se o contorno inteiro fosse tomado como marcador, este, além de não acompanhar a possível deformidade do objeto de interesse, não ofereceria opções mais eficientes de interação.
 - Esta melhoria funciona bem com contornos mal definidos ou objetos fortemente texturizados, uma vez que os marcadores são impostos próximos às bordas dos objetos a serem segmentados.
 - Os aprimoramentos baseados em contorno apresentadas nesta tese são a *amarração de marcadores* e o *ajuste pela média local dos vetores-deslocamento*.
- Baseada na **decomposição hierárquica** da máscara de segmentação: a região da máscara é segmentada hierarquicamente e os marcadores internos são selecionados entre os subconjuntos da partição gerada. O fundo é igualmente segmentado para se gerar os marcadores externos.
 - A máscara é particionada de modo a criar marcadores maiores e ao mesmo tempo fáceis de editar. Se a máscara inteira tivesse sido tomada como marcador, também não acompanharia possíveis deformações do objeto.
 - Tais marcadores tendem a cobrir consistentemente grandes regiões dentro do objeto e do fundo.
 - A variação baseada em decomposição hierárquica apresentada nesta tese é a que recebe *suporte da decomposição uniforme*, um método de segmentação hierárquica apresentado no capítulo 5.

As variações baseadas em contorno são apresentadas nos capítulos 2 e 3. A técnica baseada em decomposição hierárquica é apresentada no capítulo 5. Resumimos abaixo duas melhorias baseadas em contorno (*amarração de marcadores* e *ajuste pela média local dos vetores-deslocamento*) e uma baseada em decomposição (*com suporte da decomposição uniforme*).

Por amarração de marcadores

Dado um par de marcadores próximos (um interno ao objeto e outro externo), a heurística de amarração de marcadores é baseada em duas suposições:

1. A borda do objeto a ser segmentado cruza a região delimitada pelo par de marcadores.
2. O par de marcadores deve seguir o movimento da borda.

Considerando-se as suposições acima, ambos os marcadores do par devem ser propagados pelo mesmo vetor de movimento, ou seja, eles são propagados na mesma direção. Para o propósito de cálculo de tal vetor de movimento, a informação contida na região entre os marcadores é passada ao

estimador de movimento; o vetor calculado para esta região é atribuído aos marcadores do par. A amarração de marcadores apresenta uma melhor estimativa de movimento, pois antes o marcador era propagado em função da informação contida na região sob o marcador. Uma vez que é esperado que uma região localizada na borda do objeto de interesse no quadro seguinte tenha o melhor casamento com a região delimitada pelos marcadores no quadro corrente, espera-se que os marcadores persigam a borda do objeto.

O par de marcadores e a região delimitada por eles é criada através do processamento morfológico da máscara de segmentação do quadro corrente. Tal processamento recebe como parâmetros o comprimento dos marcadores e a distância entre os marcadores do par.

Pelo ajuste pela média local dos vetores-deslocamento

Nesta melhoria, o vetor-deslocamento associado a um marcador, responsável pela sua propagação, é corrigido através do cálculo de um *vetor médio* entre os vetores-deslocamento atribuídos a todos os marcadores impostos na vizinhança do marcador em questão (a ter seu vetor-deslocamento corrigido).

Esta heurística foi projetada de acordo com as seguintes considerações:

1. Exceto quando o objeto a ser segmentado toca as bordas da imagem (do quadro que está sendo processado), as bordas da máscara de segmentação são envolvidas por marcadores internos e externos.
2. É esperado que todos os marcadores impostos em uma região próxima a uma borda sigam o movimento desta borda. Mais, a variação entre os vetores-deslocamento calculados para os marcadores, impostos em uma vizinhança próxima a esta borda, deveria ser suave.

O cálculo do vetor-deslocamento corrigido e suavizado é feito considerando-se que o objeto é cercado por marcadores e que cada marcador tenha *marcadores vizinhos* impostos em ambos sentidos, horário e anti-horário. Nas regiões onde a máscara de segmentação toca a borda (quando este é o caso), os marcadores não tem vizinhos que extrapolem os limites da imagem.

Pela aplicação da decomposição uniforme

Outro aprimoramento é pela aplicação da decomposição uniforme, um novo método de segmentação hierárquica cuja propriedade principal é a preservação de uma dada característica para cada subconjunto da partição gerada pela decomposição. Cada um dos subconjuntos da partição respeita um critério de entrada, por exemplo, tamanho ou forma. A decomposição uniforme é comparável com a decomposição por limiarização dos valores de extinção e, além disso, oferece a vantagem da preservação do critério de decomposição para cada subconjunto da partição. Esta vantagem é explorada na utilização desta decomposição para gerar marcadores internos e externos mais consistentes que serão propagados para o próximo quadro, no *framework* de segmentação proposto.

1.3.3 Aprimoramento pelo uso de gradiente espaço-temporal

Além de melhorias para o paradigma proposto de segmentação, um aprimoramento foi estudado e aplicado: a inclusão de um gradiente espaço-temporal ao *framework* de segmentação [18]. De

posse dos conjuntos de marcadores internos e externos, impostos aos objetos de interesse em um determinado quadro, a segmentação do objeto é obtida pela aplicação do *watershed* com marcadores em um gradiente extraído de um gradiente espaço-temporal. Este gradiente espaço-temporal é uma sequência de imagens de gradiente, calculado como se a sequência de imagens de entrada fosse um volume.

O estudo deste gradiente espaço-temporal teve início em uma investigação preliminar sobre a aplicação de *watershed* interativo 3-D para segmentação de objetos em sequências de imagens. Como parte deste estudo preliminar, foi implementada uma interface de visualização e edição de marcadores para a segmentação 3-D. Os experimentos realizados com esta interface levaram a importantes observações.

Primeiro, que a informação fornecida pelo gradiente espaço-temporal gerou segmentações de muito boa qualidade para alguns objetos de interesse. Além disso, uns poucos marcadores possibilitaram a segmentação de um mesmo objeto em uma grande quantidade de quadros consecutivos.

No entanto, as desvantagens apresentadas pela segmentação interativa por *watershed* 3-D desestimularam o uso de tal método para a segmentação de objetos em sequências de imagens. Uma das razões é o longo tempo de execução de cada chamada do *watershed* 3-D. Se o custo de uma execução de tal método já é alto, o fato de ter que executá-lo sempre após a edição de algum marcador faz com que a aplicação do método através da interface proposta seja proibitivo.

Mesmo que o *watershed* 3-D fosse executado instantaneamente, ainda haveria o incômodo da inspeção do resultado após a adição ou remoção de algum marcador. Uma das vantagens da edição manual progressiva é o fato de o usuário não se preocupar mais com os resultados de segmentação para os quadros anteriores. Isto não ocorre na aplicação do *watershed* 3-D interativo para fins de segmentação de objetos em sequências de imagens devido a possíveis “vazamentos” do *watershed* em dados quadros da sequência.

Tais “vazamentos” ocorrem devido à falta de uma restrição da inundação do *watershed* na dimensão temporal. Como a inundação se dá espaço-temporalmente, pode ocorrer que a segmentação 3-D de um objeto “vaze” e se propague como uma segmentação errônea para uma grande quantidade de quadros consecutivos. A correção das segmentações geradas por tais “vazamentos” requeriria a inspeção da sequência inteira a cada edição de marcadores.

Apesar de abandonarmos a utilização do *watershed* 3-D para a segmentação de objetos em sequências de imagens, observamos que podemos utilizar o gradiente espaço-temporal no *framework* do *watershed* com marcadores propagados. O gradiente 3-D é calculado somente uma vez para o volume de imagens e, ao invés de aplicarmos o *watershed* 3-D em todo o gradiente volumétrico, aplicamos o *watershed* 2-D somente em uma fatia deste volume, referente ao quadro que está sendo processado.

Cada uma destas fatias contém informação espaço-temporal calculada em função de um elemento estruturante tridimensional. O uso de tais fatias permitiu uma segmentação mais robusta dos objetos de interesse em cada quadro. A melhoria foi obtida não só nos resultados da segmentação em si mas também na diminuição do número de intervenções do usuário para segmentar tais objetos.

O capítulo 4 apresenta mais detalhadamente a contribuição do gradiente espaço-temporal para o *framework* do *watershed* com marcadores propagados.

1.3.4 *Benchmark* para avaliação quantitativa de métodos assistidos de segmentação de objetos em sequências de imagens

Durante o desenvolvimento da tese, surgiu a necessidade de um método para comparar cada uma das técnicas desenvolvidas neste trabalho entre si como com as propostas na literatura.

Apesar de o número de trabalhos sobre avaliação de segmentação de vídeo ser uma pequena amostra do número de trabalhos sobre segmentação de vídeo, a literatura propõe várias métricas para avaliação quantitativa de segmentação [19, 20, 21, 22, 23]. Normalmente, tais métodos não consideram se a segmentação é automática ou assistida: a avaliação somente considera as máscaras de segmentação e a sequência de entrada. O desenvolvimento de métodos de avaliação é um problema em aberto [24, 23]: não há um método que seja considerado padrão para se avaliar segmentação de objetos em sequências de imagens [21].

Em relação a trabalhos sobre segmentação assistida de objetos em sequências de imagens, além do número de trabalhos na literatura ser pequeno, no melhor do nosso conhecimento não há trabalhos sobre avaliação quantitativa de tais métodos que levem em conta o histórico das intervenções do usuário. Considerando que a característica mais importante de métodos interativos é a propriedade de intervenção do usuário, um método de avaliação para segmentação assistida deveria quantificar quesitos como tempo de interação, tempo necessário para segmentar um objeto em um quadro e esforço para garantir uma segmentação desejável do objeto.

Um *benchmark* foi elaborado (e descrito em [4, 25]) para demonstrar quantitativamente a robustez e a produtividade do método de segmentação proposto. Este *benchmark*, que também serve para avaliar quantitativamente qualquer método de segmentação assistida de objetos em sequências de imagens, consiste na análise de quatro medidas correlacionadas: informação de movimento, número de interferências, tempo gasto na edição de cada quadro e erro (percentual) de segmentação. A avaliação de um método é feita em função da qualidade da segmentação e da produtividade da mesma (facilidade em que a segmentação é feita).

Detalhes adicionais sobre o *benchmark* são encontrados no capítulo 3.

1.4 Contribuições da tese

Esta tese apresenta as seguintes contribuições:

- Um paradigma de segmentação de objetos em sequências de imagens: o *watershed* com marcadores propagados;
- Duas variações deste paradigma:
 1. Baseada no contorno da máscara de segmentação;
 - Amarração de marcadores;
 - Ajuste pela média local dos vetores-deslocamento;
 2. Baseada na decomposição hierárquica da máscara de segmentação;
 - Suporte da decomposição uniforme;

- Melhoria dos resultados da segmentação do objeto, em um dado quadro, pelo uso do gradiente espaço-temporal na aplicação do *watershed* com marcadores naquele quadro;
- Um *benchmark* para avaliação quantitativa de métodos de segmentação assistida de objetos em sequências de imagens.

1.5 Organização da Tese

A tese está organizada na forma de artigos. Primeiramente, o *watershed* com marcadores propagados é apresentado, bem como uma primeira sugestão de implementação. Em seguida, são apresentadas diversas contribuições que dão suporte ao paradigma de segmentação proposto: um estudo sobre propagação de movimento e outras três variações do *watershed* com marcadores propagados. Finalmente, são apresentadas a utilização de gradiente espaço-temporal e do *benchmark* de avaliação de métodos de segmentação assistida de objetos em sequências de imagens.

Em resumo, o *watershed* com marcadores propagados é proposto somente uma vez. As demais contribuições apresentadas ao longo dos capítulos são melhorias pesquisadas para o paradigma proposto nesta tese. Além de uma proposta inicial para um método de avaliar tais melhorias.

Alguns dos capítulos que compõem esta tese consistem de artigos publicados ou submetidos. Cada um destes textos, como serão descritos a seguir, apresenta uma das contribuições supracitadas. A tese está organizada nos seguintes capítulos:

O capítulo 2 é composto pelo artigo que apresenta o *watershed* com marcadores propagados [17]. Neste artigo, o método de segmentação de objetos em sequências de imagens é proposto em sua primeira versão, baseada no contorno da máscara de segmentação. Também é apresentada uma primeira implementação do método, que foi chamada de Prontovídeo.

O capítulo 3 é composto pelo artigo aceito para publicação na revista *Image and Vision Computing* [4]. Este artigo é uma extensão do artigo apresentado no capítulo anterior [17]. O artigo que compõe este capítulo reapresenta o *watershed* com marcadores propagados e introduz duas variações - a amarração de marcadores e o ajuste pela média local dos vetores-deslocamento. Além disso, este capítulo também apresenta o *benchmark* para avaliação quantitativa de métodos assistidos de segmentação de objetos em sequências de imagens. Partes deste artigo foram publicadas em outros dois artigos de conferência: a amarração de marcadores foi publicada nos anais do *8th International Symposium on Mathematical Morphology* [26], em 2007. O *benchmark* foi publicado nos anais do *XXI Brazilian Symposium on Computer Graphics and Image Processing* [25].

O capítulo 4 apresenta a melhoria dos resultados do *watershed* com marcadores propagados pelo uso do gradiente espaço-temporal no *framework* de segmentação. O referido capítulo descreve a combinação do gradiente espaço-temporal com a amarração de marcadores. Vários experimentos foram realizados usando-se como entrada as sequências conhecidas como *Foreman* e *Carphone* e os resultados foram avaliados quantitativamente pelo *benchmark* de avaliação. O capítulo 4 foi composto por um artigo aceito para publicação nos anais da conferência VISAPP 2010 [18].

O capítulo 5 apresenta a melhoria do *watershed* com marcadores propagados pela aplicação da decomposição uniforme para a geração de conjuntos de marcadores internos e externos. Antes, porém, o capítulo apresenta a decomposição uniforme, descreve sua implementação e suas propriedades, além de propor aplicações da decomposição proposta para segmentação interativa, visão estéreo e segmentação hierárquica. O capítulo, enfim, apresenta a variação do *watershed* com marcadores propagados

obtida pelo suporte da decomposição uniforme, demonstra passo-a-passo a criação dos conjuntos de marcadores e mostra a avaliação do método pela aplicação do *benchmark*.

Finalmente, o capítulo 6 conclui a tese com uma discussão final sobre os métodos e sobre os resultados, além de propor futuros direcionamentos para este trabalho.

Capítulo 2

Object Segmentation in Image Sequences by Watershed from Markers: A Generic Approach

O *watershed* com marcadores propagados é um método que combina segmentação morfológica com estimação de movimento. Basicamente, após verificar-se que uma segmentação está completa em um determinado quadro, solicita-se a propagação de marcadores para o próximo quadro em busca da segmentação dos mesmo objetos de interesse.

Para um método de segmentação assistida de objetos em sequências de imagens, é desejável a apresentação das seguintes propriedades: i) generalidade, ii) interatividade, iii) resposta rápida e iv) edição manual progressiva. A formulação do *watershed* com marcadores propagados levou à elaboração de um método com propriedades supracitadas.

O *watershed* com marcadores propagados consiste em segmentar os objetos de interesse manualmente no quadro inicial com *watershed* interativo com marcadores e depois, para os quadros subsequentes, gerar e propagar marcadores para segmentar os mesmos objetos através da sequência. Os objetos serão segmentados com o *watershed* clássico aplicando-se os marcadores propagados e eventuais edições do usuário. A propagação, inicialmente sugerida neste primeiro capítulo, é feita por casamento de padrões.

O *watershed* com marcadores, responsável pela tarefa de segmentar os objetos em cada quadro, consiste na aplicação sequencial de operadores da Morfologia Matemática para se realizar segmentação de imagens. O uso de marcadores faz com que problemas de segmentação de imagens sejam usualmente reduzidos a problemas de se encontrar marcadores para os objetos a serem segmentados. No *framework* de segmentação proposto neste capítulo, dois tipos de marcadores são atribuídos em cada quadro: marcadores internos para os objetos de interesse e marcadores externos para o *background* e para os objetos não desejados.

O funcionamento do *watershed* com marcadores propagados depende da sinergia entre o método de segmentação e o de propagação. Quanto melhor a segmentação, melhores os marcadores gerados para serem propagados. E quanto melhor a propagação, melhor a pré-segmentação obtida no quadro seguinte. A escolha de um bom método para estimação de movimento é fundamental. Neste capítulo, foi proposto o casamento de padrões como método de propagação. A existência de métodos melhores e mais adequados motivou o estudo de outros métodos apresentados no próximo capítulo.

São apresentados neste capítulo duas versões do paradigma de segmentação.

1. Na primeira versão, os marcadores propagados são os mesmos que o usuário inseriu ou editou no quadro anterior.
2. Na segunda versão, o contornos quebrados da erosão da máscara de segmentação são usados como marcadores internos a serem propagados. Similarmente, os marcadores externos são obtidos com a quebra do contorno da dilatação da máscara de segmentação (Esta variação surgiu como uma solução para diversos problemas inicialmente combatidos na aplicação do paradigma tais como i) tamanho dos marcadores, ii) facilidade de edição dos mesmos, iii) segmentação e perseguição de bordas mal definidas e iv) custo de estimação do movimento do marcadores).

O capítulo ainda reporta a primeira implementação do método em Tcl/Tk. Foi implementada a primeira versão supracitada do método.

Este capítulo apresenta o primeiro artigo publicado sobre o *watershed* com marcadores propagados. O artigo “*Object Segmentation in Image Sequences by Watershed from Markers: A Generic Approach*” [17] foi publicado nos anais do SIBGRAPI 2003. Este artigo apresenta pela primeira vez o paradigma, além de descrever uma implementação do método.

Este artigo é resultado dos estudos iniciais do processo de doutoramento. Aluno e orientador desenvolveram idéias previamente elaboradas diante do interesse em projetos de edição de vídeo que levaram ao paradigma proposto e a uma implementação em Tcl/Tk.

2.1 Abstract

In this paper, we propose a generic method to interactive object segmentation in image sequences. This method consists in the extension of efficient techniques of static image segmentation to image sequences by combining them to fast motion estimation techniques. The proposed method has the following characteristics: interactivity, generality, rapid response and progressive manual edition.

2.2 Introduction

Image segmentation is the partitioning of an image in objects of interest. This partitioning is done according to some subjective criteria and, usually, it aims to separate the objects in the image or simply to separate a target object from the other ones.

Object segmentation in image sequences [27, 28] is the segmentation frame-to-frame of an object which semantics remains unchanged.

Object segmentation in image sequences has several applications such as:

1. Video Edition (*Video Masking*) [29, 30, 31, 27, 28];
2. Video Coding (MPEG-4) [32, 33, 34, 35, 3];
3. Biomedical Imaging [36, 37];

4. Video Surveillance [3, 38].

There are two manners to achieve object segmentation in image sequences: *automatic* (or *non-supervised*) and *assisted* (or *supervised* or also *interactive*). In the automatic segmentation, the objects are detected automatically in the initial frame (*I-frame*) and they are tracked in the following frames (*P-frames*), through application of motion estimation techniques. In the automatic segmentation there is no intervention by users in the obtained results.

In the assisted segmentation, the user is allowed to intervene in the segmentation process. The user can choose, for instance, the objects to be segmented, how they will be tracked, and he/she has the option to correct and alter the segmentation results.

Lots of works about automatic object segmentation in image sequences can be found in the literature (see [3]). However, the literature about assisted methods is not so rich.

Despite the great number of works related to object segmentation in image sequences, in practice, the computational tool commonly used to do this segmentation is still the manual one; the user draws the object contours using drawing tools. It is justified since the manual method is the more generic method applied to such segmentation: this segmentation is done according to the user's visual perception.

This paper proposes a generic method to interactive segmentation of objects in image sequences. This method consists in a combination of an efficient still-image segmentation with motion estimation techniques. The segmentation technique is tied to the motion estimation one, since the pointers to the objects to be segmented (that we will call *markers*) are also propagated to the next frames in order to track such objects.

The proposed method presents the following characteristics:

1. Generality: it can be applied to different kind of problems dealing with image sequences);
2. Interactivity: the user may intervene in the segmentation results;
3. Rapid Response: once a marker is imposed, the method must answer quickly;
4. Progressive Manual Edition: the user does not need to "look back" to check the previous segmentation; they are considered done.

Given its generality, the method can be applied to solve several problems. For instance, to segment objects in image sequences, the user can supply marker to the objects of interest and, once these objects are segmented, the markers must track their respective objects. Medical imaging is another area where this method can be applied, for instance, to 3D magnetic resonance image segmentation [36]: the user can supply markers to the first image in the image deck and call for the marker propagation in order to obtain the volumetric segmentation of an organ.

This paper is organized as follows: Section 3.3 discuss some works found in literature. Section 3.4 presents the watershed from markers, a morphological image segmentation technique. Section 2.5 describes how the markers are propagated. The proposed method to object segmentation in image sequences is proposed and discussed in Section 3.5. Finally, Section 4.6 concludes this paper.

2.3 Related Works

It were found in literature lots of works dealing with automatic segmentation of objects in image sequences [30, 27, 28, 39, 37, 3, 38, 35]. Despite their accuracy, such works present some disadvantages according the requirements we pointed above. For instance, in the automatic methods, it is not possible to correct the results interactively. Another strong disadvantage is the fact that these methods are not generic.

As stated above, the literature about interactive objects segmentation is not so rich compared to the literature about automatic objects segmentation. We found some interesting works in literature. Some of them present concepts close to what we propose in this paper.

Higgins [2] proposes a semiautomatic method to segment the right ventricle; the interactive stage consists in to separate the left ventricle and auricle manually. The next stages are automatic, consisting in 3-D morphological processing [40] of the image stack that compounds the volume. We found some problems in this technique: (i) despite it is a semiautomatic method, it is not possible a manual correction of the results, (ii) it is restricted to the problem of left ventricle segmentation, and (iii) it required many *a priori* knowledge to develop the technique.

Another interactive segmentation technique is proposed by Zanoguera and Meyer [41]. It is used a multiscale segmentation method, where the scale is given by ball with increasing diameters (the higher the diameter, the coarser the scale). The watershed technique applied here depends on the chosen ultrametrics and on such balls. The interaction with the user is done by two manners: i) the user simply choose the final number of remaining regions in the image after segmentation. (ii) an additional step where, starting from a set of regions, the user selects with a mouse the regions where the segmentation will be fine and the regions where the segmentation will be coarse.

Gu and Lee [31] propose a technique close to what we propose in this paper. Their method consists in the supervised segmentation in the I-frame and in the non-supervised segmentation in the P-frames. In the supervised step, the user draws the border with a mouse and morphological techniques correct this border, through application of watershed technique, segmenting the object. In the non-supervised step the warping of borders is computed to the next frame. It is done until occurs error, but the user may correct this error manually and restart the non-supervised step from the corrected frame. One of the problems found in this technique is the difficult to deal with the motion of large objects. Another problem is that the warping technique applied to estimate the border to the next frame is much more complex than to estimate the marker propagation to the next frame. The authors does not mention if the technique can segment various objects simultaneously.

2.4 Watershed from Markers

Let $E \subset \mathbb{Z} \times \mathbb{Z}$ be a rectangular finite subset of points. Let $K = [0, k]$ be a totally ordered set. Denote by $Fun[E, K]$ the set of all functions $f : E \rightarrow K$. An *image* is one of these functions (called graylevel functions). Particularly, if $K = [0, 1]$, f is a binary image. An *image operator* [40] (*operator*, for simplicity) is a mapping $\psi : Fun[E, K] \rightarrow Fun[E, K]$.

Morphological image segmentation via watershed from markers [42] consists in a sequential application of some morphological operators [40] to segment objects in the image. Basically, it consists in application of morphological gradient operator [40], marker selection and watershed from markers

technique [42, 43].

The *watershed from markers* can be described by flooding a topographical relief model of the gray-scale image. The markers are holes in the image relief where colored water can enter as the relief is flooded. There is one color associated to each set of markers. As the relief is uniformly flooded, different colored water may meet but cannot be mixed. When all the relief is flooded, each colored water region defines the *catchment basin* [44] associated to the marker. The classical watershed transform is when the markers are the regional minima of the image.

Watershed from markers is an efficient method to image segmentation, since it reduces the problem of objects segmentation to the problem of finding markers to these objects.

2.5 Motion Estimation by Marker Propagation

Given the imposition of a marker in a frame $f \in Fun[E, K]$ (at the instant i), the propagation of this marker to frame $g \in Fun[E, K]$ (at the instant $(i+1)$) is done by the computation of a translation vector, which will take the marker to the position where it will best “adjusted” in frame g .

Let $M \subset E$ be the set of all points, $x \in E$, belonging to the marker. Let $c \in M$ denote the point which the marker is centered. Let $S \subset E$, be the set of points, defined by,

$$s \in S \Leftrightarrow \exists x \in M : dist(s, x) \leq k,$$

where $dist(\cdot, \cdot)$ is a distance function [40] and k is a distance parameter. Note that $M \subseteq S$. The set S defines an area whose points are candidates to be the point where the marker will be centered in the next frame.

Let $M + y$ denotes the translation of marker M by the vector y , $y \in E$. The translation $M + y$ is given by $M + y = \{x + y : x \in M\}$.

Let n be the cardinality of $M \subset E$. Let $\mathbf{V}(f, M)$ be a n -dimensional vector given by, $\forall f \in Fun[E, K]$,

$$\mathbf{V}(f, M) = [f(x_1) f(x_2) \cdots f(x_n)]^T, x_i \in M.$$

The translation vector that will propagate the marker will be chosen from a set of vectors $(c + s) : s \in S$. The vector $(c + y)$, $y \in S$ that propagates the marker from $f \in Fun[E, K]$ to $g \in Fun[E, K]$ is given by

$$(c + y) : \mathbf{V}(f, M) \circ \mathbf{V}(g, M + y) < \mathbf{V}(f, M) \circ \mathbf{V}(g, M + s),$$

$\forall s \in S, S \neq y$.

The first technique we used as the “o” operation was the normalized correlation [45], in order to test the motion estimation framework, but, surely, it is not the best option. Once the n -dimensional vector $\mathbf{V}$ is computed, other techniques can be applied to do motion estimation as well.

2.6 The Proposed Method

The proposed method to object segmentation in image sequences consists, basically, in the following steps:

1. The objects of interest are segmented by the interactive watershed from markers.

18Object Segmentation in Image Sequences by Watershed from Markers: A Generic Approach

2. The markers are propagated to the next frame, through application of motion estimation to each marker. The propagated markers will be used in the watershed application to the next frame.
3. If necessary, the user interacts with the markers, doing the due corrections by deleting markers or including new ones.
4. Go to Step 2, until all sequence is processed.

Related to the kind of markers and their imposition, some experiments led us to propose the current solution. In the first experiment, the markers were inserted by the user as straight line segments, positioned with mouse clicks. Such markers are propagated by motion estimation to the next frame. This solution is very generic, since the same markers are used in the next frame. The motion estimation to each marker is very fast, because they are composed by a few pixels. However, it showed some inconveniences: i) when the object contour is not well defined or if it presents texture, it is necessary to add several markers to achieve the desired result, and these markers must be imposed close to the object borders. ii) when these markers are propagated to the next frame, the chance to occur a bad segmentation is great and it is hard to know which marker must be eliminated manually. iii) the markers are propagated just by translation; if the object of interest rotates in the scene, it is necessary to substitute the marker by another one, rotated.

In the second experiment, the first frame is segmented by interactive watershed application. The watershed result provides automatically two sets of markers; one of them obtained by eroding the segmented objects (internal to the objects) and the other set obtained by the erosion of the background (external to the objects). Both set of markers are applied to the next frame. This approach showed two disadvantages: i) due the large size of the obtained markers, the cost to find their motion estimation is computational expensive; ii) it is difficult to interact with the markers, because they form regions hard to edit.

Based on the two approaches cited above, it was developed an hybrid technique that explores just their advantages. The hybrid method to object segmentation in image sequences is describe next.

1. The objects of interest are segmented by the interactive watershed from markers.
2. Given the mask of the segmented objects, the **contour** of erosion of the object and the **contour** of the erosion of the background are obtained. Both contours are broken in short segments forming the set of inner and the set of outer markers to each object.
3. Each segment is propagated to the next frame by motion estimation. These new set of inner and outer markers are used to apply the watershed technique to the next frame.
4. If necessary, the user interacts with the markers, doing the corrections by adding or moving markers.
5. Go to Step 2, until all sequence is processed.

The hybrid technique also eliminates the disadvantages found in the first two methods. All contour of the object is wrapped by markers, and it works fine with bad defined contours or strongly textured objects. If the quality of segmentation is not approved in some frame, the user can easily move the

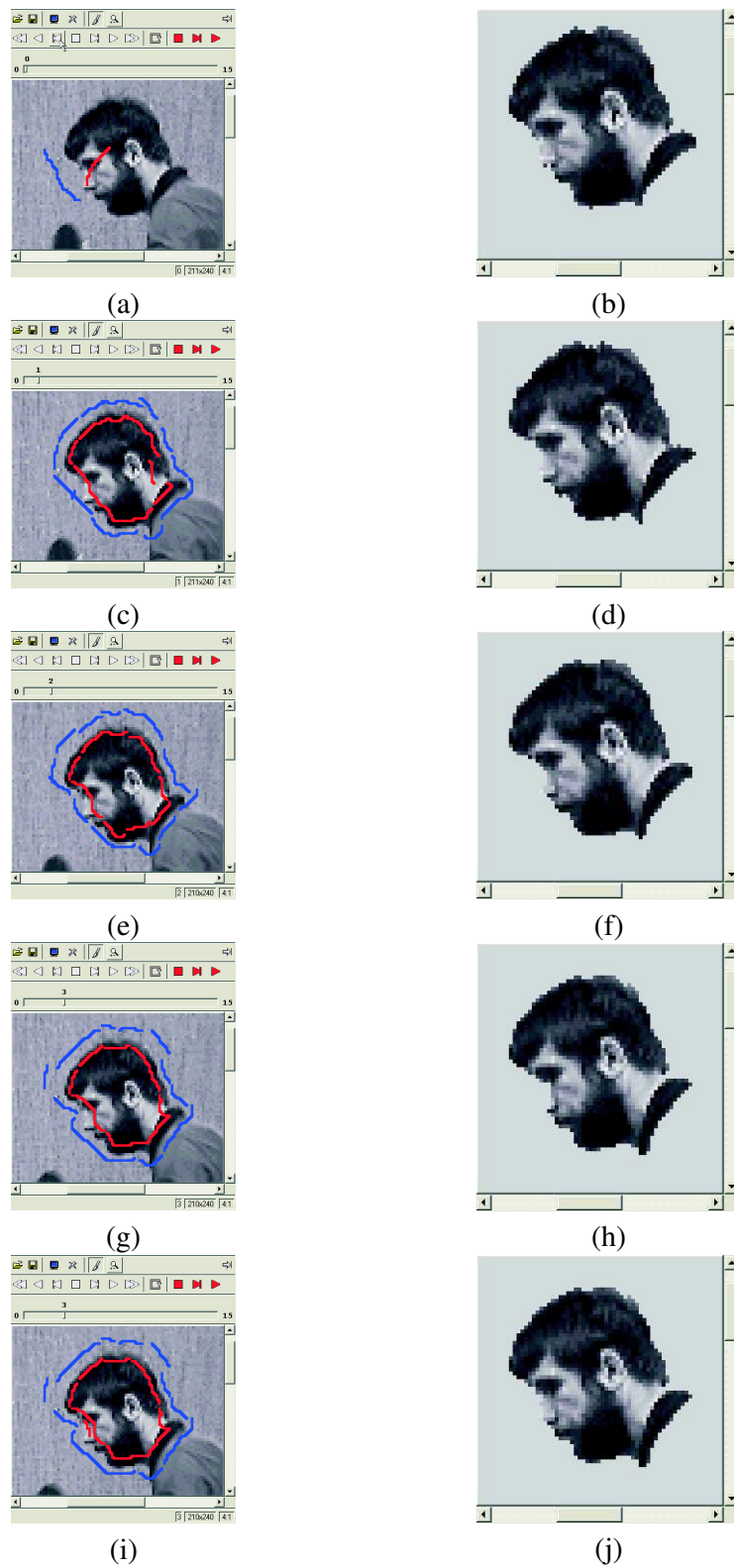


Fig. 2.1: The Proposed Method: *Left Column* - Image sequence (frames 0 to 3). *Right Column* - Object segmented for each frame.

short-segment marker. The marker propagation is very fast since each segment consists in a few points. Moreover, the contours follow the object deformation, since new markers are created from the segmentation of each frame.

Figure 2.1 illustrates the proposed method. Figure 2.1(a) shows the initial frame, with inner and outer markers imposed interactively. Figure 2.1(b) shows its respective segmentation result. Figures 2.1(c,e,g) show the frames 1 to 3 and Fig. 2.1(d,f,h) show their respective segmentation results. Note that the object (head) in the frame 3 was not segmented correctly. So, one marker was added (to the nose) to correct the segmentation (Fig. 2.1(i)). Figure 2.1(j) shows the corrected segmentation to frame 3.

2.7 Prontovideo

It was implemented a computational tool that does object segmentation in image sequences by the method proposed above. *Prontovideo* tool can be used to segment objects in a sequence by interactively placing markers with a mouse and by subsequent propagation of these markers to the following frames.

The requisites of the proposed method were satisfied: (i) Prontovideo can be used to apply the method to any kind of image sequence, (ii) the user may interact with the segmentation results, for instance, by adding new markers with the mouse or by selecting misplaced markers for deletion, (iii) the instantaneous response to the marker placement, and (iv) progressive manual edition. Once you segment manually the first frame by interactively selecting the placement of the markers, you can propagate the markers to next frame, check the quality of the segmentation, making corrections when necessary by deleting and inserting new markers. This process repeats until the the last frame.

Prontovideo was implemented in Tcl/Tk, a scripting language with a fast construction of graphical user interface. It were implemented functions to open/save sequences, to play them and to segment the objects (statically or through the sequence).

Its interface is composed by two windows: the main one and the mask one. The main window (Figure 3.1 (a)) shows the current frame to be segmented and provides access to some options as the choice of the method of propagation. In the main window the user interacts with the segmentation: all markers imposed to the current frame are available for editing (inner markers are red and outer ones are blue). Each marker is a distinct object and when the mouse pointer is on it, the marker is selected and appears highlighted.

The mask window (Figure 3.1 (b)) shows the current frame segmented according to the markers assigned to it. When the user plays the sequence he can watch it in both windows simultaneously: in the main window the user may watch the original sequence and the markers imposed to each frame can be seen. In the mask window, the user may watch the segmented sequence. The example shown in Figure 2.1 was created by Prontovideo application : left column shows a zooming of the main window and the right column, a zooming of the mask window.

The morphological operators, the watershed segmentation technique, the marker propagation techniques and other image handling functions were written in C. It were used the ADESSO, an authoring system for designing computer vision and digital image processing applications, to create automatic wrappers from Tcl/Tk to the C library.

Prontovideo is available for MS-Windows, Linux and Solaris platforms.



(a)



(b)

Fig. 2.2: Prontovideo (a) Main Window (b) Mask Window

2.8 Conclusion

This paper introduces an efficient method to object segmentation in image sequences. The main characteristics of this method is generality, interactivity, rapid response and progressive manual edition. The proposed method to segment objects in image sequences consists in a combination of classical morphological segmentation with motion estimation. Objects are segmented frame-to-frame by application of watershed from markers technique, and they are tracked by propagation of their respective markers.

Some methods related to imposition/computation and propagation of markers were tested. The best of them consists in to compute short segments close to the object borders and apply them as markers propagated to the next frame. This method avoids some problems found when dealing with borders and the motion estimation is very fast.

A prototype tool, named *Prontovideo* implements the proposed method. Prontovideo is implemented in Tcl/Tk, a scripting language with a fast construction of graphical user interface and it is available for MS-Windows, Linux and Solaris platforms.

Since the watershed technique shows itself a good solution to the segmentation step, we will do some efforts exploring alternative ways to do motion estimation. Initially, the normalized correlation had been applied as the motion estimator, but some alternatives have been investigated to boost/substitute it, since it is not the best method. Future works includes the application of some heuristics to improve the marker propagation, such as (a) the binding of pairs of internal/external markers to be propagated in the same direction and (b) the clustering of the optical flow vectors, assigned to each marker.

Capítulo 3

Watershed from Propagated Markers: An Interactive Method to Morphological Object Segmentation in Image Sequences

Como foi dito no capítulo anterior, outros métodos seriam mais adequados para a estimação de movimento do que o apresentado naquele capítulo. O que encorajou uma breve revisão de outros métodos de estimação de movimento. Dentre os testados, o método de Lucas e Kanade mostrou-se mais adequado para o *framework* do *watershed* com marcadores propagados.

Este artigo apresenta dois aprimoramentos para o método de segmentação proposto nesta tese, baseados no contorno da máscara de segmentação. No primeiro deles, chamado de amarração de marcadores, são calculados pares de marcadores ao longo da borda da máscara de segmentação. Os dois marcadores do par, um interno e outro externo, devem ser propagados pelo mesmo vetor deslocamento, calculado em função da informação contida entre os marcadores do par.

O outro aprimoramento apresentado consiste na correção de um vetor deslocamento que propaga um marcador pelo cálculo de um vetor deslocamento médio em uma dada vizinhança deste marcador.

Este artigo apresenta ainda como contribuição um *benchmark* para avaliação de métodos assistidos para segmentação de objetos em sequências de imagens. São avaliados critérios como qualidade de segmentação e a quantidade de trabalho necessária para realizar a tarefa de segmentação. Esta é uma proposta inicial do *benchmark* e outras medidas podem ser futuramente agregadas. Sobre os diversos experimentos realizados para avaliação do *watershed* com marcadores propagados, somente foram realizadas avaliações do tipo intraoperador, ou seja, executadas por um único operador [7, 8].

Este capítulo apresenta o artigo aceito para publicação na revista *Image and Vision Computing* [4]. O artigo “*Watershed from Propagated Markers: An Interactive Method to Morphological Object Segmentation in Image Sequences*” é uma extensão do artigo do SIBGRAPI 2003, com diversas contribuições como duas novas variações do *watershed* com marcadores propagados, um estudo sobre métodos para propagação de marcadores bem como a proposta de um *benchmark* de avaliação para métodos assistidos de segmentação de objetos em sequências de imagens. É o capítulo que apresenta as principais contribuições da tese.

Tal extensão se deu a convite do comitê de programa do SIBGRAPI 2003, que convidou a submissão de uma versão estendida do artigo apresentado no capítulo anterior. O processo de desenvolvimento do trabalho relatado se deu com as constantes revisões e exigências do comitê avaliador

da revista.

3.1 Abstract

In this paper, it is introduced an interactive method to object segmentation in image sequences, by combining classical morphological segmentation with motion estimation - the watershed from propagated markers. In this method, the objects are segmented interactively in the first frame and the mask generated by its segmentation provides the markers that will be used to track and segment the object in the next frame. Besides the interactivity, the proposed method has the following important characteristics: generality, rapid response and progressive manual edition. This paper also introduces a new benchmark to do quantitative evaluation of assisted object segmentation methods applied to image sequences. The evaluation is done according to several criteria such as the robustness of segmentation and the easiness to segment the objects through the sequence.

3.2 Introduction

Image segmentation is the partitioning of an image in objects of interest. This partitioning is done according to some subjective criteria and, usually, it aims to separate the objects in the image or simply to separate a target object from the other ones.

Image sequence segmentation is the temporal partitioning of a scene where such partitioning is given by a cut. *Object segmentation in image sequences* [17, 27, 28] is the segmentation frame-to-frame of an object which semantics remains unchanged. Object segmentation techniques have been applied to several frameworks, such as video edition (*Video Masking*) [29, 30, 31, 46, 27, 28], video coding (MPEG-4) [32, 33, 34, 35, 3], video surveillance [3, 38] and biomedical imaging [36, 37].

There are two manners to achieve object segmentation in image sequences: *automatic* (or *non-supervised*) and *assisted* (or *supervised* or also *interactive*). In the automatic segmentation, the objects are detected automatically in the initial frame (*I-frame*) and they are tracked in the following frames (*P-frames*), through application of motion estimation techniques. In the automatic segmentation there is no intervention by users in the obtained results.

In the assisted segmentation, the user is allowed to intervene in the segmentation process. The user can choose, for instance, the objects to be segmented, how they will be tracked, and he has the option to correct and alter the segmentation results.

Lots of works about automatic object segmentation in image sequences can be found in the literature (see [47, 3]). However, the literature about assisted methods is not so rich.

Despite the great number of works related to object segmentation in image sequences, in practice, the computational tool commonly used to do this segmentation is still the manual one; the user draws the object contours using drawing tools. It is justified since the manual method is the more generic one applied to such segmentation: this segmentation is done according to the user's visual perception.

This paper introduces a generic method to interactive segmentation of objects in image sequences - the *watershed from propagated markers* [26]. This method consists in a combination of an efficient still-image segmentation with motion estimation techniques. The segmentation technique is tied to the

motion estimation one, since the pointers to the objects to be segmented (that will be called *markers*) are also propagated to the next frames in order to track such objects.

The proposed method presents the following characteristics:

1. Interactivity: the user may intervene in the segmentation results: it must be allowed to the user to add/remove markers, to correct bad segmentation and to choose how the markers will be propagated.
2. Generality: the technique can be applied to any image sequences. It is not necessary *a priori* any knowledge about the sequence;
3. Rapid Response: once a marker is imposed or the propagation is activated, the method must answer quickly;
4. Progressive Manual Edition: the user does not need to "look back" to check the previous segmentation; they are considered done. It is not necessary to erase all markers imposed to a frame when a bad segmentation occurs; the bad segmentation is locally fixed by adding/removing markers to this region.

Given its generality, the method can be applied to solve several problems. For instance, to segment objects in image sequences, the user can supply marker to the objects of interest and, once these objects are segmented, the markers must track their respective objects [48]. Medical imaging is another area where this method can be applied, for instance, to 3D magnetic resonance image segmentation [36]: the user can supply markers to the first image in the image deck and call for the marker propagation in order to obtain the volumetric segmentation of an organ.

The evaluation of video segmentation is considered an open problem [21] and the literature about assessment techniques is new. Several evaluation methods found in literature consider just the quality of the segmentation and its impact in the solution of problems such as video coding. The amount of work required by the method is not usually assessed. Moreover, as the best of our knowledge, literature does not present any method to evaluate quantitatively the performance of an assisted video segmentation technique. Such assessment is interesting because it is expected that a good assisted object segmentation technique requires a small amount of intervention.

In order to evaluate the performance of the watershed from propagated markers, a new benchmark to quantitative evaluation of assisted video segmentation is proposed and applied on this paper. It consists in the measurement of some features such as the amount of intervention, the time spent to accomplish the task and the robustness of the method according to the quality of the segmentation (compared to a ground truth segmentation).

This paper is organized as follows: Section 3.3 discuss some works found in literature. Section 3.4 presents a well-known morphological image segmentation technique - the *watershed from markers*. The interactive method to object segmentation in image sequences by watershed from propagated markers is proposed and discussed in section 3.5. The motion estimation techniques applied to propagate the markers are presented in section 3.6 and some heuristics to boost the motion estimation results are introduce in section 3.7. Section 3.8 introduces the new benchmark to evaluation of assisted object segmentation in image sequences. Experimental results are presented and discussed in section 5.4.5. Finally, this paper is concluded in section 4.6.

3.3 Related Works

The literature offers lots of works dealing with automatic segmentation of objects in image sequences [30, 27, 37, 3, 38, 35]. Despite their accuracy, such works present some disadvantages according to the requirements pointed above. For instance, in the automatic methods, it is not possible to correct the results interactively. Another strong disadvantage is the fact that these methods are not generic.

As stated above, the literature about interactive objects segmentation is not so rich compared to the literature about automatic objects segmentation. We found some interesting works in literature [31, 41, 2, 49, 50, 51]. Some of them present concepts close to what we propose in this paper.

The proposal of Gu and Lee [31] to do interactive segmentation of objects in image sequences is close to the method proposed in this paper. The method has two steps: in the initial and supervised step, the objects are segmented manually in the I-frame by drawing their borders with a mouse and correcting through application of watershed technique. In the following and non-supervised steps, the borders of the objects of interest are found by warping the borders to the next frame. It is done until error occurs, but the user may correct this error manually and restart the non-supervised step from the corrected frame. One of the problems found in this technique is the difficulty to deal with the motion of large objects. Another problem is that the warping technique applied to estimate the border to the next frame is much more complex than to estimate the marker propagation to the next frame. The authors do not mention if the technique can segment various objects simultaneously.

Zanoguera and Meyer [41] propose another interactive segmentation technique where it is used a multiscale segmentation approach: the scale is given by balls with increasing diameters (the higher the diameter, the coarser the scale). The watershed technique applied here depends on the ultrametrics and such balls chosen. The user may intervene in the segmentation results by two manners: i) choosing the final number of remaining regions in the image after segmentation. (ii) an additional step where, starting from a set of regions, the user selects with a mouse the regions where the segmentation will be fine and the regions where the segmentation will be coarse.

In the semiautomatic method proposed by Higgins [2] to segment the right ventricle, the interaction with the user occurs in a first stage, where the left ventricle and auricle are separated manually. The next stages are automatic, and the image deck that compounds the volume is processed by 3-D morphological operators [52, 53, 40]. This technique has some problems: (i) despite it is a semiautomatic method, it is not possible a manual correction of the results, (ii) it is restricted to the problem of left ventricle segmentation, and (iii) it required much *a priori* knowledge to the develop the technique.

3.4 Watershed from Markers

Let $E \subset \mathbb{Z} \times \mathbb{Z}$ be a rectangular finite subset of points. Let $K = [0, k]$ be a totally ordered set. Denote by $Fun[E, K]$ the set of all functions $f : E \rightarrow K$. An *image* is one of these functions (called graylevel functions). Particularly, if $K = [0, 1]$, f is a binary image. An *image operator* [40] (*operator*, for simplicity) is a mapping $\psi : Fun[E, K] \rightarrow Fun[E, K]$.

Morphological image segmentation via watershed from markers [42, 54] consists in a sequential application of some morphological operators [52, 40] to segment objects in the image. Basically, it consists in application of morphological gradient operator [40], marker selection and watershed from

markers technique [42, 43]. In order to segment objects in color sequences, some metrics may be used to compute the color gradients [55, 56, 57].

The *watershed from markers* can be described by flooding a topographical relief model of the gray-scale image. The markers are holes in the image relief where colored water can enter as the relief is flooded. There is one color associated to each set of markers. As the relief is uniformly flooded, different colored water may meet but cannot be mixed. When all the relief is flooded, each colored water region defines the *catchment basin* [58, 44] associated to the marker. The classical watershed transform is when the markers are the regional minima of the image.

Watershed from markers is an efficient method to image segmentation, since it reduces the problem of objects segmentation to the problem of finding markers to these objects.

3.5 The Proposed Method

The proposed method to object segmentation in image sequences - the *watershed from propagated markers* - consists, basically, in the following steps:

1. The objects of interest are segmented by the interactive watershed from markers.
2. Given the mask of the segmented objects, the **contour** of erosion of the object and the **contour** of the erosion of the background are obtained. Both contours are broken in short segments forming the set of inner and the set of outer markers to each object.
3. Each segment is propagated to the next frame by motion estimation. These new sets of inner and outer markers are used to apply the watershed technique to the next frame.
4. If necessary, the user interacts with the markers, doing the corrections by adding or removing markers.
5. Go to Step 2, until the sequence is fully processed.

The method proposed above works fine with bad defined contours or strongly textured objects, since the markers are imposed close to the borders of the objects to be segmented. If the quality of segmentation is not approved in some frame, the user can easily move the short-segment marker. The marker propagation is very fast since each segment consists in a few points. Moreover, the contours follow the object deformation, since new markers are created from the segmentation of each frame. The object to be segmented is processed until the end of sequence or until it leaves the scene or be totally occluded. If it is partially occluded, the method makes possible the user intervention in order to regularize the process.

3.6 Motion Estimation Applied to Marker Propagation

The method introduced in the last section consists in a combination of a static segmentation method (to do object segmentation) with motion estimation (to do marker propagation). Watershed

from markers technique is a good choice to segment the objects. The motion estimation step, however, required the test of some classical techniques to compute optical flow field [5], and it was done in order to assess the accuracy of the motion estimation in the marker propagation task.

Optical flow field is a model used to represent the motion in an image. Given a sequence of frames, the optical flow is an approximation to the 2-D motion field which is computed by projecting the velocities of the 3-D surface points onto the imaging surface [59]. Once the optical flow field is computed, it may be used to estimate motion.

When image motion is measured, it is assumed that the graylevel at the image point $I(\mathbf{x}, t)$ is constant along the motion trajectories¹. Under such assumption, $I(\mathbf{x}, t)$ may be formally expressed as

$$I(\mathbf{x}, t) \approx I(\mathbf{x} + \delta\mathbf{x}, t + \delta t).$$

Using a Taylor expansion, the right side of the equation above may be rewritten as follows.

$$I(\mathbf{x}, t) = I(\mathbf{x}, t) + \nabla I \cdot \delta\mathbf{x} + \delta t I_t + \epsilon_{T_c},$$

where ∇I is the spatial gradient, I_t is the temporal gradient and ϵ_{T_c} represents the higher order terms of the Taylor expansion. Neglecting ϵ_{T_c} , subtracting $I(\mathbf{x}, t)$ from both sides of the equation and dividing them by δt , we have

$$\nabla I \cdot \mathbf{v} + I_t = 0.$$

The vector $\mathbf{v} = (u, v) = (\frac{dx_1}{dt}, \frac{dx_2}{dt})^T$ is the image velocity. Equation above is known as *optical flow constraint* (OFC). The OFC defines a single constraint on image motion, but it is not enough to compute $\mathbf{v}$. Since velocity $\mathbf{v}$ has two components, OFC is not sufficient to determine uniquely the motion field. Therefore, further constraints are needed to compute both components of $\mathbf{v}$.

There are several techniques applied to do motion estimation [3, 59, 60]. Two approaches to compute optical flow are common: the *gradient-based* (or *differential*) techniques compute velocity from spatiotemporal intensity gradients [61, 6]. The *region-based* (or *block-matching*) techniques define the velocity by selecting a displacement vector $\mathbf{d}$ from a set of candidate vectors which gives the best matching between image blocks at different times [45, 62].

One region-based optical flow estimator (where the distance measure to be minimized is the sum-of-squared difference [5]) and three gradient-based ones (Lucas-Kanade [6], Horn-Schunck [61] and based on second-order derivatives [5]) were implemented and tested. The techniques presented below are classical in the context of optical flow estimation, being regularly cited, and they are easy to implement as well.

3.6.1 Region-based method

As stated above, region-based matching techniques compute the velocity $\mathbf{v}$ as a displacement vector $\mathbf{d} = (d_x, d_y)$ which gives the best fit between regions (blocks) taken from frames at different times [3]. The displacement vector $\mathbf{d}$ is selected from a set of candidate vectors belonging to a search area $\mathcal{S}$: An image block $\mathcal{B}$ in frame k must be matched with blocks of same size within the area $\mathcal{S}$, placed in the following frame $k + 1$. The vector $\mathbf{d}$ is given by the block in frame $k + 1$ which gives the best matching within block $\mathcal{B}$ (in k).

¹This hypothesis is done in the continuous case. $I(\mathbf{x}, t)$ denotes the image graylevel at the point $\mathbf{x}$ taken at the instant t .

The candidate vector $\mathbf{d}$ is the displacement vector to be assigned to block $\mathcal{B}$ if it maximizes a similarity measure (as the normalized cross-correlation [45]) or minimizes a distance function, for instance, the sum-of-squared difference given below:

$$SSD_{k,k+1}(\mathbf{x}, \mathbf{d}) = \sum_{j=-n}^n \sum_{i=-n}^n W(i, j) \cdot [I(\mathbf{x} + (i, j), k) - I(\mathbf{x} + \mathbf{d} + (i, j), k + 1)]^2,$$

where I_k is the k -th frame and W is a discrete 2-D window function [5].

3.6.2 Second-order differential method

The conservation of spatial gradient ($d\nabla I(\mathbf{x}, t)/dt = 0$) is a constraint that can be imposed to the optical flow computation. Second-order derivatives lead to the following velocity vector computation:

$$\begin{pmatrix} \frac{\partial^2 I}{\partial x_1^2} & \frac{\partial^2 I}{\partial x_2 \partial x_1} \\ \frac{\partial^2 I}{\partial x_1 \partial x_2} & \frac{\partial^2 I}{\partial x_2^2} \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix} + \begin{pmatrix} \frac{\partial^2 I}{\partial t \partial x_1} \\ \frac{\partial^2 I}{\partial t \partial x_2} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

Often, velocity vectors estimated by second-order methods are less accurate than the vectors estimated by the first-order ones [5].

3.6.3 Horn-Schunck method

The method proposed by Horn-Schunck [61], one of most popular among the gradient-based methods, combines two constraints in the estimation of the velocity field. The first constraint is the already discussed optical flow constraint (OFC). The second one, called *global constraint*, smooths the variation of the motion field estimated over the image plane.

Let

$$\epsilon_{OFC} = \nabla I(\mathbf{x}, t) \cdot \mathbf{v} + I_t$$

and

$$\epsilon_{SC}^2 = \|\nabla u(\mathbf{x}, t)\|^2 + \|\nabla v(\mathbf{x}, t)\|^2$$

be, respectively, the error due the optical flow constraint and the error due the smoothness constraint. The velocity field $\mathbf{v}(\mathbf{x}, t) = (u(\mathbf{x}, t), v(\mathbf{x}, t))$ is estimated by the minimizing the following total error, defined over a domain D ,

$$\epsilon_{\text{total}} = \int_D (\epsilon_{OFC}^2 + \lambda^2 \epsilon_{SC}^2) dx,$$

where λ weights the influence of the smoothness term. The solution for $\mathbf{v}$ is given as a set of Gauss-Seidel equations which are solved iteratively.

3.6.4 Lucas-Kanade method

The additional constraint in the method proposed by Lucas-Kanade [6] is that, at given time, the motion field is assumed constant over a block $\mathcal{B}$ of points. Formally, this *local constraint*, is given by,

$$\mathbf{v}(\mathbf{x}, t) = \mathbf{v}, \forall \mathbf{x} \in \mathcal{B}.$$

The constant motion field $\mathbf{v}$ over $\mathcal{B}$ is given such that $\mathbf{v}$ minimizes the error due the local constraint ϵ_{LC} . The local constraint error is given by

$$\epsilon_{LC}(\mathbf{v}) = \sum_{\mathbf{x} \in \mathcal{B}} W^2(\mathbf{x}) \cdot [\nabla I(\mathbf{x}, t) \cdot \mathbf{v} + I_t]^2,$$

where $W(\mathbf{x})$ is a window function that gives more weight to the central part of the block $\mathcal{B}$. The minimization of ϵ_{LC} can be solved using least square techniques [59, 60, 63].

The details about the utilization of the techniques presented in this section to do marker propagation will be explained in section 5.4.5.

3.7 Marker Propagation Improvements

A reasonable assumption about marker propagation is that two closer markers assigned to the same object should have similar displacement vectors, i.e., both markers should follow the motion of the object. However, there are situations where it does not occur properly and some correction should be done.

Besides the study of some classical optical flow estimation methods, we designed some heuristics to improve the results given by the motion estimators. This paper introduces two of them. The first one consists in creating pairs of internal and external markers that will be propagated together [26]. The second heuristic applied is the correction of the displacement vector by computing the average among the vectors assigned to its neighbour markers. Any of the motion estimation techniques presented in last section may have its results improved by the heuristics introduced below.

3.7.1 Binding of markers

Let us consider a pair of closer markers (one internal and the other external) assigned to the same object. The heuristic proposed here [26] is based on two assumptions:

1. The border of the object to be segmented crosses the region delimited by both markers of the pair (Fig 3.1 (a)).
2. The pair of markers must follow the motion of the border.

Considering the assumptions above, both markers of the pair must be propagated by the same displacement vector, i.e. they are propagated to the same direction. For purpose of computing such displacement vector, the region delimited by both markers is considered as a "marker" (see Fig. 3.1 (a) - such region is the negated one located between the markers); the displacement vector computed to this "marker" is assigned to the pair of markers. Since it is expected that a region located at the border of the object of interest in frame $k + 1$ gives the best match to the region delimited by the bound markers (Fig. 3.2 (b)), such markers should track the border of the object.

Figure 3.1 illustrates the idea. Figure 3.1 (a-b) shows the morphological gradient of two consecutive frames (both gradients are zoomed). Figure 3.1 (a) shows a pair of markers, an internal (red) and an external (green). The area delimited by the markers was highlighted by negating the gradient

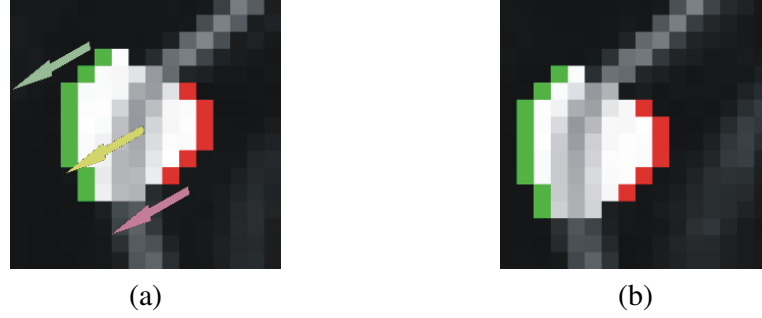


Fig. 3.1: Binding of markers. (a) The internal (red) and external (green) markers delimit a region crossed by the object border. (b) Markers are propagated by the same displacement vector, computing in function of the delimited region.

at the area. The motion estimation is done considering that area as the marker to be propagated. The displacement vector assigned to this area (as illustrated in Fig. 3.1 by the yellow vector) must be used as the displacement vector of the markers bound to this area (the red and green vectors are exemplified in Fig. 3.1 (a)). Figure 3.1 (b) shows the pair of markers propagated to the next frame.

The pair of markers assigned to the object of interest and the region used to estimate the displacement vector are created by morphological processing of the mask M of the segmented object (Fig. 3.2 (a)). Two parameters are required: the distance m between the internal and the external markers and the width of the area, given by w .

1. Let $M_{\delta,m}$ and $M_{\varepsilon,m}$ be, respectively, the dilation and the erosion of the mask M by a disk structuring element with diameter m . Both images will be used to compute a “crown”, by subtracting $M_{\varepsilon,m}$ from $M_{\delta,m}$ (Fig. 3.2 (b)).
2. The next step consists in creating seeds that will be used to separate the regions. Such seeds must be imposed on the contour of M and the geodesic distance [40] (started from a point picked from the contour) between them must be w . It is done by labeling the contour of M using the geodesic distance function and by analyzing the division of each label by w ; the points which label divided by w remain zero are the seeds (Fig. 3.2 (c)).
3. The seeds will be used as markers in an application of the watershed operator. The resulting watershed lines (Fig. 3.2 (d)) will be used to slice the “crown” of the mask.
4. The slicing is done by computing the intersection between the negation of the watershed lines and the “crown”. Fig. 3.2 (e) shows the sliced regions identified by a label (let L be the image which contains the labeled regions).
5. Given the set of all delimited areas, the creation of the markers is simple. To create the internal markers, just compute the intersection between L and the contour of $M_{\varepsilon,m}$. To create the external markers, compute the intersection between L and the contour of $M_{\delta,m}$. Fig. 3.2 (f) shows the pair of markers wrapping the borders of the mask M . Each pair of internal and external received a distinct label.

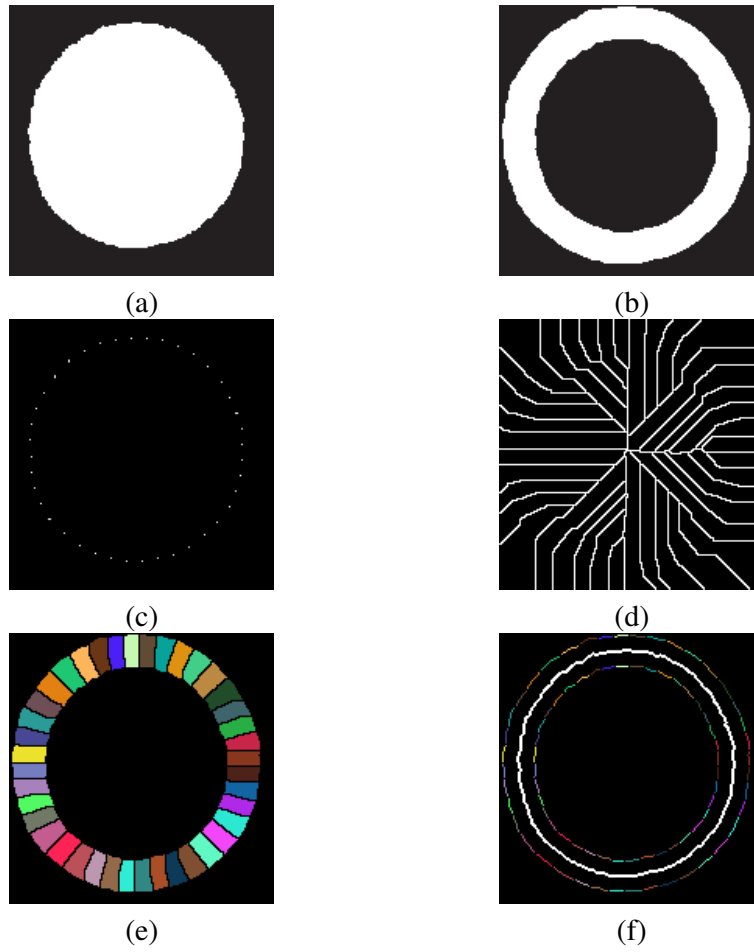


Fig. 3.2: Creation of the marker pairs and the regions delimited by them. (a) Mask of the segmented object. (b) The “crown” of the mask that must be sliced in order to obtain the delimited regions. (c) The seeds to be used to slice the “crown” of the mask. (d) Watershed lines. (e) Delimited regions (labeled). (f) Pair of markers (labeled) wrapping the borders of the mask.

3.7.2 Adjustment by local averaging of the displacement vectors

This heuristic works under the following assumptions:

1. Excepting in the case where the object to be segmented touches the frame borders (Fig. 3.3 (b)), the borders of mask of the object segmented is wrapped by internal and external markers (Fig. 3.3 (a)).
2. It is expected that all *neighbour* markers imposed close to a border should follow the motion of this border. The variation among the displacement vectors computed to the markers imposed to a *neighbourhood* close to the border of the object should be smooth.



Fig. 3.3: Wrapping of markers (a) The mask does not touch the borders of the frame. (b) The mask touches the frame borders.

To explain the heuristic itself, it is necessary to define what is a neighbour of a marker. In this definition, the kind of the marker (internal/external) is not considered; the definition works for both kind of markers.

Let $\mathcal{M}$ denotes a marker. A $\mathcal{M}_i$ is a i -th *neighbour* of $\mathcal{M}$ if it is the i -th marker to be counted, starting from the marker itself (i.e., $\mathcal{M}$ is its own $\mathcal{M}_0$ neighbour) and counting counterclockwise on the border of the mask. A $\mathcal{M}_{-i}$ is a $-i$ -th neighbour if it is the i -th marker to be counted clockwise, starting from $\mathcal{M}$ itself. One important restriction is that two markers are neighbour in one direction if and only if the object of interest does not touch the frame border. If a marker, for instance is imposed close to the frame border, this marker does not have i -neighbours in the direction of the border.

Figure 3.4 gives an example of a set of neighbours of a marker $\mathcal{M}$ (the dark blue marker impose at the border of the object). The $\mathcal{M}_1$ and $\mathcal{M}_{-1}$ markers are painted with a lighter blue and located close to the border and to $\mathcal{M}$ and markers $\mathcal{M}_2$ and $\mathcal{M}_{-2}$ can be seen, respectively, in a counterclockwise and clockwise counting around the object.

Let $\mathbf{N}_i(\mathcal{M}) = \{\mathcal{M}_{-i}, \mathcal{M}_{-(i-1)}, \dots, \mathcal{M}_{-1}, \mathcal{M}, \mathcal{M}_1, \dots, \mathcal{M}_{i-1}, \mathcal{M}_i\}$ denotes the i -neighbourhood of a marker $\mathcal{M}$ (Fig. 3.4). Let $\mathbf{v}(\mathcal{M})$ be the displacement vector that propagates the marker $\mathcal{M}$ to the next frame.

Given a marker $\mathcal{M}$, the displacement vector $\mathbf{v}_{\text{sm}}(\mathcal{M})$ that corrects (smoothes) the propagation of $\mathcal{M}$ is computed by averaging the displacement vectors of all markers in $\mathbf{N}_i(\mathcal{M})$.

Let $A_i = \{a_j, j \in [0, i] : a_j > a_{j+1}\}$ be the set of all weights assigned to the displacement vectors of the markers in $\mathbf{N}_i(\mathcal{M})$. The condition $a_j > a_{j+1}$ is taken in account since that the far a



Fig. 3.4: Marker neighbourhood $N_i(\mathcal{M})$.

marker is located from the marker which vector will be smoothed, the lower must be its influence in the averaging computation.

The displacement vector $\mathbf{v}(\mathcal{M})$ will be replaced by the smoothed vector $\mathbf{v}_{\text{sm}}(\mathcal{M})$, given by,

$$\mathbf{v}_{\text{sm}}(\mathcal{M}) = \frac{1}{S_a} \cdot (a_0 \cdot \mathbf{v}(\mathcal{M}) + \sum_{j=1}^i a_i \cdot [\mathbf{v}(\mathcal{M}_j) + \mathbf{v}(\mathcal{M}_{-j})]).$$

where $S_a = a_0 + 2 \cdot \sum_{j=1}^i a_j$.

The definition of the smoothed vector computation is given considering that the object is surrounded by markers (i.e., all markers have neighbours in both directions, clockwise and counter-clockwise). In the cases where the object of interest touches the frame border and the marker does not have i -neighbour in the direction of the border, the function above may be used but assumes that the weights to be assigned to the missing neighbours are $a_i = 0$.

3.8 Benchmark for Quantitative Evaluation of Assisted Object Segmentation Methods in Image Sequences

An important issue about object segmentation in image sequences is the evaluation of the segmentation results according to one or more criteria. Such criteria can be qualitative (for instance, how the segmentation mask represents the objects of interest) or quantitative (for instance, the gain provided by the segmentation mask in the compression process or the measurement of work need to do an assisted segmentation). The evaluation can be driven to some application [20, 22] and, in some cases, not only the quality of segmentation is assessed but also the tracking of the objects.

Considering the availability of a ground-truth (or golden standard) segmentation, the evaluation method may be classified as *relative* or *standalone* [24, 21]: it is relative when the evaluation is done by comparing the segmentation results to the ground-truth segmentation. And it is standalone when it does not occur.

One of the most popular methods to evaluate object segmentation in image sequences is by visual inspection, following subjective criteria. However, the design of computational methods to do such

evaluation has been strongly motivated because subjective evaluation is expensive and depends on a set of assessment conditions.

It may be found in literature several works that propose computational metrics for quantitative assessment of object segmentation in image sequences, such as performance measures [19, 20], contextual relevance [21], perceptual relevance [22], spatial accuracy and temporal coherence [23]. Usually, such evaluation methods do not consider if the segmentation method is automatic or assisted. They do only required the segmentation masks and the input sequences to do the measurements. Note, however, that the number of works about video segmentation evaluation is far below the number of works about video segmentation itself. Evaluation design is considered an open problem [24, 23]: there is no method considered standard to do object segmentation in image sequences [21].

As stated in the introduction, the literature about automatic segmentation is very rich. And several works about automatic segmentation also present an evaluation about the method in order to illustrate its accuracy. Video coding performance is one of the most popular ones [64, 65, 66]. Unfortunately, it does not occur with assisted segmentation literature: besides the small number of works about assisted segmentation of objects in image sequences found in literature, as the best of our knowledge, there is no work on quantitative evaluation of assisted segmentation methods.

Considering that the most important feature of the assisted methods is the user intervention property, an evaluation method to assisted segmentation results should quantify, for instance, the time response for each intervention, the time needed to segment objects in a given frame and the amount of work needed to assure a good/correct segmentation, i.e., the number of interventions needed in each frame to achieve the desired segmentation of the object.

This section proposes a benchmark to do a quantitative and relative evaluation of assisted segmentation results. It analysis the impact that user intervention has in the framework of such methods. A method is good if it provides a robust segmentation with a minimum amount of user intervention and a good time response. The benchmark consists in a set of correlated measurements that quantifies several factors such as the number of user interferences for each frame and the time spent in each frame to finish its segmentation. The segmentation error for each frame is given by the symmetrical difference between the segmentation mask in the frame and the ground truth segmentation for this frame. The amount of motion between consecutive frames, which gives a notion of difficulty to segment the sequence, is also computed.

The proposed benchmark to evaluation of assisted methods consists in the following measures:

3.8.1 Motion information

Given the ground truth segmentation, the motion information of an object in the sequence is given in function of the symmetrical difference between the segmentation masks in consecutive frames of the ground truth sequence.

Let us consider a segmentation mask as a binary image valued 1 in the pixels that belong to the segmented object (and 0, otherwise). Let gt_i be the ground truth segmentation mask for frame i . Let $\#(f)$ be the number of pixels valued 1 in a binary image f . Let ψ be the symmetrical difference between two binary images f and g , given by

$$\psi(f, g)(x) = \begin{cases} 1 & \text{if } |f(x) - g(x)| = 1 \\ 0 & \text{otherwise.} \end{cases}$$

The motion information I_i for a frame i is given by

$$I_i = \frac{\#(\psi(gt_{i-1}, gt_i))}{\#(gt_{i-1}) + \#(gt_i)}.$$

Figure 3.6 shows the motion information for *Akiyo* sequence. These values will be discussed below.

3.8.2 Number of user interferences in each frame

This measure is done by counting how many times the user intervenes in the current segmentation result in a given frame. The result of this measurement depends on the kind of interface available to the user, i.e., the options the user has to interfere in the segmentation result. In this paper, the implemented interface consists in the addition of points or line segments as internal and/or external markers. Such objects are given by mouse clicks and drags. It is also possible to select a marker, with the mouse, and call for its deletion. In this interface, all additions and removals count as an intervention.

This measure also provides the total of interventions in the sequence and the mean of interferences for each frame. Figure 3.8 (a) and Fig. 3.10 (a) show, respectively, the amount of user intervention in the ground truth segmentation and the provided by watershed from propagated markers to *Akiyo* sequence. It will be also discussed below.

3.8.3 Time spent in the edition of each frame

Given a frame from the image sequence, this measurement gives the time passed from the ending of segmentation in the previous frame (and the consequent start of the segmentation process to the current frame) to the ending of segmentation in the current frame. Many things occur in this time interval: the assessment of the initial segmentation of the frame (given some propagated information from the previous frame), the edition of the segmentation result by addition and removal of information, the segmentation algorithms themselves, the tracking algorithms, the visual inspection of the current segmentation in the frame, etc. The harder the task to segment a given object in the current frame, the greater the time needed to accomplish the task.

Time information is computed to each frame, but it is “global” to the frame, i.e., the time spent in a frame is the sum of all actions occurred until the object segmentation is completed. Time measurement also provides the total time spent to segment the object through the entire sequence, and the mean time spent in the edition of each frame. Figure 3.8 (b) and Fig. 3.10 (b) show, respectively, the time spent in the ground truth segmentation and the provided by watershed from propagated markers to *Akiyo* sequence. The time is given in seconds.

3.8.4 Segmentation error in each frame (Compared to the Ground Truth)

The last measurement is given by the error between the segmentation mask in the current frame, provided by the assessed method and its respective reference ground truth segmentation. This error is also measured in function of the number of pixels that belongs to the symmetrical difference between the segmentation masks.

Let seg_i be the segmentation mask provided by the application of the evaluated method to frame i , and let gt_i be the ground truth segmentation for frame i . The segmentation error SE_i for a frame i is given by

$$SE_i = \frac{\#(\psi(seg_i, gt_i))}{\#(seg_i) + \#(gt_i)}.$$

A robust method usually provides a good segmentation result, with a few small segmentation errors along the object border (see examples below). However, the error information computed to a given frame is considered “global” to this frame: the evaluation does not consider the segmentation errors locally in the frame.

This measure may be done after the segmentation of the object is completed, through the sequence. It also may provide the mean error for each frame. Figure 3.11 gives an example of the error measurement to the *Akiyo* sequence. Of course, segmentation error of the ground truth is zero.

All four measurements work together in the evaluation of the assisted method. It is expected that, in the frames which have high motion information, the segmentation error is also high, except if this error is fixed by user intervention. And, if it occurs, it will have impact in the measurement of the number of interferences and in the time spent to do the corrections. In other words, it is expected that frames with high motion information present high segmentation error or high rate of interference.

More, the robustness of the method (i.e., it works with small segmentation errors) is related to the number of user interferences: the lower the segmentation error rates, the lower the number of interferences (and, thus, the time spent in editions). Please compare the measurements shown in Fig. 3.8 and Fig. 3.10.

Given the popularity and the cost of the manual segmentation of objects in image sequences, this benchmark may also be used to quantify the difficulty to segment manually an image sequence. In this paper we will show the difficulty to compute the ground-truth segmentations for *Akiyo* and *Foreman* sequences.

3.8.5 Example: *Akiyo* sequence

The benchmark was applied to evaluate the segmentation of the journalist in the initial 150 frames of *Akiyo* sequence (Fig. 3.5). This sequence is very simple to segment and track, since it has a very slow movement, regions with high contrast and a static background. It is also largely tested in segmentation and coding frameworks [67, 68].

Two segmentations were done in order to illustrate the benchmark: i) manual and ii) by watershed from propagated markers. The interaction was done by insertion of points and line segments as internal/external markers and by the removal of misplaced markers. In the manual segmentation (ground truth), each object was segmented by manual imposition of all markers needed to achieve a perfect segmentation by classical watershed from markers. The second experiment adopted the binding of markers heuristic and the Lucas-Kanade motion estimator.

Figure 3.6 shows the motion information of *Akiyo* sequence, given the ground truth segmentation of the journalist. Note that, since the motion is minimal in the sequence, the error between consecutive masks is low and, thus, the motion information is also low. The mean error between consecutive frames is 0.2373%.

The manual segmentation of the journalist (ground truth) is shown in Fig. 3.7. The interference and time spent graphics are shown, respectively, in Fig. 3.8 (a) and (b). Except for a few frames, the

user intervention was not required so much. It was required 1170 interferences to do the complete ground truth segmentation - a mean of 7.8 interferences for frame. The total time spent in the manual segmentation was 3568.2 seconds = 59.4701 minutes (mean of 23.78 seconds to segment each frame).

Figure 3.9 shows the segmentation result provided by the watershed from propagated markers. Figure 3.10 (a) and (b), show, respectively, the number of interference and the segmentation time for each frame. Watershed from propagated markers was robust in a way that only four of 150 frames needed the user intervention. It required just 7 interferences in all frames - a mean of 0.0467 interventions for frame. The total time spent to segment the object in all frames was 638.2670 seconds = 10.6378 minutes (mean of 4.2551 seconds each frame). Note that there is no interference in big intervals of the interference graphics. And note that the greater peaks in the time graphics are related to the interferences.

Segmentation error is shown in Fig. 3.11. Given the easiness to segment the sequence and the robustness of the watershed from propagated markers, the segmentation error was low (mean error of 0.5987% for each frame). The source of error in this experiment is the fact that some small regions in the border of journalist hair present a low contrast, what causes bad segmentation in such regions.

The journalist in *Akiyo* sequence is very simple to segment and this experiment was done only to illustrate the benchmark. Section 3.9.3 presents the evaluation of three object segmentation methods applied to segment the man in *Foreman*, a much more complex sequence.

3.9 Experimental Results

This section presents the results of several experiments done with the proposed method to do object segmentation in image sequences. The first experiment consists in a comparison among the four optical flow estimators presented in section 3.6 applied to estimate the motion to markers. The second experiment compares both heuristics introduced in section 3.7 in order to improve the marker propagation. The third experiment applies the benchmark proposed above to evaluate three segmentation methods to video segmentation according to user intervention criteria. And the third experiment assess the choice of two parameters of the method: the length of the markers and the distance between the internal and external markers that define a pair.

3.9.1 Comparison among optical flow estimators

In this experiment, all derivatives were computed with four-point central differences for differentiation. Here, color images are converted to grayscale before the motion estimation.

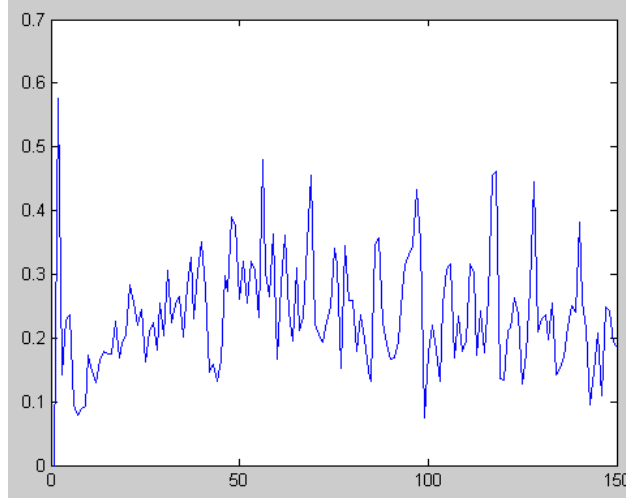
It is important to note how a marker is propagated. A marker $\mathcal{M}$ is a connected set of points of E . To propagate the marker by a displacement vector $\mathbf{v}$, each point of $\mathcal{M}$ is translated by $\mathbf{v}$, i.e., if $\mathbf{x} \in \mathcal{M}$, then $\mathbf{x}$ will be translated to point $\mathbf{x} + \mathbf{v}$.

The region-based motion estimator consider the marker $\mathcal{M}$ as a block to be matched on the next frame. Since the displacement vector is computed in function of the block, the optical flow in each point of the block is the same. So, the displacement vector $\mathbf{v}$ computed by the block-matching estimator is used to translate all marker points.

The Lucas-Kanade motion estimator applied to do marker propagation assumes that, at given time, the motion field is constant over a marker $\mathcal{M}$. As in the region-based case, the displacement



Fig. 3.5: *Akiyo* sequence (frames 1 to 150).


 Fig. 3.6: Motion information from *Akiyo* sequence.

vector $\mathbf{v}$ computed by this estimator is used to translate all marker points.

The Horn-Schunck and the second-order motion estimators compute a displacement vector *to each point of the marker*. In both implementations, it is necessary to compute a vector $\mathbf{v}_{\text{mean}}$ given by the mean of all displacement vectors assigned to the points of $\mathcal{M}$. The vector $\mathbf{v}_{\text{mean}}$ will be used to translate all points of $\mathcal{M}$.

The goal of the experiments described in this subsection is to compare the marker propagation given by the four motion estimation techniques presented above and the segmentation results provided by such propagations. In the experiments done, region-based and second-order differential estimators did not show accuracy in the markers propagation task (Fig. 3.12 (a) and (c), respectively) and so, they lead to bad segmentation of the object (Fig. 3.12 (b) and (d)). Horn-Schunck and Lucas-Kanade based-on estimators showed themselves well accurated (Fig. 3.12 (e) and (g), respectively) providing good segmentation results (Fig. 3.12 (f) and (h)).

Among the motion estimation techniques compared in this paper, we choose the Lucas-Kanade method to do marker propagation. Besides the good results achieved by application of this optical flow estimator, the assumption that the motion field is assumed constant over a region is very suitable to the method proposed in this paper to do object segmentation in image sequences, since a marker can be considered as such region and all of its points must be translated by the same displacement vector.

3.9.2 Comparison among the proposed heuristics

The goal of this experiment is to evaluate the improvement of the Lucas-Kanade estimation by application of the two heuristics introduced above. Figure 3.13 shows the propagation and segmentation results achieved by the heuristics.

Figure 3.13 (a) shows the marker propagation by Lucas-Kanade estimation, without adjustment. The length of each marker is $\mathbf{m} = 10$ pixels and the distance of each marker and the border (before propagation) is $\mathbf{w} = 10$ pixels. The result is good except for a few misplaced markers that led to a

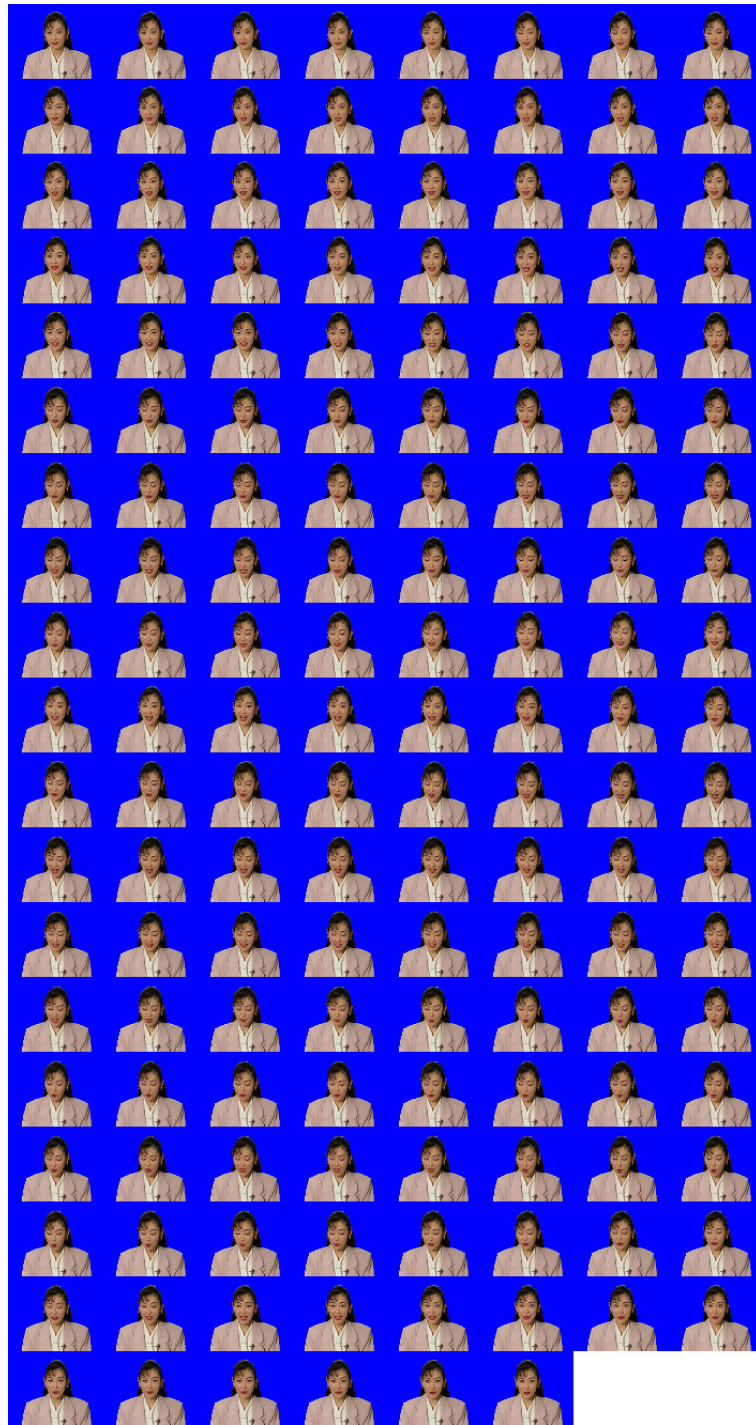
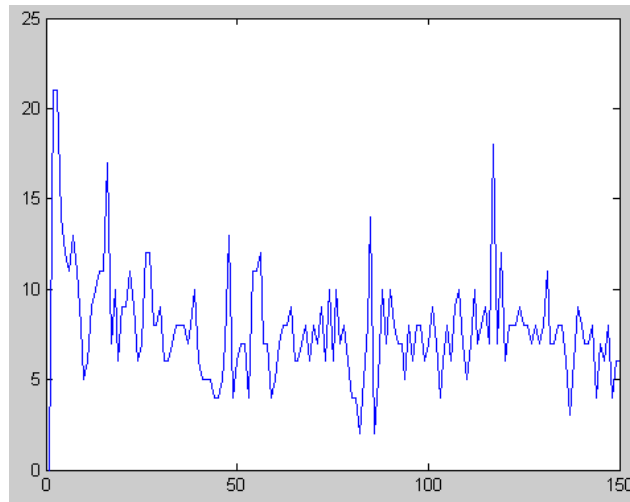
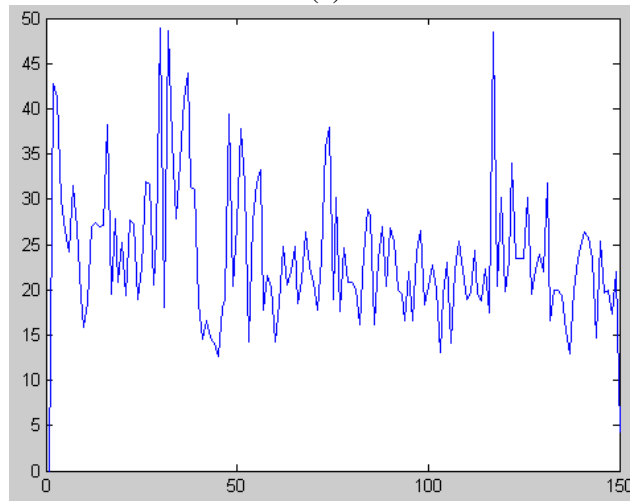


Fig. 3.7: Manual segmentation (Ground Truth) of *Akiyo* sequence (frames 1 to 150).



(a)



(b)

Fig. 3.8: Analysis of the Ground Truth segmentation (manual one) of the *Akiyo* sequence. (a) Number of user interferences in each frame. (b) Time spent in the edition of each frame (in seconds).

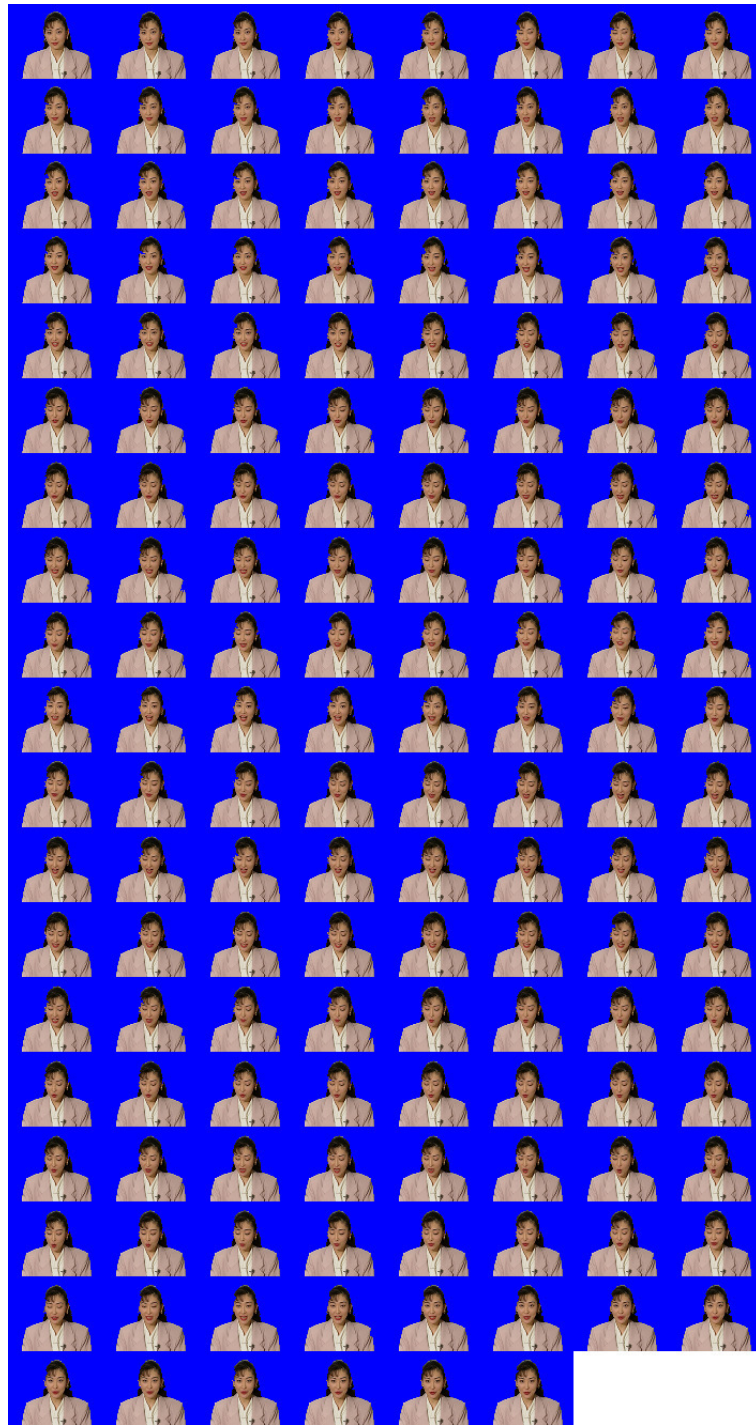
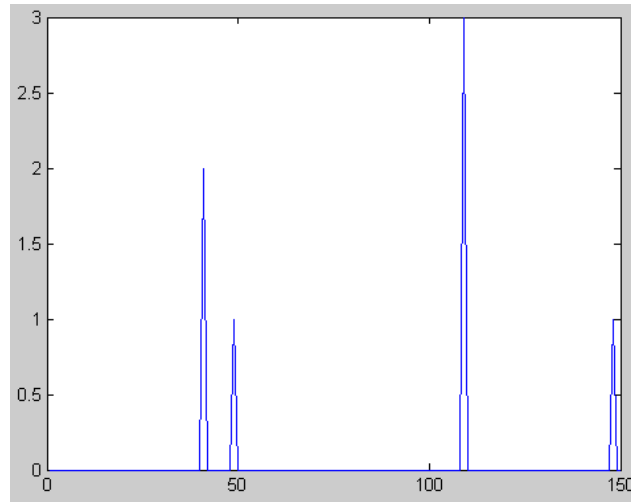
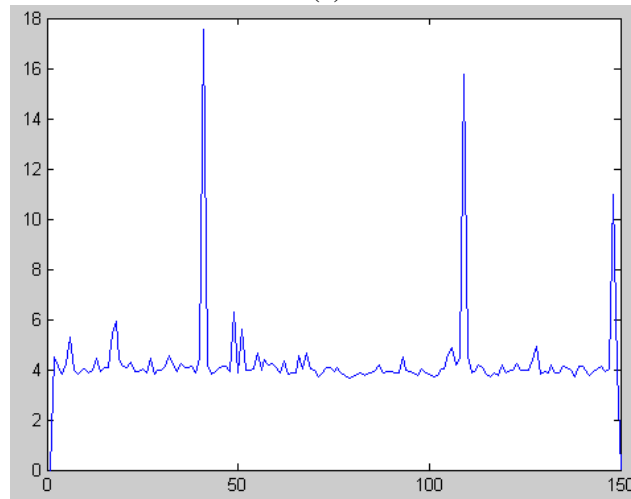


Fig. 3.9: Segmentation of *Akiyo* sequence by Watershed from Propagated Markers (frames 1 to 150).



(a)



(b)

Fig. 3.10: Analysis of the segmentation of *Akiyo* sequence by Watershed from Propagated Markers. (a) Number of user interferences in each frame. (b) Time spent in the edition of each frame (in seconds).

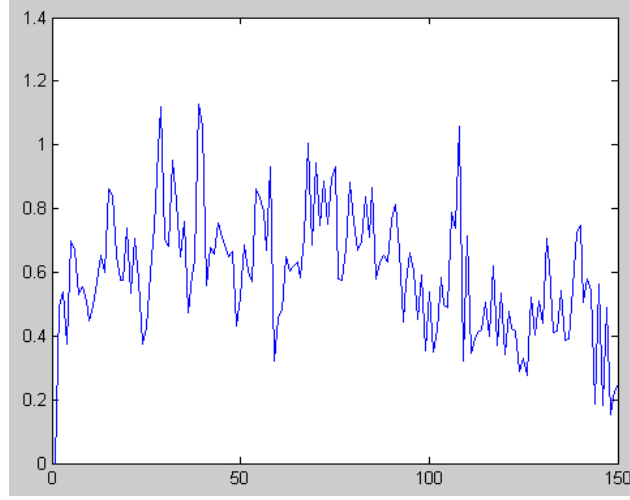


Fig. 3.11: Analysis of the segmentation of *Akiyo* sequence by Watershed from Propagated Markers: Segmentation Error in each frame (Compared to the Ground Truth.)

bad segmentation in some regions (Fig. 3.13 (b)).

Local averaging of the displacement vectors was applied to smooth the Lucas-Kanade estimation in Fig. 3.13 (c). On this experiment, the 2-neighbourhood $\mathcal{N}_2(\mathcal{M}) = \{\mathcal{M}_{-2}, \mathcal{M}_{-1}, \mathcal{M}, \mathcal{M}_1, \mathcal{M}_2\}$ and the set of weights $A_2 = \{a_0 = 1, a_1 = 0.66, a_2 = 0.33\}$ were chosen. Again, each marker are 10 pixels length and 10 pixels away from the border (before propagation). The smoothed propagation is better than the achieved by Lucas-Kanade without adjustment (Fig. 3.13 (c)), but some markers may still be misplaced, leading to a few bad segmented regions (Fig. 3.13 (d)). Note that, despite the markers propagated correctly "fix" the propagation of a misplaced marker in their neighbours, the displacement vector of the misplaced marker is a source of error to the adjustment of the good displacement vectors (in other words, the displacement vector of a misplaced marker may be considered as a "noisy vector").

The heuristics of bind pair of markers gives the best results in this experiment (Fig. 3.13 (e)). The fact that both inner and outer markers of the pair were propagated by the same displacement vector avoids the local crossing of the markers (i.e., the internal marker of the pair is not propagated outer than the external one, and vice-versa). Moreover, despite the region between the markers is greater than the markers themselves, it provides more information than the markers without significant loss of performance. The segmentation errors occurred in Fig. 3.13 (f) is due to the segmentation itself and not due to marker propagation.

3.9.3 User intervention

In order to demonstrate the benchmark and the robustness of the watershed from propagated markers, another classical sequence was segmented. As *Akiyo* sequence, *Foreman* sequence has interesting characteristics that allow us to illustrate some properties of the method proposed in this paper.

Several authors use the *Foreman* sequence to demonstrate the performance of their methods, from

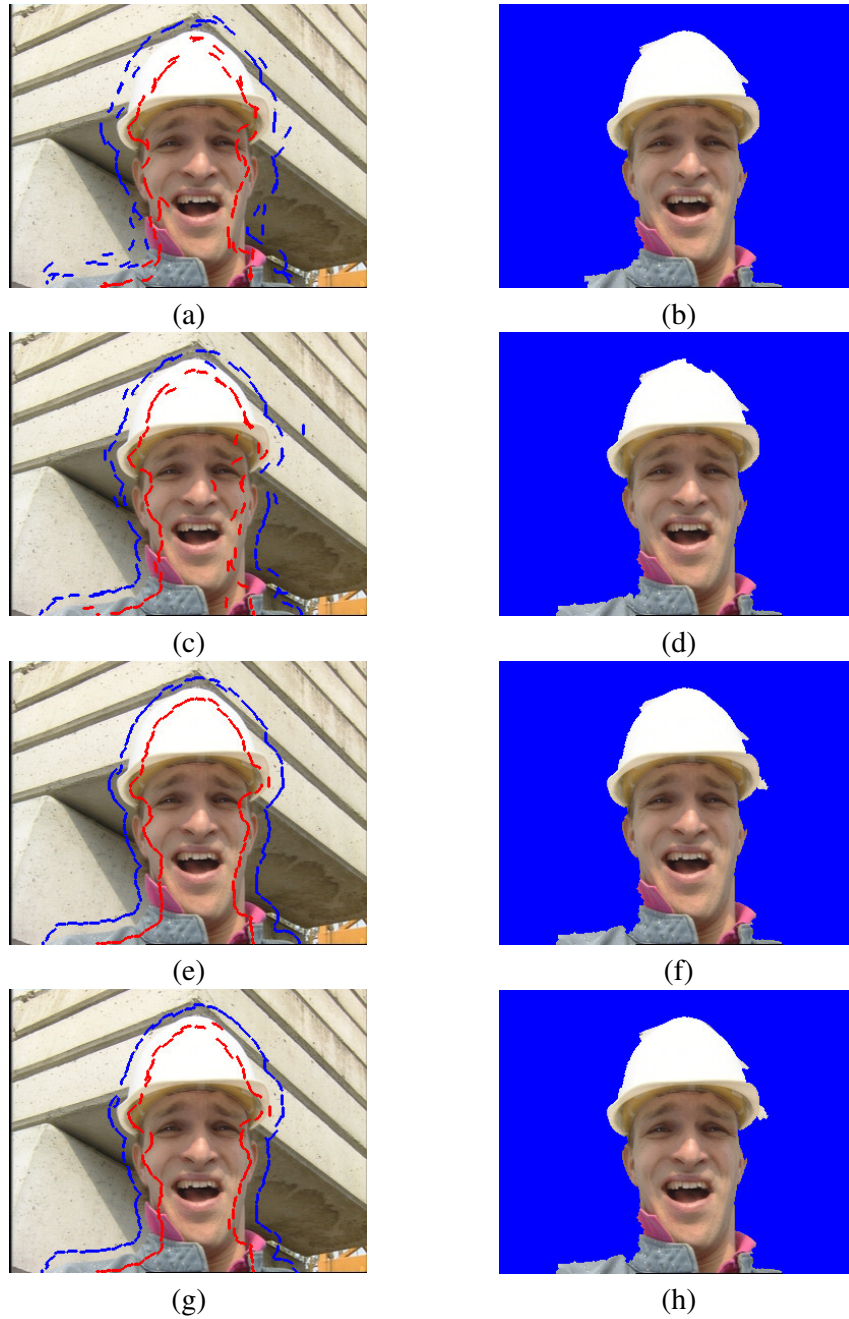


Fig. 3.12: Marker propagation comparison - Applied to *Foreman* sequence. (a) By region-based estimation. (b) Segmentation result. (c) By second-order differential estimation. (d) Segmentation result. (e) By Horn-Schunck estimation. (f) Segmentation result. (g) By Lucas-Kanade estimation. (h) Segmentation result.

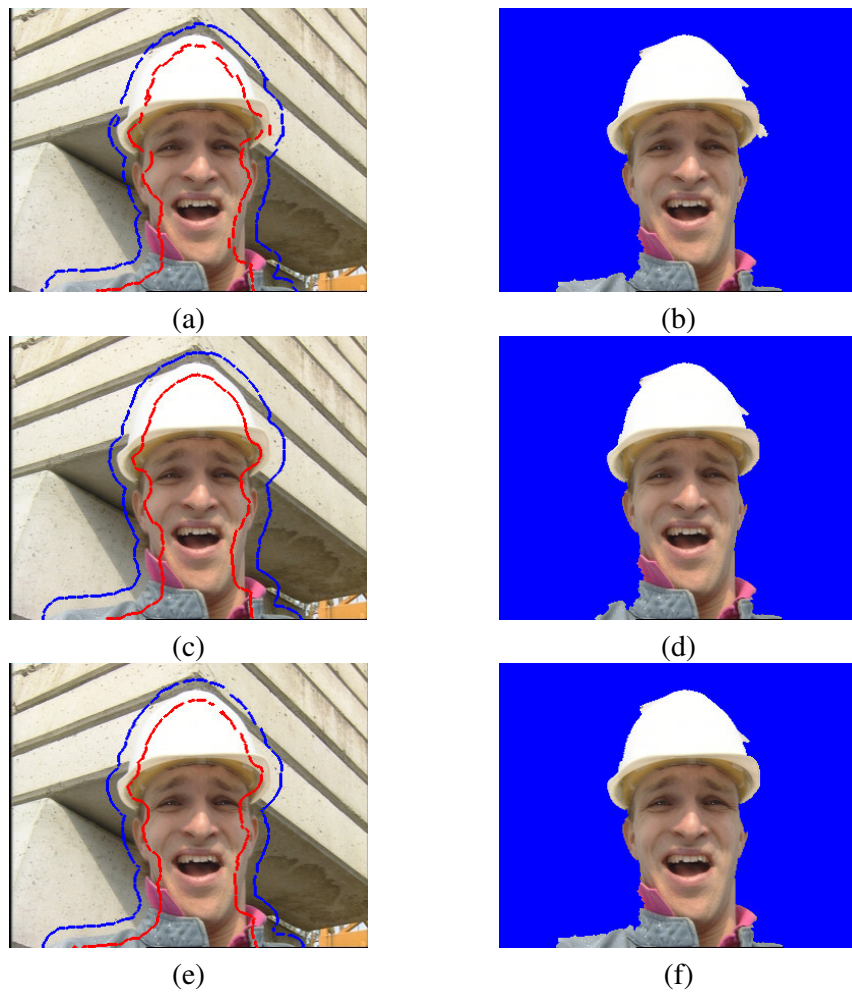


Fig. 3.13: Heuristics comparison - Applied to *Foreman* sequence. (a) No heuristics. (b) Segmentation result. (c) Local averaging of the displacement vectors. (d) Segmentation result. (e) Binding of markers. (f) Segmentation result.

video coding [69, 65, 68] to sequence segmentation techniques. Unfortunately, it is hard to do a fair comparison between our method and many other important ones that segment *Foreman* because their great majority consists in automatic segmentation methods.

Foreman is a very interesting sequence because it presents many important test situations (Fig. 3.14). The foreman himself, for instance, while he appears in the scene, moves his head in many directions (left, right and forward and backward, what provides a zooming sensation of the foreman face). Foreman head sometimes rotates, what gives a kind of deformation of the object of interest. Camera is also moving in a smooth but uncontrolled way. If the foreman is the object of interest, note that he is composed by several objects that need to be segmented and tracked: the helmet, his face and jacket. Besides, the sequence presents several regions of low contrast such as the foreman shoulder, his ears and the region where helmet meets the white concrete background. *Foreman* sequence is not a trivial task to segment appropriately.

The experiment consists in evaluating three methods applied to segment the man in the first 150 frames of *Foreman*:

- The manual method;
- The watershed from propagated markers with binding of markers heuristic and **no propagation** (i.e., the markers are bound but are not propagated; they just stay where they are imposed).
- The watershed from propagated markers with binding of markers heuristic and **Lucas-Kanade motion estimator**;

The reason to apply the second method is simple: the experiment also aims to evaluate the importance of the marker propagation according to the motion of the border. The manual method was applied to create the ground truth, and it was also assessed in order to demonstrate the accuracy of the watershed from propagated markers. The interface is the same to all segmentation methods: again, the user may insert markers as points and/or line segments with the mouse, and select markers for removal as well. Each interference is followed by the application of classical watershed from markers in order to update the segmentation. Second and third experiments ran with parameters $m = 10$ and $w = 10$.

Figure 3.15 presents the motion information of *Foreman* sequence, given its ground truth segmentation. Foreman appears to move in a reasonable velocity in almost the entire sequence (note that the difference between consecutive frames is about 1.5% in most frames. The mean error between consecutive frames is 1.7652%). The exceptions are in the end of the sequence where the foreman appears to move quickly and in the frame **90**, where the hand of the foreman appears in the scene, disappearing in the next frame (Fig. 3.14). This fact can be also observed in the assessment of the second and the third methods - see Fig. 3.19 and Fig. 3.22).

The ground truth is given by the manual segmentation of *Foreman* sequence (Fig. 3.16). It was necessary 3990 user interferences - mean of 26.6 interferences for frame - to do the manual segmentation. The total time spent to do all segmentation was 12266 seconds = 3.4073 hours. Note in Fig. 3.17 that the number of interferences and the time spent for frame is higher at the end of the sequence, because the foreman moves more quickly in that instant (see Fig. 3.15).

The watershed from propagated markers - *without propagation* - provided the segmentation results shown in Fig. 3.18. The application of this segmentation method required 98 interventions (mean of

0.6533 interventions for frame) and it was done in 851.72 seconds = 14.1953 minutes (mean of 5.6781 seconds each frame). Figure 3.19 (a) and (b), show, respectively, the number of interference and the segmentation time for each frame. Note in Fig. 3.19 (a) that there are a few intervals that did not need intervention. In addition, see the critical region at the end of both graphics where the foreman sped up. Note the amount of intervention needed in that interval and the time spent to do such intervention.

Figure 3.20 shows the segmentation error due the application of the watershed from propagated markers - *without propagation*. Besides bad segmentation may occur due the misplacement of the initial markers imposed to each frame, the segmentation of foreman is critical at some places. As stated above, there are several regions where the gradient is low and the watershed fails to segment correctly the object in that regions. Regions as the helmet, ears and shoulders require several interventions to be correctly segmented; if such interventions are minimal, as in both experiments with watershed from propagated markers shown in this section, the object is still segmented and tracked but segmentation error persists at some points. The mean segmentation error was 2.0769% for each frame.

The last experiment was done with the application of the watershed from propagated markers - *Lucas-Kanade motion estimator* - which results are shown in Fig. 3.21. Figure 3.22 (a) and (b) show, respectively, the amount of interference required and the time spent to segment the foreman in each frame. The last experiment required 80 interventions (mean of 0.5333 interventions for frame) and it was accomplished in 1001.6 seconds = 16.6927 minutes (mean of 6.6771 seconds each frame). Again, note the intervals in the sequence where intervention was not needed (Fig. 3.22 (a)). Also note that the segmentation of the foreman at the end of the sequence is still critical and required a lot of intervention.

Segmentation error for each frame, due to the Lucas-Kanade motion estimation improvement, is shown in Fig. 3.22. Error due the misplacement of propagated markers was reduced and the segmentation of some critical regions was improved, but there still appeared some bad segmentation in several critical points that required intervention. Mean segmentation error occurred due application of the watershed from propagated markers - *Lucas-Kanade motion estimator* - was 1.6434% for each frame.

Comparing the two approaches using watershed from propagated markers (both with parameters $m = 10$ and $w = 10$), we see that the approach that did not propagated markers ran faster than the Lucas-Kanade propagated one (see both time graphics in Fig. 3.19 (b) and Fig. 3.22 (b)); the last approach still needs to compute the propagation when the segmentation of the current frame ends and it does not occur in the no-propagated version. However, the approach that applies the Lucas-Kanade estimation provided a lower number of interferences and a lower segmentation error for frame. Note that the segmentation error due the misplacement of the propagated markers is lower in the Lucas-Kanade supported method, since the markers are propagated in order to follow the motion of the border; a source of segmentation error in the no-propagated version is the fact that the borders usually move toward the static markers and it may make them misplaced according to their type of marker, internal or external one.

3.9.4 Parameters

In this subsection, we discuss the choice of the parameters required by the "binding of markers" improvement heuristics combined to the Lucas-Kanade estimator. The parameters are the length of



Fig. 3.14: *Foreman* sequence (frames 1 to 150).

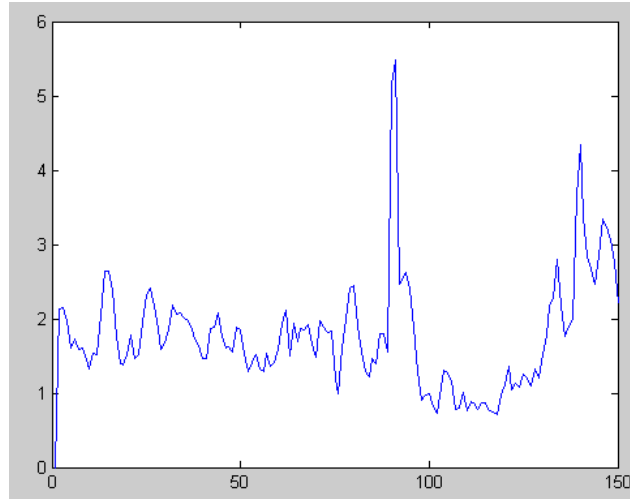


Fig. 3.15: Motion information from *Foreman* sequence.

each markers ($\mathbf{m}$) and the distance between the marker and the center of the region delimited by the pair of markers ($\mathbf{w}$). Three experiments were done: on the first one, the influence of the distance between the markers was studied, by assuming a constant length to the markers ($\mathbf{m} = 5$) and testing several distance values. The second experiment consists in studying the influence of the length of the markers, by assuming a constant distance between the marker and the center of the region delimited by the pair ($\mathbf{w} = 5$) and applying several length values. The third experiment consists in the application of the benchmark proposed above to evaluate the impact of the choice of parameters,

The experiments in this subsection illustrate the trade-off in the choice of both parameters. Segmentation results and the performance of the marker propagation are involved in this trade-off.

The suitable choice of the length and the distance values is important, since the pair of markers defines the area that will be applied to compute the displacement vector to that pair. Besides, for some images, the choice of both parameters is decisive when aiming a local improvement of the segmentation results.

Given a pair of markers, the greater the area between these markers, the greater the amount of information available to do the motion estimation, and thus, the better both markers are propagated. The gain of information is achieved by increasing the length of markers and/or the distance between them. However, the farther the markers are from the object borders, the worst is the segmentation results. Some parts of the object may not receive internal markers when the markers are imposed far from the object borders (as seen in the experiments described below).

The proposed method can be applied to segment objects with fast motion in the sequence, by choosing a distance between the markers $\mathbf{w}$ sufficient to compute the velocity of the border. However, as stated above (and illustrated below), great distance parameters may lead to bad segmentation results. One way to handle with this problem is to rely on a low displacement between consecutive frames, in order to segment fast moving objects more accurately.

Although the imposition of markers close to the border provides a better segmentation, the closeness of markers to the borders may not provide a good segmentation. Note that, if the markers in the pair are close and their length are short, the amount of information located in the area delimited by



Fig. 3.16: Manual segmentation (Ground Truth) of *Foreman* sequence (frames 1 to 150).

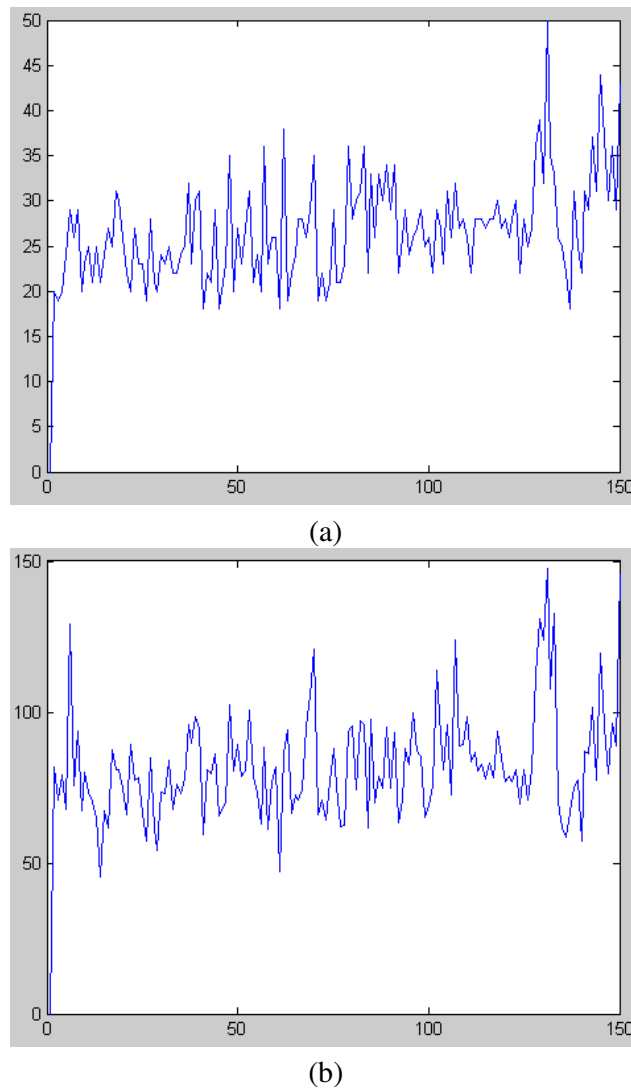


Fig. 3.17: Analysis of the Ground Truth segmentation (manual one) of the *Foreman* sequence. (a) Number of user interferences in each frame. (b) Time spent in the edition of each frame (in seconds).

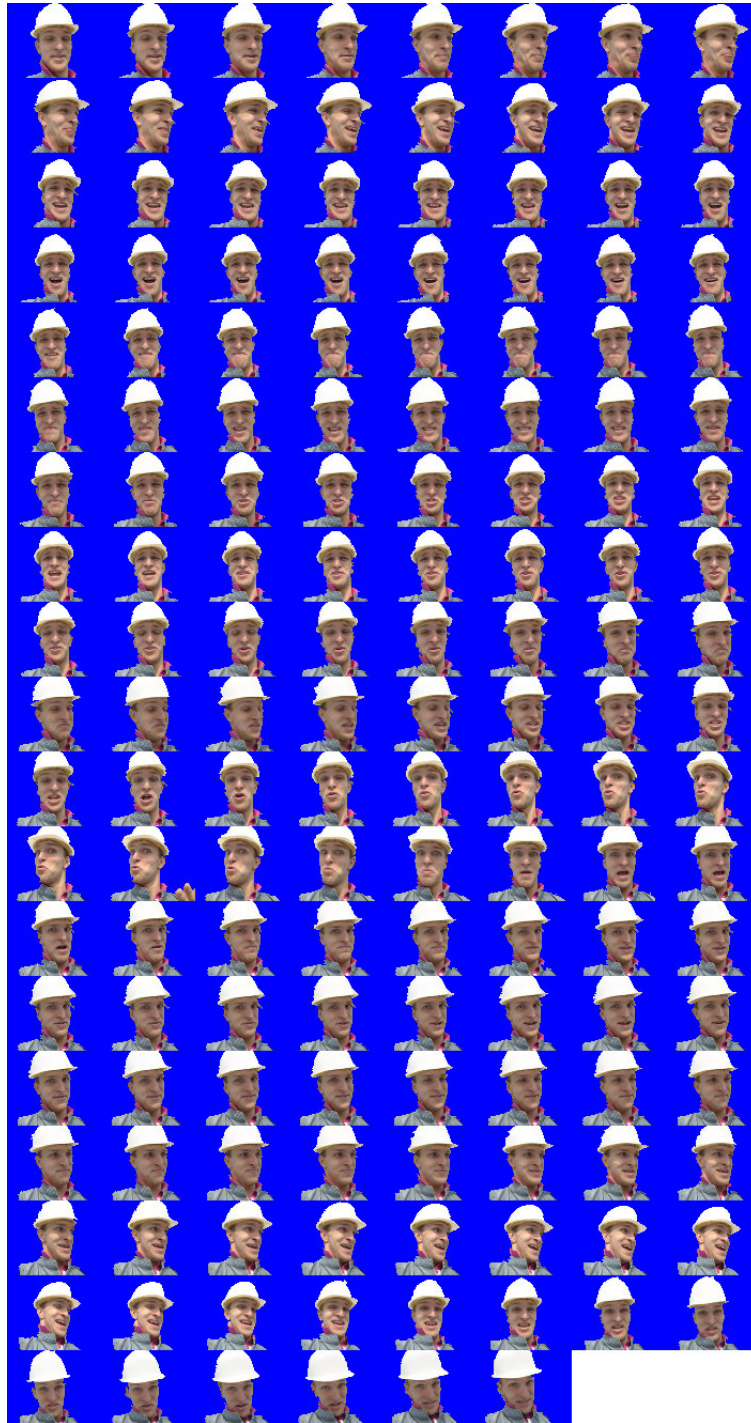


Fig. 3.18: Segmentation of *Foreman* sequence by Watershed from Propagated Markers (**No propagation**. Frames 1 to 150).

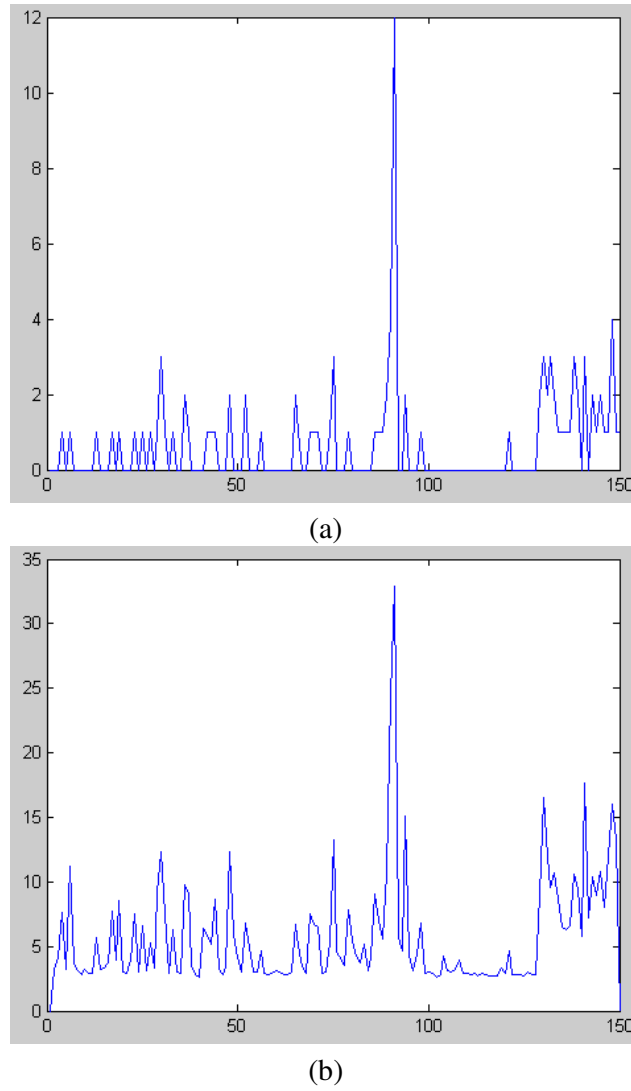


Fig. 3.19: Analysis of the segmentation of *Foreman* sequence by Watershed from Propagated Markers (**No propagation**). (a) Number of user interferences in each frame. (b) Time spent in the edition of each frame (in seconds).

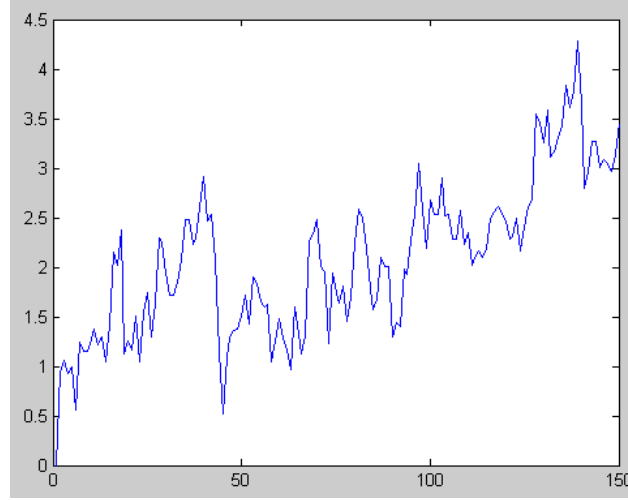


Fig. 3.20: Analysis of the segmentation of *Foreman* sequence by Watershed from Propagated Markers (**No propagation**): Segmentation Error in each frame (Compared to the Ground Truth.)

them may not be enough to compute accurately the motion of the markers (see below).

Figure 3.24 shows the results of the first experiment ($m = 5$). Note that, the farther the markers are from the borders, the better the motion is estimated (Figure 3.24 (a) shows a few misplaced pairs due the lack of information in these pairs to compute their motion, and thus, the segmentation in their regions is flickered). However, the farther the markers, the worst the segmentation results (Note that, in Fig. 3.24 (e) and Fig. 3.24 (g), some parts of the object, such as foreman's shoulders, are lost). In this experiment, markers imposed 10 pixels away from the object borders (Fig. 3.24 (c)) provided the best segmentation result.

Parameters	Total	Mean for Frame
$m = 5, w = 5$	272	1.8133
$m = 5, w = 10$	160	1.0667
$m = 5, w = 15$	440	2.9333
$m = 5, w = 20$	618	4.1200
$m = 10, w = 5$	178	1.1867
$m = 10, w = 10$	80	0.5333
$m = 15, w = 5$	125	0.8333
$m = 20, w = 5$	104	0.6933

Tab. 3.1: Quantitative Evaluation of the Choice of Parameters: Amount of Intervention.

The second experiment is illustrated in Fig. 3.25 ($w = 5$). Note that all markers are imposed close to the object borders, and thus, it is expected good segmentation results in all cases. However, notice that the quality of the segmentation is improved when the length of the markers is increased. Again, it occurs that, if the length of markers is increased, it is also increased the area delimited by the pair of markers and the propagation is computed more accurately.

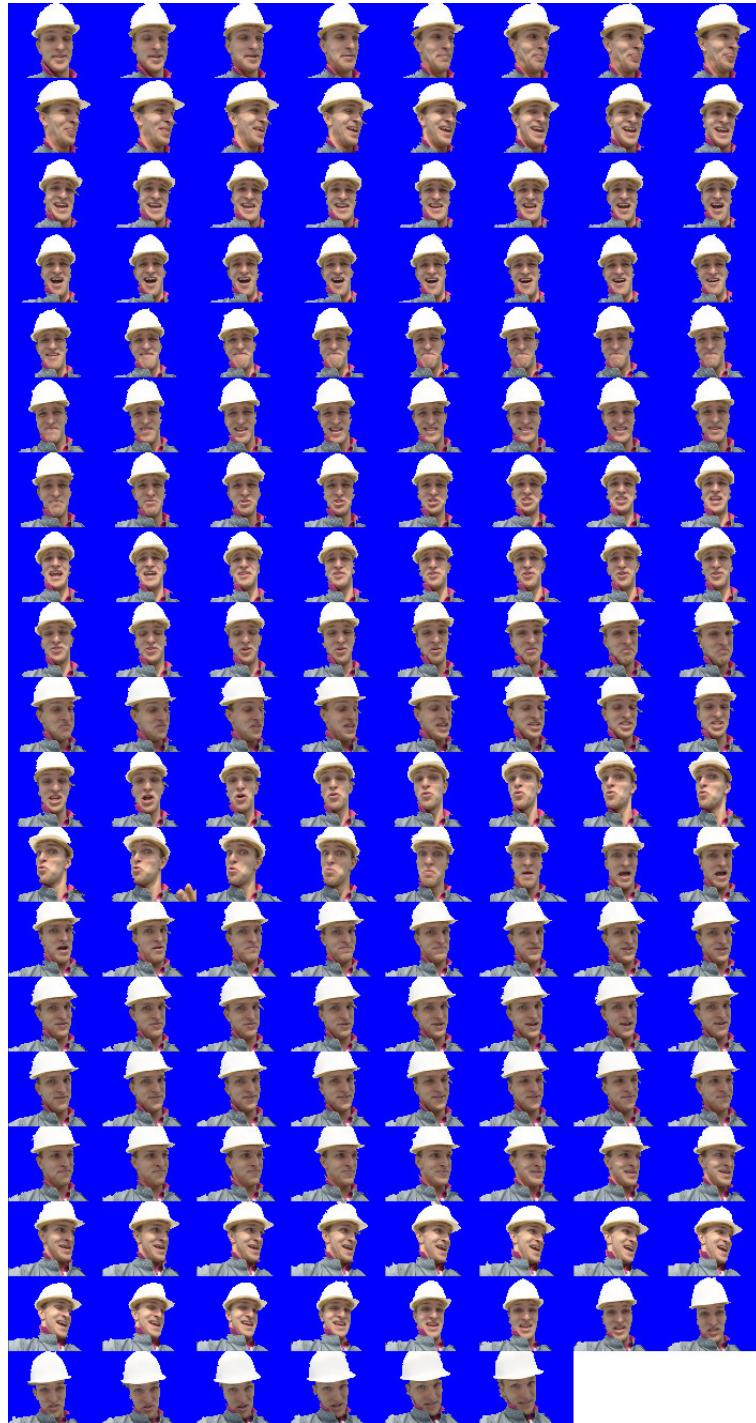


Fig. 3.21: Segmentation of *Foreman* sequence by Watershed from Propagated Markers (**Lucas-Kanade**. Frames 1 to 150).

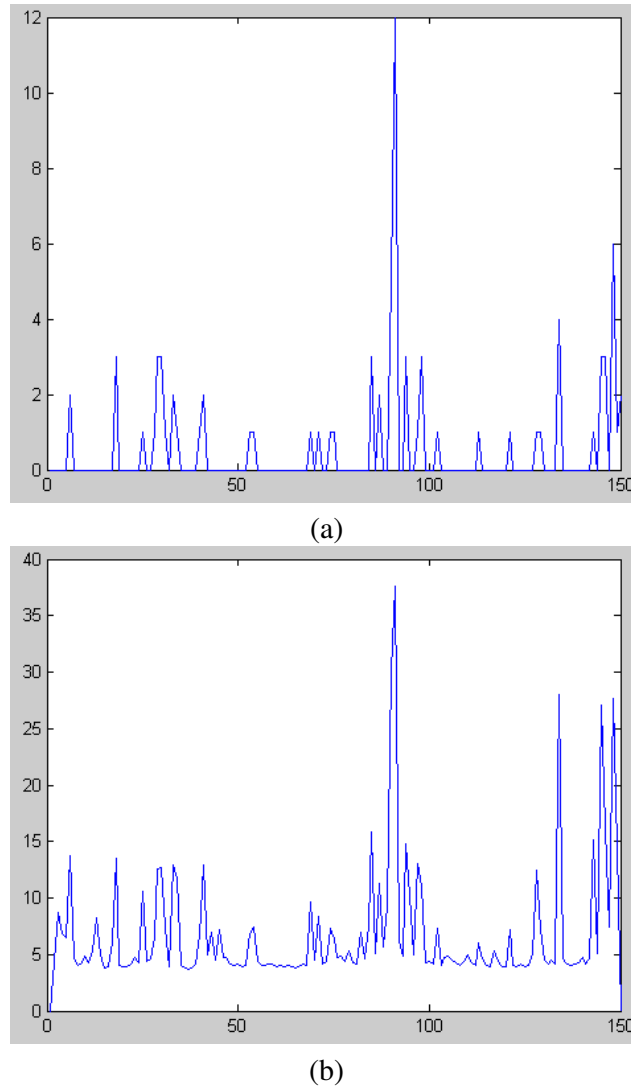


Fig. 3.22: Analysis of the segmentation of *Foreman* sequence by Watershed from Propagated Markers (**Lucas-Kanade**). (a) Number of user interferences in each frame. (b) Time spent in the edition of each frame (in seconds).

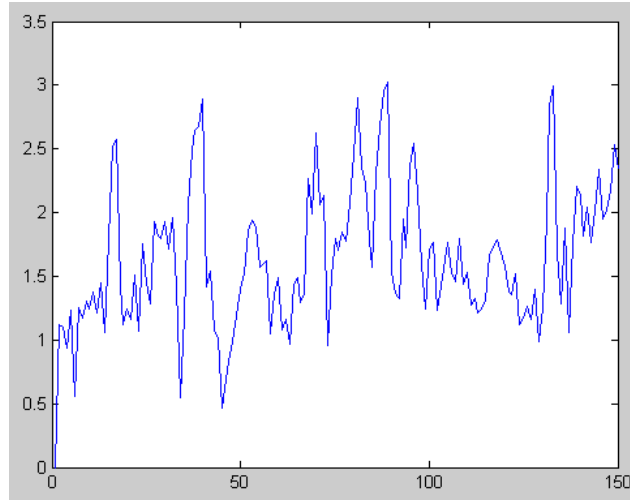


Fig. 3.23: Analysis of the segmentation of *Foreman* sequence by Watershed from Propagated Markers (**Lucas-Kanade**): Segmentation Error in each frame (Compared to the Ground Truth.)

Parameters	Total (seconds)	Total (minutes)
$\mathbf{m} = 5$, $\mathbf{w} = 5$	1817.3	30.2885
$\mathbf{m} = 5$, $\mathbf{w} = 10$	1462.2	24.3703
$\mathbf{m} = 5$, $\mathbf{w} = 15$	2112.9	35.2154
$\mathbf{m} = 5$, $\mathbf{w} = 20$	2509.1	41.8190
$\mathbf{m} = 10$, $\mathbf{w} = 5$	1251.4	20.8568
$\mathbf{m} = 10$, $\mathbf{w} = 10$	1001.6	16.6927
$\mathbf{m} = 15$, $\mathbf{w} = 5$	941.374	15.6896
$\mathbf{m} = 20$, $\mathbf{w} = 5$	833.813	13.8969

Tab. 3.2: Quantitative Evaluation of the Choice of Parameters: Time Spent to Accomplish the Segmentation Task.

The last experiment in this subsection evaluates the influence of the choice of parameters to the watershed from propagated markers. *Foreman* sequence was segmented using several choices of $\mathbf{m}$ and $\mathbf{w}$. Tables 3.1, 3.2 and 3.3 show the statistics taken from eight test cases. The cases shown in Fig. 3.24 and 3.25 were tested. Test case ($\mathbf{m} = 10$ and $\mathbf{w} = 10$) was also tested, since it was seen empirically as a good choice of parameters.

The test cases and the collected statistics provided by the application of the benchmark illustrate some features of the watershed from propagated markers:

- Short markers require many interventions to be removed when bad propagation occurs in a given region;
- Short markers provided more regions to be propagated (i.e., more displacement vectors to be computed);

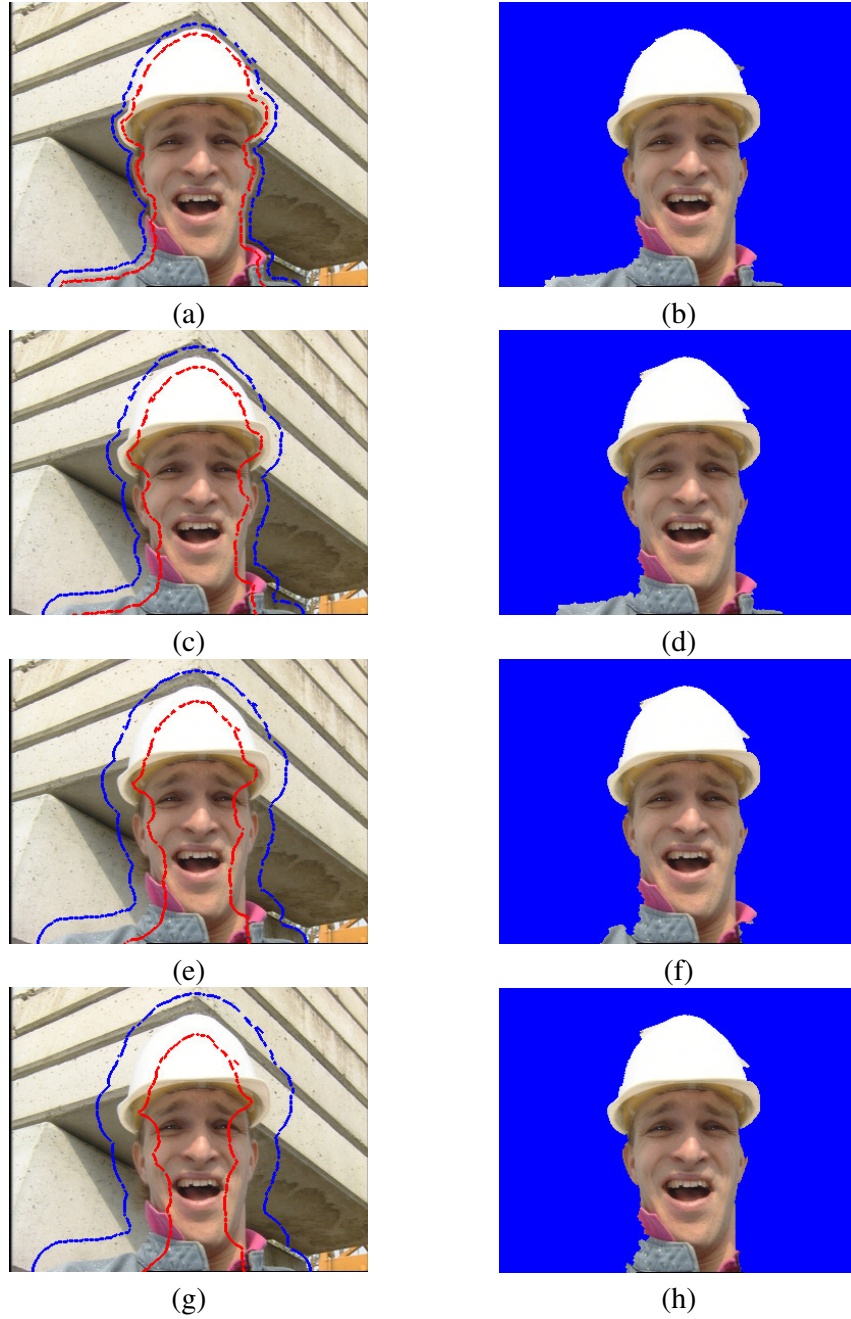


Fig. 3.24: Parameters: assuming a constant length marker ($\mathbf{m} = 5$). (a) $\mathbf{w} = 5$. (b) Segmentation result. (c) $\mathbf{w} = 10$. (d) Segmentation result. (e) $\mathbf{w} = 15$. (f) Segmentation result. (g) $\mathbf{w} = 20$. (h) Segmentation result.

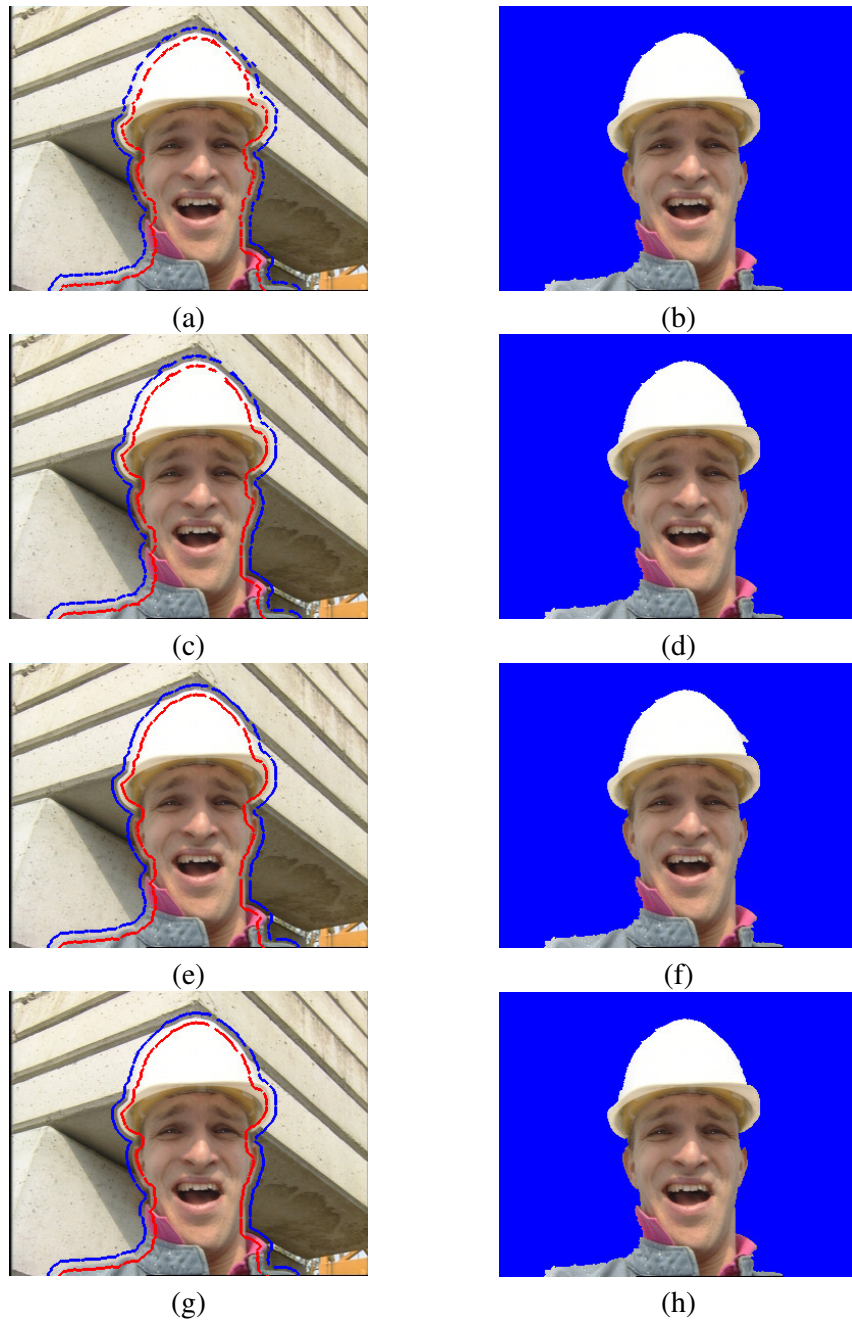


Fig. 3.25: Parameters: assuming a constant marker distance ($w = 5$). (a) $m = 5$. (b) Segmentation result. (c) $m = 10$. (d) Segmentation result. (e) $m = 15$. (f) Segmentation result. (g) $m = 20$. (h) Segmentation result.

Parameters	Mean Time Spent	Mean Segmentation Error
$\mathbf{m} = 5, \mathbf{w} = 5$	12.1154	1.2888%
$\mathbf{m} = 5, \mathbf{w} = 10$	9.7481	1.9046%
$\mathbf{m} = 5, \mathbf{w} = 15$	14.0862	1.8617%
$\mathbf{m} = 5, \mathbf{w} = 20$	16.7276	1.8736%
$\mathbf{m} = 10, \mathbf{w} = 5$	8.3427	1.6061%
$\mathbf{m} = 10, \mathbf{w} = 10$	6.6771	1.6434%
$\mathbf{m} = 15, \mathbf{w} = 5$	6.2758	1.7208%
$\mathbf{m} = 20, \mathbf{w} = 5$	5.5588	1.6712%

Tab. 3.3: Quantitative Evaluation of the Choice of Parameters: Mean Time Spent for Frame (in seconds) and Mean Segmentation Error

- Segmentation results require more intervention when markers are farther from the borders;
- Test cases that required more interference also took more time to be done;
- Combination of lengthy markers (lesser markers to be edited) and short distances (markers close to the border provide better segmentation) seem to be a good choice;
- Test case ($\mathbf{m} = 10$ and $\mathbf{w} = 10$) took more time to be finished than test cases ($\mathbf{m} = 15$ and $\mathbf{w} = 5$) and ($\mathbf{m} = 20$ and $\mathbf{w} = 5$). However, test case ($\mathbf{m} = 10$ and $\mathbf{w} = 10$) required less intervention and achieved a lower segmentation error. Parameters ($\mathbf{m} = 10$ and $\mathbf{w} = 10$) is a good choice for the segmentation of foreman in the homonymous sequence.

It could be appealing to choose lengthy markers and a short distance from them to the borders. However, there are situations when you may need greater distances or a set of short markers. Great distances from the border allow the tracking of objects with faster motion. Short markers are useful to segment deformable objects.

3.10 Conclusion

This work introduces an interactive method to object segmentation in image sequences, by combining classical morphological segmentation with motion estimation. Except for the I-frame, where the objects of interest are segmented manually, the starting segmentation of the objects in the current frame is computed in function of a set of markers given by the segmentation of the same objects in the previous frame. After the interactive addition/removal of markers to fix some bad segmentation, the desired segmentation is obtained to the current frame and it is used to compute markers to be propagated to the next frame. Besides the interactivity and the progressive manual edition, this method also presents two other main characteristics: the generality and the rapid response.

The watershed from propagated markers consists in computing short segments close to the object borders and apply them as markers propagated to the next frame. This method avoids some problems found when dealing with borders and the deformation of the object of interest. The manual edition of the markers is easy and their propagation is very fast.

Since the watershed from markers technique shows itself a good solution to the segmentation step, some efforts were done in the investigation of alternative ways to do motion estimation. Classical optical flow estimation techniques to do marker propagation were investigated and also some heuristics were developed to boost the result of the motion estimator. Several experiments were done to compare the performance of the optical flow estimators and to compare the results of the heuristics as well.

The experiments led us to choose the combination of Lucas-Kanade estimator with the binding of markers heuristics as the technique to propagate the markers; Lucas-Kanade assumption that the motion field is assumed constant over a region matches to our object segmentation framework, since a marker can be considered as such region and all of its points must be translated by the same displacement vector. In addition, the results achieved by this estimator are very good. The adding of the bound markers constraint improved the results, since it provides more information to estimate the displacement vectors and avoid problems as the locally crossing of inner and outer markers.

It was also introduced in this paper a new benchmark to quantitative evaluation of assisted object segmentation in image sequences. This benchmark aims to assess robustness of the method and the amount of intervention required to segment an object. The measurements of performance were applied to evaluate the segmentation of *Akiyo* and *Foreman* sequences. These experiments illustrate the robustness and the easiness of segmentation provided by the watershed from propagated markers.

Several experiments were also done in order to test the watershed from propagated markers with Lucas-Kanade estimator and the heuristics of binding of markers [26]. The influence of the choice of parameters was studied. It was shown that there is a trade-off in the choice of the length of markers and the distance between the marker in the pair. This trade-off is related to the segmentation results and the performance of the marker propagation.

Future works include improvements in the computation of propagation vectors and correction of bad segmentation by locally weighting the gradient or tightening the pair of markers to the local border of the object.

3.11 Acknowledgements

Thanks to Leticia Rittner (PhD student at DCA-FEEC-UNICAMP, Brazil) for valuable discussions about Lucas-Kanade estimation.

Capítulo 4

Watershed from Propagated Markers Improved by the Combination of Spatio-Temporal Gradient and Binding of Markers Heuristics

Como foi explicado no capítulo 1, o estudo iniciado sobre a aplicação de *watershed* interativo 3-D para segmentação de objetos em sequências de imagens levou à observação de que o uso de um gradiente espaço-temporal proporcionava melhores resultados de segmentação, além de reduzir o número de interferências do usuário. O *watershed* 3-D não foi utilizado, mas a informação do gradiente espaço-temporal mostrou-se de grande utilidade.

Nesta proposta de melhoria do *framework* de segmentação de objetos em sequências de imagens, o gradiente é calculado somente uma vez para o volume de imagens e, ao invés de aplicar-se o *watershed* em todo o gradiente volumétrico, aplica-se somente em uma fatia deste volume, referente ao quadro que está sendo processado. Cada uma destas fatias contém informação espaço-temporal calculada em função de um elemento estruturante tridimensional.

O uso de tais fatias permitiu uma segmentação mais robusta dos objetos de interesse em cada quadro. A melhoria foi obtida não só nos resultados da segmentação em si mas também na diminuição do número de intervenções do usuário para segmentar tais objetos. Tais melhorias no paradigma do *watershed* com marcadores propagados justificou a elaboração do artigo que compõe este capítulo, bem como sua importância no corpo desta tese.

Este capítulo apresenta o artigo “*Watershed from Propagated Markers Improved by the Combination of Spatio-Temporal Gradient and Binding of Markers Heuristics*” [18], aceito para publicação nos anais do VISAPP 2010, em Angers, na França.

4.1 Abstract

This paper presents the improvement of the watershed from propagated markers, a generic method to interactive segmentation of objects in image sequences, by the inclusion of a spatio-temporal gradient to the segmentation framework. Segmentation is done by applying the watershed from markers

to a gradient image extracted from the spatio-temporal gradient sequence and using markers provided by the binding of markers heuristics. The performance of the improved method is demonstrated by application of a benchmark that supports a quantitative evaluation of assisted segmentation of objects in image sequences. Experimental results provided by the combination of spatio-temporal gradient with the binding of markers heuristics show that the proposed improvement can decrease the number of human interferences and the time required to process the sequences.

4.2 Introduction

The assisted segmentation of objects in image sequences by application of an extension of the watershed transform to the 3-D case appears to be a good solution to this kind of problem. The watershed definition is extensible to the 3-D case by applying it to a spatio-temporal gradient computed considering the image deck as an image volume and using a 3-D structuring element. With well placed markers, the 3-D segmentation may provide very good results.

However, its application is not viable for interactive segmentation, for several reasons. First, the 3-D watershed application is very time consuming: the greater the image sequence, the longer the 3-D watershed takes to segment all sequence. More, it is even worst if this segmentation process needs to be achieved every time the user makes an interference. Second, despite the good results 3-D watershed may provide, it is also the source of a kind of interframe error segmentation: the 3-D segmentation of an object may flow back and forth in the sequence to regions that does not represent the object in other instant times. Third, this process is not a progressive edition: every time a user makes an interference, the segmentation in the sequence needs to be whole inspected.

It could be interesting to exploit the good side of the 3-D assisted morphologic segmentation presented above avoiding those side effects. The idea is to include the spatio-temporal gradient, that holds spatio-temporal information from the image sequence, to the *watershed from propagated markers*, a generic method to interactive segmentation of objects in image sequences [4, 17]. This method consists in a combination of the watershed from markers [42, 70] with motion estimation techniques [59, 5]. The segmentation technique is tied to the motion estimation one, since the markers to the objects of interest are propagated to the next frames in order to track such objects.

The *binding of markers* [26] is an improvement to the watershed from propagated markers. It consists in to compute pairs of markers along the border, and each pair is composed by an internal marker and an external one. Both markers in the pair must be propagated by the same displacement vector, and this vector is computed by the motion estimation of the area *between* the pair of markers. This heuristics provides more information to the motion estimation and helps the motion of the pair of markers to follow the motion of the border that crosses the region between them in the previous frame.

This paper introduces the combination of spatio-temporal gradient with the binding of markers heuristics, in the watershed from propagated markers context. The segmentation of a given frame is done by applying the watershed from markers to a gradient extracted from a spatio-temporal gradient sequence and using markers provided by the binding of markers heuristics.

Experiments were done in order to demonstrate the performance of such combination. The demonstration was done by application of a benchmark that supports a quantitative evaluation of assisted segmentation of objects in image sequences [25]. The benchmark application showed a sig-

nificant decrease in the number of interferences and a increase in the segmentation results.

The paper is organized as follows: Section 4.3 presents some preliminary concepts. Section 5.5 reviews the watershed from propagated markers framework and the binding of markers heuristics and presents the spatio-temporal gradient added to the segmentation framework. Section 4.5 shows and discusses some experimental results and, finally, Section 4.6 concludes the paper with a brief discussion.

4.3 Preliminary Review

Let us define some sets used in the formalizations introduced along the paper. Let $E = \mathbb{Z} \times \mathbb{Z}$ be an *image domain*. Let $F = E \times \{1, 2, \dots, n\}$ be the *sequence domain*, where $n \in \mathbb{Z}_+$. Let $E = F$, when $n = 1$. Let $K = [0, k]$ be a totally ordered set and let $C = K \times K \times K$.

Let $Fun[E, K]$ be the set of all functions $gs : E \rightarrow K$, denoting the set of all *grayscale images*. Let $Fun[E, C]$ be the set of all functions $cs : E \rightarrow C$, that denotes the set of all *color (or multiband) images*. Let $Fun[F, K]$ be the set of all *grayscale sequences* and let $Fun[F, C]$ be the set of all *color sequences*.

Let $i \in \{1, 2, \dots, n\}$. Let $gs \in Fun[F, K]$ and $cs \in Fun[F, C]$. The set of all $gs(x, i)$, $\forall x \in E$ denotes the *i-th frame* of the grayscale sequence gs . The set of all $cs(x, i)$ denotes the *i-th frame* of the multiband sequence cs , $\forall x \in E$.

4.3.1 3-D structuring elements

Let us define the 3-D structuring elements (s.e.) that defines the 3-D connectivity in this paper:

$$\begin{aligned}
 B_5 &= \left[\begin{pmatrix} \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \end{pmatrix}, \begin{pmatrix} \cdot & \bullet & \cdot \\ \cdot & \underline{\bullet} & \cdot \\ \cdot & \bullet & \cdot \end{pmatrix}, \begin{pmatrix} \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \end{pmatrix} \right] \\
 B_6 &= \left[\begin{pmatrix} \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \end{pmatrix}, \begin{pmatrix} \cdot & \bullet & \cdot \\ \cdot & \underline{\bullet} & \cdot \\ \cdot & \bullet & \cdot \end{pmatrix}, \begin{pmatrix} \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \end{pmatrix} \right] \\
 B_{17} &= \left[\begin{pmatrix} \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \end{pmatrix}, \begin{pmatrix} \cdot & \bullet & \bullet \\ \cdot & \underline{\bullet} & \bullet \\ \cdot & \bullet & \bullet \end{pmatrix}, \begin{pmatrix} \cdot & \bullet & \bullet \\ \cdot & \bullet & \bullet \\ \cdot & \bullet & \bullet \end{pmatrix} \right] \\
 B_{26} &= \left[\begin{pmatrix} \cdot & \bullet & \bullet \\ \cdot & \bullet & \bullet \\ \cdot & \bullet & \bullet \end{pmatrix}, \begin{pmatrix} \cdot & \bullet & \bullet \\ \cdot & \underline{\bullet} & \bullet \\ \cdot & \bullet & \bullet \end{pmatrix}, \begin{pmatrix} \cdot & \bullet & \bullet \\ \cdot & \bullet & \bullet \\ \cdot & \bullet & \bullet \end{pmatrix} \right]
 \end{aligned}$$

Each s.e. is denoted by three 3×3 structures denoting a $3 \times 3 \times 3$ subspace. The center of this subspace is underlined. If the s.e. is centered at frame i , the first 3×3 structure touches frame $i - 1$,

the second structure touches frame i and the last one touches frame $i + 1$. A dot means that the point does not belongs to the s.e., while a bullet means that the point belongs to it.

The s.e. B_6 and B_{26} represents, respectively, the classical 6 and 26-connectivity. The first one is 4-connected in the frame where it is centered and touches the previous and next frames with one point connected to the center of the s.e. along the time axis. The 26-connect s.e. is a cube that touches the frame the s.e. is centered, the previous frame and the next one. The B_5 s.e. is similar to B_6 , except that B_5 does not have the point that touches the previous frame. The B_{17} s.e. is similar to B_{26} , except B_{17} does not have any points touching the previous frame: it is a $3 \times 3 \times 2$ parallelepiped s.e..

All of it means that the spatio-temporal gradient and the 3-D watershed computed using the B_6 and B_{26} s.e. consider three frames in their computations: the current frame, the previous and the next ones. When the cited methods applies B_5 and B_{17} to do the spatio-temporal segmentation, just the current and the next frames are considered.

4.3.2 Band sequence extraction

In order to compute the spatio-temporal gradient for color sequences, it is convenient to define an operator that receives a multiband sequence and the desired band and outputs the sequence of the selected band.

Let $\beta : Fun[F, C] \times \{1, 2, 3\} \rightarrow Fun[F, K]$ be the operator thta extracts the band sequence from a multiband sequence, given by,

$$\beta(cs, i)(x) = s_i,$$

where $cs(x) = (s_1, s_2, s_3), \forall cs \in Fun[F, C], \forall x \in F$.

4.3.3 Spatio-temporal gradient

Let $gs \in Fun[F, K]$. The operator $\nabla_{B_{3D}} : Fun[F, K] \rightarrow Fun[F, K]$ computes the graylevel 3-D morphological gradient, given by,

$$\nabla_{B_{3D}}(gs)(x) = \bigvee_{y \in B_{3D_x}} gs(y) - \bigwedge_{y \in B_{3D_x}} gs(y).$$

where B_{3D} is a 3-D s.e., $\forall x \in F$. B_{3D_x} is the s.e. B_{3D} translated to the point $x \in F$.

Let $cs \in Fun[F, C]$. The operator $\nabla_{B_{3D}}^c : Fun[F, C] \rightarrow Fun[F, K]$ computes the multiband 3-D morphological gradient, given by,

$$\nabla_{B_{3D}}^c(cs)(x) = \bigvee_{i \in \{1, 2, 3\}} \nabla_{B_{3D}}(\beta(cs, i))(x),$$

where B_{3D} is a 3-D s.e., $\forall x \in F$.

4.3.4 Sequence sampling

Let $\varsigma_i : Fun[F, K] \rightarrow Fun[E, K]$ be the operator that extracts a frame from a gs sequence, given by,

$$\varsigma_i(gs)(x, y) = gs(x, y, i),$$

where $(x, y) \in E$. This function is used to extract the gradient from the i -th frame of the spatio-temporal gradient image deck in order to apply the watershed from markers to it.

4.4 Watershed From Propagated Markers

The watershed from propagated markers [17] consists, basically, in the following steps:

1. The objects of interest are segmented by the interactive watershed from markers, in the initial frame.
2. The segmentation mask, given by the segmented objects, is broken in pieces closer to the border of the mask. These pieces are regions that form the set of inner markers.
3. In a similar way, the inner markers are computed, the background region is also broken in pieces closer to the border of the mask. The set of outer markers will be formed by these regions.
4. Each marker is propagated to the next frame by motion estimation [6, 61, 45]. These new set of inner and outer markers are used to apply the watershed technique to the next frame. Go to the next frame and apply the watershed from markers with the set of computed markers.
5. If necessary, the user interacts with the markers, doing the corrections by adding or moving markers.
6. If the sequence is not fully processed, then go to Step 2.

(In the first proposal of the watershed from propagated markers, the inner markers were computed (Step 2) by taking the contour of erosion of the mask and breaking it in short segments. Similarly, the outer markers were obtained (Step 3) by splitting the contour of erosion of the background region in short segments).

The method proposed above works fine with bad defined contours or strongly textured objects, since the markers are imposed close to the borders of the objects to be segmented. If the quality of segmentation is not approved in some frame, the user can easily move the short-segment marker. The marker propagation is very fast since each segment consists in a few points. Moreover, the contours follow the object deformation, since new markers are created from the segmentation of each frame. The object to be segmented is processed until the end of sequence or until it leaves the scene or be totally occluded. If is partially occluded, the user may intervene to regularize the process.

4.4.1 Binding of markers

The *binding of markers* [26] is an improvement to the watershed from propagated markers framework. It consists in to compute pairs of markers along the border, and each pair is composed by an internal marker and an external one. Both markers in the pair must be propagated by the same displacement vector, and this vector is computed by the motion estimation of the area *between* the pair of markers. Besides to provide more information to the motion estimation (both markers *and* the area between them), the binding of markers heuristics helps the motion of the pair of markers to follow the motion of the border that crosses the region between them in the previous frame.

4.4.2 Spatio-temporal gradient improvement

FOREMAN				
Method	# Interferences	Time (Secs.)	Time *	Segm. Error *
Manual	3990	12266	81.7733 secs.	-
Binding of Markers	80	1001.6	6.6771 secs.	1.6434 %
∇_{B_5} and $\mathcal{W}_{B_4}$	44	856.2670	5.7084 secs.	1.1340 %
∇_{B_6} and $\mathcal{W}_{B_4}$	46	883.7500	5.8917 secs.	1.2281 %
$\nabla_{B_{17}}$ and $\mathcal{W}_{B_8}$	67	924.4520	6.1630 secs.	1.4288 %
$\nabla_{B_{26}}$ and $\mathcal{W}_{B_8}$	51	869.9540	5.7997 secs.	1.7078 %
CARPHONE				
Method	# Interferences	Time (Secs.)	Time *	Segm. Error *
Manual	2533	10429	69.5282 secs.	-
Binding of Markers	90	1064.1	7.0962 secs.	0.78 %
∇_{B_5} and $\mathcal{W}_{B_4}$	35	802.2650	5.3484 secs.	0.9697 %
∇_{B_6} and $\mathcal{W}_{B_4}$	30	773.7030	5.1580 secs.	0.9144 %
$\nabla_{B_{17}}$ and $\mathcal{W}_{B_8}$	28	770.0460	5.1336 secs.	1.1171 %
$\nabla_{B_{26}}$ and $\mathcal{W}_{B_8}$	33	777.2950	5.1820 secs.	1.1625 %

Tab. 4.1: Quantitative assessment of binding of markers improvements. Applications to *Foreman* and *Carphone* sequences (150 frames each sequence) (* : Mean / Frame).

The watershed from markers applied to in Step 4 of the framework described in the beginning of this Section uses the markers provided by the binding of markers heuristics. Since the first proposal of the watershed from propagated markers, the watershed operator was always applied directly to the gradient computed from the frame to be segmented at instant i . Let s be the input sequence and $s(i)$ the frame to be segmented. If $s(i) \in Fun[E, K]$ (i.e., a grayscale image), the morphological gradient was computed by the classical way. If $s(i) \in Fun[E, C]$ (i.e., a color image), some metrics were applied to compute the color gradient [55, 56]. Anyway, the gradient computation was an intra-frame process, computed by using just the information contained in the frame of interest. It was not exploited any temporal information in the gradient computation.

The spatio-temporal gradient improvement consists in a 3-D morphological processing of the input sequence. Instead to compute the intra-frame gradient in Step 4 each time the watershed is applied to, the spatio-temporal gradient is computed at once, at the beginning of the watershed from propagated markers framework, before Step 1. Let s be an image sequence. The spatio-temporal gradient $tg \in Fun[F, K]$ is given by,

$$tg = \begin{cases} \nabla_{B_{3D}}^c(s), & \text{if } s \text{ is a color sequence,} \\ \nabla_{B_{3D}}(s), & \text{otherwise,} \end{cases}$$

where B_{3D} is a 3-D s.e. such as the examples in Section 4.3.

As stated above, the spatio-temporal gradient tg is a function from F to K , i.e., it is a grayscale sequence. The watershed transform could be applied to tg but, as discussed in the Introduction, it is an expensive process, propagates interframe error and requires an inspection in the whole segmentation sequence every time it is applied to. Even when the sequence was sampled, in previous experiments, with two or three consecutive frames, the interframe error worsened the segmentation result.

The solution is to sample just the i -th frame of the spatio-temporal gradient when it is desired to segment the i -th frame of the input sequence. The sample is an image in $Fun[E, K]$ but contains spatio-temporal information obtained in the spatio-temporal gradient computation. Since it is just a single frame, the interframe error propagation does not occur.

Let $tg \in Fun[F, K]$ be a spatio-temporal gradient. The gradient $g \in Fun[E, K]$ where the watershed from markers will be applied to in Step 4 of the watershed from propagated markers is given by

$$g = \varsigma_i(tg),$$

where i is the index to the i -th frame to be segmented.

4.5 Experimental Results

Two experiments were done in order to illustrate the gains of productivity and segmentation quality provided by the application of the spatio-temporal gradient to the binding of markers framework. First experiment shows the gains achieved to the first 150 frames of *Foreman* sequence. The second experiment shows the experimental results to the first 150 frames of *Carphone* sequence.

Both experiments apply an benchmark proposed in [25] to do a quantitative assessment of interactive object segmentation to image sequences. The benchmark evaluates several measurements such as quality of segmentation and time spent to complete the segmentation task. In that work, several applications of the binding of markers supported by the Lucas-Kanade [6] motion estimator were tested and compared to. The application that presented the best overall results was the one with parameters $\mathbf{m} = \mathbf{w} = 10$. These parameters and the cited motion estimator were used in all experiments done with *Foreman* and *Carphone* sequences in this paper.

The experiment, for each sequence, consists in to compare several applications of the binding of markers supported by the spatio-temporal gradient to the manual segmentation and to the simple binding of markers application (the one proposed in [26]). Table 4.1 shows the quantitative results for both sequences. Four 3-D s.e. were tested to compute the spatio-temporal gradients: B_5 , B_6 , B_{17} and B_{26} (see Section 4.3). When the s.e. used to compute the spatio-temporal gradient were B_5 or B_6 , the chosen connectivity for the watershed operator was 4 (the cross s.e.). When the spatio-temporal gradient was computed using B_{17} or B_{26} , the chosen connectivity for the watershed was 8 (the box s.e.). The manual segmentations were used as the ground-truth segmentations. The application of the spatio-temporal gradient is denoted in Table 4.1 by $\nabla_{B_{3D}}$ and the watershed by a connectivity s.e. is denoted by $\mathcal{W}_{B_c}$. In all experiments, an user interference consists in the addition of points or lines segments as markers or in the removal of misplaced markers.

The experiments show that the application of the spatio-temporal gradient provides a decrease in the number of interferences, also providing in some cases a improvement in the segmentation results.

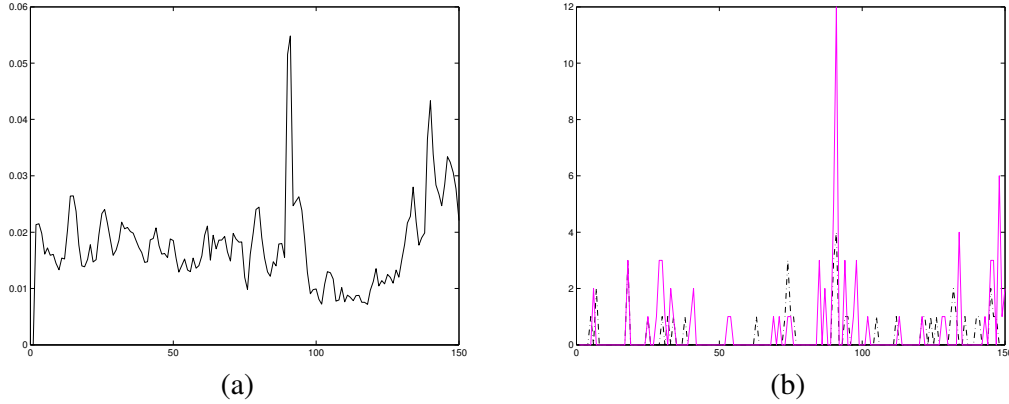


Fig. 4.1: Quantitative evaluation for *Foreman* sequence. (a) Motion information from *Foreman* sequence. (b) Number of interferences for each frame.

4.5.1 Foreman sequence

Several authors use the *Foreman* sequence to demonstrate the performance of their methods, from sequence segmentation techniques to video coding [69, 65, 68]. Unfortunately, the great majority of such methods is automatic and it is hard to do a fair comparison among these methods using the proposed benchmark since it does not apply to them.

Foreman is a very interesting sequence because it presents many important test situations. The foreman himself, for instance, while he appears in the scene, moves his head in many directions (left, right and forward and backward, what provides a zooming sensation of the foreman face). Foreman head also sometimes rotates, what gives a kind of deformation of the object of interest. Camera is also moving in a smooth but uncontrolled way. If the foreman is the object of interest, note that he is composed by several objects that need to be segmented and tracked: the helmet, his face and jacket. More, the sequence presents several regions of low contrast such as the foreman shoulder, his ears and the region where helmet meets the white concrete background. *Foreman* sequence is not a trivial task to segment appropriately.

Table 4.1 - Part One shows the evaluation of six experiments: the ground-truth provided by manual segmentation, the simple binding of markers and four applications of the binding of markers supported by spatio-temporal gradients. Except for the third spatio-temporal case ($\nabla_{B_{17}}$ and $\mathcal{W}_{B_8}$), all spatio-temporal cases required about a half of user effort to complete the task, compared to the simple binding of markers framework. And, except for the forth spatio-temporal case ($\nabla_{B_{26}}$ and $\mathcal{W}_{B_8}$), all spatio-temporal cases provided a better segmentation result, again compared to the simple binding of markers approach. Note that all spatio-temporal approaches were faster than the simple binding of markers to segment the foreman.

Figures 4.1 and 4.2 show some graphical information about *Foreman* sequence and the application of the binding of markers in two versions: the simple one and the improved by the spatio-temporal gradient - this one, using ∇_{B_5} and $\mathcal{W}_{B_4}$ as s.e.. Figure 4.1 (a) shows the amount of motion in *Foreman* sequence (computed by the symmetrical difference between consecutive frames) and Fig. 4.1 (b) shows the number of user interferences for the two examples: solid magenta plotting shows the

number of interferences required in the simple binding of markers application and the dot-dashed black plotting shows the number of interferences for the spatio-temporal case.

Figure 4.2 (a) shows the plotting of the time needed to segment each frame for the two example cases (solid magenta plotting for the simple case, dot-dashed black plotting for the spatio-temporal one). Note that the amount of efforts and time shown, respectively in Fig. 4.1 (b) and Fig. 4.2 (a) are consistent with the plotting with the image motion in Fig. 4.1 (a). Finally, Fig. 4.2 (b) shows the plotting of the segmentation error for each frame. In overall, the spatio-temporal case had a lower error in most of the cases.

Figure 4.3 shows the segmentation of *Foreman* sequence using ∇_{B_5} and $\mathcal{W}_{B_4}$. It is shown the segmentation of the first 150 frames of the sequence.

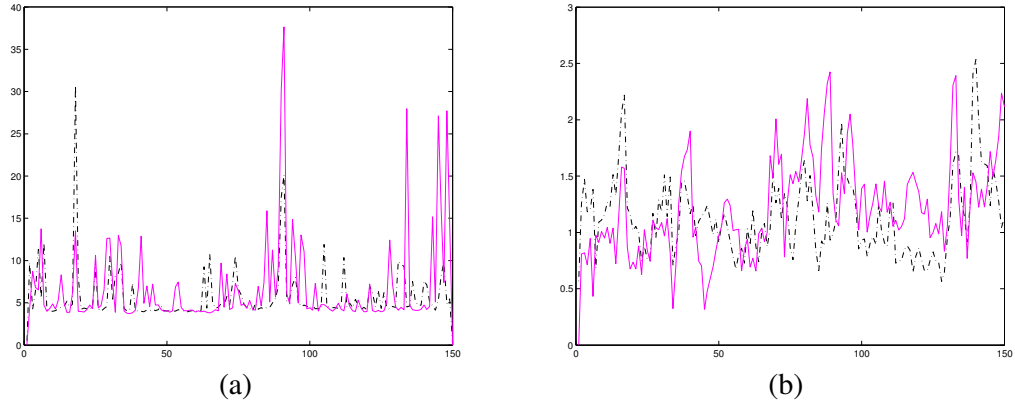


Fig. 4.2: Quantitative evaluation for *Foreman* sequence. (a) Time spent in each frame (secs.). (b) Segmentation error (Percentile. Compared to the ground-truth).

4.5.2 Carphone sequence

The segmentation of the man in *Carphone* sequence is also a difficult task. *Carphone* is also a very used sequence in the demonstration of image sequence processing techniques. The goal in this experiment was to segment the passenger from the remain of the scene. The passenger moves his body in several directions through the sequence, rotating slightly the body and approaching the face to the camera in a given instant. There are moments when the separation of the left shoulder and the ear regions from the background is difficult. More, there is also movement outside the car window. The passenger is composed by several objects: his head, which has difficult regions to segment and his hair, and the body, that is a easiest part to segment. However, the region where the left shoulder meets the window has a low contrast and a strong source of segmentation error.

Table 4.1 - Part Two shows the segmentation evaluation of six experiments, the same applied to the *Foreman* sequence. Note that all mean segmentation errors were close to 1%. The common binding of markers provided the best mean segmentation result, but not too much better than the ones provided by the spatio-temporal supported applications. Note, however, that the number of interferences decreased drastically in the spatio-temporal gradient approaches: simple binding of markers approach required 90 interferences. The spatio-temporal approaches required a third less

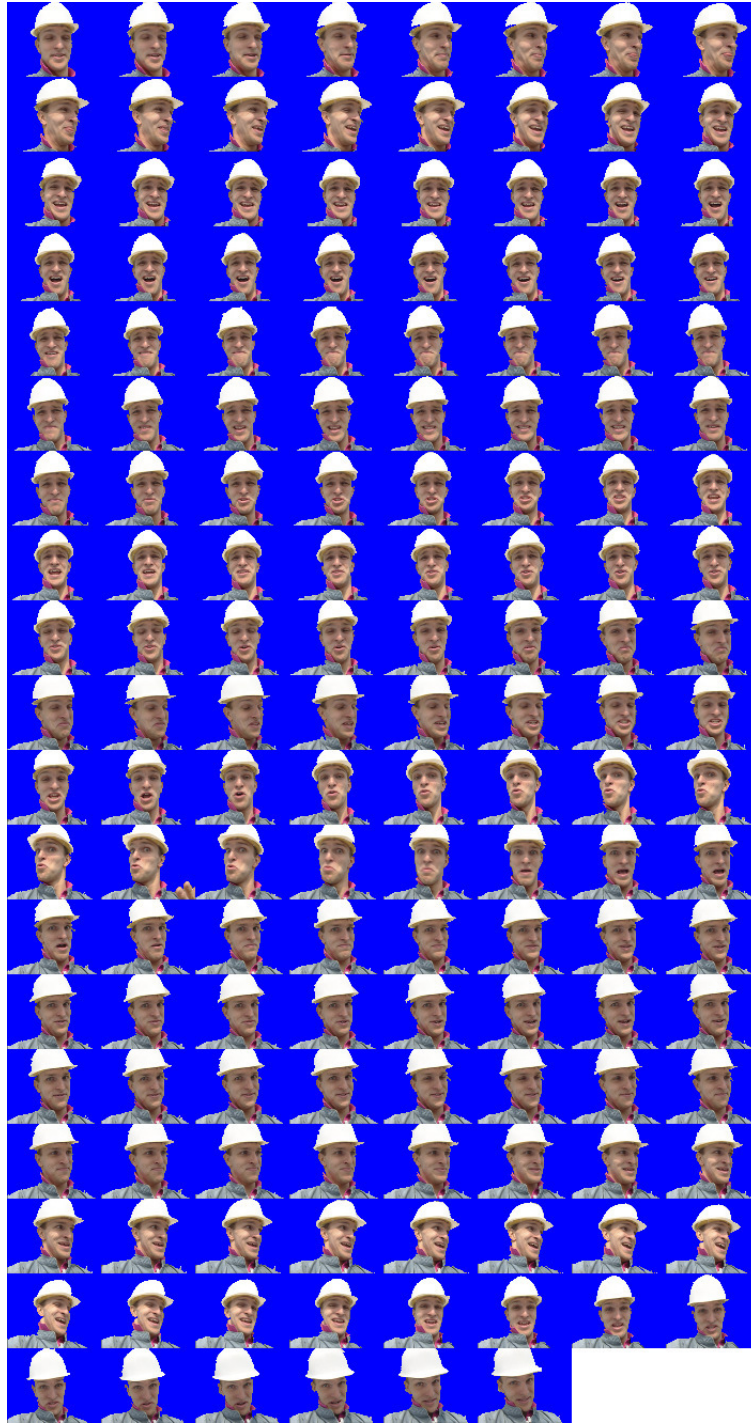


Fig. 4.3: Segmentation of *Foreman* sequence (frames 1 to 150). Using ∇_{B_5} and $\mathcal{W}_{B_4}$.

effort to complete the task: they required about 30 interferences, a very good improvement compared to the simple binding of markers approach.

Figure 4.4 shows the segmentation of *Carphone* sequence using $\nabla_{B_{17}}$ and $\mathcal{W}_{B_8}$. It is shown the segmentation of the first 150 frames of the sequence.

4.6 Conclusion

This paper introduces an improvement to the watershed from propagated markers: the combination of binding of markers heuristics with the spatio-temporal gradient. The main idea consists in, for a given frame, to apply the watershed from markers to an image extracted from an 3-D gradient sequence, computed by a morphological processing using a 3-D s.e.. The applied markers were computed by the binding of markers heuristics.

Experiments were done in order to assess the impact of the addition of the spatio-temporal gradient to the watershed from propagated markers framework and this addition provided a increasing in the segmentation performance. The decrease in the number of interferences was significant and the quality of segmentation was improved even with a lower number of user interferences. Time segmentation decreased as well.

Future works include the study of the spatio-temporal gradient properties in order to improve the segmentation results provided by the watershed from propagated markers technique.

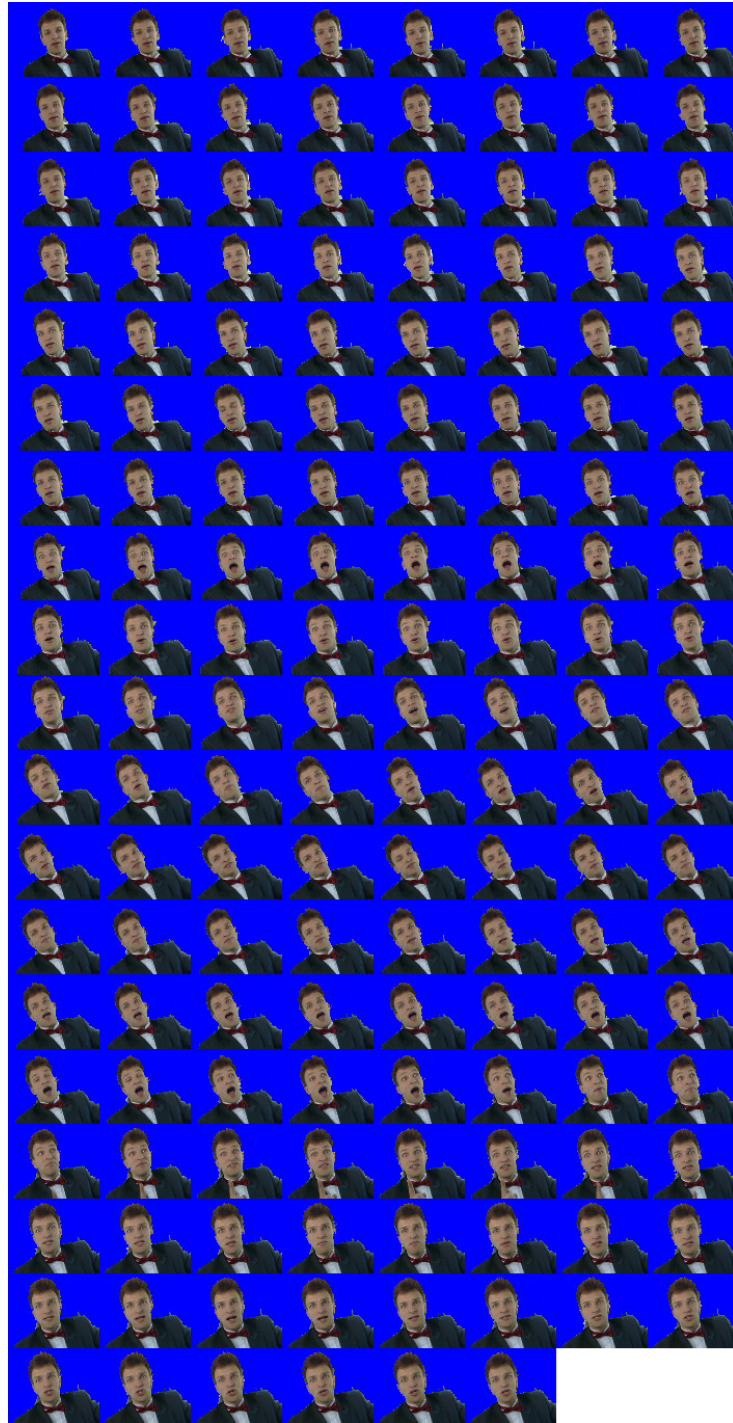


Fig. 4.4: Segmentation of *Carphone* sequence (frames 1 to 150). Using $\nabla_{B_{17}}$ and $\mathcal{W}_{B_8}$.

Capítulo 5

A Decomposição Uniforme: Uma Nova Decomposição Hierárquica Aplicada à Segmentação Morfológica de Objetos em Sequências de Imagens

Conforme apresentado na Introdução, foram desenvolvidas duas técnicas de geração de marcadores, uma baseada em contorno, explicada nos capítulos 2 e 3, e a baseada em regiões resultado da decomposição hierárquica do objeto e do fundo. Este capítulo apresenta mais uma melhoria baseada na decomposição hierárquica da máscara de segmentação. A melhoria apresentada aqui recebe suporte da decomposição uniforme, um novo método de segmentação hierárquica.

Uma abordagem considerada, antes do início dos trabalhos desta tese, para o *watershed* com marcadores propagados consistia simplesmente em erodir a máscara de segmentação e usá-la como marcador interno (similarmente, erodir o *background* e usá-lo como marcador externo). No entanto, esta idéia não é praticável. Apesar do marcador maior e mais consistente, este seria de difícil edição e não funcionaria bem para objetos deformáveis (ou composto de outros objetos). Além disso, objetos compostos de partes diferentes (e com prováveis diferentes deslocamentos) receberiam somente um vetor deslocamento.

O uso de segmentação hierárquica, no entanto, permite a criação de marcadores menores e mais consistentes de modo a minimizar as desvantagens apontadas no parágrafo anterior. A máscara de segmentação é dividida em regiões homogêneas e cada uma destas regiões será usada como marcador interno (o mesmo acontece com o *background* para os marcadores externos).

O uso de métodos clássicos para segmentação hierárquica permite a criação de marcadores homogêneos e mais definidos. No entanto, nada podemos afirmar sobre o formato de cada subconjunto das partições geradas por tal segmentação. Foi necessária a criação de um método de segmentação hierárquica onde *forma e tamanho dos subconjuntos gerados para a partição* fossem levados em conta. Desta necessidade, surgiu a decomposição uniforme.

A decomposição uniforme é um método incremental onde regiões são divididas em cada iteração em dois subconjuntos que respeitem critérios de entrada como tamanho ou forma. Caso não seja possível encontrar uma divisão que respeite tais critérios, a região não é dividida. No final do processo, temos um particionamento da imagem de modo que não haja subconjunto tal que sua medida, em

relação ao critério escolhido, seja menor que um parâmetro p .

Esta propriedade é bastante útil quando queremos projetar marcadores com uma determinada característica no *framework* do *watershed* com marcadores propagados. Por exemplo, se desejamos marcadores que não desapareçam após uma erosão realizada para um melhor ajuste dentro do objeto, escolhemos decompor a imagem em função de um elemento estruturante disco. Se queremos marcadores com uma determinada área mínima, escolhemos decompor a imagem em função de um parâmetro de área. Mais detalhes sobre esta decomposição no texto que se segue.

Este capítulo, além de propor uma forma de se calcular uma decomposição uniforme para uma imagem, sugere algumas aplicações desta segmentação hierárquica, tais como possível pré-processamento para visão estéreo e para segmentação interativa de imagens estáticas.

A contribuição deste capítulo é uma melhoria do *watershed* com marcadores propagados, baseada na decomposição hierárquica da máscara de segmentação e uma outra forma de edição dos marcadores (baseada na segmentação interativa supracitada). A própria decomposição uniforme é uma outra contribuição, consequente do desenvolvimento da melhoria do *watershed* com marcadores propagados proposta neste capítulo. As contribuições citadas são apresentadas neste capítulo em formato de artigo e em português.

5.1 Resumo

A decomposição uniforme é um novo método de segmentação hierárquica cuja propriedade principal é a preservação de uma dada característica para cada subconjunto da partição gerada pela decomposição. A decomposição é gerada incrementalmente e, em cada iteração, uma região candidata é dividida somente se for encontrada uma divisão válida onde cada novo subconjunto respeita um critério de entrada, por exemplo, tamanho ou forma. A decomposição uniforme é comparável com a decomposição clássica dada pela limiarização dos valores de extinção para cada mínimo regional. Mais, é comparável enquanto mantém o critério para cada subconjunto da partição. Tal aplicação de critério é vantajosa quando se quer gerar ou preservar regiões de acordo com alguma propriedade. Por exemplo, pode se querer particionar a imagem em regiões que possuam uma área mínima, de modo que todas sejam preservadas após algum tipo de filtragem por área. Este capítulo ainda propõe várias aplicações da decomposição uniforme tais como possível pré-processamento para visão estéreo e segmentação interativa. Em destaque, este capítulo apresenta a aplicação da referida decomposição como suporte para uma nova versão do *watershed* com marcadores propagados.

5.2 Introdução

Segmentação de imagens é um dos campos de pesquisa mais desafiadores em processamento digital de imagens. Isto consiste na separação de objetos contidos em uma imagem de entrada através do particionamento da mesma, de acordo com um dado critério ou método, em subconjuntos disjuntos. Dada a representação dos objetos por cada subconjunto do particionamento de saída, é possível comparar e/ou classificar objetos entre si ou apenas destacar um objeto em particular. Este último caso é usualmente visto como a separação entre objeto e fundo em uma imagem.

A literatura apresenta vários métodos desenvolvidos para segmentação de imagens. Entre eles,

a Morfologia Matemática merece menção devido à contribuição dada por este campo de pesquisa com vários operadores projetados para segmentação de imagens. Em particular, o *watershed* com marcadores [42, 54] é uma eficiente técnica de segmentação, que reduz o problema de segmentar objetos ao problema de se encontrar apontadores para os objetos de interesse (estes apontadores são chamados *marcadores*).

A transformada *watershed* com seleção de marcadores é também a base para um modo de realizar *segmentação hierárquica* [71]. Uma imagem é hierarquicamente segmentada em função de uma hierarquia de partições aninhadas. Através de uma sistemática seleção de marcadores inicialmente impostos em todos os mínimos regionais de uma imagem), a segmentação de uma imagem pode ser refinada ou simplificada. Segmentação hierárquica possibilita a representação dos objetos em uma imagem em vários níveis de detalhes e tem sido aplicada para resolver problemas tais como segmentação de imagens RM [72], segmentação de células [73] e codificação de vídeo [65].

Um modo de se obter segmentação hierárquica é pela aplicação de uma função de extinção [74] para cada mínimo regional de uma imagem e então descartar todos os mínimos cujos valores de extinção sejam menores que um valor limiar. É também possível segmentar uma imagem em n regiões mais significativas - de acordo com uma função de extinção - tomando os mínimos regionais com os n maiores valores de extinção e aplicando-os como marcadores para a transformada *watershed*.

O método descrito acima é clássico no *framework* da segmentação morfológica hierárquica. Porém, o método supracitado não oferece um meio de controlar as propriedades de cada subconjunto de uma dada partição. Tal método, por exemplo, possibilita particionar a imagem em n regiões, mas nada pode ser afirmado sobre qualquer subconjunto da partição. Poderia ser interessante obter-se uma segmentação morfológica hierárquica onde cada subconjunto da partição respeitasse um dado critério, tais como tamanho ou forma. Poderia ser útil, por exemplo, em situações onde os subconjuntos das partições devessem ser preservados após uma filtragem por área ou uma erosão por um dado elemento estruturante.

Este capítulo da tese introduz um novo método para segmentação hierárquica que computa partições tais que todos os seus subconjuntos respeitam um dado critério: a *decomposição uniforme*. A construção desta decomposição é incremental e, em cada iteração, uma região candidata é quebrada em duas regiões somente se ambas as regiões forem aprovadas por uma função critério. Desta forma, o método garante que todas as regiões na imagem decomposta respeitem o critério de entrada. Além do critério e da imagem de entrada, a decomposição uniforme recebe como dados de entrada um parâmetro numérico para a função critério e um domínio de aplicação que possibilita a decomposição de somente uma região específica da imagem.

Uma das vantagens oferecidas pela decomposição uniforme é a geração de regiões que respeitam alguma propriedade. Pode ser desejável a preservação das regiões da imagem após um determinado processamento como a erosão morfológica ou a filtragem por área.

Além de introduzir a decomposição uniforme, este capítulo também apresentará várias aplicações da decomposição, tais como possível pré-processamento para visão estéreo e segmentação hierárquica interativa. Particularmente, este capítulo apresenta a aplicação da decomposição uniforme como um importante passo em uma nova variação do *watershed* com marcadores propagados [4]. Tais aplicações ilustram a vantagem descrita no parágrafo anterior.

Este capítulo está organizado desta forma: seção 5.3 apresenta uma revisão de conceitos importantes para o *framework* proposto de decomposição. Seção 5.4 introduz a decomposição uniforme e apresenta várias aplicações desta segmentação hierárquica. Seção 5.5 apresenta uma nova versão do

watershed com marcadores propagados com suporte da decomposição uniforme.

5.3 Revisão Preliminar

Seja $E \subset \mathbb{Z} \times \mathbb{Z}$ um subconjunto retangular finito de pontos. Seja $K = [0, k]$ um conjunto totalmente ordenado. Denote por $Fun[E, K]$ o conjunto de todas as funções $f : E \rightarrow K$. Uma *imagem* é uma destas funções (chamadas funções em níveis de cinza). Particularmente, se $K = [0, 1]$, f é uma imagem binária. Um *operador entre imagens* (*operador*, por simplicidade) é um mapeamento $\psi : Fun[E, K] \rightarrow Fun[E, K]$.

Seja $N(x)$ o conjunto que contém a *vizinhança* de x , $x \in E$. Define-se o *caminho* de x a y , $x, y \in E$ como uma sequência $P(x, y) = (p_0, p_1, \dots, p_n)$ de E , onde $p_0 = x$, $p_n = y$ e $\forall i \in [0, n-1], p_i \in N(p_{i+1})$.

Um *subconjunto conexo* de E é um subconjunto $X \subset E$ tal que, $\forall x, y \in X$, há um caminho C inteiramente contido em X .

Seja $f \in Fun[E, K]$. Uma *zona plana* de f é um subconjunto conexo $Z \subset E$, tal que $f(x) = f(y)$, $\forall x, y \in Z$. O conjunto de todas as zonas planas adjacentes a Z é dado por $\mathcal{F}_Z = \{ \text{zona plana } Y : \exists y \in Y, z \in Z : y \in N(z) \}$.

Operadores conexos pertencem a uma família de operadores ψ tais que, se $x \in N(y)$, $f(x) = f(y) \Rightarrow \psi(f)(x) = \psi(f)(y)$, $\forall f \in Fun[E, K]$. Sua principal propriedade é a simplificação da imagem (por agrupamento de zonas planas adjacentes) sem a introdução de bordas (uma vez que eles não quebram zonas planas) [75].

Seja $f \in Fun[E, k]$. Um *mínimo regional* de f é uma zona plana Z tal que $f(z) < f(y)$, $z \in Z$, $y \in Y$, $Y \in \mathcal{F}_Z$. Seja $\text{Min}(f)$ o subconjunto de E que denota o conjunto de todos os mínimos regionais de $f \in Fun[E, K]$.

Seja $B_c \subset E$ um *elemento estruturante* (e.e.) que define a *conectividade* dos operadores morfológicos conexos usados neste capítulo (4-conectividade é dada pelo e.e. cruz elementar 3×3 . 8-conectividade é dada por um e.e. quadrado 3×3).

5.3.1 Funções de extinção

Esta seção revisa as noções básicas das *funções de extinção*, um conjunto de medidas de persistência aplicado aos mínimos e máximos regionais de uma imagem f . Tais medidas avaliam os extremos de acordo com características estruturais tais como área, volume ou contraste. Estes valores ajudam em tarefas tais como a remoção de estruturas irrelevantes de uma imagem e este é um dos importantes conceitos por trás do *framework* de segmentação hierárquica.

Apesar dos conceitos de valores e funções de extinção serem aplicados em mínimos e máximos regionais, este capítulo discutirá apenas o caso dos mínimos regionais. Os máximos regionais não são considerados no *framework* de segmentação hierárquica introduzido na seção 5.4.

Uma *anti-granulometria* é uma família decrescente de operadores $\Lambda = (\rho_\lambda)_{\lambda \geq 0}$ com as seguintes propriedades, $\forall \lambda \geq 0, \forall f, g \in Fun[E, K]$:

- *crescente*: $f \leq g \Rightarrow \rho_\lambda(f) \leq \rho_\lambda(g)$.
- *extensividade*: $\rho_\lambda(f) \geq f$.

- $\forall \mu \geq 0, \rho_\lambda(\rho_\mu(f)) = \rho_\mu(\rho_\lambda(f)) = \rho_{\max(\lambda, \mu)}(f)$.
- *idempotência*: $\rho_\lambda(\rho_\lambda(f)) = \rho_\lambda(f)$ (devido à propriedade acima).

Devido à propriedade dos operadores conexos, zonas planas podem se juntar após a aplicação de tais operadores. Como consequência, zonas planas, incluindo mínimos regionais, podem ser estendidos. Uma vez que operadores conexos também não quebram zonas planas, eles não criam novos mínimos regionais. Ambas as propriedades dos operadores conexos fazem com que sejam adequados para serem aplicados no *framework* de avaliação dos mínimos. Isto justifica a escolha de fechamentos por reconstrução (ou sup-reconstruções) para compor as famílias de anti-granulometrias que serão aplicadas na avaliação citada.

Seja Λ uma anti-granulometria por fechamentos por reconstrução. Na análise anti-granulométrica, o parâmetro λ aumenta incrementalmente e, para cada valor de λ , a quantidade de estruturas peneiradas por ρ_λ é computada. Uma vez que a quantidade de estruturas peneiradas aumenta à medida que λ aumenta, a curva granulométrica para Λ é uma função crescente. O espectro anti-granulométrico é uma outra função que mede, para um dado λ , a quantidade de informação peneirada quando λ é incrementado. Este espectro é dado pela derivada da curva anti-granulométrica.

Note que as medidas descritas no parágrafo anterior são globais no contexto da análise anti-granulométrica. É também possível usar a anti-granulometria Λ para realizar uma análise local em uma imagem.

Por exemplo, dado um mínimo regional $M \in \text{Min}(f)$, é possível avaliar o quanto M persiste sob a aplicação sequencial de Λ . A anti-granulometria Λ será aplicada em f , começando de $\lambda = 0$, até o mínimo regional em $\rho_\lambda(f)$ associado a M desaparecer. Suponha que $\rho_{\lambda+1}$ remova M inteiramente. A *valorização da extinção* de M será feita pela avaliação do resultado de ρ_λ , que provê o último mínimo regional associado a M antes que este desapareça.

Uma vez que ρ_λ é extensivo $\forall \lambda \geq 0$, o mínimo regional de $\rho_\mu(f)$ associado com M é sempre uma expansão do próprio M , $\forall 0 \leq \mu \leq \lambda$. Uma vez que M é peneirado na aplicação de $\rho_{\lambda+1}$, M não estará contido em nenhum mínimo regional de $\rho_{\lambda+1}(f)$. Formalmente, a *função de extinção* é dada por

$$\mathcal{E}_\Lambda(M) = \max\{\lambda \geq 0 : \forall 0 \leq \mu \leq \lambda, M \subseteq \text{Min}(\rho_\mu(f))\}. \quad (5.1)$$

Note que, nas definições acima, a propriedade dos operadores conexos é a única restrição aplicada na anti-granulometria. O tipo dos fechamentos que compõem a família de tais operadores não foi especificada. Neste capítulo, valores de extinção e funções de extinção foram definidos conceitualmente, independente de critério de análise específico. Vachier [76] define valores e funções de extinção usando (aberturas e) fechamentos por reconstrução. Breen e Jones apresentam os filtros por atributo [77], onde eles propõem vários critérios crescentes para o projeto de aberturas (e fechamentos) por atributos.

Vários critérios podem ser aplicados na construção de uma família Λ de fechamentos. Três critérios, clássicos no contexto da segmentação morfológica hierárquica, foram aplicados neste capítulo: a *dinâmica* [78], o *valor de extinção por área* [74, 79] e o *valor de extinção por volume* [79].

As Figuras 5.1, 5.2 e 5.3 ilustram o funcionamento do uso de valores de extinção por área na filtragem de uma função f com três mínimos regionais, M_1 , M_2 e M_3 . A Figura 5.1 (a) mostra tal função de entrada e uma legenda com valores associados a diferentes cores. Os três vales da função

estão preenchidos com “degraus” de diversas cores e cada cor representa a área do vale naquela altitude. Por exemplo, no vale que contém o mínimo regional M_1 , a área da região onde se localiza este mínimo é 30. No nível imediatamente acima, a área do vale naquela altitude vale 35 e, no nível seguinte, a área vale 40. Os “degraus” hachurados mostram o quanto foi filtrado de cada vale em um determinada filtragem.

Denotemos, neste exemplo, por $\rho_\mu(f)$ a filtragem por fechamento por área com parâmetro de área μ . A Figura 5.1 (b) mostra o resultado de $\rho_{40}(f)$. Tal operador filtrou metade do vale que contém M_1 e o fundo do vale que contém M_2 . Note que $M_1, M_2 \subseteq \text{Min}(\rho_{40}(f))$. A Figura 5.1 (c) mostra o resultado de $\rho_{59}(f)$. Nesta filtragem, $M_1 \not\subseteq \rho_{59}(f)$. Como $M_1 \subseteq \rho_{45}(f)$, o valor de extinção por área do mínimo regional M_1 é 45.

A Figura 5.2 (b) mostra o resultado de $\rho_{93}(f)$. Note que, aqui, $M_1 \subseteq \rho_{93}(f)$, no entanto, o fato de $M_1 \not\subseteq \rho_{59}(f)$ desqualifica os demais valores $\mu \geq 45$ como valores de extinção por área para M_1 . Note que ainda $M_2 \subseteq \rho_\mu(f), \forall 0 \leq \mu \leq 93$.

A Figura 5.3 mostra o momento em que o mínimo regional M_3 foi extinto. Veja que $M_3 \not\subseteq \rho_{131}(f)$. A última filtragem que possuía um mínimo que contivesse M_3 foi $\rho_{130}(f)$. Assim, o valor de extinção por área para M_3 é 130. Finalmente, como M_2 não foi extinto para nenhum parâmetro de área, o valor de extinção atribuído a M_2 é $+\infty$.

Note que, dada uma função $f \in \text{Fun}[E, K]$ e seus mínimos regionais $\text{Min}(f)$, a ordenação dos mínimos em $\text{Min}(f)$ varia de acordo com o critério aplicado para computar os valores de extinção. Veja, por exemplo, a função f na Fig. 5.4 e seus três marcadores, rotulados como M_1, M_2 e M_3 . A Tabela 5.1 mostra os valores de extinção computados para cada mínimo de acordo com os critérios supracitados. Veja na primeira linha da Tabela 5.1 que o mínimo com a mais significativa dinâmica é M_2 , seguido por M_1 e M_3 . De acordo com o critério de volume (terceira linha), novamente, M_2 tem o valor de extinção mais significativo mas, neste caso, é seguido por M_3 e, então, por M_1 .

Critério	M_1	M_2	M_3
Dinâmica	30	$+\infty$	20
Área	4	8	$+\infty$
Volume	110	$+\infty$	210

Tab. 5.1: Valores de extinção para os mínimos regionais da função na Fig. 5.4.

5.3.2 Segmentação hierárquica

Uma *partição* $\mathcal{X}$ of E é uma família de subconjuntos $X_i \in E$, tais que $X_i \cap X_j = \emptyset, i \neq j$, e $\bigcup_i X_i = E$. Seja $\mathcal{X} = (\mathcal{X}_i)$ uma família de partições tais que $\mathcal{X}_1 = E$ e

$$X_j \in \mathcal{X}_i \Rightarrow \exists X_k \in \mathcal{X}_{i-1} : X_j \subseteq X_k, \quad (5.2)$$

$\forall i > 1$. A família de partições $\mathcal{X}$ define uma *hierarquia* [71] de partições aninhadas: a união de regiões adjacentes em partições mais refinadas forma uma região em uma partição menos refinada. E o contorno de uma região em uma partição menos refinada também pertencerá às partições mais refinadas.

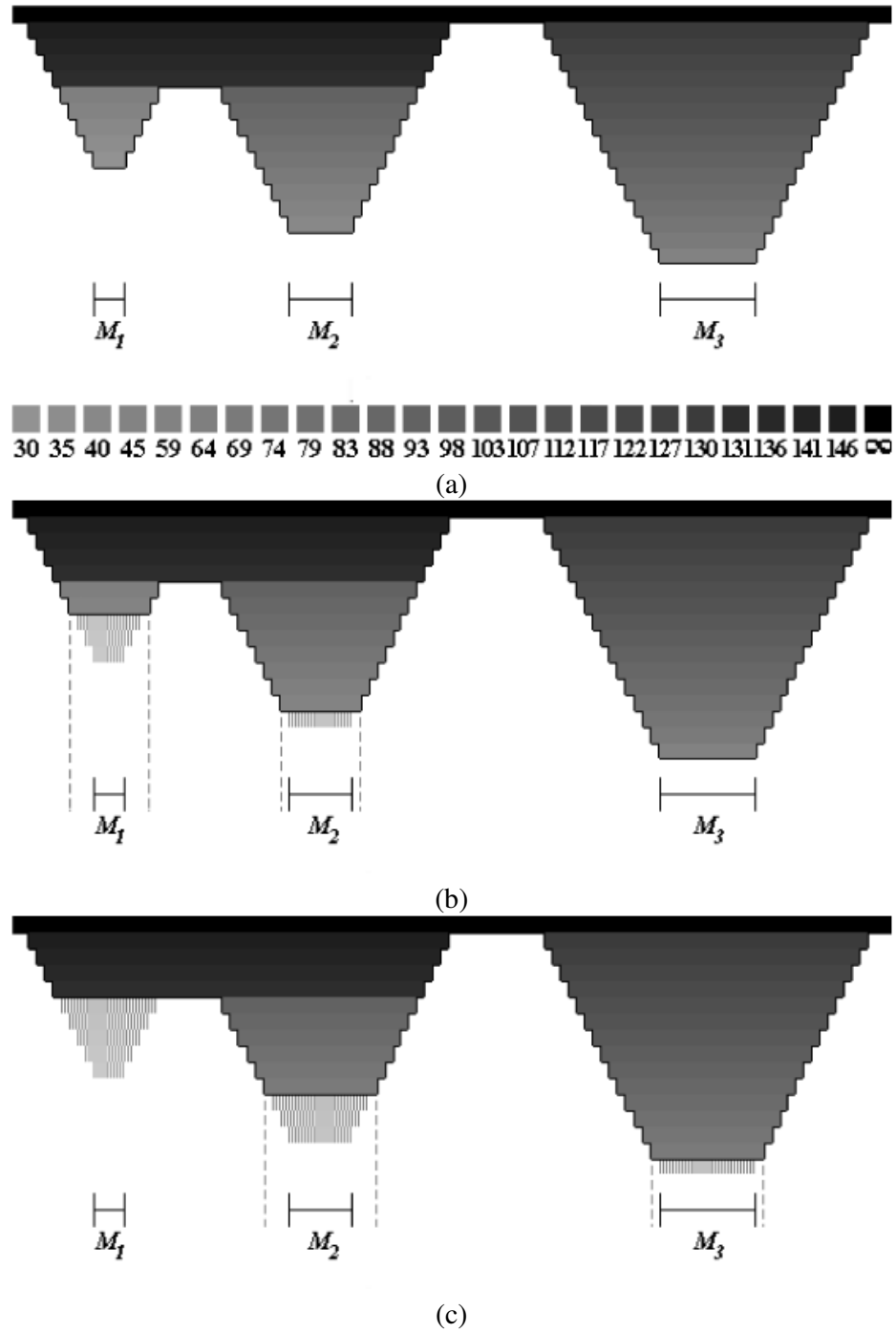


Fig. 5.1: Exemplo da limiarização por valores de extinção por área. (a) Função inicial e legenda com os valores de extinção. (b) Aplicação de limiar 40. (c) Aplicação de limiar 59.

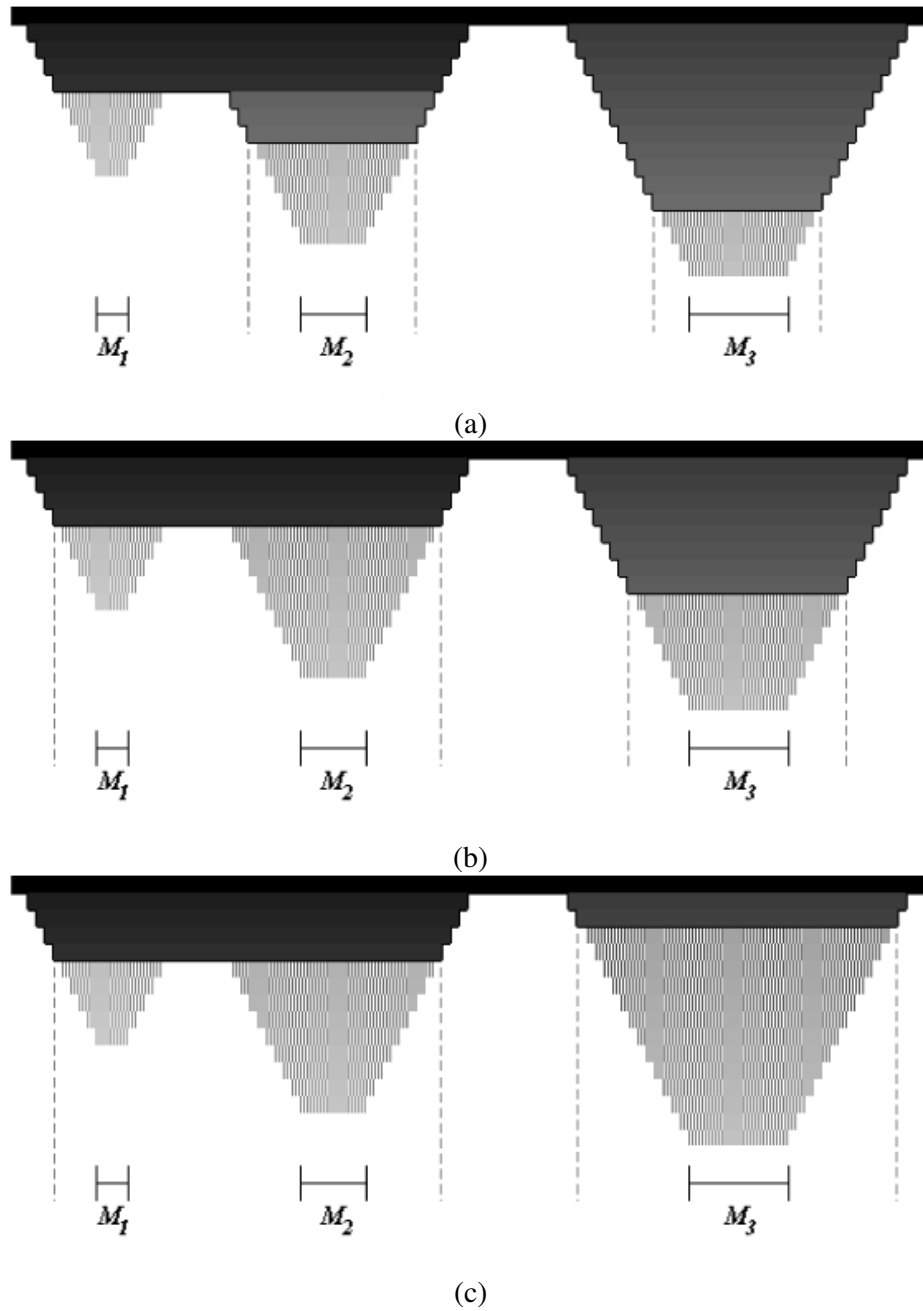


Fig. 5.2: Exemplo da limiarização por valores de extinção por área. (a) Aplicação de limiar 79. (b) Aplicação de limiar 93. (c) Aplicação de limiar 127.

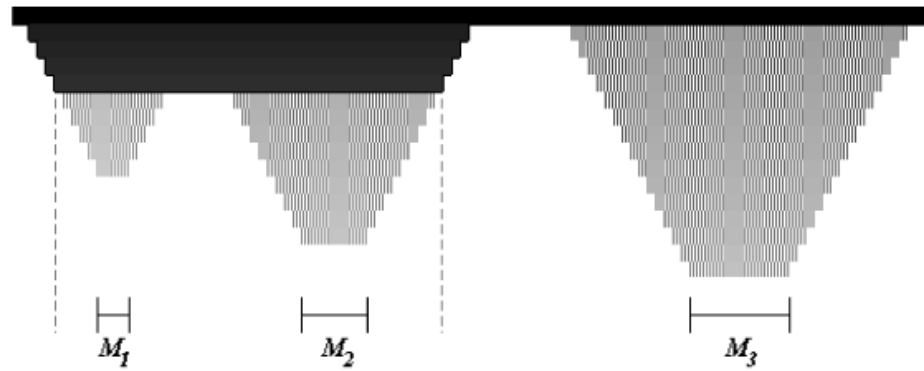


Fig. 5.3: Exemplo da limiarização por valores de extinção por área. (a) Aplicação de limiar 131.

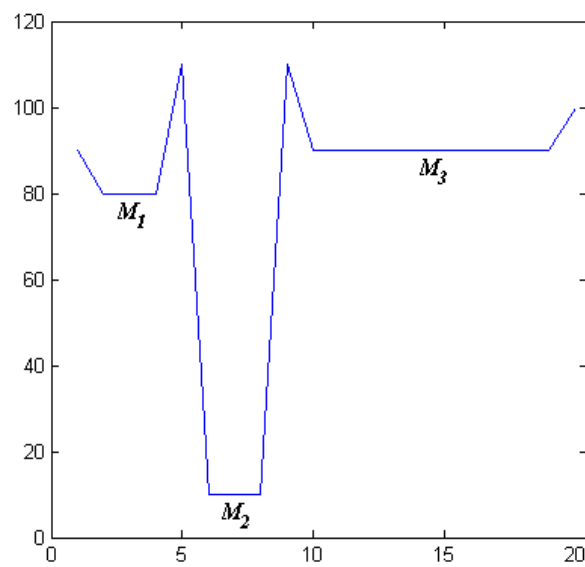


Fig. 5.4: Exemplo de função com marcadores rotulados.

Segmentação hierárquica é a segmentação de uma imagem em função de uma hierarquia de partições aninhadas. Dada uma família de tais partições $\mathcal{X} = (\mathcal{X}_i)$, atribuídas a uma imagem f , a segmentação hierárquica de f pode ser dada por uma partição $\mathcal{X} \in \mathcal{X}$.

Uma maneira de produzir uma segmentação hierárquica é por aplicação da técnica de *watershed* com marcadores. A idéia é computar as partições aninhadas através do descarte de mínimos regionais que não são relevantes e da preservação dos mínimos importantes, antes da aplicação do operador *watershed*. Uma hierarquia pode ser definida pelo modo que os mínimos regionais de uma imagem são qualificados e selecionados.

O cálculo de valores de extinção é um maneira clássica de qualificar os mínimos regionais de uma imagem para segmentação hierárquica. Esta segmentação é usualmente realizada pela aplicação de uma função de extinção em cada mínimo e pela seleção de todos os mínimos cujos valores de extinção sejam maiores que um dado valor de limiar.

5.4 A Decomposição Uniforme

Esta seção introduz a decomposição uniforme, algumas de suas propriedades e o algoritmo da decomposição proposta. Para definir a decomposição uniforme, alguns conceitos auxiliares devem ser definidos *a priori*.

5.4.1 Conceitos preliminares

O primeiro conceito a ser introduzido é o de *função critério*. Esta é uma função que será aplicada para avaliar subconjuntos conexos de acordo com um dado critério crit e um parâmetro p . Sua fórmula geral é dada por

$$\mathbb{C}_{\text{crit}}(C, p) = \begin{cases} 1, & \text{se } C \text{ satisfaz o critério } \text{crit}, \\ 0, & \text{caso contrário,} \end{cases} \quad (5.3)$$

onde $C \subseteq E$ e p é um parâmetro numérico.

Este capítulo usará duas funções critério: o critério de *área* e o de *disco*. A função de critério de área devolve 1 se a área do subconjunto conexo C é maior ou igual a p . É dada por,

$$\mathbb{C}_{\text{area}}(C, p) = \begin{cases} 1, & \text{se } \#(C) \geq p, \\ 0, & \text{caso contrário,} \end{cases} \quad (5.4)$$

onde $\#(C) : E \rightarrow \mathbb{Z}_+$ devolve a cardinalidade de $C \subseteq E$.

A função de critério de disco devolve 1 se $C \subseteq E$ resiste à erosão por um e.e. disco com raio p . É dado por,

$$\mathbb{C}_{\text{disco}}(C, p) = \begin{cases} 1, & \text{se } \varepsilon_{B_p}(C) \neq \emptyset, \\ 0, & \text{caso contrário,} \end{cases} \quad (5.5)$$

onde B_p é o e.e. disco com raio p .

Uma *fila* é uma lista sequencial $Q = \{q_1, q_2, q_3, \dots, q_n\}$ de elementos arbitrários cujas propriedades principais são a inserção de elementos no final da fila (anexado após o n -ésimo elemento q_n) e a remoção de elementos do início da estrutura (somente o primeiro elemento de Q pode ser

removido, i.e., q_1). Uma fila Q pode ter, pelo menos, zero elementos (neste caso, é uma fila vazia). Um conjunto de funções é usualmente associado à fila Q : as que são aplicadas no algoritmo de decomposição são descritas abaixo:

- $\text{InicializaFila}(Q)$: esta função inicializa Q com zero elementos;
- $\text{FilaVazia}(Q)$: retorna **VERDADEIRO** quando Q tem zero elementos, e **FALSO**, caso contrário;
- $\text{InsereFila}(Q, x)$: o elemento x é inserido no final de Q , se tornando o último elemento de Q . Se Q fosse uma fila vazia, x também se torna o primeiro elemento da sequência.
- $x \leftarrow \text{RemoveFila}(Q)$: remove o primeiro elemento de Q , atribuindo-o a x . Se Q tinha somente um elemento, Q se torna uma fila vazia; caso contrário, q_2 se torna o primeiro elemento de Q .

Sejam $A, B \subset E$ dois subconjuntos conexos e disjuntos. Seja $\mathcal{W}_{A,B}(g)$ o *operador watershed com marcadores*, que computa a partição de E em função de g , aplicando A e B como marcadores. Esta partição é dada por dois subconjuntos E_A e E_B , tais que

- $E_A \cup E_B = E$;
- $E_A \cap E_B = \emptyset$;
- $A \subseteq E_A$;
- $B \subseteq E_B$.

Seja $D \subseteq E$ um *domínio de validação*. Dado uma função de critério $\mathbb{C}_{\text{crit}}$ e um parâmetro p , os conjuntos $A, B \in D$ são dois *marcadores válidos* se,

$$\mathbb{C}_{\text{crit}}(E_A \cap D, p) = \mathbb{C}_{\text{crit}}(E_B \cap D, p) = 1, \quad (5.6)$$

onde $E_A, E_B \in E$ são os dois subconjuntos que denotam uma partição de E computada por $\mathcal{W}_{A,B}$. O conceito de *validação* introduzido acima será aplicado na decomposição, para checar se um subconjunto da segmentação hierárquica pode ser quebrado em função de um par de marcadores. A quebra deve ocorrer somente se os dois novos subconjuntos satisfazem o critério de decomposição.

O algoritmo a seguir é usado pelo algoritmo de decomposição introduzido na próxima subseção. Ele recebe como dados de entrada dois mínimos regionais $M_a, M_k \in E$, a imagem de gradiente $g \in \text{Fun}[E, K]$, o domínio de validação $D \in E$, o critério crit e seu parâmetro numérico relacionado $\text{par} \in \mathbb{Z}_+$. O algoritmo **Crit** recebe dois mínimos regionais candidatos e verifica se eles compõem um par de marcadores válidos. Para tal verificação, aplica-se a função critério $\mathbb{C}_{\text{crit}}$ em ambos os subconjuntos computados pela intersecção do resultado do *watershed* com o domínio de validação e soma-se os resultados. Se o resultado da soma for 2, então os marcadores compõem um par válido de marcadores. O algoritmo $\mathbb{C}_{\text{crit}}$ devolve o resultado da soma.

Crit($M_a, M_k, g, C, \text{crit}, \text{par}$);

- 1 $w_k \leftarrow \mathcal{W}_{M_a, M_k}(g);$
- 2 $E_a \leftarrow \{x \in E : w_k(x) = 1\} \cap C;$
- 3 $E_k \leftarrow \{x \in E : w_k(x) = 2\} \cap C;$
- 4 $\underline{\text{Crit}} \leftarrow \mathbb{C}_{\text{crit}}(E_a, par) + \mathbb{C}_{\text{crit}}(E_k, par);$

5.4.2 O algoritmo

O algoritmo **DecomposicaoUniforme** recebe os seguintes dados de entrada:

- a imagem de entrada $img \in Fun[E, K];$
- o subconjunto $D \in E$ que define o domínio de aplicação da decomposição. Este domínio de aplicação será a região da imagem onde a decomposição uniforme será aplicada. O domínio de aplicação pode ser dado inteiramente pelo domínio da própria imagem (o que define que a imagem inteira será decomposta) ou, ainda, ser dado por uma máscara de segmentação;
- o critério crit , aplicado na quebra iterativa de regiões; e
- o parâmetro $par \in \mathbb{Z}_+$ usado quando o critério acima é aplicado.

O algoritmo devolve uma imagem $w \in Fun[E, \mathbb{Z}_+]$ contendo a imagem decomposta em um ou mais subconjuntos. Cada subconjunto recebe um rótulo distinto $lbl \in \mathbb{Z}_+$ para ser unicamente identificado.

O algoritmo de decomposição também usa os seguintes dados auxiliares:

- a imagem de gradiente $g \in Fun[E, K]$, cujos mínimos regionais localizados fora do domínio de aplicação foram removidos;
- um inteiro $lbl \in \mathbb{Z}_+$, usado para armazenar o último rótulo atribuído a w ;
- uma fila Q , que armazena os rótulos das regiões que estão sendo correntemente processadas;
- a região $C \in E$, que denota o domínio de validação corrente (a região onde o algoritmo buscará um par válido de marcadores na iteração corrente);
- um inteiro $Elem$ que armazena o rótulo do domínio de validação corrente;
- o conjunto $M \in E$ que armazena o conjunto de mínimos regionais localizados dentro do domínio de validação corrente;
- a lista ordenada A de mínimos regionais, cujos mínimos estão ordenados decrescentemente em função de seus valores de extinção;
- três identificadores M_a , M_b e M_k para mínimos regionais;
- uma imagem auxiliar $w_i \in Fun[E, K]$ que recebe a partição de E , de acordo com g , M_a e M_b , e o conjunto $W_i \in E$ que define a região que é atualizada em w quando um par de marcadores válidos é encontrado na iteração corrente;

- dois identificadores inteiros $n, k \in \mathbb{Z}_+$.

As funções que são aplicadas no algoritmo de decomposição são descritas abaixo:

- $\nabla : Fun[E, K] \rightarrow Fun[E, K]$ é a função que computa o gradiente da imagem de entrada. O gradiente é um dos parâmetros da função *watershed*;
- $\text{Min} : Fun[E, K] \rightarrow E$ é a função que computa o conjunto de mínimos regionais da imagem de gradiente. Esta função é citada na seção 5.3;
- a função *watershed* com marcadores $\mathcal{W}_{A,B}$ que devolve uma partição de E de acordo com o gradiente e os dois marcadores de entrada A e B . O conjunto E é particionado em dois conjuntos disjuntos, tal como foi definido na subseção anterior.

DecomposicaoUniforme(*img*, *D*, *crit*, *par*);

```

1    $w : E \rightarrow \mathbb{Z}_+$ ;
2    $w(x) \leftarrow 1 : x \in D$ ;
3    $w(x) \leftarrow 0 : x \notin D$ ;
4    $g \leftarrow \nabla(\text{img})$ ;
5    $g(x) \leftarrow \max\{g(y) : y \in D\} : x \notin D$ ;
6    $lbl \leftarrow 1$ ;
7   InicializaFila(Q);
8   se  $\mathbb{C}_{\text{crit}}(D, \text{par}) = 1$  então
8.1   InsereFila(Q, 1);
9   fim se;
10  enquanto NAO FilaVazia(Q) faça
10.1    $Elem \leftarrow \text{RemoveFila}(\text{Q})$ ;
10.2    $C \leftarrow \{x \in D : w(x) = Elem\}$ ;
10.3    $M \leftarrow \text{Min}(g) \cap C$ ;
10.4    $n \leftarrow (\text{número de mínimos regionais em } M)$ ;
10.5    $A \leftarrow \{M_j \in M : \mathcal{E}_\Lambda(M_j) \geq \mathcal{E}_\Lambda(M_{j+1}), j \in [1, n]\}$ ;
10.6    $M_a \leftarrow M_1 \in A$ ;
10.7    $M_b \leftarrow \emptyset$ ;
10.8    $k \leftarrow 2$ ;
10.9   enquanto  $(M_b = \emptyset)$  E  $(k \leq n)$  faça
10.9.1   se  $\mathbb{C}_{\text{crit}}(M_a, M_k \in A, g, C, \text{crit}, \text{par}) = 2$  então
10.9.1.1    $M_b \leftarrow M_k$ ;
10.9.2   fim se;
10.9.3    $k \leftarrow k + 1$ ;
10.10  fim enquanto;
10.11  se  $M_b \neq \emptyset$  então
10.11.1    $lbl \leftarrow lbl + 1$ ;
10.11.2    $w_i \leftarrow \mathcal{W}_{M_a, M_b}(g)$ ;

```

```

10.11.3       $W_i \leftarrow \{x \in C : w_i(x) = 2\};$ 
10.11.4       $w(x) \leftarrow lbl : x \in W_i;$ 
10.11.5       $\text{InsereFila}(Q, Elem);$ 
10.11.6       $\text{InsereFila}(Q, lbl);$ 
10.12      fim se;
11      fim enquanto;
12      DecomposicaoUniforme  $\leftarrow w;$ 

```

O algoritmo **DecomposicaoUniforme** é descrito a seguir: Linhas 1-3 são três abusos de notação. Eles são usados para descrever a inicialização da imagem de saída w . Linha 1 declara que w é uma imagem que pertence ao conjunto $Fun[E, \mathbb{Z}_+]$. Na Linha 2, w recebe o rótulo 1 em todos os pontos dentro do domínio de aplicação. Fora do domínio de aplicação, w recebe o valor zero. Isto é feito na Linha 3 e significa que a região fora do domínio não pertence ao resultado da decomposição. Nos casos onde o domínio de aplicação é a imagem inteira, w não recebe zeros em ponto algum.

A Linha 4 computa o gradiente da imagem de entrada. Linha 5 apresenta um outro abuso de notação: significa que g receberá o mais alto nível de cinza do gradiente em todos os pontos fora do domínio de aplicação, para descartar todos os mínimos regionais fora do domínio em questão.

A Linha 6 inicializa lbl com “1” - o último rótulo atribuído a w . A Linha 7 inicializa a fila Q - neste ponto, Q está vazia. A Linha 8 testa o domínio de aplicação: se este não passa pela função critério, o domínio não deve ser particionado e o algoritmo devolve o próprio domínio de aplicação rotulado com “1”. Se o domínio de aplicação passa pela função critério, seu rótulo deve ser inserido na fila - esta inserção é feita na Linha 8.1.

A estrutura **enquanto** nas Linhas 10 a 11 controla as iterações da decomposição. Enquanto a fila armazenar elementos, uma nova iteração deve ser iniciada. A Linha 10.1 remove um rótulo da fila e o armazena em $Elem$. A Linha 10.2 computa a região C tomando todos os pontos em w que possuam rótulo $Elem$. A região C é usada como o domínio de validação corrente e como a região a ser quebrada se um par válido de marcadores for encontrado na iteração corrente.

A Linha 10.3 computa o conjunto de todos os mínimos regionais localizados na região C e a Linha 10.4 armazena em n o número de mínimos regionais em M . A Linha 10.5 toma todos os mínimos em M e os insere na sequência ordenada A , de acordo com seus valores de extinção. As Linhas 10.6 até 10.8 inicializam os identificadores envolvidos na busca de um par válido de marcadores: A Linha 10.6 armazena em M_a o mínimo regional com o maior valor de extinção. A Linha 10.7 inicializa M_b como o conjunto vazio: isto significa que, neste ponto, não foi encontrado nenhum marcador que componha um par válido de marcadores com M_a . A Linha 10.8 inicializa k com 2, para iniciar a checagem dos mínimos em A a partir do mínimo com o segundo maior valor de extinção.

A estrutura **enquanto** nas Linhas 10.9 a 10.10 busca por um marcador em $A \setminus \{M_1\}$ que componha um par válido com M_1 . A Linha 10.9.1 aplica o algoritmo **Crit** em M_a e $M_k \in A$ (o k -ésimo elemento de A) para verificar a validação do par. Se **Crit** devolve 2, então o par é válido e M_b denotará M_k (Linha 10.9.1.1). A Linha 10.9.3 incrementa k . Se a execução retorna para a Linha 10.9 e $M_b \neq \emptyset$, então a execução da estrutura **while** encerra e segue para a Linha 10.11. Se a execução retorna para a Linha 10.9 e M_b é ainda o conjunto vazio mas k é maior que n , a execução também pára e vai para a Linha 10.11. É importante explicar a razão pela qual um dos marcadores do par é dado por M_1 e este é testado com os demais, ao invés de se testar todas as possibilidades, duas a duas.

Isto foi feito em função da complexidade de tal busca: se todas as possibilidades fossem testadas, a busca por um par válido teria, no pior caso, um custo computacional quadrático em função do número de marcadores contidos em C . Do modo como foi implementado, descarta-se muitas possibilidades, porém, a busca tem um custo computacional linear em função do número de marcadores contidos em C .

Se, na iteração corrente, a execução atinge a Linha 10.11 e M_b é ainda o conjunto vazio, significa que nenhum par válido de marcadores foi encontrado e a execução deve pular as Linhas 10.11.1 até Linha 10.12. Se M_b denota outro marcador, o bloco **se** nas Linhas 10.11 até 10.12 é executado. A Linha 10.11.1 incrementa lbl para calcular o novo rótulo a ser atribuído a w . A Linha 10.11.2 aplica o operador *watershed* com marcadores e particiona a imagem em função do par válido de marcadores. A Linha 10.11.3 realiza a intersecção da partição computada na Linha anterior com C para encontrar a região W_i onde w será atualizada com o novo rótulo lbl . Um novo abuso de notação ocorre na Linha 10.11.4, onde ocorre a atualização: w recebe o rótulo lbl apenas na região dada por W_i . Finalmente, as Linhas 10.11.5 e 10.11.6 inserem $Elem$ e lbl em Q , para processar as regiões que possuem estes rótulos *a posteriori*. Note que, nas iterações onde o bloco **se** não é executado, o comprimento da fila decresce em uma unidade. Este é o modo que a fila Q pode ser esvaziada e que possibilita com que o algoritmo chegue ao fim, devolvendo w na Linha 12.

5.4.3 A propriedade hierárquica da decomposição uniforme

Vamos discutir a construção de w para demonstrar a propriedade hierárquica da decomposição proposta. Seja $\mathcal{W}_i$ o particionamento da imagem de entrada na iteração i dentro do domínio de aplicação, onde cada subconjunto $W_j \in \mathcal{W}_i$ é dado por uma zona plana do estado corrente de w na i -ésima iteração. Seja $\mathcal{W} = (\mathcal{W}_i)$ uma família de partições tal que $\mathcal{W}_1 = D$ (i.e., considera-se aqui que o próprio domínio de aplicação é o produto da primeira iteração).

Note que um dos subconjuntos conexos $W_j \in \mathcal{W}_{i-1}$ é recuperado na iteração i e usado como o conjunto C , o i -ésimo domínio de validação e a região candidata a ser quebrada naquela iteração. Uma vez que somente a região C é processada na iteração i , os outros subconjuntos $W_k \in \mathcal{W}_{i-1}$ são preservados em $\mathcal{W}_i$. Cada um destes subconjuntos respeita a hierarquia: se $W_k \in \mathcal{W}_i$ também pertence a $\mathcal{W}_{i-1}$, então existe um subconjunto $W' \in \mathcal{W}_{i-1}$ tal que $W_k \subseteq W'$ e W' é o próprio conjunto W_k .

Agora, consideremos o subconjunto C e a busca por um par válido de marcadores dentro deste subconjunto. Se não for encontrado um par válido de marcadores dentro de C , então este não deve ser quebrado em dois novos subconjuntos. Uma vez que C permanece em $\mathcal{W}_i$ do modo como foi recuperado de $\mathcal{W}_{i-1}$, a hierarquia é preservada quando não ocorre a quebra de C .

Se um par válido de marcadores é encontrado na iteração i , então C deve ser quebrado e as Linhas 10.11 até 10.12 do algoritmo **DecomposicaoUniforme** executam esta quebra, onde o novo estado da imagem de saída w é computado. Sejam $E_a, E_b \in E$ dois subconjuntos disjuntos que particionam E em duas partes, dadas pela aplicação de $\mathcal{W}_{M_a, M_b}(g)$ na Linha 10.11.2. A região C será quebrada em duas novas regiões: sejam $C_a = C \cap E_a$ e $C_b = C \cap E_b$. Note que, uma vez que E_a e E_b são disjuntos e quebram E em apenas dois subconjuntos, então $C_a \cup C_b = C$ e $C_a \cap C_b = \emptyset$. Note que $C_a, C_b \subset \mathcal{W}_i$ e $C_a, C_b \subset C \in \mathcal{W}_{i-1}$, assim, a propriedade de hierarquia também é preservada quando ocorre a quebra.

5.4.4 Outras propriedades da decomposição uniforme

Além da propriedade hierárquica, a decomposição uniforme também apresenta outras características importantes:

- Dada uma imagem $f \in Fun[E, K]$ e um conjunto de domínios de aplicação $D_i \subseteq E : D_i \cap D_j = \emptyset, i \neq j$. A decomposição proposta pode ser aplicada independentemente em cada domínio D_i e um critério $crit_i$ e um parâmetro par_i podem ser ajustados para cada D_i . Esta vantagem é explorada na aplicação apresentada na seção 5.5.
- Todos os subconjuntos da partição dada pela decomposição respeitam o critério e o parâmetro aplicados no domínio de aplicação. I.e., todos as medidas dos subconjuntos são maiores que par de acordo com o critério $crit$. Isto provê uma decomposição com uma distribuição mais balanceada (ou uniforme) de subconjuntos.
- Uma vez que a distribuição de subconjuntos é mais balanceada, os objetos mais importantes tendem a ter uma representação mais robusta.
- A aplicação de critérios no cálculo da decomposição possibilita o projeto de operadores de imagens que preservem as estruturas representadas pelos subconjuntos da decomposição. Por exemplo, se o critério de área é aplicado, todos os subconjuntos da decomposição devem resistir à aplicação de uma filtragem por área (com parâmetro par). Um outro exemplo, com relação ao critério de disco, é que todos os subconjuntos resistem à erosões com certo e.e. (i.e., discos com diâmetro menor ou igual a par).

5.4.5 Resultados experimentais

Esta subseção apresenta vários resultados experimentais dados pela aplicação da decomposição uniforme. Resultados em segmentação hierárquica, possível pré-processamento para visão estéreo e segmentação interativa são apresentados e discutidos. A seção 5.5 é inteiramente dedicada à descrição de uma aplicação da decomposição uniforme como suporte para uma nova versão do *watershed* com marcadores propagados.

Segmentação hierárquica

A primeira aplicação apresentada neste capítulo é o uso da decomposição uniforme para segmentação hierárquica. Uma vez que esta é uma aplicação direta da decomposição, é preferível apresentar a aplicação com uma comparação com a clássica segmentação hierárquica por seleção de mínimos regionais através da limiarização de valores de extinção. Será demonstrado que a decomposição uniforme é comparável à segmentação hierárquica clássica em questão, além de manter o critério de decomposição para cada subconjunto da partição obtida.

A comparação será realizada entre três decomposições: a decomposição uniforme segundo o critério de área, a decomposição uniforme segundo o critério de disco e o método clássico de segmentação hierárquica. As três decomposições requerem diferentes tipos de parâmetros, assim, foi decidido realizar a comparação de acordo com o número de regiões geradas pelos métodos de segmentação. Os três métodos serão aplicados de modo a produzir o mesmo número de regiões e assim

os resultados de segmentação serão avaliados subjetivamente e quantitativamente pela aplicação de duas funções projetadas para avaliar o erro de uma segmentação.

A decomposição clássica é o modo mais fácil para segmentar uma imagem em n regiões: após computar o valor de extinção para cada mínimo regional, o n -ésimo maior valor de extinção é tomado como um valor de limiarização para selecionar os n mínimos regionais com os maiores valores de extinção¹. Não há um método direto para segmentar uma imagem em n regiões pela aplicação da decomposição uniforme e esta é uma desvantagem do método. Assim, para comparar a decomposição uniforme com a clássica, foram escolhidos parâmetros de área e disco que geram segmentações com o mesmo número de regiões e, depois, este mesmo número foi usado para segmentar a imagem usando o método clássico.

Três experimentos foram realizados. No primeiro, duas decomposições uniformes foram realizadas pela aplicação de um parâmetro de área igual a 5000 e um parâmetro de disco igual a 33. Ambas as decomposições geraram 13 regiões (Fig. 5.5 (a) e (b)). Para completar o primeiro experimento, o método clássico foi aplicado - usando-se uma função de extinção por área - para obter uma segmentação com 13 regiões (Fig. 5.5 (c)). O segundo experimento aplicou parâmetro de área igual a 1600 e parâmetro de disco igual a 20. As duas segmentações geraram 34 regiões (Fig. 5.5 (d) e (e)) e foram comparadas com as 34 regiões geradas pela aplicação do método clássico que também utilizou uma função de extinção por área (Fig. 5.5 (f)). Finalmente, o terceiro experimento comparou três decomposições que geraram 101 regiões (Fig. 5.5 (g) a (i)). Os parâmetro de área e disco foram, respectivamente, 500 e 10. No terceiro experimento, o método clássico também usou a função de extinção por área.

Seguindo um critério subjetivo, pode-se ver que a segmentação gerada pela decomposição uniforme é comparável com a gerada pelo método clássico. Além da decomposição uniforme apresentar resultados similares aos apresentados pelo método clássico, os subconjuntos contidos nos resultados uniformes respeitam os critérios aplicados para gerá-los. Por exemplo, na Fig. 5.5 (d), nenhuma região tem área menor que 1600 pixels e, na Fig. 5.5 (e), todas as regiões resistem a uma erosão com e.e. de raio igual a 10. Tais preservações de estruturas não ocorrem na decomposição clássica.

Para demonstrar a similaridade entre ambas as decomposições uniformes e a clássica, foi realizado um experimento de avaliação. As avaliações das segmentações foram feitas com a aplicação de duas funções propostas por Borsotti et al. [80] para avaliar segmentação de imagens coloridas. Ambas as funções avaliam a segmentação de imagens coloridas de acordo com critérios heurísticos tais como homogeneidade e simplicidade (veja [80] para maiores detalhes).

Quando comparando os resultados das segmentações, os menores resultados são dados pelas melhores segmentações de acordo com os critérios supracitados. Ambas as funções são definidas abaixo:

$$F'(I) = \frac{1}{10000 \times N \times M} \times \sqrt{\sum_{A=1}^{Max} [R(A)]^{1+1/A}} \times \sum_{i=1}^R \frac{e_i^2}{\sqrt{A_i}} \quad (5.7)$$

and

$$Q(I) = \frac{1}{10000 \times N \times M} \times \sqrt{R} \times \sum_{i=1}^R \left[\frac{e_i^2}{1 + \log A_i} + \left(\frac{R(A_i)}{A_i} \right)^2 \right], \quad (5.8)$$

¹ Assuma, por simplicidade, que cada mínimo regional tem um valor de extinção distinto.

onde N e M são, respectivamente, a altura e a largura da imagem, $R(A)$ é o número de regiões que possuem área igual a A , Max é a maior área segmentada, R é o número de regiões em que a imagem foi segmentada, A_i é a área da região i e e_i é o erro médio de cor para a região i .

Tabelas 5.2 e 5.3 mostram, respectivamente, os resultados da avaliação dados pela aplicação das funções $F'(I)$ e $Q(I)$ propostas por Borsotti et al. Note que os valores são muito similares entre si e cada decomposição vence pelo menos em uma situação: a decomposição uniforme por área venceu para 101 regiões com a função $Q(I)$; a decomposição uniforme por disco venceu para 13 regiões com ambas as funções $F'(I)$ e $Q(I)$; e a decomposição clássica venceu os demais três casos. Porém, nenhuma condição de vitória foi significativa o bastante para destacar uma decomposição em particular, pois as diferenças entre os erros foram muito pequenas. Quantitativamente, as três segmentações foram equivalentes. As E, lembrando, a decomposição uniforme ainda mantém as vantagens dadas pelos critérios.

Método \ Regiões	13	34	101
Clássica	57,3692	30,8142	23,0695
Área	57,4575	49,3614	23,1594
Disco	55,3578	49,4325	51,5475

Tab. 5.2: Erros de segmentação avaliados pela função $F'(I)$ de Borsotti et al.

Método \ Regiões	13	34	101
Clássica	581,4834	229,1602	110,9979
Área	535,1354	319,6443	103,0404
Disco	524,6157	325,5783	270,1857

Tab. 5.3: Erros de segmentação avaliados pela função $Q(I)$ de Borsotti et al.

Possível pré-processamento para visão estéreo

A segunda aplicação apresentada neste capítulo é o casamento de um par de imagens estéreo, para processá-las antes da estimação do deslocamento de cada objeto no contexto do registro de imagens. Basicamente, o método de pré-processamento proposto objetiva segmentar os objetos contidos em uma imagem e usá-los como marcadores para segmentar os mesmo objetos na outra imagem. Não é inserida nenhuma semântica ao método de pré-processamento: a segmentação dos objetos é dada pela decomposição uniforme da Imagem Um, usando como critério o disco com raio p . Os subconjuntos são então erodidos separadamente por um e.e. disco com raio não maior que $p/2$. Estes subconjuntos erodidos serão os marcadores aplicados para segmentar a Imagem Dois via *watershed* com marcadores.

Este *framework* de casamento proposto aqui foi projetado sob a suposição de que os objetos contidos na Imagem Um estejam também contidos na Imagem Dois, considerando-se um deslocamento razoável. Dado um objeto representado por um subconjunto da partição da Imagem Um, é esperado

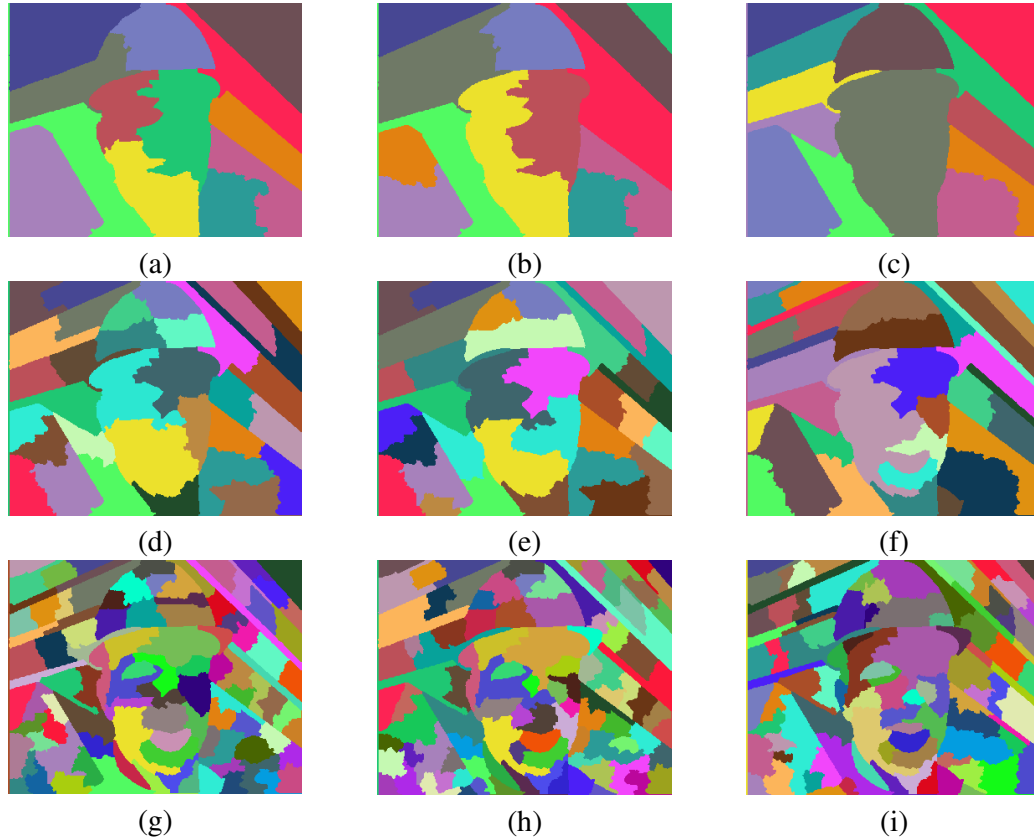


Fig. 5.5: Segmentação hierárquica. *Primeira linha* : 13 regiões. (a) Critério de área. (b) Critério de disco. (c) Clássica (limiarização dos valores de extinção). *Segunda linha* : 34 regiões. (d) Critério de área. (e) Critério de disco. (f) Clássica (limiarização dos valores de extinção). *Terceira linha* : 101 regiões. (g) Critério de área. (h) Critério de disco. (i) Clássica (limiarização dos valores de extinção).

que a erosão deste subconjunto, por um e.e. cujo raio é escolhido em função do deslocamento da imagem, proveja um marcador localizado sobre o mesmo objeto na Imagem Dois.

A seguir, é apresentado o resultado do pré-processamento descrito nos parágrafos anteriores. O método é aplicado em um par de imagens extraídas de um dos *datasets* disponíveis em:

<http://vision.middlebury.edu/stereo/>

A Fig. 5.6 ilustra a aplicação da decomposição uniforme para realizar o casamento das imagens. A Figura 5.6 (a) mostra o deslocamento das duas Imagens Um e Dois (este deslocamento foi calculado pela diferença simétrica entre as versões em níveis de cinza de ambas as imagens). A Fig. 5.6 (c) mostra a decomposição uniforme da Imagem Um, usando como critério um disco com raio 20. Todas as regiões foram erodidas separadamente por um disco com raio 10 e usadas como um conjunto de marcadores (Fig. 5.6 (b)) para segmentar a Imagem Dois (Fig. 5.6 (e)).

As Fig. 5.6 (c) e (e) são mostradas em uma coluna para facilitar para o leitor a percepção do deslocamento horizontal da Imagem Um para a Imagem Dois e os rótulos atribuídos aos subconjuntos que representam o mesmo objeto são também os mesmos. Note que a segmentação da Imagem Dois em função dos marcadores gerados pela decomposição da Imagem Um é muito boa. As linhas de ambas as segmentações foram extraídas e sobrepostas às imagens originais para tornar possível ver quais objetos foram casados. As Imagens Um e Dois sobrepostas são mostradas respectivamente nas Fig. 5.6 (d) e (f). Novamente, elas são mostradas de modo a enfatizar o deslocamento horizontal do par de imagens.

Segmentação multiescala por quebra interativa de marcadores

Nesta aplicação, a decomposição uniforme é usada para a segmentação interativa de imagens estáticas. A idéia nesta abordagem não é adicionar ou remover marcadores para segmentar objetos através da aplicação do *watershed* com marcadores. As regiões da imagem são segmentadas pelo controle local da hierarquia em cada região.

O usuário aponta o cursor do *mouse* para uma dada região e seleciona as operações desejadas com os botões do *mouse*. A ativação de tais operações depende do mode corrente em que a interface se encontra. Há dois modos de interface: no *modo de decomposição*, o usuário segmenta hierarquicamente a região apontada pelo cursor do *mouse* - que será usada como um domínio local de aplicação - usando-se os botões do *mouse* para controlar a hierarquia. O botão esquerdo diminui a hierarquia em uma região enquanto o botão direito aumenta a hierarquia. No *modo de edição*, o usuário pode escolher duas operações: quebrar uma região em duas partes usando a mesma decomposição ou definir, através de um chaveamento, se uma região pertence ao objeto a ser segmentado ou ao fundo da imagem. Exceto no modo de decomposição onde o usuário sobe ou desce na hierarquia de uma dada região, não há junção (união) de regiões neste *framework* de segmentação. Uma vez que a operação de decomposição é considerada terminada em uma região, não é possível reunir as regiões quebradas nas operações seguintes.

As Fig. 5.7 amostram vários instantes de uma sequência de operações sobre a imagem de entrada extraída da sequência *Foreman*. Cada amostra apresenta um instante significativo onde parte do operário é segmentada. A Fig. 5.7 (a) mostra a decomposição e chaveamentos de uma região na região superior esquerda da imagem, após uma decomposição inicial e global. As Fig. 5.7 (b-d)

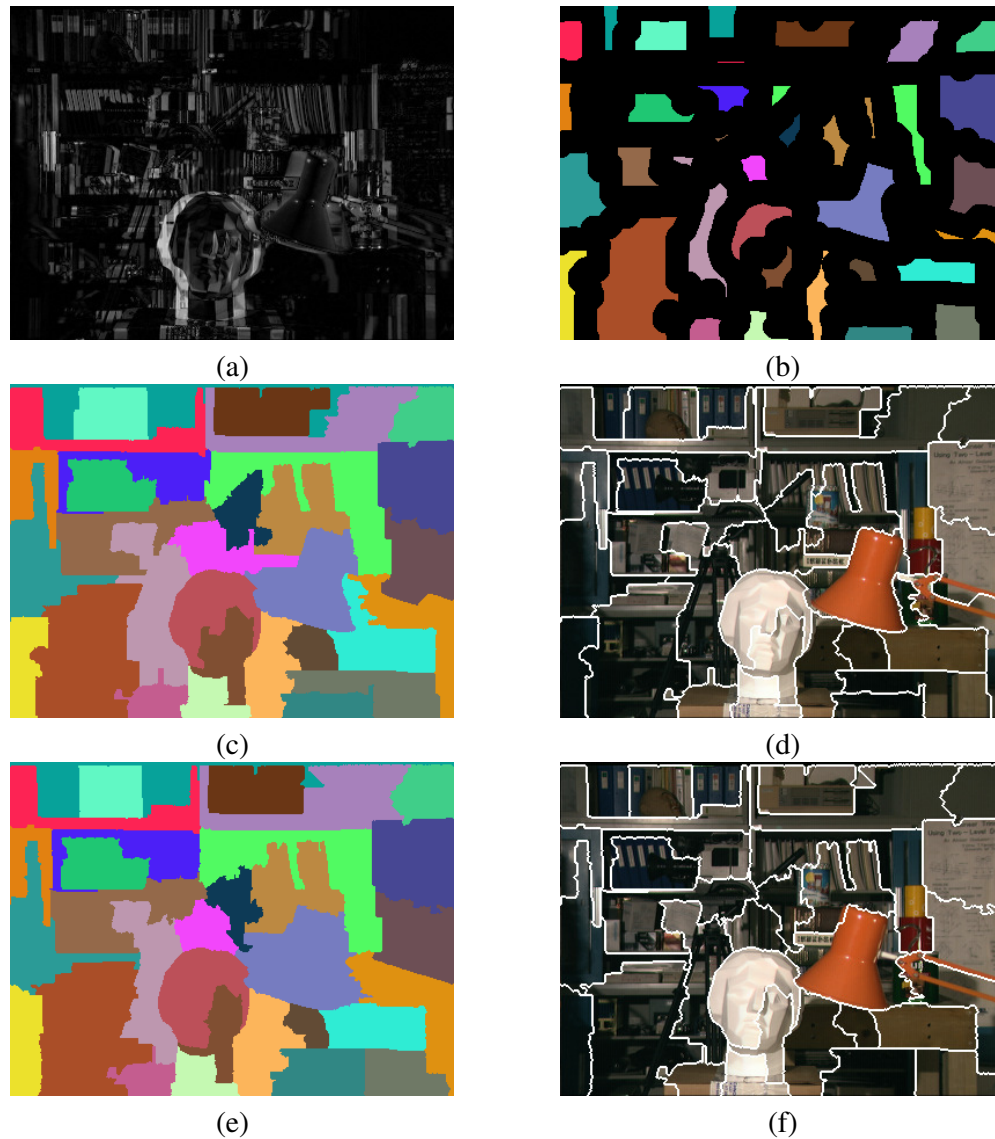


Fig. 5.6: Pré-processamento ao registro de imagens. (a) Deslocamento da imagem. (b) Marcadores aplicados para segmentar a Imagem Dois. (c) Decomposição uniforme da Imagem Um. (d) Imagem Um (com o resultado da segmentação sobreposto). (e) Segmentação da Imagem Dois. (f) Imagem Dois (com o resultado da segmentação sobreposto).

mostram as decomposições e chaveamentos de outras regiões da imagem. A Fig. 5.7 (d) mostra a composição do resultado final com a imagem de entrada.

5.5 A Decomposição Uniforme e o *Watershed* com Marcadores Propagados

A primeira proposta do *watershed* com marcadores propagados computava segmentos pequenos próximos às bordas dos objetos e os aplicava como marcadores [17]. Um aperfeiçoamento da técnica do *watershed* com marcadores propagados foi proposta em [4, 26] : o *amarramento de marcadores*. Ele consiste na computação de pares de marcadores ao longo da borda, e cada par é composto por um marcador interno outro externo. Ambos os marcadores do par devem ser propagados pelo mesmo vetor deslocamento, e este vetor é computado pela estimação de movimento da área *entre* o par de marcadores. Esta heurística provê mais informação para a estimação de movimento e auxilia com que o movimento do par de marcadores siga o movimento da borda que cruza a região entre os marcadores no quadro anterior.

A seguir, é introduzido um novo aprimoramento para o *watershed* com marcadores propagados. Neste aprimoramento, o *watershed* com marcadores propagados é aprimorado pela aplicação da decomposição uniforme, que é aplicada para computar os conjuntos de marcadores internos e externos.

1. O gradiente temporal (veja Capítulo 4) é calculado para toda a sequência.
2. Os objetos de interesse são segmentados pelo *watershed* interativo com marcadores, aplicado no gradiente temporal correspondente ao quadro corrente (a Fig. 5.8 (a) mostra o quadro corrente onde o *Foreman* será segmentado).
3. A máscara do objeto segmentado define o domínio de aplicação D_f que é a região onde a decomposição uniforme será aplicada (Fig. 5.8 (b)). Sejam $crit_f$ e par_f , respectivamente, o critério e o parâmetro atribuídos ao objeto de interesse. A partição de D_f resulta em um conjunto de pré-marcadores P_f atribuídos ao objeto de interesse (Fig. 5.8 (c)).
4. Muitos subconjuntos de P_f não tocam as bordas do objeto segmentado e, assim, são descartados uma vez que os marcadores restantes são suficientes para perseguir o objeto, já que são próximos às bordas do objeto. O conjunto de marcadores internos M_f a serem atribuídos ao objeto de interesse é dado por estes pré-marcadores remanescentes (Fig. 5.8 (e)).
5. Considere a negação da máscara de segmentação como o domínio de aplicação D_b (Fig. 5.8 (d)). Aplique os mesmos procedimentos descritos nos Passos 2 e 3 para o fundo (*background*), considerando D_b , $crit_b$ e par_b para computar o conjunto de marcadores externos M_b (Fig. 5.8 (f)).
6. Os marcadores em M_f e M_b são propagados pela aplicação da técnica de Lucas e Kanade [6], levando em conta o quadro corrente e o quadro seguinte. Dado um marcador M de M_f ou M_b , seu vetor de movimento é computado considerando-se a região inteira definida por M . Porém, o marcador propagado para o quadro seguinte é dado pela erosão de M por um e.e disco, cujo diâmetro é um parâmetro de velocidade (Fig. 5.9 (a) and (b)).

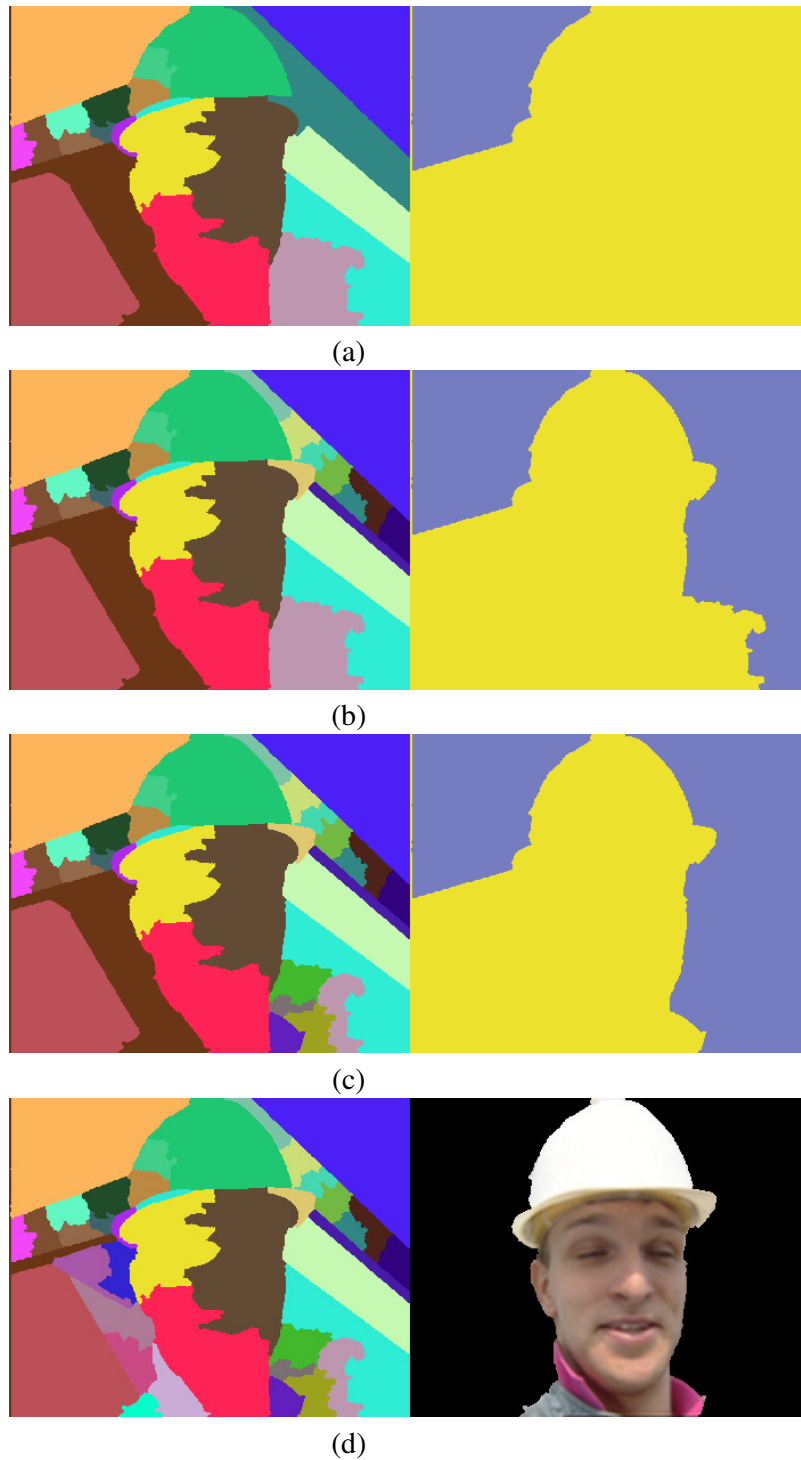


Fig. 5.7: Segmentação interativa. Decomposições e chaveamentos da (a) região superior esquerda; (b) região superior direita; (c) região inferior direita; (d) região inferior esquerda (composta com a imagem original).

7. A colisão de marcadores internos com externos pode levar a resultados ruins de segmentação. Isto é resolvido pelo cálculo da diferença simétrica entre os dois conjuntos de marcadores, dilatando-se esta diferença e subtraindo-se esta dilatação de ambos os conjuntos M_f e M_b .
8. Vá para o quadro seguinte e aplique os marcadores propagados (Fig. 5.9 (c)) para segmentar a máscara inicial para este novo quadro (Fig. 5.9 (d)). A segmentação é realizada pela aplicação do *watershed* com marcadores, usando-se os marcadores propagados para o novo quadro e o gradiente temporal calculado para o referido quadro. Se necessário, o usuário pode melhorar os resultados de segmentação pela interação com os marcadores.
9. Vá para o Passo 3, até a sequência ter sido inteiramente processada.

A decomposição uniforme é aplicada ao *framework* do *watershed* com marcadores propagados para que marcadores maiores e mais consistentes sejam computados. Este aprimoramento apresenta diversas vantagens:

- Os marcadores tendem a resistir à erosão feita em função da velocidade dos objetos - desde que os parâmetros estejam corretamente ajustados.
- Uma vez que o tamanho dos marcadores tende a ser maior que o dos marcadores nas versões prévias do *watershed* com marcadores, muito mais informação é disponibilizada para o estimador de movimento. Além do tamanho, a região definida pelo marcador é também mais homogênea.
- Uma vez que um marcador tende a representar um objeto na cena, a estimação de movimento por Lucas-Kanade é uma boa escolha para a propagação de marcadores, porque este assume que o movimento dentro de uma região é constante.
- Os marcadores são maiores e mais ajustados às bordas do objeto, o que melhora os resultados da segmentação quando são bem propagados.
- Se um marcador não é bem propagado, o seu tamanho facilita a tarefa de edição ou remoção do marcador mal propagado.
- A segmentação hierárquica interativa pode ser facilmente integrada ao passo interativo do *watershed* com marcadores propagados, quando a interferência do usuário é requerida.

5.5.1 Resultados experimentais

O experimento descrito abaixo mostra uma avaliação quantitativa da segmentação da sequência *Foreman* usando o *watershed* com marcadores propagados aprimorado pela aplicação da decomposição uniforme. Esta avaliação é feita pela aplicação do *benchmark* proposto em [25] que avalia várias características tais como o número de interferências do usuário, o tempo requerido para segmentar a sequência e a quantidade de movimento da mesma. Os resultados experimentais cujos dados são apresentados nas Tabelas abaixo podem ser assistidos nos vídeos disponíveis no website:



Fig. 5.8: Aplicação da decomposição uniforme no *framework* do *watershed* com marcadores propagados. (a) Quadro corrente. (b) Composição com a máscara corrente de segmentação D_f . (c) Pré-marcadores internos computados pela decomposição uniforme do quadro corrente de acordo com D_f . (d) Pré-marcadores externos. (e) Conjunto de marcadores internos M_f . (f) Conjunto de marcadores externos M_b .

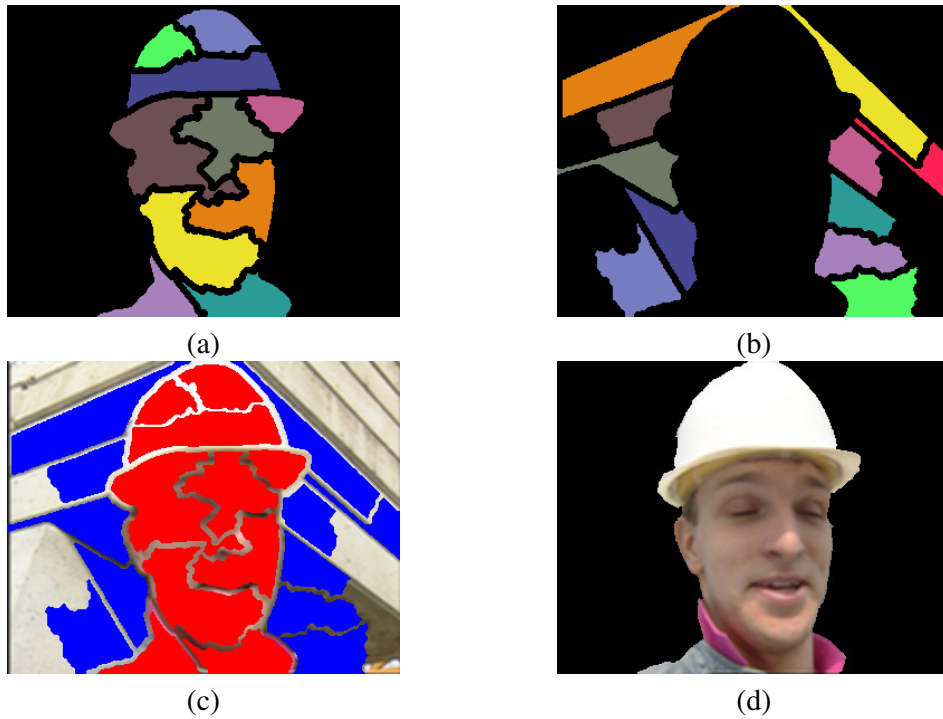


Fig. 5.9: Aplicação da decomposição uniforme no *framework* do *watershed* com marcadores propagados. (a) Erosão dos marcadores M_f . (b) Erosão dos marcadores M_b . (c) Marcadores internos e externos propagados e sobrepostos ao quadro seguinte. (d) Segmentação do quadro seguinte.

<http://calhau.dca.fee.unicamp.br/wiki/index.php/Usuário:Fcflores>

Para demonstrar o *benchmark* e a robustez do *watershed* com marcadores propagados, o operário foi segmentado da sequência *Foreman*. Esta sequência tem características interessantes que permitem ilustrar algumas propriedades do método proposto neste capítulo para a segmentação de objetos em sequências de imagens.

Vários autores usam a sequência *Foreman* para demonstrar a performance de seus métodos, desde codificação de vídeo [69, 65, 68] até técnicas de segmentação de sequências. Infelizmente, é difícil realizar uma comparação justa entre nosso método e muitos outros importantes que segmentam *Foreman* porque tais métodos, em sua grande maioria, consistem em segmentações automáticas.

Foreman é uma sequência interessante porque apresenta muitas situações de teste importantes (Fig. 5.10). O próprio operário, por exemplo, enquanto aparece na cena, move sua cabeça em muitas direções (esquerda, direita, para frente e para trás, o que gera uma sensação de *zooming* da face do operário). A cabeça do operário às vezes gira, o que resulta em um tipo de deformação do objeto de interesse. A câmera também se move de modo suave porém não controlado. Se o operário é o objeto de interesse, note que ele é composto de vários objetos que precisam ser segmentados e perseguidos: o capacete, sua face e sua jaqueta. Além disso, a sequência apresenta várias regiões de baixo contraste tais como o ombro do operário, suas orelhas e a região onde o capacete se encontra com o fundo branco de concreto. Segmentar apropriadamente a sequência *Foreman* não é uma tarefa trivial.

O experimento consiste na avaliação de dois métodos aplicados para segmentar o operário nos primeiros 150 quadros da sequência *Foreman*:

- O método manual;
- O *watershed* com marcadores propagados com suporte da decomposição uniforme e da estimação de movimento pelo método de Lucas-Kanade;

A Fig. 5.11 apresenta a informação de movimento da sequência *Foreman*, dada a sua segmentação Padrão-Ouro. O operário se move em uma velocidade razoável por quase toda a sequência (note que a diferença entre quadros consecutivos é aproximadamente 1,5% na maioria dos quadros. O erro médio entre quadros consecutivos é 1,7652%). As exceções ocorrem no fim da sequência onde o operário se move mais rapidamente e no quadro **90**, onde a mão do operário entra na cena, desaparecendo no quadro seguinte (Fig. 5.10. Este evento também pode ser observado na avaliação do segundo método - veja a Fig. 5.15).

O Padrão-Ouro é dado pela segmentação manual da sequência *Foreman* (Fig. 5.12). Foram necessárias 3990 interferências do usuário - média de 26,6 interferências por quadro - para completar a segmentação manual. O tempo total gasto para completar toda a segmentação foi de 12266 segundos = 3,4073 horas. Note na Fig. 5.13 que o número de interferências e o tempo gasto por quadro é maior no final da sequência, porque o operário se move mais rapidamente naquele instante (veja a Fig. 5.11).

A segunda avaliação foi feita com a aplicação do *watershed* com marcadores propagados - com suporte da decomposição uniforme e do estimador de movimento por Lucas-Kanade (Fig. 5.14). As Fig. 5.15 (a) e (b) mostram, respectivamente, a quantidade de interferência requerida e o tempo gasto para segmentar o operário em cada sequência. A segunda segmentação requereu 132 interferências

(média de 0,88 interferências por quadro) e foi totalmente realizada em 1241,786 segundos = 20,6964 minutos (média de 8,2786 segundos por quadro). Note os intervalos na sequência onde a interferência não foi necessária (Fig. 5.15 (a)).

O erro de segmentação para cada quadro é mostrado na Fig. 5.16. O erro devido à má propagação de marcadores foi baixo e a segmentação de algumas regiões críticas foi melhorada, mas ainda alguns pontos críticos, onde a segmentação foi ruim requereram alguma interferência. O erro médio de segmentação ocorrido devido à aplicação do *watershed* com marcadores propagados, com suporte da decomposição uniforme e do estimador Lucas-Kanade, foi de 1,1642% por quadro.

5.6 Conclusão

Este capítulo apresentou uma nova variação para o *framework* do *watershed* com marcadores propagados. Esta variação baseia-se na aplicação da decomposição uniforme para a geração de marcadores internos e externos. Tais marcadores, maiores, mais consistentes e homogêneos promovem uma diferente maneira de interação com resultados de segmentação que são, usualmente, de boa qualidade.

Os marcadores gerados pela decomposição uniforme são mais consistentes pois esta segmentação hierárquica permite o controle da decomposição da imagem em função de um critério. O critério de área, por exemplo, permite a construção de marcadores maiores e mais homogêneos. O critério de disco, por sua vez, permite que os marcadores gerados resistam a aplicação de erosão com certos elementos estruturantes em formato de disco (para ser mais exato, discos com diâmetro menor que o parâmetro usado com o critério de disco).

A decomposição uniforme é uma segmentação hierárquica cuja propriedade principal é a preservação de uma dada característica para cada subconjunto da partição gerada pela decomposição. É um método incremental onde, em cada iteração, uma região candidata é dividida em dois subconjuntos que respeitem critérios como tamanho ou forma. A decomposição uniforme devolve uma partição da imagem tal que nenhum dos subconjuntos desta partição desrespeite o critério de entrada aplicado. Em outras palavras, nenhum subconjunto da partição, após medido segundo o critério escolhido, terá medida menor que o parâmetro numérico associado ao critério.

A decomposição proposta neste capítulo é integrada no contexto do *watershed* com marcadores propagados da seguinte maneira: os marcadores internos são calculados considerando-se a máscara de segmentação como domínio da aplicação da decomposição. Os subconjuntos desta partição são, então, erodidos e propagados. Similarmente, os marcadores externos são calculados considerando-se a negação da máscara de segmentação como domínio de aplicação da decomposição uniforme e os subconjuntos obtidos são, também, erodidos e propagados. Por fim, marcadores internos e externos que se interceptem após as propagações são corrigidos pela eliminação de suas intersecções.

O *benchmark* proposto no capítulo 3 foi aplicado para avaliar o *watershed* com marcadores propagados com suporte da decomposição uniforme e com estimação de movimento pelo método de Lucas e Kanade. Nesta avaliação, foi segmentado o operário nos 150 primeiros quadros da sequência *Foreman*. Foi requerida, em média, 0,88 interferências por quadro, em um tempo de 20,6964 minutos (média de 8,2786 segundos por quadro). O erro de segmentação em algumas regiões críticas e em função de má propagação de marcadores foi baixo (erro médio de 1,1642% por quadro). Alguns pontos críticos, em alguns quadros, requereram interferência do usuário.

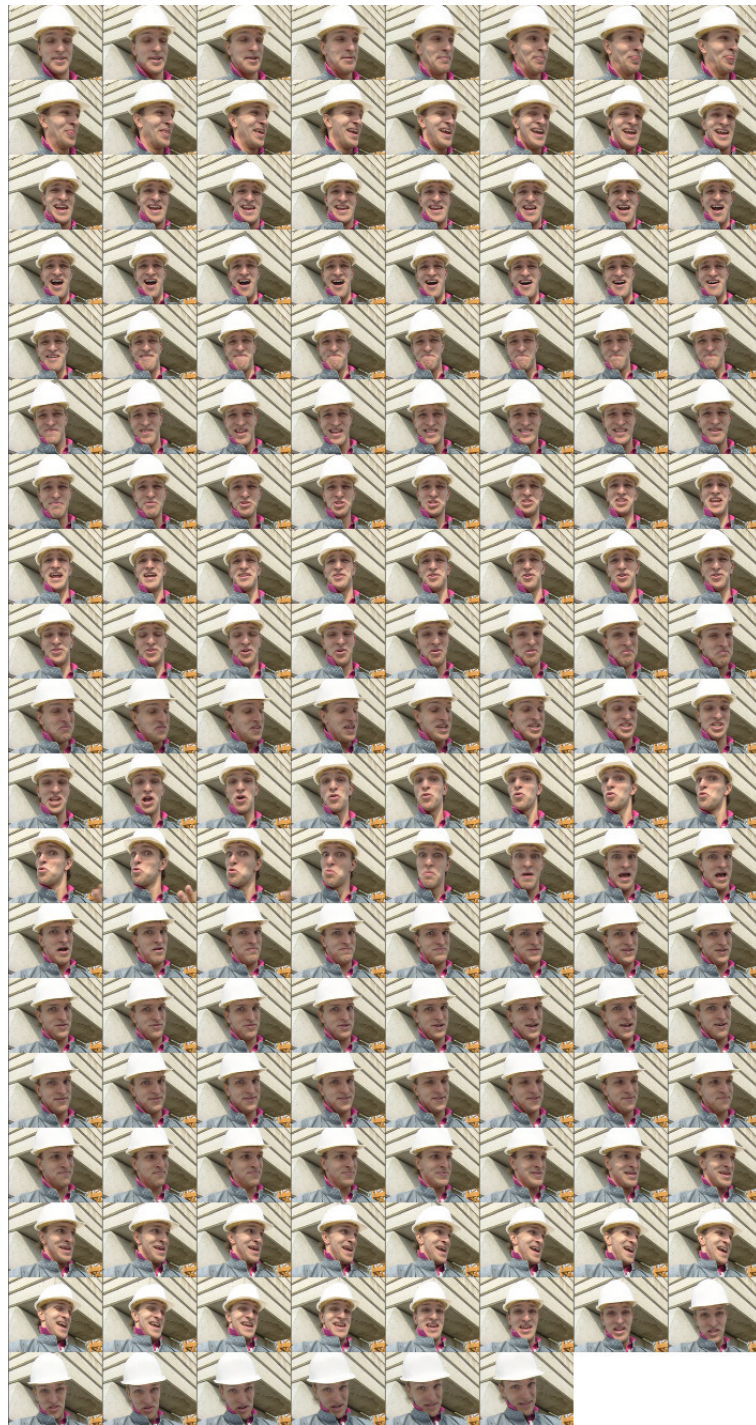


Fig. 5.10: Sequência *Foreman* (quadros 1 a 150).

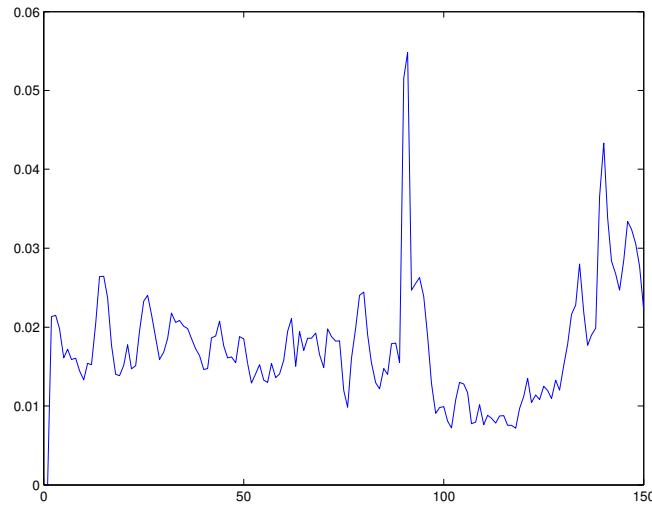


Fig. 5.11: Informação de movimento da sequência *Foreman*.

A principal contribuição deste capítulo é uma variação do *watershed* com marcadores propagados que apresenta resultados robustos de segmentação e com baixo trabalho do usuário, além de prover um conjunto de marcadores mais consistentes e mais fáceis de editar. A decomposição uniforme, responsável pela geração de marcadores no *watershed* com marcadores propagados proposto neste capítulo, também é uma contribuição deste trabalho e que merece desenvolvimentos futuros.

Além da aplicação no *framework* do *watershed* com marcadores propagados, apresentamos outras aplicações para a decomposição uniforme, tais como possível pré-processamento para visão estéreo, segmentação interativa - que pode ser usada também no processo interativo do *watershed* com marcadores propagados - e segmentação hierárquica. neste último exemplo, experimentos mostraram que a decomposição uniforme compete quantitativamente com a segmentação hierárquica clássica por valores de extinção, e tem sobre esta a vantagem de preservar as propriedades dos subconjuntos da partição de acordo com o critério escolhido.

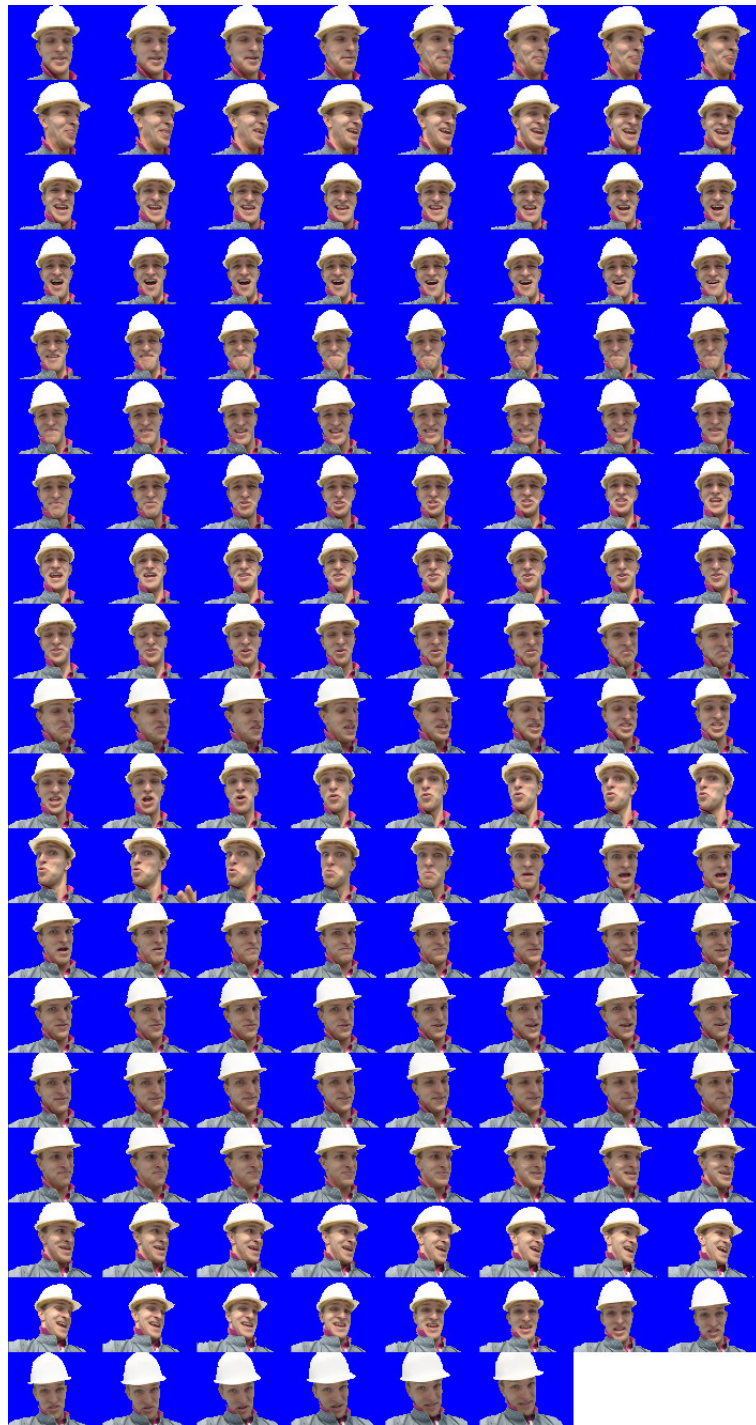


Fig. 5.12: Segmentação manual (Padrão-Ouro) da sequência *Foreman* (quadros 1 a 150).

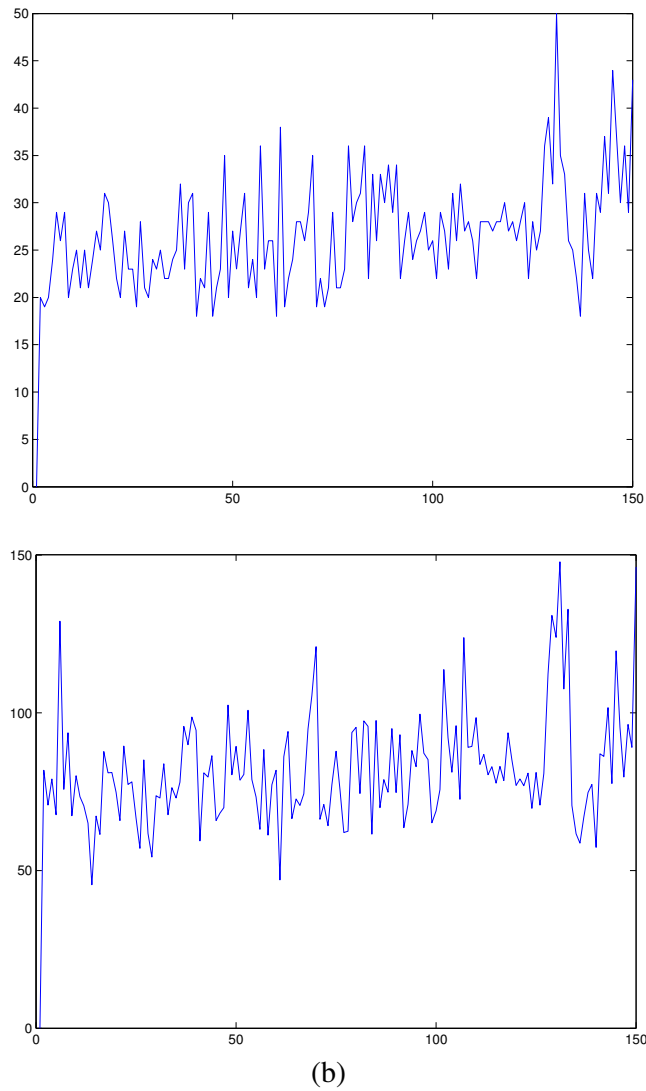


Fig. 5.13: Análise da segmentação Padrão-Ouro (manual) da sequência *Foreman*. (a) Número de interferências do usuário em cada quadro. (b) Tempo gasto na segmentação de cada quadro (em segundos).



Fig. 5.14: Segmentação da sequência *Foreman* por *watershed* com marcadores propagados - com suporte da decomposição uniforme e da estimação de movimentos por Lucas-Kanade (quadros 1 a 150).

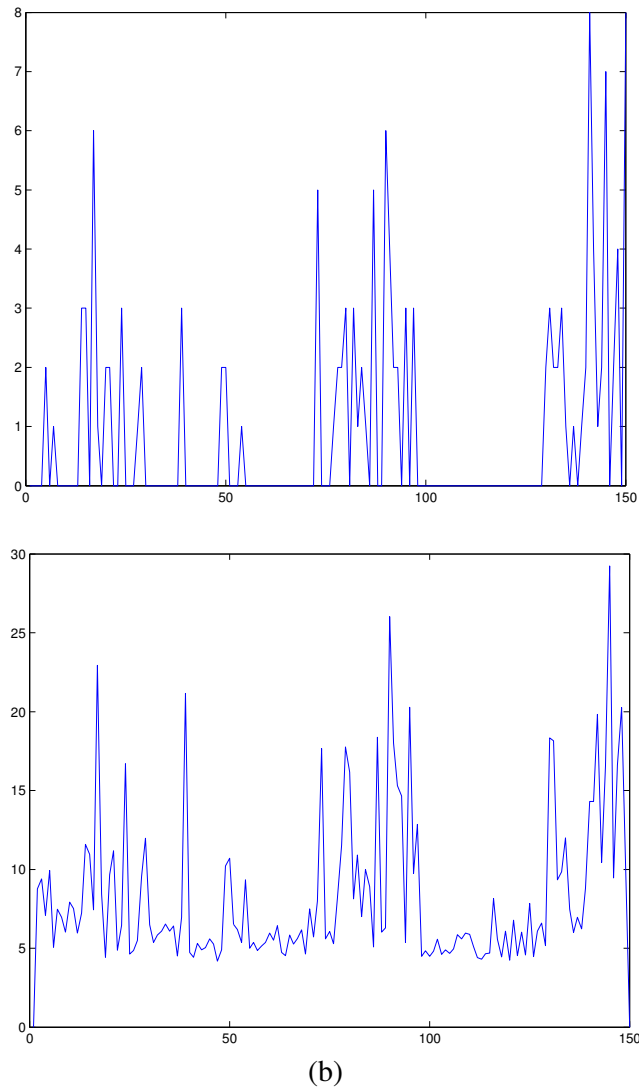


Fig. 5.15: Análise da segmentação da sequência *Foreman* por *watershed* com marcadores propagados - com suporte da decomposição uniforme e da estimação de movimentos por Lucas-Kanade. (a) Número de interferências do usuário em cada quadro. (b) Tempo gasto na segmentação de cada quadro (em segundos).

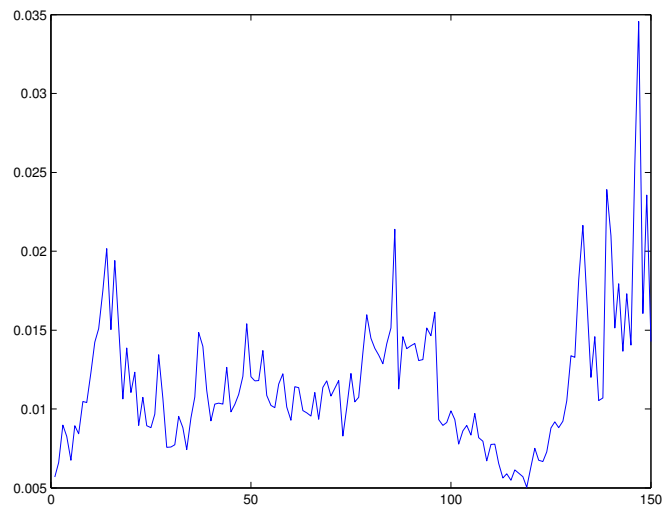


Fig. 5.16: Análise da segmentação da sequência *Foreman* por *watershed* com marcadores propagados - com suporte da decomposição uniforme e da estimação de movimentos por Lucas-Kanade. Erro de segmentação em cada quadro (comparado com o Padrão-Ouro.)

Capítulo 6

Conclusão

Segmentação assistida de objetos em sequência de imagens é a segmentação interativa de um ou mais objetos, definidos semanticamente, ao longo de uma sequência de imagens. É permitido ao usuário, em tal segmentação, definir os objetos a serem segmentados e interferir nos resultados de segmentação, seja para corrigir ou melhorar tais resultados. Um método assistido para segmentação de objetos em sequências de imagens deve apresentar importantes características tais como a interatividade, a generalidade, resposta rápida e edição manual progressiva.

Esta tese teve como objetivo principal apresentar um método para segmentação assistida de objetos em sequências de imagens, formulado de modo a apresentar as desejáveis propriedades supracitadas. Tal método, denominado **watershed com marcadores propagados**, consiste na criação de marcadores próximos às bordas do objeto de interesse e na propagação destes marcadores para o próximo quadro de modo a segmentar o mesmo objeto de interesse no quadro em questão.

Basicamente, os objetos de interesse em um dado quadro são segmentados da seguinte forma: com exceção do primeiro quadro, onde os objetos de interesse são segmentados manualmente, a segmentação inicial dos objetos no quadro corrente é realizada pelo uso de marcadores computados em função da segmentação dos mesmos objetos segmentados no quadro anterior. A segmentação inicial no quadro corrente é corrigida por uma eventual inserção/remoção interativa de marcadores e a segmentação desejada para o referido quadro é obtida. A seguir, esta segmentação é usada para calcular os marcadores a serem propagados para o quadro seguinte. A segmentação dos objetos em um dado quadro é realizada pela aplicação do clássico método de *watershed* com marcadores e a propagação é feita através da aplicação de técnicas de estimação de movimento.

Para realizar a tarefa da segmentação dos objetos em um determinado quadro, escolhemos o método de *watershed* com marcadores por ser um método de partição ótima da imagem [58], adequado à segmentação interativa e que permite implementação muito eficiente.

Em relação à propagação dos marcadores, experimentos foram realizados para comparar alguns métodos de estimação de movimento. O método escolhido, nesta tese, para ser aplicado como estimador de movimento foi o proposto por Lucas e Kanade. Além de ter provido os melhores resultados experimentais, o método de Lucas-Kanade foi proposto, e é aplicado, sob a suposição de que o movimento da imagem dentro de uma determinada região é constante. Como nesta tese consideramos que os marcadores não se deformam, todos os pontos que compõem um marcador devem ser transladados pelo mesmo vetor-deslocamento, ou seja, podemos assumir que o movimento localizado na região ocupada pelo marcador, antes da propagação, é constante. O que entra em consonância com a su-

posição do método de propagação.

6.1 Principais resultados

Esta tese apresenta um método assistido para segmentação de objetos em sequências de imagens - o *watershed* com marcadores propagados. O método foi apresentado nesta tese em duas modalidades de criação de marcadores: i) baseada no contorno da máscara de segmentação e ii) baseada na decomposição. A primeira modalidade apresenta marcadores curtos, próximos à borda, mais adaptados a objetos deformáveis e de fácil edição. A segunda modalidade, por sua vez, gera marcadores maiores, mais consistentes e que provêm mais informação para o estimador de movimento. Experimentos mostraram que a combinação do método de amarração de marcadores com gradiente espaço-temporal e com estimação de movimento proposta por Lucas e Kanade geraram melhores resultados. Isso será discutido abaixo.

6.1.1 Variações do paradigma de segmentação proposto

A melhoria por amarração de marcadores, baseada no contorno da máscara de segmentação, é a que apresentou melhores resultados entre as melhorias propostas nesta tese. Esta técnica, baseada no contorno da máscara de segmentação, consiste no cálculo de pares de marcadores ao longo da borda e cada par é composto por um marcador interno e outro externo que devem ser propagados pelo mesmo vetor-deslocamento. A idéia é fornecer ao estimador de movimento uma quantidade maior de informação que, nesta contribuição, é extraída da região entre os marcadores do par. A amarração de marcadores recebe como parâmetros o comprimento de cada marcador e a distância entre os marcadores que compõem o par. Experimentos mostraram que existe um *trade-off* na escolha destes parâmetros: quanto maior a área entre os marcadores, melhor a propagação e pior o resultado de segmentação.

A variação por ajuste pela média local dos vetores-deslocamento, também baseada no contorno da máscara de segmentação, consiste no ajuste da propagação de cada marcador pelo cálculo da média local dos vetores-deslocamento. Esta variação apresentou resultados melhores que os apresentados pelo paradigma como foi proposto inicialmente. No entanto, por não apresentar resultados tão eficientes quanto os obtidos pela variação por amarração de marcadores, não foi explorada em maior profundidade.

A proposta baseada na segmentação hierárquica da máscara de segmentação consiste no uso da decomposição uniforme para o cálculo dos marcadores internos e externos a serem propagados. Neste *framework*, usa-se a máscara de segmentação como domínio de aplicação para a criação dos marcadores internos: aplica-se a referida decomposição e, após eliminar os subconjuntos que não tocam a borda do objeto, erode-se os subconjuntos restantes em função de um parâmetro de velocidade. Processo similar se aplica para a geração dos marcadores externos. Finalizados estes processos, os marcadores internos e externos são, enfim, propagados. Além de prover segmentações muito boas, esta abordagem permite a geração de marcadores maiores e mais homogêneos. Estes marcadores, além de prover uma quantidade ainda maior de informação ao estimador de movimento, também possibilita uma edição manual muito mais confortável dos marcadores.

6.1.2 Uso do gradiente espaço-temporal

Em relação à qualidade da segmentação, melhorias substanciais foram obtidas pelo uso de informação espaço-temporal para o cálculo dos gradientes de cada quadro da sequência de imagens. Esta informação, responsável pelos bons resultados que se obtém pela aplicação do *watershed* com marcadores no caso 3-D, é computada inicialmente através do processamento tridimensional da sequência de entrada (considerada como uma imagem volumétrica).

Este processamento consiste no cálculo de um gradiente 3-D em função de um elemento estruturante tridimensional. Este gradiente, que contém informação espaço-temporal da sequência de entrada, é usado parcialmente nas etapas subsequentes onde o *watershed* com marcadores é aplicado para segmentar os objetos de interesse no quadro corrente: para a segmentação em um determinado quadro, usa-se a fatia temporal do gradiente volumétrico relativa ao quadro em questão.

Observou-se que o uso de tal gradiente não só melhorou os resultados de segmentação como também resultou em uma diminuição no número de interferências do usuário. Em especial, os resultados obtidos pelo uso do gradiente espaço-temporal em conjunto com o aprimoramento por amarração de marcadores, conforme pode ser conferido na Tabela 4.1.

6.1.3 Um *benchmark* para avaliação quantitativa de métodos de segmentação assistida de objetos em sequências de imagens

Durante o desenvolvimento da tese, avaliações quantitativas do *watershed* com marcadores propagados fizeram-se necessárias. Assim, foi proposto um método de *benchmark* para avaliação de métodos assistidos de segmentação de objetos em sequências de imagens [4, 25]. De acordo com o nosso conhecimento, não existe na literatura métodos que realizem tais avaliações. Este *benchmark* possibilita, por exemplo, analisar o impacto que a intervenção do usuário tem no *framework* de tais métodos. Desta forma, este *benchmark* possibilita que trabalhos futuros de segmentação assistida sejam comparados, por exemplo, com os resultados apresentados nos capítulos 3, 4 e 5.

A avaliação de um método é feita em função da qualidade da segmentação e da produtividade da mesma (facilidade em que a segmentação é feita). Um método é bom se provê uma segmentação robusta com uma quantidade mínima de intervenção e com um gasto menor de tempo. O *benchmark* proposto consiste em um conjunto de medidas correlacionadas que quantificam vários fatores tais como o histórico de intervenções do usuário para cada quadro e o tempo gasto em cada quadro para se finalizar uma segmentação. O erro de segmentação para cada quadro, também levado em conta pelo *benchmark*, é dado pela diferença simétrica entre a máscara de segmentação computada para o quadro pelo método avaliado e sua respectiva segmentação padrão-ouro. Além disso, a quantidade de movimento dos objetos de interesse entre quadros consecutivos também é calculada para dar uma noção da dificuldade de se segmentar os objetos na sequência.

6.2 Considerações finais

Dados os resultados experimentais apresentados nesta tese, o *watershed* com marcadores propagados com variação por amarração por marcadores e por uso do gradiente espaço-temporal mostrou-se a melhor opção entre as estudadas.

O casamento da melhoria por amarração de marcadores com o uso do gradiente espaço-temporal gerou ganhos importantes em comparação com a melhoria em questão sem o suporte do referido gradiente. A adição do gradiente espaço-temporal ao *framework* de amarração gerou ganhos no número de interferências e, conseqüentemente, no tempo gasto para executar a tarefa. Em geral, houve reduções de até dois terços no número de interferências. Em relação ao erro de segmentação, houve casos onde o erro obtido pelo casamento supracitado foi menor do que o obtido somente pelo método de amarração de marcadores.

Comparando-se a amarração de marcadores com o suporte da decomposição uniforme, nota-se uma redução drástica no número de interferências e no tempo gasto para a realização da tarefa, com ligeira vantagem para a amarração de marcadores. Os erros de segmentação decorridos das aplicações dos dois métodos ficaram um pouco acima de 1%, com uma mínima vantagem para a variação com suporte da decomposição uniforme. Tais observações foram feitas aplicando-se as variações **sem** o uso do gradiente espaço-temporal.

A situação muda quando o gradiente espaço-temporal é utilizado na aplicação dos dois métodos comparados no parágrafo anterior. A aplicação do gradiente espaço-temporal ao *framework* baseado na decomposição uniforme não se mostrou vantajosa. A combinação da decomposição uniforme com o gradiente espaço-temporal não resultou em um ganho quantitativo substancial.

Outra desvantagem do aprimoramento por decomposição uniforme está na atual implementação desta decomposição, que é muito lenta. A avaliação deste método teve de ser realizada desconsiderando-se do tempo total de trabalho, os tempos gastos por todas as aplicações da decomposição uniforme. Considerando-se estes tempos gastos, o *watershed* com marcadores propagados com suporte da decomposição uniforme torna-se proibitivo.

6.3 Trabalhos futuros

Esta tese apresenta, finalmente, as seguintes oportunidades de trabalhos futuros:

- O estudo de outras variações do paradigma proposto para segmentação de objetos em sequências de imagens. Tais variações dizem respeito à criação dos conjuntos de marcadores internos e externos a serem propagados.
- O estudo de outros métodos de propagação de marcadores. O método de Lucas e Kanade mostra uma boa opção para realizar a propagação mas outros métodos também podem ser investigados e testados.
- A aplicação de técnicas que visam a melhoria dos resultados de segmentação tais como o gradiente espaço-temporal.
- A implementação de outros métodos de interação. O método correntemente implementado é o de inserção de marcadores pontuais e/ou formato de segmentos de reta.
- A aplicação do *watershed* com marcadores propagados para a solução de problemas de processamento de imagens e visão computacional.

Apêndice A

Sobre a Amarração de Marcadores

Este Apêndice apresenta dois tópicos adicionais sobre a variação do *watershed* com marcadores propagados por amarração de marcadores. O primeiro deles é uma variação do método de amarração. O segundo tópico descreve uma variação do método de amarração que procura garantir, salvo para um caso particular, que cada região gerada possua dois marcadores, um interno e outro externo.

A.1 Variação

O método de amarração de marcadores apresenta um caso particular. O Passo 2 da descrição do método de amarração consiste na criação de sementes ao redor da máscara de segmentação através da rotulação do contorno pela aplicação da função distância geodésica. A partir de um ponto inicial, os pontos do contorno recebem como rótulo a sua distância do ponto inicial e, em seguida, todos os pontos cujas distâncias não sejam exatamente divisíveis pelo parâmetro numérico w são descartados.

Porém, quando a máscara de segmentação toca as bordas da imagem em mais de um local (Fig. A.1 (a)), o contorno da máscara é quebrado em pelo menos dois pedaços. O problema desta quebra é que, quando o ponto inicial da distância geodésica está em um dos pedaços, os pontos pertencentes ao(s) outro(s) pedaço(s) não são alcançáveis e, portanto, não são rotulados (Fig. A.1 (b)). Isto resulta na imposição de sementes somente em um dos pedaços do contorno da máscara de segmentação (Fig. A.1 (c)).

Um simples redimensionamento da imagem possibilita a imposição das sementes ao longo de todo o contorno que não toca os limites da imagem: esta é aumentada a partir de suas bordas, recebendo “zeros” ao seu redor de modo que a máscara de segmentação não toque mais as bordas da imagem. A distância geodésica agora pode ser calculada por todo o contorno da máscara de segmentação (Fig. A.2 (a)). As sementes são impostas normalmente em função da distância geodésica (Fig. A.2 (b)). Finalmente, a imagem é recortada para voltar às suas dimensões originais (Fig. A.1 (d)). As sementes localizadas fora da região recortada são, desta maneira, descartadas.

A.2 Criação de “Pares Completos” de Marcadores

No método de amarração descrito no Capítulo 3, após a geração de cada região ao longo da borda da máscara de segmentação, os marcadores internos (externos) são dados pela interseção das regiões

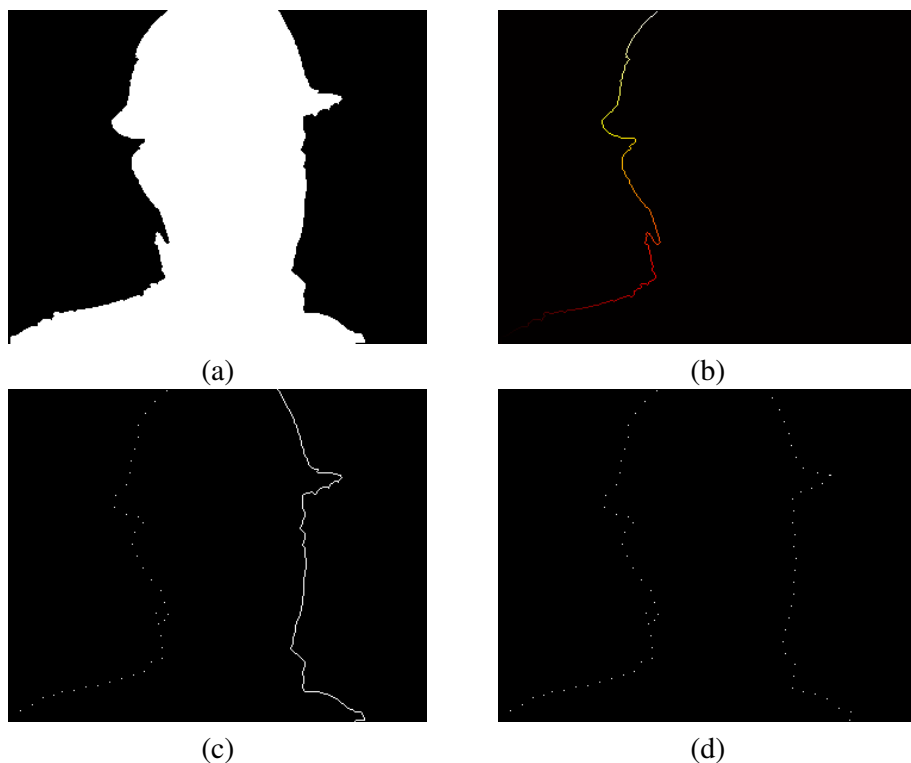


Fig. A.1: Variação da amarração de marcadores. (a) Máscara de segmentação tocando a borda da imagem em dois lugares. (b) Distância geodésica de um dos pedaços do contorno da máscara. (c) Sementes atribuídas a um dos pedaços do contorno. O outro pedaço não recebeu sementes. (d) Sementes atribuídas por todo o contorno (que não toca a borda da imagem).

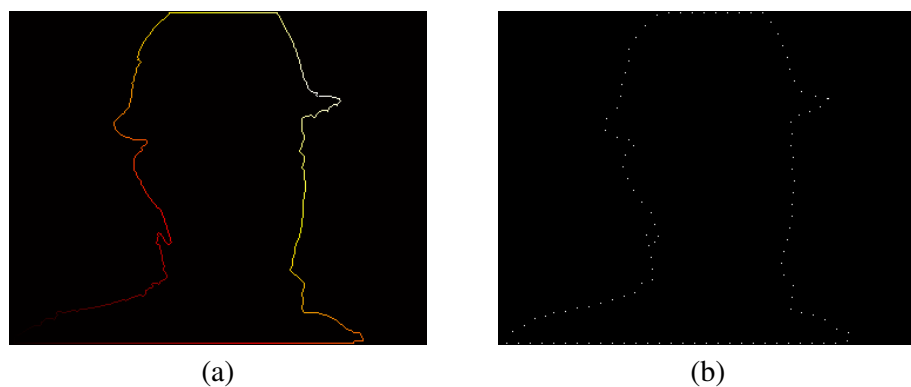


Fig. A.2: Variação da amarração de marcadores. (a) Distância geodésica calculada ao redor do contorno da máscara. (b) Sementes atribuídas em função da distância geodésica ilustrada na Fig A.2 (a).

com a borda interna (externa) da “coroa” da máscara de segmentação.

Contudo, dependendo os parâmetros escolhidos para a amarração, é usual que regiões calculadas em partes convexas da máscara de segmentação não tenham interseção com a borda interna da “coroa”. Da mesma maneira, também pode ocorrer que regiões calculadas em partes côncavas da máscara de segmentação não tenham interseção com a borda externa da “coroa”. Como algumas interseções não ocorrem, alguns marcadores internos e externos são “eventualmente perdidos”.

Esta é uma particularidade do método: a eventualidade de gerar “pares” de marcadores contendo somente um marcador interno ou um externo. Este não é um erro de método, uma vez que pares vizinhos que contenham um “par completo” de marcadores compensam a falta de um marcador na região afetada. No entanto, pode-se ainda desejar que cada região receba exatamente dois marcadores, um de cada tipo. Esta Seção apresenta uma variação do método de amarração, que atribui um par de marcadores para cada região calculada, exceto no caso particular da bordas do objeto, onde a imposição de um par completo é incorreta.



Fig. A.3: Criação de “Pares Completos” de Marcadores. (a) Máscara de segmentação. (b) Sementes.

1. As imagens da máscara de segmentação e das sementes calculadas ao redor do contorno da máscara (Fig. A.3 (a) e (b), respectivamente) são novamente aumentadas através da replicação da informação das bordas de cada imagem (Fig. A.4 (a) e (b)). Isto é feito para evitar o cálculo de marcadores desnecessários nas regiões onde a máscara de segmentação toca a borda da imagem. A replicação é feita em função do parâmetro w : a imagem é expandida ($w + 10$) linhas para cima e para baixo, e expandida ($w + 10$) colunas para a esquerda e para a direita.
2. A “coroa” da máscara expandida é calculada da mesma maneira que a descrita no Passo 1 da descrição do método no Capítulo 3 (Fig A.4 (c)).
3. Linhas de *watershed* são calculadas em função da imagem expandida de sementes, do mesmo modo que o Passo 3 da descrição no Capítulo 3 (Fig. A.4 (d)).
4. Estas linhas são usadas para fatiar a “coroa”, assim como descrito no Passo 4 do método descrito no Capítulo 3 (Fig. A.4 (e)).

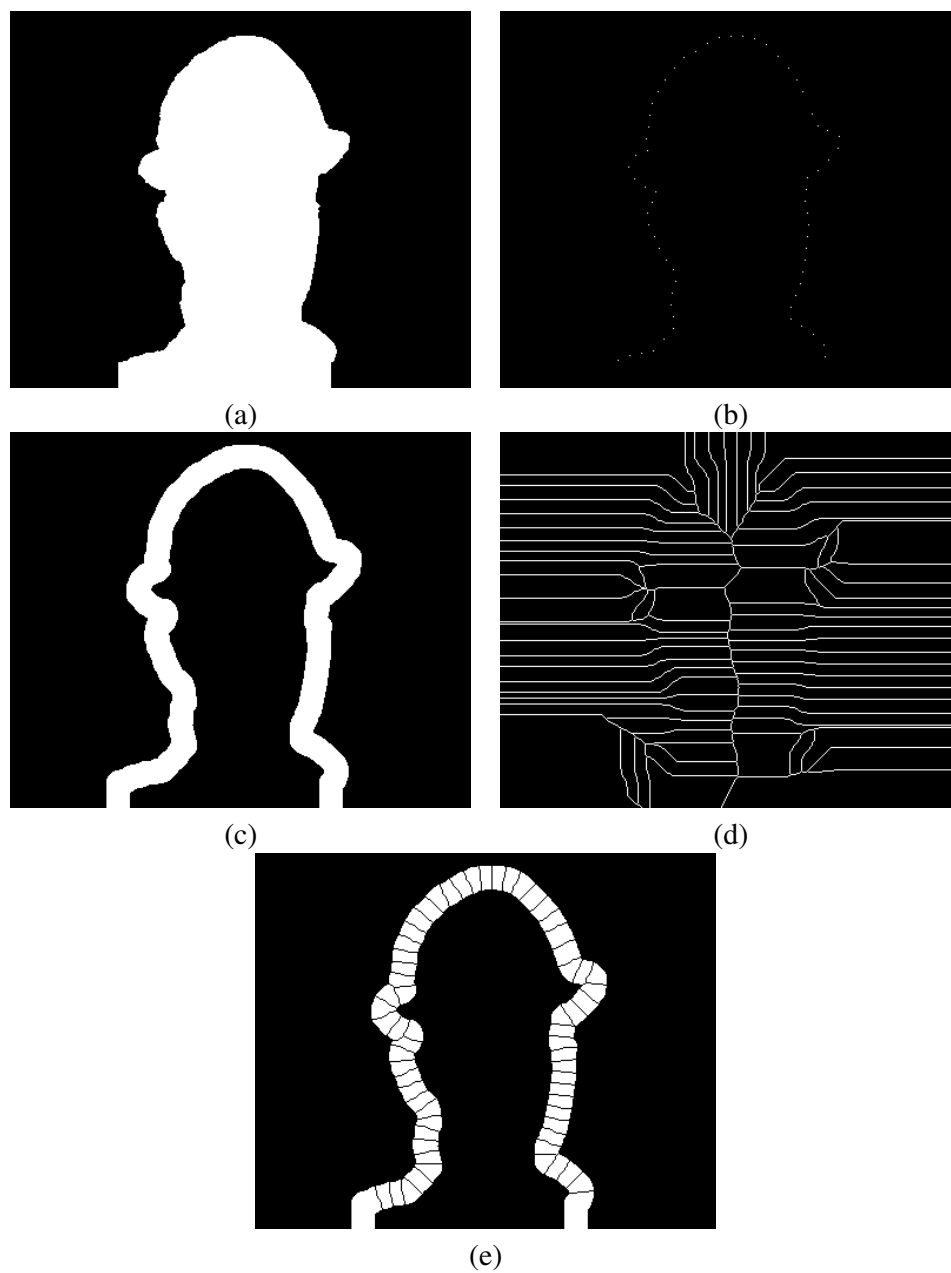


Fig. A.4: Criação de “Pares Completos” de Marcadores. (a) Máscara de segmentação expandida. (b) Imagem expandida das sementes. (c) “Coroa” da máscara expandida. (d) Linhas de *watershed* que fatiam a “coroa” da máscara expandida. (e) “Coroa” fatiada.

Note na Fig. A.4 (e) que uma das regiões geradas não tem interseção com a borda interna da “coroa”. Outras quatro regiões não tem interseção com a borda externa da “coroa”. Para que isto não ocorra, as sementes precisam ser selecionadas de modo que as linhas de *watershed* fatiem a “coroa” de maneira apropriada. O processo é, então, finalizado do seguinte modo:

1. Reconstrói-se a imagem da borda interna da “coroa”, condicionada à imagem com as regiões fatiadas. O resultado é o conjunto de regiões fatiadas que possuem um marcador interno (Fig. A.5 (a)).
2. Similarmente, reconstrói-se a imagem da borda externa da “coroa”, condicionada à imagem com as regiões fatiadas. Obtem-se, assim, o conjunto de regiões que possuem um marcador externo (Fig. A.5 (b)).
3. Faz-se a interseção destes dois conjuntos, o que gera o conjunto de regiões fatiadas que possuem um marcador interno e um externo (Fig. A.5 (c)).
4. A seleção das sementes é feita pela interseção das mesmas com o conjunto computado no Passo anterior. Estas sementes serão usadas no cálculo das linhas de *watershed* que serão usadas para fatiar novamente a “coroa”. A Fig. A.5 (d) mostra as sementes selecionadas e, em destaque, as sementes que foram descartadas.
5. As linhas de *watershed* são calculadas e usadas para o fatiamento definitivo da “coroa”.
6. Os marcadores internos e externos associados a cada nova região são calculados conforme descrito no Passo 5 da descrição do método de amarração apresentado no Capítulo 3.
7. Finalmente, a imagem com as regiões é recortada para voltar às suas dimensões originais (Fig. A.6 (a)). Do mesmo modo são recortadas as imagens com os marcadores internos e externos (Fig. A.6 (c) e (d), respectivamente).

A Fig. A.6 (b) apresenta cada um dos pares de marcadores envolvendo uma região ao redor do contorno da máscara de segmentação. É importante notar, porém, que algumas poucas regiões tiveram marcadores perdidos na fase de recorte das imagens para as dimensões originais. Isto é intencional. Por exemplo, note que na Fig. A.6 (b), para o parâmetro w escolhido, não é possível impor marcadores externos na região superior do capacete. Estes, embora até sejam calculados, são descartados no ato de recorte.

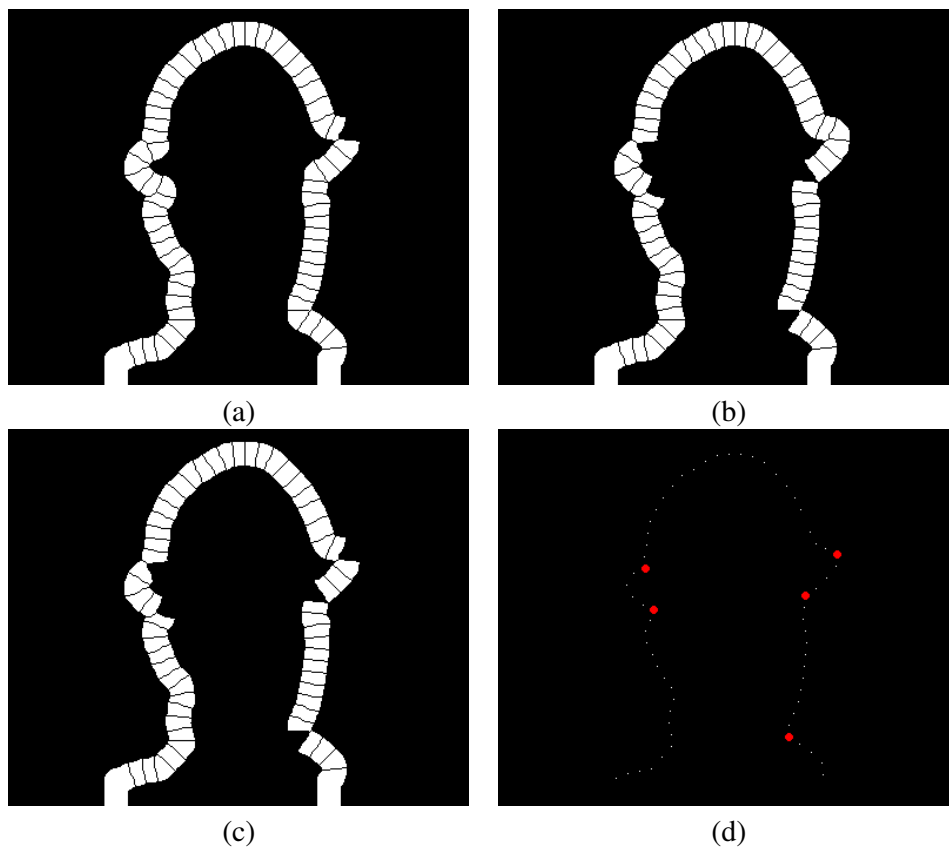


Fig. A.5: Seleção das sementes. (a) Remoção das regiões que não possuem marcador interno. (b) Remoção das regiões que não possuem marcador externo. (c) Interseção das imagens Fig. A.5 (a) e Fig. A.5 (b) contendo as regiões que possuem marcador interno e externo. (d) Sementes selecionadas (menores) e (em destaque) as sementes removidas.

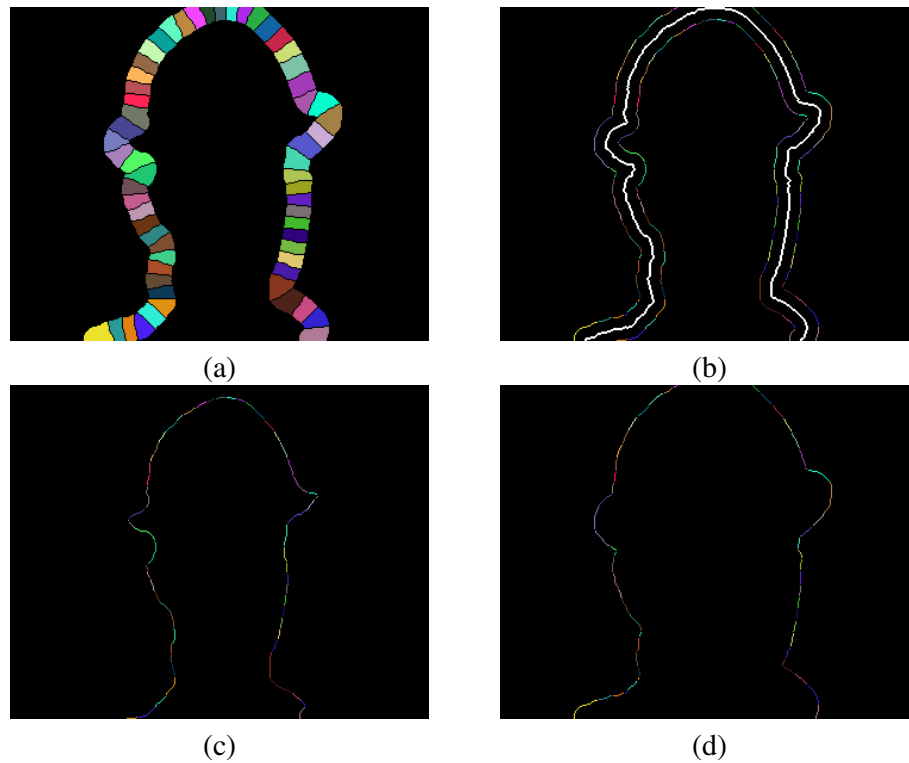


Fig. A.6: Marcadores amarrados. (a) Região associada a um par de marcadores. (b) Pares de marcadores envolvendo o contorno da máscara de segmentação. (c) Marcadores internos. (d) Marcadores externos.

Referências Bibliográficas

- [1] J. M. Rondina; R. A. Lotufo and M. A. Gutierrez. Um Sistema de Segmentação Interativa do Ventrículo Esquerdo em Sequências de Imagens de Ressonância Magnética (*cine MR*). *submetido à Revista Brasileira de Engenharia Biomédica*, 2001.
- [2] W. E. Higgins; N. Chung and E. L. Ritman. Extraction of Left-Ventricular Chamber from 3-D CT Images of the Heart. *IEEE Transactions on Medical Imaging*, 9(4):384–395, December 1990.
- [3] A. Mitiche and P. Bouthemy. Computation and Analysis of Image Motion: A Synopsis of Current Problems and Methods. *International Journal of Computer Vision*, 19(1):29–56, 1996.
- [4] F. C. Flores and R. A. Lotufo. Watershed from Propagated Markers: An Interactive Method to Morphological Object Segmentation in Image Sequences. *To Appear on Image and Vision Computing*, 2010.
- [5] J. L. Barron; D. J. Fleet and S. S. Beauchemin. Performance of Optical Flow Techniques. *International Journal of Computer Vision*, 12(1):43–77, 1994.
- [6] B. Lucas and T. Kanade. An interactive image registration technique with an application to stereo system. In *Proceedings of DARPA Image Understanding Workshop*, pages 121–130, 1981.
- [7] A.X. Falcão; J.K. Udupa; S. Samarasekera; S. Sharma; R.A. Lotufo. User-Steered Image Segmentation Paradigms: Live Wire and Live Lane. *Graphical Models and Image Processing*, (60):233–260, 1998.
- [8] E.N. Mortensen and W.A. Barrett. Interactive Segmentation with Intelligent Scissors. *Graphical Models and Image Processing*, (60):349–384, 1998.
- [9] D.B. Goldman; C. Gonterman; B. Curless; D. Salesin and S.M. Seitz. Video object annotation, navigation, and composition. In *UIST '08: Proceedings of the 21st annual ACM symposium on User interface software and technology*, pages 3–12, New York, NY, USA, 2008. ACM.
- [10] X. Bai; J. Wang; D. Simons and G. Sapiro. Video SnapCut: robust video object cutout using localized classifiers. In *SIGGRAPH '09: ACM SIGGRAPH 2009 papers*, pages 1–11, New York, NY, USA, 2009. ACM.
- [11] C.J. Armstrong; B.L. Price and W.A. Barrett. Interactive segmentation of image volumes with Live Surface. *Comput. Graph.*, 31(2):212–229, 2007.

- [12] J-K. Ahn and C-S. Kim. Real-time segmentation of objects from video sequences with non-stationary backgrounds using spatio-temporal coherence. In *15th IEEE International Conference on Image Processing, 2008. ICIP 2008.*, pages 1544–1547, San Diego, USA, October 2008.
- [13] X. Bai and G. Sapiro. A Geodesic Framework for Fast Interactive Image and Video Segmentation and Matting. In *ICCV 2007. IEEE 11th International Conference on Computer Vision*, pages 1–8, 2007.
- [14] A. Souza; J.K. Udupa; G. Grevera; D. Sun and N. Suri. Iterative Live Wire and Live Snake: New User-Steered 3D Image Segmentation Paradigms. In *SPIE Medical Imaging 2006*, pages 1159–1165, March 2006.
- [15] J. Liu; J.K. Udupa; P.K. Saha; D. Odhner; B.E. Hirsch; S. Siegler; S. Simon and B.A. Winkelstein. Rigid model-based 3D segmentation of the bones of joints in MR and CT images for motion analysis. *Medical Physics*, 35(8):3637–3649, 2008.
- [16] M. Poon; G. Hamarneh and R. Abugharbieh. Efficient interactive 3D Livewire segmentation of complex objects with arbitrary topology. *Computerized Medical Imaging and Graphics*, 32(8):639–650, 2009.
- [17] F. C. Flores and R. A. Lotufo. Object Segmentation in Image Sequences by Watershed from Markers: A Generic Approach. In *IEEE Proceedings of SIBGRAPI'2003*, pages 347–352, Sao Carlos, Brazil, October 2003.
- [18] F. C. Flores and R. A. Lotufo. Watershed from Propagated Markers Improved by the Combination of Spatio-Temporal Gradient and Binding of Markers Heuristics. In *To appear on Proceedings of VISAPP 2010*, Angers, France, May 2010.
- [19] C. Erden; B. Sankur; A. Tekalp. Performance Measures for Video Object Segmentation and Tracking. *IEEE Transactions on Image Processing*, 13(7):937–951, July 2004.
- [20] K. McKoen; R. Navarro-Prieto; E. Durucan; B. Duc; F. Ziliani; T. Ebrahimi. Evaluation of segmentation methods for surveillance applications. In *EUSIPCO 2000*, pages 1045–1048, Kobe, Japan, September 2000.
- [21] P. Correia and F. Pereira. Video Object Relevance Metrics for Overall Segmentation Quality Evaluation. *EURASIP Journal on Applied Signal Processing*, 2006:1–11, Article ID 82915 2006.
- [22] P. Villegas and X. Marichal. Perceptually-weighted evaluation criteria for segmentation masks in video sequences. *IEEE Transactions on Image Processing*, 13(8):1092–1103, August 2004.
- [23] R. Mech and F. Marques. Objective Evaluation Criteria for 2-D Shape Estimation Results of Moving Objects. *EURASIP Journal on Applied Signal Processing*, 2002:401–409, 2002.
- [24] P. Correia and F. Pereira. Objective Evaluation of Video Segmentation Quality. *IEEE Transactions on Image Processing*, 12(2):186–200, February 2003.

- [25] F. C. Flores and R. A. Lotufo. Benchmark for Quantitative Evaluation of Assisted Object Segmentation Methods to Image Sequences. In *IEEE Proceedings of SIBGRAP'2008*, pages 95–102, Campo Grande, Brazil, October 2008.
- [26] F. C. Flores and R. A. Lotufo. Watershed from Propagated Markers Improved by a Marker Binding Heuristic. In J. Barrera G.J.F. Banon and U.M. Braga-Neto, editors, *Mathematical Morphology and its Applications to Image and Signal Processing*, Proc. ISMM'07, pages 313–323. MCT/INPE, 2007.
- [27] F. C. Flores, R. Hirata Jr., J. Barrera, R. A. Lotufo, and F. Meyer. Morphological Operators for Segmentation of Color Sequences. In *IEEE Proceedings of SIBGRAP'2000*, pages 300–307, Gramado, Brazil, October 2000.
- [28] R. Hirata Jr., J. Barrera, F. C. Flores, and R. A. Lotufo. Automatic Design of Morphological Operators for Motion Segmentation. In J. Stolfi and C. L. Tozzi, editors, *IEEE Proc. of Sibgrapi'99*, pages 283–292, Campinas, SP, Brazil, 1999.
- [29] Na Li; Shipeng Li; Wen-Yin Liu and Chun Chen. A novel framework for semi-automatic video object segmentation. In *IEEE International Symposium on Circuits and Systems, 2002. ISCAS 2002.*, volume 3, pages 811–814, 2002.
- [30] I. Patras; E. A. Hendriks and R. L. Lagendijk. Video Segmentation by MAP Labeling of Watershed Segments. *IEEE Transactions on Pattern Analysis and Machine Learning*, 23(3):326–332, March 2001.
- [31] C. Gu and M.-C. Lee. Semiautomatic Segmentation and Tracking of Semantic Video Objects. *IEEE Transactions on Circuits and Systems for Video Technology*, 8(5):572–584, September 1998.
- [32] T. Yoshida and T. Shimosato. Motion image segmentation using 3-D watershed algorithm. In *IEEE International Conference on Image Processing, 2001. Proceedings. 2001*, volume 2, pages 773–776, 2001.
- [33] M. Maziere; F. Chassaing; L. Garrido and P. Salembier. Segmentation and tracking of video objects for a content-based video indexing context. In *ICME 2000. 2000 IEEE International Conference on Multimedia and Expo, 2000.*, volume 2, pages 1191–1194, 2000.
- [34] P. Salembier and J. Ruiz. Connected operators based on reconstruction process for size and motion simplification. In *IEEE International Conference on Acoustics, Speech, and Signal Processing, 2002*, volume 4, pages 3289–3292, 2002.
- [35] P. Salembier and M. Pardàs. Hierarchical Morphological Segmentation for Image Sequence Coding. *IEEE Transactions on Image Processing*, 3(5):639–651, September 1994.
- [36] R. Mosges and S. Lavallee. *Computer Integrated Surgery - Technology and Clinical Applications*, chapter 1 - Multimodal Information for Computer-Integrated Surgery, pages 5–19. MIT Press, 1996.

- [37] N. Ayache *et al.* *Computer Integrated Surgery - Technology and Clinical Applications*, chapter 4 - Segmentation of Complex Three-Dimensional Medical Objects: A Challenge and a Requirement for Computer-Assisted Surgery Planning and Performance, pages 59–74. MIT Press, 1996.
- [38] S. Gil, R. Milanese, and T. Pun. Comparing Features for Target Tracking in Traffic Scenes. *Pattern Recognition*, 29(8):1285–1296, August 1996.
- [39] A. Pommert *et al.* *Computer Integrated Surgery - Technology and Clinical Applications*, chapter 9 - Three-Dimensional Imaging in Medicine: Methods and Applications, pages 155–174. MIT Press, 1996.
- [40] H. J. A. M. Heijmans. *Morphological Image Operators*. Academic Press, Boston, 1994.
- [41] F. Zanoguera; B. Marcotegui and F. Meyer. A Toolbox for Interactive Segmentation Based on Nested Partitions. In *Proceedings of IEEE International Conference on Image Processing, ICIP'99*, Japan, October 1999.
- [42] S. Beucher and F. Meyer. *Mathematical Morphology in Image Processing*, chapter 12. The Morphological Approach to Segmentation: The Watershed Transformation, pages 433–481. Marcel Dekker, 1992.
- [43] 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):583–598, June 1991.
- [44] R. A. Lotufo and A. X. Falcão. The Ordered Queue and the Optimality of the Watershed Approaches. In J. Goutsias, L. Vincent, and D. S. Bloomberg, editors, *Mathematical Morphology and its Applications to Image and Signal Processing*, volume 18, pages 341–350. Kluwer Academic Publishers, 2000. Fifth ISMM.
- [45] R. C. Gonzalez and R. E. Woods. *Digital Image Processing*. Addison-Wesley Publishing Company, 1992.
- [46] C. Finch. *Special Effects: Creating Movie Magic*. Abbeville Press, 1984.
- [47] P. Salembier and F. Marques. Region-based representations of image and video: Segmentation tools for multimedia services. *IEEE Transactions on Circuits and Systems for Video Technology*, 9(8):1147–1167, December 1999.
- [48] R. A. Lotufo; R. Machado; F. C. Flores; A. X. Falcao; R. Koo; G. S. Mazzela and R. M. Costa. Prontovideo: An Image Sequence Segmentation Tool Applied to Video Edition. In *IEEE Proceedings of SIBGRAPI'2001*, page 404, Florianopolis, Brazil, October 2001.
- [49] A.B.V. Graciano; R.M. Cesar-Jr and I. Bloch. Graph-based Object Tracking Using Structural Pattern Recognition. In *SIBGRAPI*, pages 179–186, Belo Horizonte, Brazil, October 2007.

- [50] L. A. Consularo, R. M. Cesar-Jr, L. H. Figueiredo and I. Bloch. Oversegmentation Control for Inexact Graph Matching: First Results. In J. Barrera G.J.F. Banon and U.M. Braga-Neto, editors, *Mathematical Morphology and its Applications to Image and Signal Processing*, Proc. ISMM'07, pages 375–386. MCT/INPE, 2007.
- [51] B. Marcotegui;F. Zanoguera;P. Correia;R. Rosa;F. Marques;R. Mech;M. Wollborn. A Video Object Generation Tool Allowing Friendly User Interaction. In *ICIP'99*, pages 391–395, Kobe, Japan, October 1999.
- [52] J. Serra. *Image Analysis and Mathematical Morphology*. Academic Press, 1982.
- [53] J. Serra. *Image Analysis and Mathematical Morphology. Volume 2: Theoretical Advances*. Academic Press, 1988.
- [54] F. Meyer and S. Beucher. Morphological Segmentation. *Journal of Visual Communication and Image Representation*, 1(1):21–46, September 1990.
- [55] F. C. Flores, A. M. Polidório and R. A. Lotufo. The Weighted Gradient: A Color Image Gradient Applied to Morphological Segmentation. *Journal of the Brazilian Computer Society*, 11(3):53–63, December 2006.
- [56] F. C. Flores, A. M. Polidório and R. A. Lotufo. Color Image Gradients for Morphological Segmentation: The Weighted Gradient Improved by Automatic Imposition of Weights. In *SIB-GRAPI*, pages 146–153, Curitiba, Brazil, October 2004.
- [57] R. Hirata Jr., F. C. Flores, J. Barrera, R. A. Lotufo, and F. Meyer. Color Image Gradients for Morphological Segmentation. In *IEEE Proceedings of SIBGRAPI'2000*, pages 316–322, Gramado, Brazil, October 2000.
- [58] 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, January 2004.
- [59] S. S. Beauchemin and J. L. Barron. The Computation of Optical Flow. *ACM Computing Surveys*, 27(3):433–467, 1995.
- [60] G. Castellano. *Investigation of a Complex Wavelet Algorithm for the Estimation of Optical Flow and Application to Moving Target Detection*. PhD thesis, King's College, University of London, June 1999.
- [61] B.K.P. Horn and B.G. Schunck. Determining Optical Flow. *Artificial Intelligence*, 17:185–204, 1981.
- [62] W. K. Pratt. *Digital Image Processing*. John Wiley and Sons, 1991.
- [63] L. Rittner. Identification and Transformation of Outliers as Confidence Measures of Lucas and Kanade Method for Optical Flow Estimation. Master Thesis (in Portuguese), DCA - FEEC - State University of Campinas, BRAZIL, February 2004.

- [64] Z. He. Dynamic programming framework for automatic video object segmentation and vision-assisted video pre-processing. In *IEE Proceedings of Vision, Image and Signal Processing*, pages 597–603, October 2005.
- [65] P. Salembier; F. Marques; M. Pardas; J.R.Morros; I. Corset; S. Jeannin; L. Bouchard; F. Meyer; B. Marcotegui. Segmentation-Based Video Coding System Allowing the Manipulation of Objects. *IEEE Transactions on Circuits and Systems for Video Technology*, 7(1):60–74, February 1997.
- [66] P. Salembier and J. Ruiz. On Filters by Reconstruction for Size and Motion Simplification. In H. Talbot and R. Beare, editors, *Mathematical Morphology and its Applications to Image and Signal Processing*, Proc. ISMM'02, pages 425–434. CSIRO Publishing, 2002.
- [67] S.H. Hong and M.M-O. Lee. User-Assisted Segmentation and Tracking of Video Objects. In *Proceedings of 6th IEEE International Conference HSNMC 2003*, pages 275–283, Estoril, Portugal, July 2003.
- [68] F. Moscheni; Sushil Bhattacharjee and Murat Kunt. Spatiotemporal Segmentation Based on Region Merging. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(9):897–915, September 1998.
- [69] P. Smith; T. Drummond and R. Cipolla. Layered Motion Segmentation and Depth Ordering by Tracking Edges. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(4):479–494, April 2004.
- [70] P. Soille and L. Vincent. Determining Watersheds in Digital Pictures via Flooding Simulations. In *Visual Communications and Image Processing*, pages 240–250. SPIE, 1990. volume 1360.
- [71] F. Meyer. Hierarchies of Partitions and Morphological Segmentation. In *Proceedings of the Third International Conference on Scale-Space and Morphology in Computer Vision*, volume 2106, pages 161–182, 2001.
- [72] M.A.G. Carvalho, R. A. Lotufo and M. Couprie. Spatiotemporal Segmentation of MR Image Sequence Based on Hierarchical Analysis. In *Proceedings of Seventh International Symposium on Signal Processing and Its Applications - ISSPA 2003*, pages 677–680, Paris, France, July 2003.
- [73] M.A.G. Carvalho, R. A. Lotufo and M. Couprie. Segmentation of Images of Yeast Cells by Scale-Space Analysis. In *IEEE Proceedings of SIBGRAP'2003*, pages 376–380, Sao Carlos, Brazil, October 2003.
- [74] C. Vachier and F. Meyer. Extinction Value: A New Measurement of Persistence. In *Proceedings of 1995 IEEE Workshop on Nonlinear Signal and Image Processing*, volume I, pages 254–257. IEEE, 1995.
- [75] H. J. A. M. Heijmans. Introduction to Connected Operators. In E. R. Dougherty and J. T. Astola, editors, *Nonlinear Filters for Image Processing*, pages 207–235. SPIE–The International Society for Optical Engineering,, 1999.

-
- [76] C. Vachier and L. Vincent. Valuation of Image Extrema Using Alternating Filters by Reconstruction. In *Proceedings of SPIE - Neural, Morphological, and Stochastic Methods in Image and Signal Processing*, volume 2568, pages 94–103. SPIE, July 1995.
- [77] E. J. Breen and R. Jones. Attribute Openings, Thinnings and Granulometries. *Computer Vision and Image Understanding*, 64(3):377–389, 1996.
- [78] M. Grimaud. A New Measure of Contrast: the Dynamics. In SPIE, editor, *Image Algebra and Morphological Image Processing III*, volume 1769, pages 292–305, 1992.
- [79] C. Vachier. *Extraction de caractéristiques, segmentation d'image et morphologie mathématique*. PhD thesis, Ecole des Mines de Paris, 1995.
- [80] M. Borsotti, P. Campadelli and R. Schettini. Quantitative Evaluation of Color Image Segmentation Results. *Pattern Recognition Letters*, 19(1998):741–747, 1998.