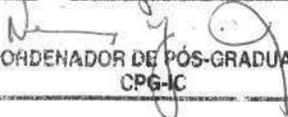


Este exemplar corresponde à redação final da
Tese/Dissertação devidamente corrigida e defendida
por: Elisângela de Araújo
Rodrigues
e aprovada pela Banca Examinadora.
Campinas, 30 de Julho de 2011

COORDENADOR DE PÓS-GRADUAÇÃO
CPG-IC

**Segmentação de Imagens Tomográficas
usando Contornos Deformáveis
com Segmentos de Custo Mínimo**

Elisângela de Araújo Rodrigues

Dissertação de Mestrado

Segmentação de Imagens Tomográficas usando Contornos Deformáveis com Segmentos de Custo Mínimo

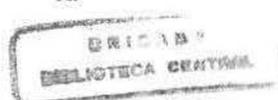
Elisângela de Araújo Rodrigues

Maio de 2001

Banca Examinadora:

- Prof. Dr. Alexandre Xavier Falcão
IC - Universidade Estadual de Campinas (Orientador)
- Dr. Gilson Antônio Giraldi
COPPE - Universidade Federal do Rio de Janeiro
- Prof. Dr. Neucimar Jerônimo Leite
IC - Universidade Estadual de Campinas
- Prof. Dr. Paulo Lício de Geus (suplente)
IC-Universidade Estadual de Campinas

ia



UNIDADE De
N.º CHAMADA:
T/UNICAMP
R618s
TOMBO BC/ 46234
PROC. 16-39201
C D
PREÇO R\$ 11,00
DATA 13/09/01
N.º CPD

CM00159630-4

**FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DO IMECC DA UNICAMP**

Rodrigues, Elisângela de Araújo

R618s Segmentação de imagens tomográficas usando contornos deformáveis com segmentos de custo mínimo / Elisângela de Araújo Rodrigues -- Campinas. [S.P. :s.n.], 2000.

Orientador : Alexandre Xavier Falcão

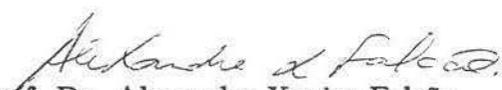
Dissertação (mestrado) - Universidade Estadual de Campinas, Instituto de Computação.

1. Processamento de imagens. 2. Tomografia. I. Falcão, Alexandre Xavier. II. Universidade Estadual de Campinas. Instituto de Computação. III. Título.

Segmentação de Imagens Tomográficas usando Contornos Deformáveis com Segmentos de Custo Mínimo

Este exemplar corresponde à redação final da Dissertação devidamente corrigida e defendida por Elisângela de Araújo Rodrigues e aprovada pela Banca Examinadora.

Campinas, 28 de Maio de 2001.


Prof. Dr. Alexandre Xavier Falcão
IC - Universidade Estadual de Campinas
(Orientador)

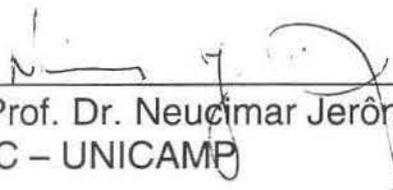
Dissertação apresentada ao Instituto de Computação, UNICAMP, como requisito parcial para a obtenção do título de Mestre em Ciência da Computação.

TERMO DE APROVAÇÃO

Tese defendida e aprovada em 22 de dezembro de 2000, pela Banca Examinadora composta pelos Professores Doutores:



Prof. Dr. Gilson Antônio Giraldi
COPPE - UFRJ



Prof. Dr. Neucimar Jerônimo Leite
IC - UNICAMP



Prof. Dr. Alexandre Xavier Falcão
IC - UNICAMP

*“Aquele que sai andando e chorando,
enquanto semeia,
voltará cheio de júbilo
trazendo consigo os seus molhos”*

(Sl 126,6)

A Deus, que faz muito mais do que
tudo quanto pedimos ou cremos ou pensamos.
Aos meus pais, pelo muito que sempre lutaram.
Aos meus irmãos, com muito amor.

Agradecimentos

A Deus, que me tem dado a vida e a graça de viver este momento. *“Porque dEle e por meio dEle e para Ele são todas as coisas.”* (Rm 11.36)

Aos meus pais, Idenir e José, e meus irmãos, Elisandra e Thiago, a quem agradeço por todo o amor. A saudade teria sido razão suficiente para eu nem sair em busca deste sonho, não fosse eles terem-no sonhado junto comigo. Obrigada por não me deixarem desistir em momentos que parecia tão difícil continuar!

Ao Dário, que se tornou um dos melhores presentes que eu poderia receber. Seu amor, carinho e apoio têm sido essenciais. E, em diversos momentos, sua ajuda ou simples companhia foram indispensáveis para a realização deste trabalho.

Ao Alexandre Falcão, pela orientação ao longo deste trabalho. E ao Gilson Giraldi, pela disposição em ensinar, com paciência, o que eu não compreendia.

A amigos que, apesar de distantes fisicamente, estavam presentes nas lembranças, nos telefonemas, e-mails... e ajudavam a “matar” a saudade de casa, especialmente a Karla e a Jaqueline.

A Kelcy Matsuda, Gianfrancesca, Sandro (Danilo), Aldecir, ... e a cada um daqueles que fazem parte da turma MsC/98 e tornaram-se amigos. A ajuda de vocês foi fundamental para que este trabalho fosse concluído. Não esquecendo da companhia do Mogami (Fernando) no início do processo de aprendizado de processamento de imagens, pois sua ajuda foi necessária para que outras etapas pudessem ser concluídas.

A cada um dos demais que conheci no IC e se dispuseram a ser amigos, como Cláudio Márcio, Silvio Jamil, Marta Duarte, Delano, Vera Ragazzi, Daniel Capeleto...

Às pessoas com quem tive o prazer de morar, como D.Lúcia, Everly Herr, Débora, Viviane,... Cada uma delas em um momento, a seu modo, tornaram-se especiais em minha vida. E ao Dr. Ross Douglas, que não só ajudou-me neste processo de encontrar moradia, mas durante um tempo permitiu-me ter sua casa como minha.

Às igrejas que tive oportunidade de conhecer e encontrar abrigo, nas quais encontrei pessoas que me permitiram outra vez sentir-me em família: Obrigada Carmem, Sandra, Glícia, Patrícia, André, Berna, Mateus, Aninha, D. Ana,... À Comunidade Cristã Koinonia, pelas orações; em especial ao Tom, pela disposição em ouvir e aconselhar, e a Iolanda

e Paul Freston pelo carinho que têm oferecido. Aos amigos e irmãos da Aliança Bíblica Universitária que sonharam junto, que incentivaram, e compartilham alegrias e tristezas... E tantos outros cujo carinho e amizade, trago guardado no coração. Deus abençõe a cada um de vocês. Obrigada a todos que têm me ajudado a encontrar e permanecer no Caminho.

A Dra. Silvia Helena Cardoso e Dr. Renato Sabbatini, pela oportunidade que me foi dada para trabalhar com informática biomédica e através daquele trabalho, confirmar o interesse de pesquisar nesta área.

Aos professores e colegas da graduação, em especial ao Altigran Soares da Silva, pelo incentivo que dá aos seus alunos a realizarem o mestrado.

A Carla, Fabiana, e a cada profissional e criança que conheci no Semi-Internato Alternativo. Os conselhos, carinho e experiência adquiridos durante o tempo que estivemos juntos também contribuiu para que este trabalho fosse concluído.

E, com carinho, agradeço a Flávia Munhoz (CPqD), pela compreensão e apoio, indispensáveis nesta etapa final.

Resumo

Imagens tomográficas consistem de uma seqüência de imagens de cortes paralelos contendo informações internas da anatomia (ou função) de estruturas tridimensionais. A segmentação é a operação que "extrai" das imagens tomográficas as estruturas de interesse em uma dada aplicação. Métodos de contornos deformáveis têm sido bastante investigados como uma solução para problemas de segmentação de imagens. Estes métodos, porém, apresentam problemas de convergência e estabilização do contorno sobre a borda da estrutura desejada. Este trabalho apresenta uma revisão sobre as principais técnicas de contornos deformáveis e uma solução eficiente para os problemas de convergência e estabilização através do uso de segmentos de custo mínimo. Os resultados do método proposto são ilustrados e analisados com respeito a segmentação de imagens tomográficas do corpo humano.

Abstract

Tomographic images contain important information about the anatomy (or function) of three-dimensional structures of the human body. Image segmentation is the operation that “extracts” the structures of interest from the tomographic images for a given application. Deformable contours are among the most investigated methods for image segmentation. Two traditional problems in such an approach are the convergence and the stabilization of the deformable model over a desired object boundary. This work presents an overview about deformable contours and proposes an efficient solution to the convergence and stabilization problems. Our solution essentially splits the contour in minimum-cost segments which snap onto the object boundary when the deformable contour gets close to it. The results of the work are illustrated with respect to image segmentation tasks in medicine.

Conteúdo

Agradecimentos	vii
Resumo	ix
Abstract	x
1 Introdução	1
1.1 Motivação	1
1.2 Contornos Deformáveis	2
1.3 Objetivos	2
1.4 Organização do Texto	2
2 Modelos de Contornos Deformáveis	4
2.1 Introdução	4
2.2 Modelo Snakes Original	4
2.3 Discretização	6
2.3.1 Análise das Dificuldades	8
2.4 <i>Snakes</i> : extensões ao modelo original	9
2.4.1 Modelos Contínuos Paramétricos	9
2.4.2 Modelos Contínuos Implícitos	10
2.4.3 Modelos Dinâmicos	11
2.5 Modelos Híbridos	12
2.5.1 Dual T-Snakes	12
2.6 Conclusão	15
3 Detecção de Bordas através de Segmentos de Custo Mínimo	17
3.1 Introdução	17
3.2 <i>Live-Wire</i>	18
3.2.1 Conceitos básicos	19
3.2.2 <i>Processo de segmentação no Live-Wire</i>	22

3.2.3	<i>Extensões do modelo</i>	27
3.3	Conclusão	28
4	Contornos Deformáveis usando Segmentos de Custo Mínimo	31
4.1	O modelo de Contornos Deformáveis usando Segmentos de Custo Mínimo	32
4.1.1	Deformação do Contorno	32
4.1.2	A orientação da borda	36
4.1.3	Condições de parada do contorno	38
4.1.4	Segmentos de Custo Mínimo	40
4.2	Trabalhos Relacionados	41
4.3	Conclusão	44
5	Resultados	46
5.1	Definição de parâmetros	47
5.2	Pré-processamento	49
5.3	Deformação do contorno	50
5.4	Deformação por segmentos de custo mínimo	54
5.5	Reinicialização na próxima fatia	57
5.6	Conclusão	62
6	Conclusão	65
	Bibliografia	68

Lista de Figuras

2.1	Exemplo de inicialização da Dual T-Snakes em três objetos distintos. Observe que há um contorno interno para cada objeto de interesse e um único contorno externo envolvendo a todos.	14
2.2	Ilustração da malha de triângulos utilizada pelo modelo Dual-T-Snakes. . .	14
3.1	Borda de um objeto em uma cena 2D definida como um contorno fechado, conectado e orientado formado por arestas de pixel.	19
3.2	Um elemento da borda (<i>bel</i>) é uma aresta de pixel orientada. Os quatro possíveis tipos de <i>bels</i> em uma cena 2D são mostrados. Para qualquer orientação da borda, um <i>bel</i> $b = (p, q)$ é sempre definido tal que p está à esquerda de b e q está à direita. Para uma borda orientada no sentido anti-horário, o pixel p está no interior do objeto e o pixel q no exterior. . .	20
3.3	(a) Um grafo $G(V, E)$ definido para a cena C após a classificação. Um caminho $P = (e_1, e_2, \dots, e_5)$ em G entre os vértices v_0 e v_5 . (b) Cada vértice $v \in V$ define um conjunto $A(v) = u_1, u_2, u_3, u_4$ de no máximo 4 vértices vizinhos 4-adjacentes.	22
3.4	A figura ilustra o processo de segmentação usando <i>live-wire</i>	23
3.5	A figura ilustra a importância das características sensíveis à orientação nos métodos <i>live</i> . (a) A orientação da borda é assumida no sentido anti-horário. Um segmento de <i>live-wire</i> entre dois vértices é obtido usando características sensíveis à orientação. A figura mostra uma região de interesse onde o segmento de contraste similar não atrai o <i>live-wire</i> visto que sua orientação não é favorável. (b) Um segmento de <i>live-wire</i> partindo do mesmo vértice inicial de (a), mas com orientação no sentido contrário, ilustra a sensibilidade do método à orientação previamente adotada para a borda.	24

3.6	A figura ilustra o algoritmo <i>DPS</i> . (a) Um grafo de entrada representando uma cena <i>C</i> após classificação, onde valores de custo conjunto de $[0,10]$ são associados aos <i>bels</i> de <i>C</i> e o vértice inicial é representado pelo círculo escuro. As figuras de (b) a (f) ilustram respectivamente a situação do grafo de saída antes da primeira iteração, após a primeira iteração, após a segunda iteração, após a sétima iteração e após quinze iterações. Nestas figuras, o próximo vértice a sair da fila é sempre representado pelo círculo parcialmente escuro.	26
3.7	A figura mostra um corte ortogonal ao plano das fatias de uma cena 3D. O contorno que representa a borda do objeto na cena 2D resultante do corte ortogonal gera pontos sobre a borda do objeto nas fatias da cena 3D. . . .	29
4.1	Ilustração da formação da Força Interna. Calcula-se a Força Elástica $F_{el}(A)$ (entre v_i e v_{i-1}) e $F_{el}(B)$ (entre v_i e v_{i+1}). A soma de ambas resulta na Força Interna no snaxel v_i	34
4.2	Ilustração de um pixel i e seus 4-vizinhos (pixeis q_1, q_2, q_3 e q_4)	35
4.3	Imagem de Contorno Inicial para segmentação do Quadrado. O mesmo pode expandir a fim de segmentar o quadrado mais externo ou contrair, para segmentação do quadrado mais interno.	37
4.4	O objeto de interesse é o quadrado mais interno. (a) A segmentação foi feita pelo lado de fora do objeto. (b) A segmentação foi feita pelo lado de dentro do objeto.	37
4.5	(a) Resultado da Segmentação do Quadrado Interno utilizando o algoritmo de caminhos mínimos. (b) Resultado da Segmentação do Quadrado Interno sem utilizar o algoritmo de caminhos mínimos.	42
4.6	(a) Resultado da Segmentação do Quadrado Externo utilizando o algoritmo de caminhos mínimos. (b) Resultado da Segmentação do Quadrado Externo sem utilizar o algoritmo de caminhos mínimos.	42
5.1	(a) Contorno inicial para segmentação do osso; (b) Contorno inicial para segmentação da pele; (c) Contorno inicial para segmentação do tecido mole 1; (d) Contorno inicial para segmentação do tecido mole 2.	48
5.2	Contorno final resultante. Os erros ocorrem devido a má escolha dos parâmetros. O contorno inicial foi o mesmo utilizado no tecido mole 2 (e este era o objeto de interesse). O contorno pára em pontos de borda, porém não os do objeto desejado.	49
5.3	Energia da Imagem.	50
5.4	Limiarização da Energia da Imagem do joelho para bordas fortes.	51
5.5	Limiarização da Energia da Imagem do joelho para bordas fracas.	51

5.6	Contorno resultante da segmentação do osso, sem o uso de segmentos de custo mínimo (a) Pelo lado de dentro da borda. (b) Pelo lado de fora da borda.	53
5.7	Contorno resultante da segmentação da pele que envolve o joelho, pelo lado de fora da borda, sem o uso de segmentos de custo mínimo.	53
5.8	Contorno resultante da segmentação do tecido mole sem o uso de segmentos de custo mínimo.	54
5.9	Contorno resultante da segmentação da pele que envolve o joelho, pelo lado de fora da borda, utilizando segmentos de custo mínimo para concluir a segmentação.	55
5.10	Contorno resultante da segmentação do osso, usando segmentos de custo mínimo. (a) Pelo lado de dentro da borda. (b) Pelo lado de fora da borda.	55
5.11	Resultado da segmentação de um objeto de tecido mole na imagem do joelho, com o uso de segmentos de custo mínimo.	56
5.12	Resultado da segmentação de um segundo objeto de tecido mole na imagem do joelho, com o uso de segmentos de custo mínimo.	56
5.13	Resultado final da segmentação pele usando contornos deformáveis sem segmentos de custo mínimo. A maior sequência de imagens que foi possível segmentar sem uma nova intervenção do usuário foi de 5 fatias: (a) primeira, (b) segunda, (c) terceira, (d) quarta e (e) quinta.	58
5.14	Usando segmentos de custo mínimo foi possível gerar toda a sequência de 20 fatias com apenas uma intervenção do usuário na segmentação da pele. A Figura ilustra cinco fatias desta sequência, as quais são: (a) quarta; (b) oitava; (c) décima sexta e (d) vigésima. O contorno final de uma fatia era exatamente o contorno inicial da seguinte.	59
5.15	Energia da imagem do pé.	60
5.16	(a) Limiarização da Energia da Imagem com valores 100 e 200 para segmentação do talus; (b) Contorno inicial para segmentação do calcâneo.	60
5.17	(a) Contorno final da segmentação do calcâneo usando apenas contornos deformáveis; (b) Contorno final usando segmentos de custo mínimo. As Figuras (c) e (d) mostram a propagação do erro nas fatias seguintes.	61
5.18	(a) Contorno final da segmentação do calcâneo usando as distâncias 3 e 6 pixels (máxima e mínima, respectivamente); (b) Contorno final usando valores 0 e 100 para a limiarização da Energia da Imagem.	62
5.19	Segmentação do tecido mole. (a) Contorno inicial para a segunda fatia de uma sequência de imagens, usando uma constante de contração. (b) Contorno final obtido na segunda fatia. (c) Contorno final obtido na terceira fatia.	63

Capítulo 1

Introdução

Na radiologia de muitos hospitais, equipamentos tomográficos (CT, MRI, SPECT, etc.) produzem diariamente um conjunto de imagens de cortes (ou fatias) contendo informações anatômicas e funcionais de estruturas 3D do corpo humano. A visualização, manipulação e análise dessas estruturas no computador formam o principal objetivo da computação de imagens médicas [UH91]. Suas aplicações compreendem o diagnóstico por imagens, o planejamento de cirurgias e radioterapia, a educação médica, o estudo de diversos fenômenos que alteram a anatomia e/ou função de órgãos do corpo humano, a análise de movimento de articulações, etc. O sucesso da visualização, manipulação e análise em tais aplicações depende principalmente de um método eficiente de “extrair” as estruturas de interesse do conjunto de imagens. Este processo é referido como segmentação [Fal96].

1.1 Motivação

A segmentação de imagens envolve basicamente duas tarefas: reconhecimento e delimitação. O reconhecimento, neste contexto, consiste em determinar a localização do objeto de interesse e distinguir este objeto de outras estruturas similares na imagem. A delimitação consiste em definir com precisão os limites do objeto na imagem. A etapa de reconhecimento é melhor desempenhada por especialistas no domínio da aplicação do que pelo computador. Isto ocorre devido à dificuldade de transformar o conhecimento global relevante sobre o objeto em operações computáveis. Por outro lado, a tarefa de delimitação é melhor desempenhada por algoritmos de computador, que conseguem ser mais repetitivos, mais rápidos e mais exatos do que o ser humano [Fal96].

Devido às dificuldades na etapa de reconhecimento, métodos de segmentação completamente automáticos falham e em muitas situações a única alternativa imediata é o uso da segmentação manual. Isto requer muito esforço e tempo de envolvimento do usuário e muitas vezes inviabiliza o uso da aplicação clínica na rotina do hospital. Isto tem motiva-

do o estudo e desenvolvimento de técnicas interativas que tentam minimizar o esforço e a participação do usuário. Entre os diversos métodos interativos, os métodos de contornos deformáveis têm se tornado populares.

1.2 Contornos Deformáveis

Na literatura de segmentação, métodos de contornos deformáveis têm sido intensamente investigados nos últimos anos [Bri93, CC93, HUS96, Lai94, LL93, LV95, OS97]. Entre estes métodos, o mais conhecido é o *Snakes* [KWT87]. *Snakes* são curvas planares abertas ou fechadas que têm associado um funcional de energia e que se deformam sob a ação de forças internas e externas com o objetivo de minimizar este funcional. A idéia básica é que a partir de uma localização inicial na imagem, a *snake* se deforme até atingir um valor mínimo de energia que deve coincidir com a situação em que a *snake* adere à borda do objeto desejado na imagem. Este modelo é usado para representar uma borda 2D de um objeto na imagem. Muito embora existam extensões deste modelo para segmentar objetos 3D a partir de um conjunto de fatias tomográficas [CC90, CC92, CC93], este trabalho adota a estratégia de segmentar estes objetos fatia por fatia. (Mesmo porque não existe nenhuma evidência na literatura que mostre que, independente da aplicação, a abordagem 3D é mais eficiente do que a abordagem 2D).

1.3 Objetivos

Entre os maiores problemas em contornos deformáveis pode-se citar a dificuldade de convergência e estabilização do contorno sobre a borda do objeto desejado. Este trabalho apresenta uma revisão das principais técnicas de contornos deformáveis e propõe uma solução eficiente para o problema de convergência e estabilização através de segmentos de custo mínimo. O método proposto foi denominado Contorno Deformável usando Segmentos de Custo Mínimo.

Uma outra contribuição é a definição de orientação de borda para contornos deformáveis. A origem desta idéia parte de um método interativo de segmentação, chamado *Live Wire*, o qual também é descrito neste trabalho junto com outros métodos derivados.

1.4 Organização do Texto

Esta dissertação encontra-se organizada da seguinte forma: O Capítulo 2 apresenta uma visão geral sobre métodos de Contornos Deformáveis. Inicia-se com o método *Snakes* original, sua formulação, discretização e sobre diversas extensões que surgiram visando

solucionar os problemas existentes nesta metodologia. O Capítulo 3 apresenta o método *Live Wire* e suas extensões, facilidades, e também dificuldades; mostrando o uso de algoritmos de custo mínimo e de borda orientada na segmentação de imagens médicas. O Capítulo 4 descreve o método de segmentação desenvolvido ao longo deste trabalho, que utiliza contornos deformáveis para a tarefa de reconhecimento e segmentos de custo mínimo para a etapa de delimitação. O Capítulo 5 discute os resultados obtidos, detalhando as etapas e analisando as dificuldades para obtenção dos mesmos. A conclusão do trabalho e proposta de trabalhos futuros, são apresentados no Capítulo 6.

Capítulo 2

Modelos de Contornos Deformáveis

2.1 Introdução

Métodos de Contornos Deformáveis foram originalmente propostos por Kass [KWT87], através do modelo *Snakes*¹. Este tornou-se bastante difundido em segmentação de imagens, levando à criação de várias extensões da idéia original [Bri93, CCK95, CC93, HUS96, Lai94, LL93, Mac95, NFK94, OS97, TK95].

A *snake* é uma linha ou contorno fechado do tipo *spline* de continuidade controlada sob a influência de **forças externas** (provenientes da imagem) e **forças internas** (referentes a características da própria *spline*). As forças internas atuam impondo suavidade ao contorno, enquanto as forças externas empurram a *snake* para características da imagem que se assemelhem a bordas ou segmentos de bordas.

Com base no princípio de minimização de energia, a estratégia da *snake* é encontrar mínimos globais que representem a borda desejada. O modelo recebeu este nome devido ao modo como se dá o comportamento da curva no processo de segmentação.

Este capítulo está dividido da seguinte forma: Na primeira parte, apresenta-se o modelo *Snakes* original [KWT87]. Em seguida, faz-se um *overview* sobre as principais categorias onde se encontram trabalhos relacionados a *snakes*. E, na última parte, a conclusão do capítulo.

2.2 Modelo Snakes Original

O modelo *Snakes* (também conhecido por Contornos Ativos), pode ser definido no plano como um contorno paramétrico fechado composto de pontos, $\mathbf{v}(s) = (\mathbf{x}(s), \mathbf{y}(s))$, onde

¹Durante a apresentação deste trabalho, os termos “método” e “modelo” serão utilizados como sinônimo

$(\mathbf{x}, \mathbf{y}) \in \mathbb{R}^2$, $s \in [0, 1]$ e $\mathbf{v}(0) = \mathbf{v}(1)$ [KWT87]. A equação geral da energia deste modelo é escrita em termo das Energias Interna e Externa:

$$\mathbf{E}_{snake}(\mathbf{v}(s)) = \int_0^1 \{\mathbf{E}_{int}(\mathbf{v}(s)) + \mathbf{E}_{ext}(\mathbf{v}(s))\} ds \quad (2.1)$$

onde,

$$\mathbf{E}_{int}(\mathbf{v}(s)) = \frac{\alpha(s)\|\mathbf{v}'(s)\|^2 + \beta(s)\|\mathbf{v}''(s)\|^2}{2} \quad (2.2)$$

e

$$\mathbf{E}_{ext}(\mathbf{v}(s)) = \mathbf{E}_{image}(\mathbf{v}(s)). \quad (2.3)$$

O objetivo é encontrar a solução que minimiza E_{snake} . O termo \mathbf{E}_{int} representa a Energia Interna do contorno cujo diferencial gera a **Força Interna** que impõe regularidade ao contorno. Os termos v' e v'' representam as derivadas de primeira e segunda ordem, respectivamente. O coeficiente $\alpha(s)$ (onde $\alpha(s) \in [0, 1]$) é o termo que controla a 'tensão' entre os pontos do contorno, enquanto $\beta(s)$, $\beta(s) \in [0, 1]$, é responsável pela 'elasticidade' do contorno. A Força Interna, quando atua sozinha, pode levar a curva a tornar-se um único ponto, por isto deve ser balanceada pela Força Externa, a qual resulta do diferencial da Energia Externa e atrai o contorno para a borda desejada. A Energia Externa é calculada com base no gradiente da imagem para que a *snake* seja atraída para pontos de borda.

A Energia da Imagem pode ser escrita da seguinte forma:

$$\mathbf{E}_{ext}(\mathbf{v}(s)) = \mathbf{E}_{image}(\mathbf{v}(s)) = -(\nabla^2(\mathbf{G}_\sigma * \mathbf{I}(\mathbf{v}(s))))^2 \quad (2.4)$$

onde ∇ representa o operador gradiente, e $\mathbf{G}_\sigma * \mathbf{I}(\mathbf{v}(s))$ denota a imagem trabalhada por um filtro de suavização (Gaussiano), cuja característica de largura σ controla a extensão dos mínimos de $\mathbf{E}_{ext}(\mathbf{v}(s))$. Note que $\mathbf{E}_{ext}(\mathbf{v}(s))$ deve ser baixa onde o gradiente é alto.

Kass [KWT87] menciona também que a intervenção do usuário pode ser levada em conta no cálculo da Energia Externa. Quando isto ocorre, a equação da Energia Externa (2.3) é representada por:

$$\mathbf{E}_{ext}(\mathbf{v}(s)) = \mathbf{E}_{image}(\mathbf{v}(s)) + \mathbf{E}_{user}(\mathbf{v}(s)) \quad (2.5)$$

onde E_{user} representa a energia do usuário².

2.3 Discretização

Embora a maior parte dos modelos *Snakes* sejam apresentados em uma formulação contínua, a discretização das equações é inevitável quando o modelo vai ser implementado. A discretização do modelo original é apresentada a seguir.

$$E_{snake}(\mathbf{v}) = \sum_{i=1}^n (E_{int}(\mathbf{v}_i) + E_{ext}(\mathbf{v}_i)) \quad (2.6)$$

onde o contorno é representado por um conjunto V de n pontos $v_i = (x_i, y_i)$, $v_1 = v_n$, $E_{int}(v_i)$ é a Energia Interna do contorno no ponto v_i e $E_{ext}(v_i)$ é a Energia Externa avaliada no mesmo ponto. As equações de Euler, que garantem o valor mínimo do funcional da equação 2.6, podem ser escritas como:

$$\alpha_i(\mathbf{v}_i - \mathbf{v}_{i-1}) - \alpha_{i+1}(\mathbf{v}_{i+1} - \mathbf{v}_i) + \beta_{i-1}(\mathbf{v}_{i-2} - 2\mathbf{v}_{i-1} + \mathbf{v}_i) - 2\beta_i(\mathbf{v}_{i-1} - 2\mathbf{v}_i + \mathbf{v}_{i+1}) + \beta_{i+1}(\mathbf{v}_i - 2\mathbf{v}_{i+1} + \mathbf{v}_{i+2}) + (\mathbf{f}_x(\mathbf{v}_i), \mathbf{f}_y(\mathbf{v}_i)) = 0 \quad (2.7)$$

onde $f_x(v_i) = \partial E_{ext}/\partial x_i$ e $f_y(v_i) = \partial E_{ext}/\partial y_i$, e o restante decorre das derivadas parciais de $E_{int}(v_i)$.

Assumindo valores constantes para α e β , e isolando os termos referentes às Forças Interna e Externa, pode-se reescrever a equação 2.7 da seguinte forma:

$$\mathbf{F}_{int}(\mathbf{v}_i) = \alpha(\mathbf{v}_i - \mathbf{v}_{i-1}) - \alpha(\mathbf{v}_{i+1} - \mathbf{v}_i) + \beta(\mathbf{v}_{i-2} - 2\mathbf{v}_{i-1} + \mathbf{v}_i) - 2\beta(\mathbf{v}_{i-1} - 2\mathbf{v}_i + \mathbf{v}_{i+1}) + \beta(\mathbf{v}_i - 2\mathbf{v}_{i+1} + \mathbf{v}_{i+2}) = -(\mathbf{f}_x(\mathbf{v}_i), \mathbf{f}_y(\mathbf{v}_i)) = -\mathbf{F}_{ext}(\mathbf{v}_i) \quad (2.8)$$

Quando ocorre esta derivação nas energias Interna e Externa, passa-se a falar em Força Interna e Externa, respectivamente.

Se os termos da Força Interna forem separados, chamando de $F_{int_1}(v_i)$ a parte que contém os elementos dependentes de α e de $F_{int_2}(v_i)$ aqueles que apresentam β , tem-se:

$$\begin{aligned} F_{int_1}(v) &= \alpha(v_i) - \alpha(v_{i-1}) - \alpha(v_{i+1}) + \alpha(v_i) \\ &= -\alpha(v_{i-1}) + 2\alpha(v_i) - \alpha(v_{i+1}) \end{aligned} \quad (2.9)$$

²A energia E_{user} não é utilizada nas bibliografias que citam o trabalho de Kass; tão pouco será utilizada como parte da formulação no decorrer deste trabalho

$$\begin{aligned}
F_{int_2}(v) &= \beta(v_{i-2} - 2v_{i-1} + v_i) - 2\beta(v_{i-1} - 2v_i + v_{i+1}) + \beta(v_i - 2v_{i+1} + v_{i+2}) \\
&= \beta v_{i-2} - 2\beta v_{i-1} + \beta v_i - 2\beta v_{i-1} + 4\beta v_i - 2\beta v_{i+1} + \beta v_i - 2\beta v_{i+1} + \beta v_{i+2} \\
&= \beta v_{i-2} - 4\beta v_{i-1} + 6\beta v_i - 4\beta v_{i+1} + \beta v_{i+2} \quad (2.10)
\end{aligned}$$

Assim, reunindo outra vez $F_{int_1}(v_i)$ e $F_{int_2}(v_i)$, podemos escrever a equação geral da seguinte forma:

$$F_{int}(\mathbf{v}_i) = -\alpha \mathbf{v}_{i-1} + 2\alpha \mathbf{v}_i - \alpha \mathbf{v}_{i+1} + \beta \mathbf{v}_{i-2} - 4\beta \mathbf{v}_{i-1} + 6\beta \mathbf{v}_i - 4\beta \mathbf{v}_{i+1} + \beta \mathbf{v}_{i+2} \quad (2.11)$$

Colocando os valores de v em evidência, tem-se:

$$F_{int}(\mathbf{v}_i) = \mathbf{v}_{i-2}(\beta) + \mathbf{v}_{i-1}(-\alpha - 4\beta) + \mathbf{v}_i(2\alpha + 6\beta) + \mathbf{v}_{i+1}(-\alpha - 4\beta) + \mathbf{v}_{i+2}(\beta) \quad (2.12)$$

$$\begin{pmatrix} \beta & (-\alpha - 4\beta) & (2\alpha + 6\beta) & (-\alpha - 4\beta) & \beta \end{pmatrix} \begin{pmatrix} v_{i-2} \\ v_{i-1} \\ v_i \\ v_{i+1} \\ v_{i+2} \end{pmatrix} = -F_{ext}(\mathbf{v}_i) \quad (2.13)$$

A partir desta equação, obtêm-se o seguinte sistema linear:

$$AV + F = 0 \quad (2.14)$$

sendo:

A uma matriz pentadiagonal,

V a matriz de posições v_i

e **F** a matriz com a Força Externa nos pontos v_i .

A estrutura deste sistema linear é melhor compreendida no exemplo abaixo. Seja o conjunto V de pontos (x, y) , sendo n o número de elementos de V , com $n = 7$. $V = \{v_1, v_2, v_3, v_4, v_5, v_6, v_7\}$.

A estrutura da matriz A para uma *snake* fechada é:

$$\begin{pmatrix} (2\alpha + 6\beta) & (-\alpha - 4\beta) & \beta & 0 & 0 & \beta & (-\alpha - 4\beta) \\ (-\alpha - 4\beta) & (2\alpha + 6\beta) & (-\alpha - 4\beta) & \beta & 0 & 0 & \beta \\ \beta & (-\alpha - 4\beta) & (2\alpha + 6\beta) & (-\alpha - 4\beta) & \beta & 0 & 0 \\ 0 & \beta & (-\alpha - 4\beta) & (2\alpha + 6\beta) & (-\alpha - 4\beta) & \beta & 0 \\ 0 & 0 & \beta & (-\alpha - 4\beta) & (2\alpha + 6\beta) & (-\alpha - 4\beta) & \beta \\ \beta & 0 & 0 & \beta & (-\alpha - 4\beta) & (2\alpha + 6\beta) & (-\alpha - 4\beta) \\ (-\alpha - 4\beta) & \beta & 0 & 0 & \beta & (-\alpha - 4\beta) & (2\alpha + 6\beta) \end{pmatrix} \quad (2.15)$$

A formulação gerada faz com que o comportamento da curva seja comparável a um conjunto de massas unidas por molas de comprimento zero (quando totalmente contraída). Consequentemente, se não há Força Externa ($F_{ext} = 0$), a curva se contrai a um único ponto.

Uma outra abordagem para discretização da *snake* é apresentada por Cohen [CC93], utilizando-se do método de elementos finitos, ao invés de diferenças finitas.

2.3.1 Análise das Dificuldades

Uma das principais dificuldades do modelo *Snakes* original é encontrar a melhor localização para o contorno inicial. Se este não estiver próximo ao objeto de interesse, ele não é atraído para a borda do objeto podendo se reduzir a um ponto na imagem ou se prender a pontos de mínimo local de energia. Para resolver este problema, [Lai94] propõe um procedimento automático baseado em Transformada de Hough para gerar o contorno inicial. Na prática, porém, a forma mais eficiente é a seleção manual feita pelo usuário.

Esta sensibilidade do contorno à sua localização inicial deve-se ao fato da energia ser, em geral, não convexa. Este problema, por sua vez, está diretamente relacionado com a escolha dos parâmetros do modelo. Isto porque, são os parâmetros que determinam a importância de cada termo do funcional de energia. Esta alta sensibilidade dos resultados à escolha dos parâmetros tem sido bastante discutida na literatura [LL93, GN97].

A escolha certa dos parâmetros de elasticidade (β) e rigidez (α) do modelo também tem um forte efeito sobre o comportamento da curva, tornando difícil a aderência em protusões e indentações da borda.

Outro problema do modelo *snakes* original é não dar suporte a mudanças de topologia da curva, pois este não realiza operações de *split* e *merge*, limitando sua aplicação a objetos totalmente conexos.

A condição de parada do contorno é outra dificuldade pertinente ao modelo. Devido ao fato de que critérios de parada são diretamente ligados ao funcional de energia da curva, a curva pode ficar oscilando entre mínimos locais ou ainda, se estiver numa região homogênea da imagem, continuar seu deslocamento e tornar-se um único ponto. Isto pode ocorrer se o objeto de interesse tiver suas bordas fracas, o que leva o contorno a comportamentos instáveis.

Estas dificuldades têm sido a principal motivação de muitas outras propostas de Contornos Deformáveis [Boi88, MT96].

Na seção a seguir, apresenta-se três principais categorias onde se concentram diversas extensões propostas ao modelo *Snakes* original.

2.4 Snakes: extensões ao modelo original

As principais extensões ao modelo original podem ser classificadas como:

- a) Modelos Contínuos Paramétricos
- b) Modelos Contínuos Implícitos
- c) Modelos Dinâmicos

2.4.1 Modelos Contínuos Paramétricos

Esta é, provavelmente, a classe de contornos deformáveis que reúne o maior número de extensões propostas. Eles consideram como ponto de partida o método *Snakes* original.

Nos modelos Contínuos Paramétricos, o princípio de minimização de energia é tratado através de forças, derivando a equação geral da *snake* para obter uma Força Resultante ($FR(c(s))$) nula em pontos de mínimo. O funcional de energia (equação 2.1) é resolvido através da equação de Euler-Lagrange, gerando o seguinte sistema de equações:

$$\frac{\delta E_{snake}(v(s))}{\delta s} = -(\alpha v'(s))' + (\beta v''(s))'' + \frac{\delta E_{image}(v(s))}{\delta s} = 0 \quad (2.16)$$

sendo que:

$$\frac{\delta E_{snake}(v(s))}{\delta s} = FR(c(s)) = 0 \quad (2.17)$$

onde:

$$FR(c(s)) = \sum (F_{int}(v_i) + F_{ext}(v_i)) \quad (2.18)$$

Dentre os destaques, há o método de Cohen [Coh91]. Cohen propôs o uso de uma força adicional, chamada *Força Balão* a fim de resolver o problema da não convexidade da curva para as bordas. Esta atua especialmente em regiões homogêneas, onde a Força Externa é nula.

Força Balão

Uma das propostas mais interessantes para melhorar o comportamento da *snake* é a utilização da Força Balão ($F_{bal}(v(s))$) [Boi88, Bri93, Mac95] baseada em propostas anteriores como a já apresentada por Cohen e Cohen [CC90, CC92, CC93]. Esta é uma componente da F_{ext} e objetiva mover os pontos na direção normal à curva em cada ponto na direção

inversa de F_{int} , evitando assim que o contorno se reduza a um único ponto ou fique preso em mínimos locais de energia.

Observe que a Força Balão pode ser usada em um contorno deformável fechado para forçar o contorno a expandir ou reduzir, na ausência de influências externas. A motivação é que um contorno inicializado dentro de uma imagem uniforme de objeto irá expandir sob a influência da Força Balão até que este se aproxime da borda do objeto (quando a Força Externa começa a afetar seu movimento).

Em [CC93] o modelo do balão é estendido para imagens tridimensionais de Ressonância Magnética (MRI), aplicando-se equações diferenciais parciais para modelar a deformação de superfícies. Neste trabalho Cohen também apresenta uma comparação entre os resultados obtidos e desempenho do algoritmo utilizando método de diferenças finitas e o método de elementos finitos, visando generalizar o uso do balão para 3D.

Cohen comenta ainda que o uso de diferenças finitas só permite saber o valor da função em pontos discretos e não há informação entre os snaxels. Para garantir a melhor precisão possível nos resultados, as distâncias entre snaxels sucessivos devem ser bem pequenas. Entretanto, quando se trabalha com métodos de elementos finitos, funções contínuas são utilizadas, e assim os valores da função são considerados em qualquer lugar na imagem.

2.4.2 Modelos Contínuos Implícitos

Os modelos Contínuos Implícitos têm a proposta de tratar mudanças de topologia durante a segmentação. Nos modelos Contínuos Implícitos, o contorno deformável é definido não simplesmente como uma curva no plano bidimensional, mas pela intersecção de uma superfície Ψ com o plano xy . Desta forma, a evolução do contorno está vinculada à evolução da superfície (e vice-versa), sendo o movimento determinado por equações do tipo *Hamilton-Jacobi* [Kec98, OS88] para a função $\Psi(x, y, t)$.

Neste contexto, mudanças topológicas são facilmente observadas, uma vez que a intersecção da superfície com o plano pode mudar sua topologia de forma contínua e suave durante a evolução.

Um dos principais exemplos desta abordagem é apresentado por Malladi et.al. em [MSV95], que utiliza o conceito de Conjuntos de Níveis (*Level sets*).

O modelo de Malladi

O modelo apresentado por Malladi et. al. [MSV95] auxilia na segmentação de objetos em movimento numa dada sequência de imagens, garantindo a segmentação dos mesmos ainda que, durante a evolução, alterem sua conectividade e partam-se. Isto porque a topologia do contorno final independe de seu formato inicial. Além disso, o algoritmo permite o surgimento de ramos no contorno à medida que este se move.

Uma instância única do modelo tem a capacidade de se repartir livremente para representar mais de um objeto presente em seu interior. O contorno é fechado, sem intersecções e move-se ao longo do campo do gradiente com velocidade constante ou uma velocidade dependente da curvatura.

Esta abordagem apresenta como um dos principais problemas[MSV95] o seguinte aspecto: pequenos erros nas posições das partículas computadas são ampliados à medida que a curva se movimenta. Por outro lado, uma das principais vantagens deste método é tratar mudanças de topologia, quebras, união de partes, formação de cantos, enquanto a superfície se desenvolve.

Metodologias que seguem o paradigma “contínuos implícitos” são de fato melhores para a detecção de objetos de formas complexas e topologias desconhecidas. Entretanto, devido à formulação complexa, estes modelos não são tão convenientes nem tão intuitivos quanto os modelos paramétricos, tão pouco permitem uma maior interação com o usuário.

2.4.3 Modelos Dinâmicos

Nos modelos Dinâmicos, procurou-se resolver o problema da não convexidade da curva. A solução proposta foi imergir o modelo original em um contexto dinâmico, onde a curva possui uma aceleração que lhe permite não se prender a mínimos locais.

Neste tipo de modelo a energia do contorno é formulada sobre uma dimensão finita e os mínimos (bordas) desejados são refinados localmente, num trabalho que se assemelha ao de esculpir a borda. Como exemplos de *snakes* nesta abordagem, têm-se D-NURBS [QT96] e *Splines Ativas* [BY93].

Usando a mecânica Lagrangeana, obtém-se um modelo de contornos deformáveis dinâmico que unifica forma e movimento, tornando possível realizar a segmentação de uma forma no decorrer do tempo.

Neste contexto, as curvas deformáveis são representadas como um conjunto de pontos, chamados de pontos de controle, sobre os quais são aplicadas forças linearmente independentes.

À medida que o tempo passa ocorrem mudanças no sistema, e o ponto que o caracteriza no hiperespaço se desloca, traçando uma curva que define a evolução do sistema. A expressão *movimento do sistema* corresponde ao movimento do ponto, gerando a curva parametrizada na forma:

$$\mathbf{c}(t) = (\mathbf{q}_1(t), \mathbf{q}_2(t), \dots, \mathbf{q}_n(t)) \quad (2.19)$$

Como critérios de parada, existem basicamente duas propostas na literatura: A primeira delas, proposta por Cohen [Coh91] e a segunda por Leymarie [LL93]. Ambas partem do princípio que as bordas são vales nos quais a curva deve ficar confinada, i.e., assumem que a curva apresentará uma baixa velocidade na vizinhança da borda.

De fato, uma importante característica destes métodos é que a *snake* apresenta uma inércia, podendo assim escapar de mínimos locais, aumentando suas chances de chegar a um mínimo global. Entretanto, para uma real eficiência dos métodos é necessário encontrar o melhor compromisso entre os parâmetros do modelo, o que na prática é um desafio.

2.5 Modelos Híbridos

Não se pode afirmar que a intersecção entre as abordagens acima propostas seja nula. Mas, além das características em comum nos princípios matemáticos que utilizam, há um crescente interesse dos pesquisadores em criarem modelos híbridos, visando uma formulação mais simplificada, ganho de desempenho e um campo de aplicação cada vez maior.

Um exemplo disto é o modelo *Dual Topologically Adaptable Snakes* (Dual T-Snakes), proposto por Giraldi [GGA00]. Sua origem está fundamentada em dois outros métodos de *Snakes*: O *Dual Active Contour*, proposto por Gunn e Nixon em [GN97], e no trabalho de McInerney, o qual desenvolveu o método *Topologically Adaptable Snakes* (T-Snakes) [MT95, MT97, McI97]. A T-Snakes teve como um de seus métodos inspiradores o modelo de Malladi [MSV95].

2.5.1 Dual T-Snakes

Inicialmente, é necessário compreender sobre o que trata a abordagem “Dual” e o que caracteriza uma “T-Snakes”; isto porque o modelo Dual T-Snakes utiliza-se de duas T-Snakes, uma Interna e outra Externa ao objeto de interesse, que se deslocam em direções opostas, até se encontrarem, transformando-se num único contorno. Esta estratégia visa proporcionar maior estabilidade do método e convergência da curva em direção à borda.

A abordagem “Dual”

Esta abordagem consiste de dois contornos iniciais: um inicializado por fora do objeto de interesse, que se contrai à medida que a *snake* evolui, e outro inicializado dentro, o qual se expande a cada evolução. A borda é definida quando os dois contornos se fundem, tornando-se apenas um. Isto ocorre por ser estabelecida uma correspondência entre os dois contornos, interligando-os por uma força adicional (*driving force*), utilizada para retirar o contorno de mínimos locais (Figura 2.1).

Dentre as vantagens desta abordagem tem-se: (a) não existe preocupação de vincular a *snake* a um protótipo; (b) evita-se que a curva fique presa em mínimos locais. Como uma das principais desvantagens tem-se: a necessidade de correspondência entre os pontos das

snakes impõe restrições às formas das mesmas, pois se os dois contornos não têm a mesma forma, encontrar uma correspondência entre os pontos pode ser uma tarefa complicada.

Em [Gun96] é proposta uma outra formatação para este modelo, utilizando programação dinâmica.

T-Snakes

Inspirado nos modelos implícitos, McInerney propôs um modelo chamado de “Topologically Adaptable Snakes”, mais conhecido como T-Snakes [MT95, MT97, McI97], o qual permitiu incorporar mudanças topológicas às *snakes* porém com uma abordagem numérica bem mais simples do que aquela utilizada por Malladi [MSV95] e mais genérica que no modelo Dual [GN97].

Este modelo parte do princípio de colocar a *snake* sobre uma malha triangular (a imagem é vista como sendo formada por um conjunto de triângulos. Este mecanismo é chamado de triangularização) (Figura 2.2). Os *snaxels* (pontos sobre a *snake*) são tomados a partir de intersecções entre a *snake* e a malha de triângulos. Isto leva a uma reparametrização da *snake* baseada na subdivisão do espaço no qual a mesma está imersa. **Não há uma preocupação explícita em minimizar um funcional de energia** pois a evolução da *snake* acontece baseada em informações extritamente locais [McI97].

A reparametrização da curva sobre os vértices dos triângulos gera uma perturbação na *snake*, o que se torna desvantagem principalmente próximo às bordas desejadas. Isto gera portanto uma deficiência no método quando pensamos, por exemplo, numa segmentação para reconstrução e consequente visualização do objeto em 3D. Por outro lado, é a reparametrização que permite a implementação simples de mudanças topológicas.

Outras vantagens deste método são: (a) sua generalidade, pois permite tanto o agrupamento (*merge*) quanto o particionamento (*split*) da *snake*; (b) sua facilidade de extensão para superfícies, tendo em vista que os elementos básicos utilizados são eficientemente extensíveis para 3D [McI97].

Quando acontece um *split* numa *snake*, cada uma das *snakes* geradas serão parte de um novo *dual snake*. Cada conjunto *dual snake* é independente dos outros, permitindo que sejam processados em paralelo.

Uma T-Snake é definida como um conjunto de N partículas (*snaxels*) conectadas entre si de modo a formarem um contorno fechado.

Cada par de *snaxel* (chamado *elemento do modelo*) é ligado por molas, definidas por uma constante de elasticidade $\alpha(v_i)$ e um comprimento $l(v_i)$. A Força Elástica correspondente é adicionada a uma Força de Rigidez, uma Força Normal (Balão) e a Força da Imagem. Estas forças se apresentam da seguinte forma:

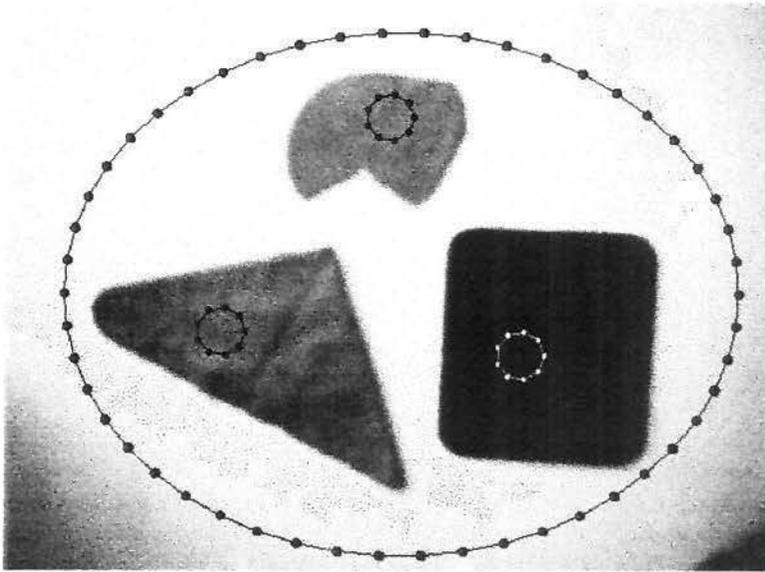


Figura 2.1: Exemplo de inicialização da Dual T-Snakes em três objetos distintos. Observe que há um contorno interno para cada objeto de interesse e um único contorno externo envolvendo a todos.

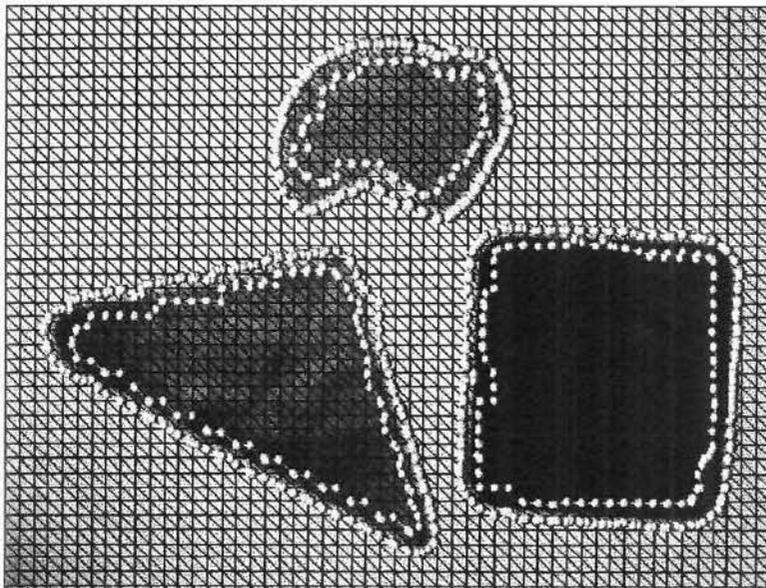


Figura 2.2: Ilustração da malha de triângulos utilizada pelo modelo Dual-T-Snakes.

Força Elástica:

$$\mathbf{F}_{el}(\mathbf{v}_i) = \alpha(\mathbf{v}_i)\mathbf{e}(\mathbf{v}_i)\mathbf{r}(\mathbf{v}_i)(t) - \alpha(\mathbf{v}_{i-1})\mathbf{e}(\mathbf{v}_{i-1})\mathbf{r}(\mathbf{v}_{i-1})(t) \quad (2.20)$$

onde $\|\mathbf{r}(\mathbf{v}_i)(t)\| = \|v_{i+1} - v_i\|$ e $e(v_i) = \|\mathbf{r}(\mathbf{v}_i)(t)\| - l(v_i)$.

Força de Rigidez: utilizada a fim de minimizar a curvatura local do contorno. Esta Força procura minimizar a distância entre o ponto v_i e o centróide de seus vizinhos.

$$\mathbf{F}_{rig}(\mathbf{v}_i) = \beta(\mathbf{v}_i)(\mathbf{v}_i - \frac{1}{2}(\mathbf{v}_{i-1} + \mathbf{v}_{i+1})) \quad (2.21)$$

Força Normal: atua como a Força Balão, visando empurrar a *snake* em direção aos segmentos de borda da imagem até que esta seja oposta à Força da Imagem. É definida por:

$$\mathbf{F}_{normal}(\mathbf{v}_i) = k\mathbf{n}(\mathbf{v}_i), \quad (2.22)$$

sendo $n(v_i)$ a normal no snaxel v_i e k o parâmetro da Força que carrega um sinal, positivo ou negativo, definido de acordo com as propriedades do objeto na imagem, como um limiar de brilho [McI97].

Todas as três equações acima (2.20, 2.21, 2.22) são referentes à Força Interna.

A Força Externa, baseada em dados da imagem, é definida por:

$$\mathbf{F}_{ext}(\mathbf{v}_i) = -\gamma(\mathbf{v}_i)\nabla\|\nabla I\|^2, \quad (2.23)$$

sendo I a imagem, ∇ o gradiente da imagem, e γ um parâmetro para controle da Força.

A equação de evolução de cada T-Snake torna-se:

$$\mathbf{v}_i(t + \Delta t) = \mathbf{v}_i(t) + (\mathbf{F}_{el}(\mathbf{v}_i) + \mathbf{F}_{rig}(\mathbf{v}_i) + \mathbf{F}_{normal}(\mathbf{v}_i) + \mathbf{F}_{ext}(\mathbf{v}_i)), \quad (2.24)$$

onde é assumido que $i = 0, \dots, N - 1$.

Maiores detalhes sobre o funcionamento do modelo podem ser encontrados em [GGA00].

2.6 Conclusão

Snakes são um tipo de modelo deformável. Originalmente proposto por Kass et.al. [KWT87], este modelo, também chamado de Contornos Ativos, tem sido largamente discutido e investigado. Ele parte do princípio que, dado um contorno inicial, este se deforma em direção à borda do objeto desejado em função da atuação de duas forças: uma Interna, referente a características físicas do próprio contorno, e outra Externa, baseada em

características da imagem. Através da minimização do funcional de energia, o contorno é conduzido à borda desejada.

Dentre as dificuldades apresentadas por este modelo está a questão da inicialização do contorno, que requer que seja feita próxima da borda desejada, além de requerer múltiplas inicializações, uma por objeto de interesse, etc. Outra questão é a não convergência do contorno à borda desejada, que pode ficar preso em mínimos locais não desejados. Outra questão é causada pela atuação da Força Interna que tem a tendência natural de fazer o contorno se contrair, podendo levá-lo a tornar-se um único ponto se a Força Externa não for forte o bastante. Encontrar portanto um compromisso bom o bastante entre os parâmetros do contorno, é um desafio difícil de se resolver na prática.

Várias abordagens tem sido utilizadas visando resolver problemas das *Snakes*. Basicamente os modelos de *Snakes* podem ser classificados em três categorias: Modelos Paramétricos, Modelos Implícitos e Modelos Dinâmicos. Vantagens e desvantagens destes modelos são [MT96, GO99]:

Os Modelos Paramétricos são melhores para tratar com objetos de topologias mais simples. A topologia das estruturas de interesse deve ser conhecida previamente, uma vez que o modelo matemático utilizado não lida com mudanças topológicas. Além disso, os parâmetros da *snake* são muito sensíveis a suas condições iniciais devido a não convexidade do funcional de Energia e da Força de Contração, que cresce devido ao termo da Energia Interna.

Os Modelos Implícitos tratam a *snake* como sendo uma intersecção da curva com um plano. Este tipo de modelo é mais adequado a segmentações de objetos de topologia complexa ou desconhecida. Entretanto, sua formulação não é tão simples quanto a dos métodos paramétricos, tão pouco é um método muito recomendável quando se pensa em interatividade.

Modelos Dinâmicos têm uma preocupação maior em evitar problemas causados pela convergência em mínimos locais não desejados. Entretanto, eles têm sua desvantagem em nível de desempenho.

É certo que esta divisão em categorias não gera modelos mutuamente exclusivos. Apesar de ser possível afirmar que os modelos paramétricos têm sido usados em um grande número de aplicações por sua "simplicidade" de formulação, os modelos dinâmicos por oferecerem a vantagem de se trabalhar com mínimos locais, e os modelos implícitos por terem maior flexibilidade a mudanças topológicas, de forma crescente os pesquisadores têm procurado desenvolver modelos híbridos, de maneira que possam contar com as vantagens de mais de uma dessas categorias, e.g. [GGA00, MT96].

Capítulo 3

Detecção de Bordas através de Segmentos de Custo Mínimo

3.1 Introdução

Além dos modelos de contornos deformáveis, uma outra abordagem que serviu de base para o desenvolvimento do modelo utilizado neste trabalho é a de algoritmos em grafos para detecção de bordas. Esta idéia foi originalmente proposta por Martelli [Mar71] e recentemente retomada por Falcão [Fal96].

A estratégia utilizada nesta metodologia é tratar a imagem como um grafo onde elementos da imagem (pixels ou vértices de pixels) são os nós do grafo e uma relação de adjacência define os arcos. A partir disto, o problema de detectar um segmento de borda passa a ser o de encontrar um caminho de custo mínimo no grafo ligando dois pontos.

No método apresentado por Martelli [Mar71, Mar72] um segmento de borda é detectado usando o algoritmo A^* partindo de um pixel x_a e chegando a um pixel x_b (ambos selecionados sobre a borda do objeto na imagem). Este segmento é representado pelo caminho "ótimo" que parte de um nó x_a e chega ao nó x_b em um grafo G direcionado. Uma função de custo $f(x_i) = \tilde{g}(x_i) + \tilde{h}(x_i)$ representa o custo do caminho entre x_a e x_b passando por x_i . A parte $\tilde{g}(x_i)$ refere-se ao custo do caminho que parte de x_a e chega a x_i . A parte $\tilde{h}(x_i)$ refere-se à estimativa do custo de um caminho que parte de x_i e chega a x_b . Qualquer informação heurística usada para resolver o problema encontra-se em $\tilde{h}(x_i)$. Se $\tilde{h}(x_i) = 0$ (indicando que nenhuma heurística é utilizada) o algoritmo A^* coincide com o algoritmo de Dijkstra [Dij59], o qual sempre garante um caminho ótimo como o caminho de custo mínimo global em G [Nil80]. Se $\tilde{h}(x_i) \neq 0$, A^* nem sempre garante o caminho ótimo mas pode encontrar a solução do problema mais rapidamente que o algoritmo de Dijkstra.

Existem discussões sobre variações ao algoritmo utilizado por Martelli [BB82], visando

reduzir o tempo de processamento do mesmo usando diferentes heurísticas. O grafo utilizado porém nem sempre garante que existe pelo menos um caminho partindo de x_a e chegando a x_b .

Nos métodos *live* [Fal96], x_a e x_b são vértices de pixel e o grafo G utilizado sempre garante que para qualquer nó inicial de G existe pelo menos um caminho que parte deste nó e chega a qualquer outro nó de G . O algoritmo utilizado é baseado na implementação de Dial do algoritmo de Dijkstra, onde a solução é encontrada em tempo real utilizando interação com o usuário e sem qualquer restrição de contorno ou heurística [FUM99].

Tendo em vista que o objetivo deste capítulo é introduzir conceitos necessários para compreensão do método de **Contornos Deformáveis utilizando Segmentos de Custo Mínimo (CDSCM)** e contextualizá-lo entre outros métodos que adotam a mesma abordagem de algoritmos em grafos na tarefa de segmentação de imagens, o enfoque será dado na apresentação do método *live-wire*, o original dentre os métodos *live* e aquele que tem sofrido alterações, visando garantir uma melhor segmentação mesmo em meio a falta de informações na imagem (como ruídos ou ausência de borda) [FUM99]. Um outro motivo para esta concentração no “Live- Wire” deve-se ao fato de que os outros métodos *live* (*Live-Lane* e *3D-Live-Wire* [Fal96]) são na verdade extensões do primeiro, utilizando o mesmo algoritmo de caminhos mínimos, mas adotando estratégias diferentes na tarefa de reconhecimento da borda; estas estratégias serão explanadas na seção 3.2.3.

3.2 Live-Wire

O método *live-wire* foi inicialmente utilizado na segmentação de imagens médicas. Sendo assim, faz-se necessário algumas considerações iniciais. Quando se realiza um exame tomográfico de uma determinada estrutura do corpo humano, como resultado tem-se uma sequência de imagens (fatias) bidimensionais equidistantes. Neste trabalho, cada fatia é referida como uma cena 2D; ao conjunto destas fatias denomina-se uma cena 3D. E, quando considerada variações do conjunto no tempo denomina-se cena 4D.

O método *live-wire* é classificado como interativo. Nele, uma borda de objeto 2D, 3D ou 4D em uma fatia (cena 2D) é definida como um contorno “fechado”, “conectado” e “orientado” formado pelas arestas dos pixels desta fatia (Figura 3.1).

- (a) **Borda Fechada** Entende-se como borda fechada aquela que separa o espaço digital em duas componentes disjuntas (interior e exterior) de tal forma que qualquer “caminho” iniciado no interior e terminado no exterior, cruza a borda.
- (b) **Borda Orientada** É a característica que permite que uma das componentes seja identificada como constituindo o interior do objeto e a outra como o exterior.

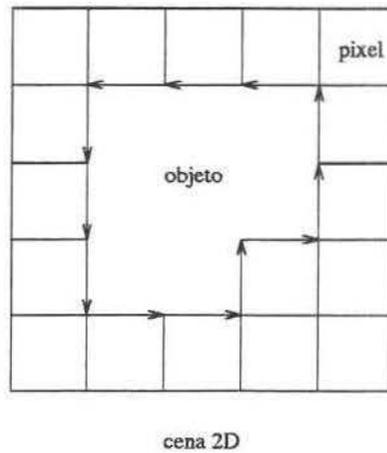


Figura 3.1: Borda de um objeto em uma cena 2D definida como um contorno fechado, conectado e orientado formado por arestas de pixel.

- (c) **Borda Conectada** Esta propriedade garante que há uma única curva conectada e esta não se cruza em nenhum ponto.

Com estas características, pode-se afirmar que o contorno a definir a borda é uma curva digital de Jordan [Her92]. Assim, dado um objeto numa cena 3D ele pode ser representado por um ou mais contornos fechados, orientados e conectados. Segmentar este objeto é delinear todas suas bordas, fatia por fatia.

3.2.1 Conceitos básicos

Nos métodos *live*, extrair a borda de um objeto significa encontrar segmentos de borda ótimo na cena, o que é transformado no problema de encontrar caminhos de custo mínimo entre dois pares de vértices no grafo correspondente. Isto ocorre porque uma imagem é vista como um grafo direcionado, onde os vértices dos pixels são os nós do grafo e as arestas orientadas dos pixels são os arcos (i.e. cada aresta de pixel define dois arcos entre nós adjacentes no grafo, mas com orientações opostas).

A estratégia utilizada divide a segmentação em duas etapas: **Classificação** e **Conectividade**.

Classificação: nesta etapa associa-se um valor de custo para cada arco do grafo.

Conectividade: nesta outra etapa, encontra-se o caminho de custo mínimo entre dois vértices do grafo.

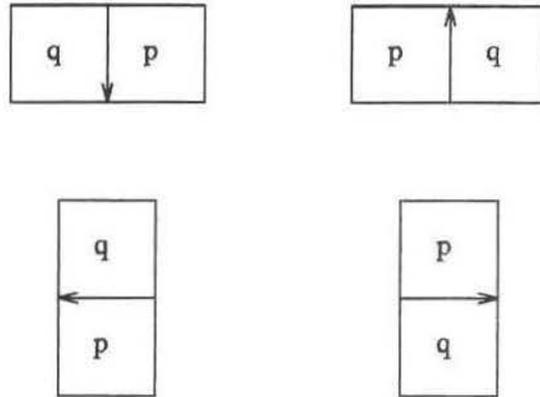


Figura 3.2: Um elemento da borda (*bel*) é uma aresta de pixel orientada. Os quatro possíveis tipos de *bels* em uma cena 2D são mostrados. Para qualquer orientação da borda, um *bel* $b = (p, q)$ é sempre definido tal que p está à esquerda de b e q está à direita. Para uma borda orientada no sentido anti-horário, o pixel p está no interior do objeto e o pixel q no exterior.

Classificação

Na etapa de **classificação**, para cada aresta orientada de pixel são associadas características de imagem, tais como intensidade de pixel e gradiente. Os valores destas características são convertidos em um único valor de custo no intervalo $[0, 1]$ por aresta orientada. Associa-se **custos baixos** para as arestas orientadas que têm propriedades similares à borda orientada desejada (a orientação da borda é definida no início do processo) e **custos altos** às demais.

De maneira mais formal, tem-se: Uma cena 2D, denotada C , é um par (C, g) que consiste de uma matriz C de pixels, finita, retangular, chamada “domínio da cena” e uma função $g(p) : C \rightarrow [L, H]$, chamada “intensidade da cena” que associa a cada pixel p de C um valor de intensidade dentro do intervalo $[L, H]$. Cada aresta de pixel em C tem associada duas possíveis orientações. Toda aresta de pixel orientada em C é em potencial um elemento de borda chamado “*bel*” (*boundary element*). Um *bel* de C é, portanto, um par ordenado (p, q) de pixels que compartilha uma aresta. Todo *bel* $b = (p, q)$ de C tem uma localização e uma orientação. A localização de b é determinada como sendo a única aresta comum a p e q . A orientação de b é assumida tal que p está sempre à esquerda de b e q está sempre à direita (Figura 3.2).

O objetivo é associar a cada *bel* b em C um valor no intervalo $[0, 1]$ que representa o custo de ter b como elemento da borda desejada. Uma “boa” classificação deve associar custos baixos aos *bels* da borda desejada e custos altos às demais.

Seleção de Características da Imagem. Existem várias funções que podem ser utilizadas para mapear características em custos na classificação dos *bels*. Basicamente,

podem ser divididas em dois grupos: *funções sensíveis à orientação da borda* e *funções independentes da orientação da borda*. Duas possíveis orientações para delinear uma borda de objeto em C são: sentido horário e sentido anti-horário. Se uma borda no sentido anti-horário é escolhida, por exemplo, os pixels p dos *bels* $b = (p, q)$ que formam esta borda estão no interior do objeto e os pixels q estão no exterior (Figuras 3.1, 3.2). Em outras palavras, para um dado *bel* b de C , candidato a elemento da borda desejada, o interior do objeto é o assumido como estando à esquerda de b e o exterior à direita. Levando em conta esta suposição, e associando valores diferentes de custo para *bels* de orientações opostas que compartilham o mesmo par de pixels, a função é dita ser sensível à orientação da borda. Isto é muito útil para diferenciar segmentos de borda com contrastes similares e orientações opostas. Por outro lado, funções independentes da orientação da borda atribuem valores iguais de custo para *bels* que compartilham o mesmo par de pixels. Um conjunto de funções independentes da orientação e sensíveis à orientação, podem ser vistas em [Fal96].

Conectividade

Na **conectividade**, para quaisquer dois pontos (vértices de pixel) especificados na borda, o problema de encontrar o segmento de borda ótimo (como um conjunto de arestas de pixel orientadas e conexas) entre os dois pontos dados é transformado no problema de encontrar o caminho de custo mínimo entre os dois vértices dados no grafo.

Seja G um **grafo direcionado** (Figura 3.3a), sendo V um conjunto finito de $|V|$ vértices (ou nós) e E um conjunto de $|E|$ arcos. Cada $v \in V$ pode ter até 4 vértices $u_k \in V$, $k = 1, 2, 3, 4$, distintos e vizinhos 4-adjacentes, formando o conjunto $A(v)$ dos vértices adjacentes a v em G (Figura 3.3b).

Um **arco** $e = (v, u)$ em E é um par ordenado de vértices adjacentes em G , iniciando em v , terminando em $u \in A(v)$ e tendo associado um **valor de comprimento** (ou custo) $l(v, u)$, ou $l(e)$, no intervalo $[0, 1]$. Há dois possíveis arcos entre dois vértices adjacentes, v e $u \in V$, onde $l(u, v)$ independe do valor de $l(v, u)$.

Um **caminho** $P = \{e_1, e_2, \dots, e_n\}$ em G é formado por uma sequência de arcos conexos $e_i = (v_{i-1}, v_i)$, $i = 1, 2, \dots, n$, onde o vértice final de um arco é o vértice inicial do próximo arco. P é dito um **caminho simples** em G quando não repete nenhum vértice, $v_i \neq v_j$, para todo $i \neq j$. A soma do comprimento dos arcos de P é chamado de **comprimento do caminho** P , i.e., $d(P) = l(e_1) + l(e_2) + \dots + l(e_n)$. $P = \langle v, v' \rangle$ é chamado de **caminho mais curto** de um vértice $v \in V$, para todo $v' \in V$ se, considerando todos os possíveis caminhos simples de v a v' , $d(P)$ possui o **comprimento mínimo**.

Na etapa da conectividade, uma cena $C = (C, g)$ é vista como um grafo $G = (V, E)$ onde os vértices dos pixels de C correspondem aos vértices de G e as arestas de pixels orientadas de C (*bels*) formam os arcos de G . O custo conjunto $c(b)$ associado a cada *bel*

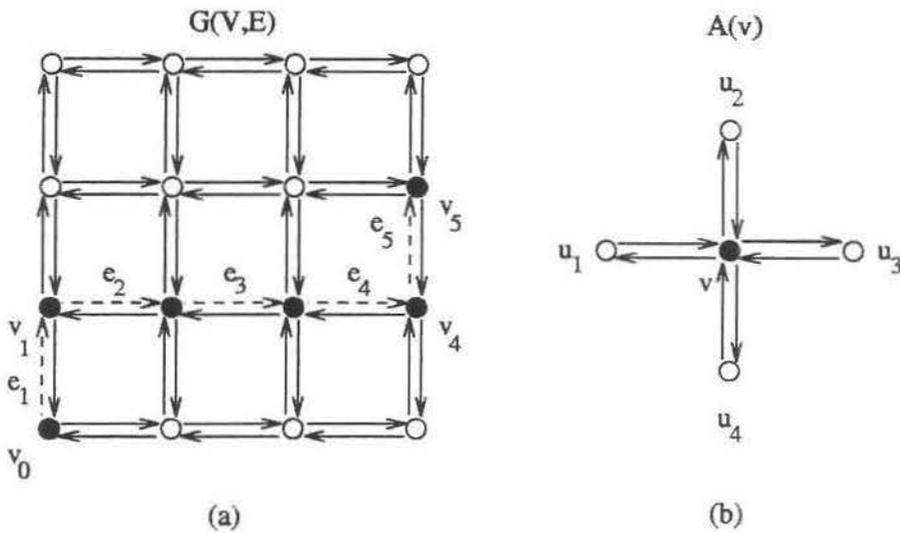


Figura 3.3: (a) Um grafo $G(V, E)$ definido para a cena C após a classificação. Um caminho $P = (e_1, e_2, \dots, e_5)$ em G entre os vértices v_0 e v_5 . (b) Cada vértice $v \in V$ define um conjunto $A(v) = u_1, u_2, u_3, u_4$ de no máximo 4 vértices vizinhos 4-adjacentes.

b de C na etapa de classificação equivale ao comprimento $l(e)$ de cada arco e de G .

Com esta nova visão dada a uma cena, encontrar o segmento ótimo (ou de custo mínimo) da borda de um objeto em C passa a ser tratado como encontrar o **caminho mais curto** no grafo G partindo de um v_i chegando a v_f , ambos vértices marcados sobre a borda. Este problema é referido na literatura como *the shortest path problem* [AMO93, Nil80, CLR91].

Nos métodos *live* a solução adotada utiliza a implementação do algoritmo de Dijkstra proposta por Dial [Fuk90]. Maiores detalhes sobre este algoritmo, estão apresentadas na seção 3.2.2.

3.2.2 Processo de segmentação no Live-Wire

Para segmentar um objeto em uma imagem usando o *live-wire*, o usuário seleciona um ponto inicial sobre a borda desejada usando o cursor (ponto v_0 na Figura 3.4). Para qualquer posição subsequente do cursor na imagem, dado que o usuário pode mover o cursor para qualquer lugar da imagem, uma curva - caminho de custo mínimo - conectando o ponto inicial ao ponto que corresponde a posição atual do cursor na imagem é calculada e apresentada na tela do computador em tempo real (daí o nome *live-wire* - caminhos de v_0 a v_1 na Figura 3.4). Quando o cursor é posicionado próximo a borda desejada, o segmento de *live-wire* adere à borda. Se este segmento descrever adequadamente a borda desejada, o usuário deposita o cursor cuja localização passa a ser a do novo ponto de partida (ponto v_1

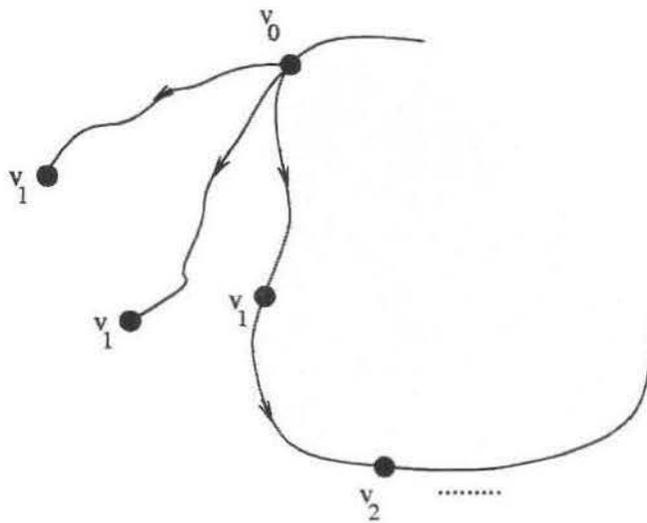


Figura 3.4: A figura ilustra o processo de segmentação usando *live-wire*.

na Figura 3.4). A borda completa na imagem (contorno fechado, conectado e orientado) é especificada via um conjunto de segmentos de live-wire desta maneira. Este processo é repetido para cada borda de objeto fatia por fatia. Normalmente de 2 a 5 pontos são necessários para completar uma borda em uma fatia.

Duas características do comportamento do método descrito são importantes de enfatizar:

- Os vértices pelos quais passam os segmentos de *live wire* previamente encontrados não são incluídos na determinação do segmento de *live-wire* atual. Isto evita que a curva que está sendo traçada cruze a si própria afetando a definição da borda como uma curva de Jordan.
- A tendência do algoritmo é escolher caminhos Euclidianos mais curtos onde existe deficiência de informação na borda. E isto pode ser visto como uma vantagem.

Optar pelo caminho mais curto pode ser uma solução interessante em casos onde ocorra deficiência de informação para se completar a borda. Entretanto, há casos em que o caminho mais curto não representa obrigatoriamente a borda desejada. Nestes casos, o usuário pode intervir sem que haja qualquer comprometimento na qualidade do resultado final.

Outra importante contribuição do método para a segmentação de imagens é a característica de orientação. Na Figura 3.5 observa-se como ter uma borda sensível à orientação pode favorecer uma melhor delineação do objeto. Ela apresenta uma imagem do pé obtida por MRI, onde a borda de interesse é o do talus. A Figura 3.5(a) mostra um segmento entre dois vértices sem levar em conta a orientação. Observe que o segmento do *live-wire*

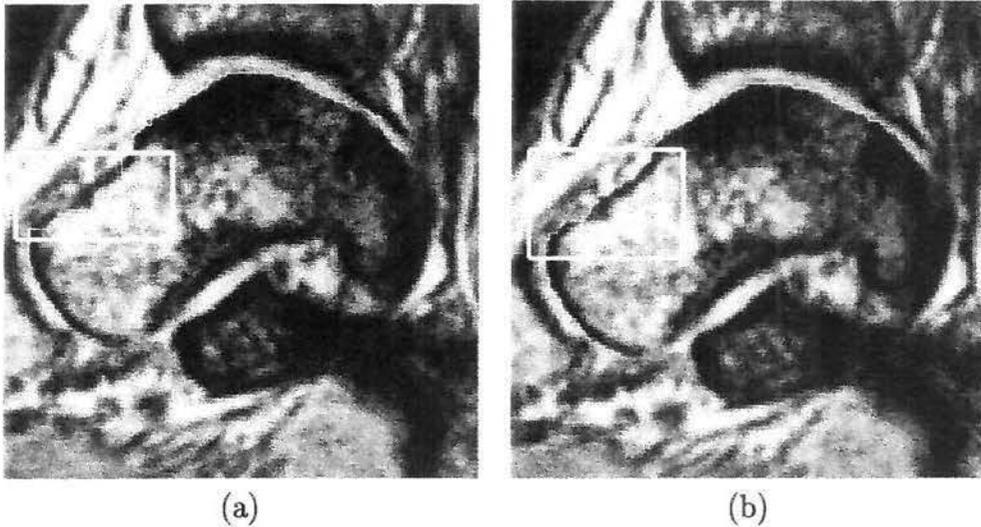


Figura 3.5: A figura ilustra a importância das características sensíveis à orientação nos métodos *live*. (a) A orientação da borda é assumida no sentido anti-horário. Um segmento de *live-wire* entre dois vértices é obtido usando características sensíveis à orientação. A figura mostra uma região de interesse onde o segmento de contraste similar não atrai o *live-wire* visto que sua orientação não é favorável. (b) Um segmento de *live-wire* partindo do mesmo vértice inicial de (a), mas com orientação no sentido contrário, ilustra a sensibilidade do método à orientação previamente adotada para a borda.

em alguns pontos (como na região identificada pelo retângulo) é atraído pelo segmento de contraste similar ao da borda desejada. O mesmo não ocorre na Figura 3.5b quando é usada orientação. Isto porque a orientação deste segmento e da borda são opostas e esta informação foi armazenada na geração de custos durante a fase de classificação. Com as características sensíveis à orientação, o usuário facilmente percebe que o segmento do *live-wire* favorece apenas um sentido (horário ou anti-horário).

O algoritmo de Caminhos Mínimos

A implementação proposta por Dial [Fuk90] para o algoritmo de Dijkstra calcula o caminho de custo mínimo para todos os nós partindo de um único vértice, mas como o interesse a cada passo é encontrar apenas o caminho entre v_i e v_f , o algoritmo utilizado pelo *live-wire* [FUM99] traz esta modificação.

Entrada: um vértice inicial v_i ; um vértice final v_f ; uma função de custo c ;

Saída: um caminho ótimo entre v_i e v_f .

Estruturas Auxiliares: uma fila circular Q ; uma matriz de custo acumulativo cc ; uma matriz de direção dir ; uma lista L de vértices já processados.

início

1. inicialize $cc(v)$ com ∞ e $dir(v)$ com $null$ para todos os vértices $v \in V$ e $L \leftarrow vazio$;
 2. inicialize $cc(v_i)$ com 0 e coloque v_i em Q ;
 3. enquanto $v_f \notin L$ faça
 - a. retire um vértice v de Q tal que $cc(v) = \min_{v' \in Q} cc(v')$, e coloque v em L ;
 - b. para cada vértice v' tal que v' está no subconjunto dos vizinhos 4-adjacentes de v e $v' \notin L$ faça
 - (i) calcule $cc_{tmp} = cc(v) + c(e')$ onde e' é o arco que parte de v' para v e $c(e')$ é o custo de e' , calculado na etapa de classificação;
 - (ii) se $cc_{tmp} < cc(v')$ então
 - a. atualize $cc(v')$ com cc_{tmp} e $dir(v')$ para a direção de v' para v ;
 - b. se $v' \notin Q$ então insira v' em Q ;
senão atualize v' em Q ;
4. começando em v_f trace recursivamente o caminho ótimo seguindo a direção dos arcos em dir até que v_i seja encontrado;

fim

Observe que, pelo algoritmo acima, a fila Q está sempre ordenada em ordem crescente de $cc(v)$. A Figura 3.6 ilustra um exemplo do funcionamento do algoritmo, sendo a imagem representada pelo grafo da Figura 3.6a. As Figuras 3.6b e 3.6f mostram os vetores de custo acumulados cc e direções dir combinados em uma só representação. O próximo vértice a sair da fila é sempre ilustrado pelo círculo parcialmente escuro. A situação antes da primeira iteração é mostrado na Figura 3.6b. As Figuras 3.6c e 3.6d apresentam a situação após a primeira e segunda iteração respectivamente. A Figura 3.6e

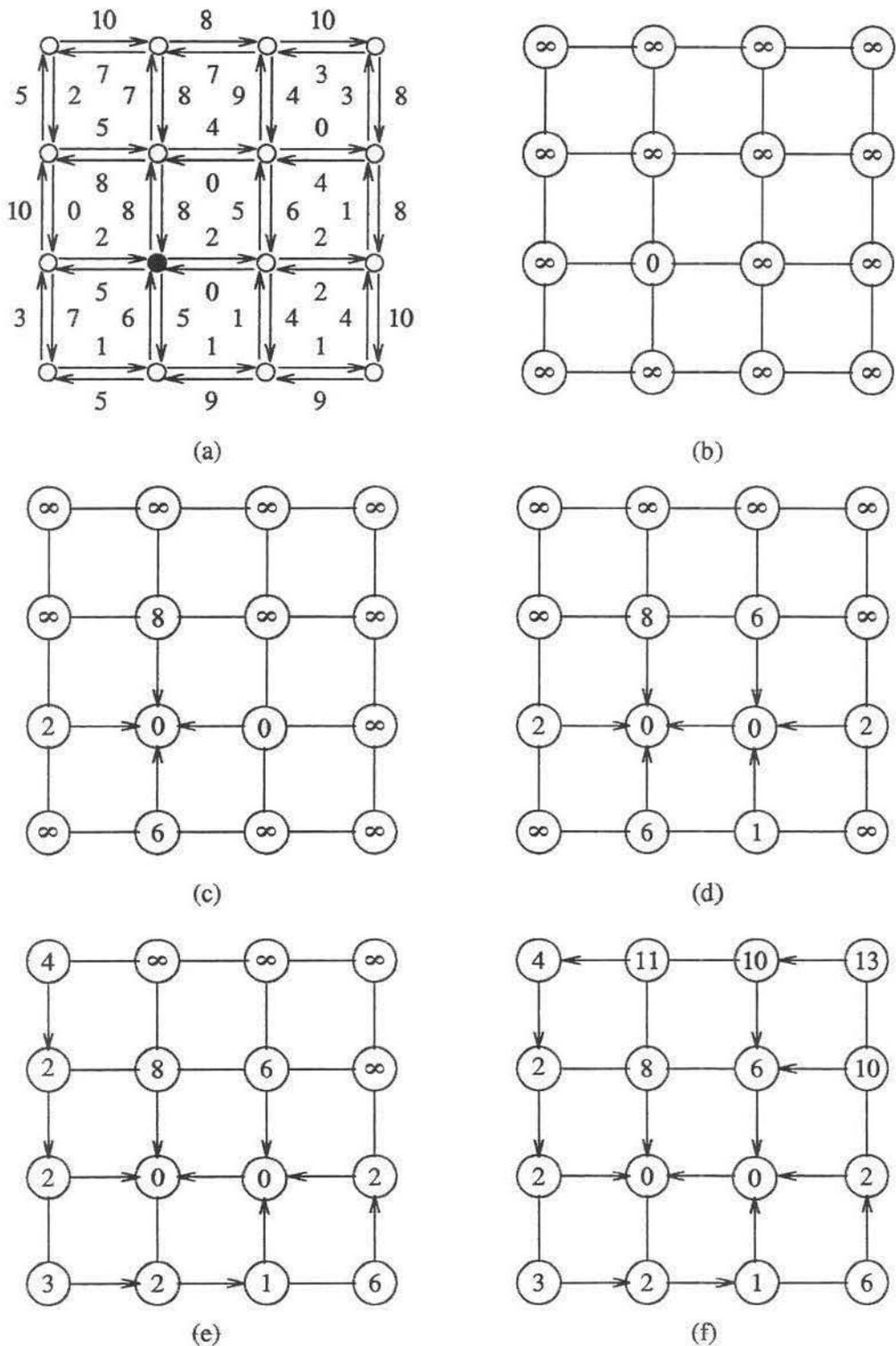


Figura 3.6: A figura ilustra o algoritmo *DPS*. (a) Um grafo de entrada representando uma cena *C* após classificação, onde valores de custo conjunto de $[0,10]$ são associados aos *bels* de *C* e o vértice inicial é representado pelo círculo escuro. As figuras de (b) a (f) ilustram respectivamente a situação do grafo de saída antes da primeira iteração, após a primeira iteração, após a segunda iteração, após a sétima iteração e após quinze iterações. Nestas figuras, o próximo vértice a sair da fila é sempre representado pelo círculo parcialmente escuro.

reflete a situação após sete iterações e a Figura 3.6f exibe o resultado final após quinze iterações. É interessante observar que se v_f for previamente determinado como sendo, por exemplo, o vértice do canto superior esquerdo do grafo da Figura 3.6a, o caminho ótimo seria encontrado logo após a sétima iteração (Figura 3.6e) e o algoritmo terminaria o processamento antes de visitar todos os vértices do grafo.

3.2.3 Extensões do modelo

O modelo *live-wire* não é a única alternativa para segmentação usando a mesma abordagem de grafo apresentada pelo método *live*. Duas outras extensões a este modelo foram definidas por Falcão [Fal96], descritas resumidamente a seguir.

Live-Lane

Ao utilizar o método *live-lane* o usuário tem um maior envolvimento com as ações do computador. Como no *live-wire*, o usuário seleciona um ponto inicial com o cursor do mouse; para que o traçado seja feito o usuário deve mover o cursor próximo à borda do objeto. À medida que o cursor é deslocado, segmentos de *live-wire* entre sucessivos pontos são calculados e apresentados na tela em tempo real, dentro de uma certa faixa de largura (daí o nome *live-lane*). Esta largura da faixa pode variar ao longo da borda, sendo menor em regiões onde a borda é mal definida e maior onde existe forte evidência da borda do objeto. Sendo a largura da faixa maior, o método tem um comportamento próximo ao do *live-wire*; se a faixa tiver largura muito estreita, o método quase se torna manual. Critérios para definição e variação da largura da faixa, podem ser vistos em [Fal96].

O método *live-lane* diferencia do *live-wire* basicamente em três aspectos:

- O mecanismo de seleção dos vértices v_i de V ;
- A extensão do grafo utilizado para encontrar os caminhos ótimos entre dois vértices subsequentes de V ;
- O processo de interação do usuário com a máquina.

3D-Live-Wire

No método *3D-live-wire*, a segmentação é feita para cada cena 3D (ou subcena 3D de uma cena 4D) isoladamente. Dada uma cena 3D, o usuário deve inicialmente selecionar cortes ortogonais ao plano das fatias e traçar a borda do objeto nas imagens deste corte usando o *live-wire*. (Nada impede que se utilize o *live lane* ao invés do *live-wire*). Em seguida, para todas as fatias C , ortogonais, da cena, os vértices v_i são selecionados e ordenados

automaticamente sobre a borda do objeto, sendo em seguida delineada automaticamente fatia por fatia.

Se há mudança topológica ao longo do eixo z , esta informação é obtida dividindo-se a cena 3D em intervalos I de fatias consecutivas, nos quais a topologia não apresenta variações. Estes intervalos são chamados *slabs* de topologia constante.

Uma vez separada uma cena 3D em *slabs* I , resolve-se a segmentação de cada estrutura S do objeto de interesse separadamente. Observe que a intersecção de qualquer plano de corte em xyz define uma subcena 2D. Sendo assim, poderia-se questionar a utilidade do método *3D live-wire*, se é necessário o uso do *live-wire* em cada plano de corte selecionado pelo usuário. A resposta disto está no fato de que o tempo do usuário é reduzido, uma vez que não precisará realizar a segmentação fatia por fatia, mas sim em um número k de cortes, em geral muito inferior ao número de fatias de I .

O processo de segmentação utilizando o *3D live-wire* pode ser basicamente dividido em cinco etapas:

- Definição e seleção de *slabs*
- Seleção de estruturas
- Seleção de cortes
- Ordenação dos vértices
- Detecção de borda.

A Figura 3.7 mostra um exemplo do uso deste método. Observe que os pontos de um contorno representando uma borda em uma imagem de corte ortogonal ao plano das fatias, geram pelo menos dois vértices v_0 e v_1 sobre as bordas de cada fatia de I . Nestas fatias, segmentos de *live-wire* $\langle v_0, v_1 \rangle$ e $\langle v_1, v_0 \rangle$, definem um contorno fechado, conectado e orientado, encontrado automaticamente fatia por fatia.

Certamente, a razão entre o número de cortes necessários e o número de fatias pode comprometer o desempenho do *3D live wire* com relação a outras abordagens 2D. Mas, em casos onde poucos cortes ortogonais (máximo de seis) sejam necessários, o *3D live-wire* reduz consideravelmente o tempo de envolvimento do usuário no processo.

3.3 Conclusão

Neste capítulo apresentou-se basicamente um paradigma de segmentação interativa: os métodos *live*. A estratégia básica utilizada é transformar o problema de detecção de segmentos de borda como o problema de encontrar o caminho de custo mínimo dados

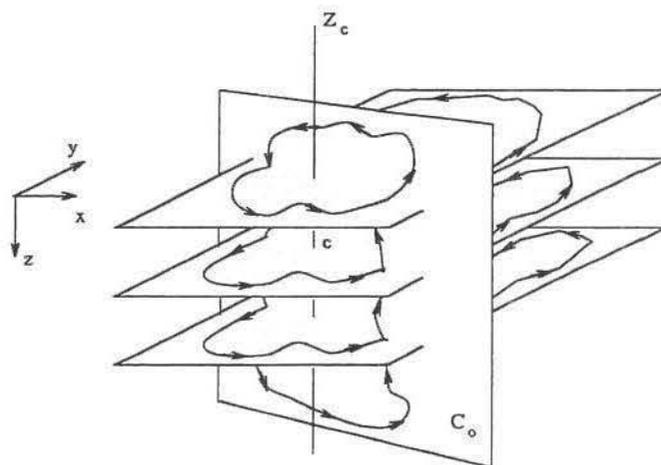


Figura 3.7: A figura mostra um corte ortogonal ao plano das fatias de uma cena 3D. O contorno que representa a borda do objeto na cena 2D resultante do corte ortogonal gera pontos sobre a borda do objeto nas fatias da cena 3D.

dois vértices de um grafo. Para tanto, a imagem é tratada como um grafo, os vértices dos pixels são os nós do grafo e as arestas dos pixels, os arcos.

O uso de algoritmos em grafos para encontrar bordas tem sido estudado por muitos anos [Mar71, Mon71, PPGC84, FHW81], quase sempre sem garantias de sucesso e impondo restrições ao contorno. Entretanto os métodos *live* além de partirem deste princípio também demonstram ser possível resolver o problema em tempo real usando um modelo de grafo que (i) não impõe qualquer restrição na forma ou tamanho dos objetos a serem segmentados; (ii) considera todas as possíveis bordas em toda a imagem e independe de pontos iniciais selecionados; (iii) considera os elementos da borda como uma curva orientada de Jordan de tal maneira que todas as bordas têm claramente um lado interno e um externo.

Certamente, como qualquer método de segmentação até hoje existente, vantagens e desvantagens existem no uso dos métodos acima expostos [Fal96]. A principal desvantagem dos métodos *live-wire* e *live-lane* está no fato de requerem a segmentação fatia por fatia. A principal vantagem, em particular do *live-wire* está na garantia de uma segmentação precisa, mesmo na presença de ruídos na imagem. O uso do *3D-live-wire* seria melhor justificado na segmentação de objetos em que se observa de antemão a necessidade de um número pequeno de cortes, número este menor que o de fatias contendo o objeto, diminuindo o tempo de interação do usuário no processo. Por outro lado este método requer uma certa habilidade com o mesmo, a fim de que os cortes selecionados sejam efetivos.

Uma grande vantagem no método *live-wire* é seu tempo de resposta. Quando o usuário seleciona um ponto sobre a borda e move o cursor, o segmento de *live-wire* é calculado e

apresentado em tempo real, partindo do ponto selecionado para qualquer lugar na imagem, seja ela pequena ou grande, mas os cálculos outrora feitos para um ponto previamente selecionado são aproveitados, não precisando iniciá-los do zero. Para um método interativo, tempo de resposta é um fator fundamental para garantir maior usabilidade do mesmo na prática.

Capítulo 4

Contornos Deformáveis usando Segmentos de Custo Mínimo

Segmentação de imagens, como pode ser observado nos capítulos anteriores, é um tema bastante investigado. De acordo com a aplicação desejada uma técnica pode ser mais apropriada que outra.

Quando se fala em segmentação de imagens, é necessário enfatizar que esta tarefa se divide em duas etapas fundamentais: Reconhecimento e Delineação. A primeira, compreende a etapa de localizar o objeto de interesse e sua posição na imagem, em meio a tantos outros objetos e artefatos ali existentes. Esta tarefa tem sido feita com muito mais êxito pelo homem do que pelo computador. A segunda etapa, refere-se a tarefa de delinear ou contornar a borda do objeto. A delineação tem sido melhor desempenhada pelo computador, isto porque a precisão, exatidão e capacidade de reproduzir o mesmo resultado exige do homem algumas vezes a percepção de detalhes que o olho humano não consegue ver, ou que a mão não consegue contornar com precisão sem que o cansaço (entre outros fatores, como falta de habilidade ou coordenação motora) afetem no resultado, reduzindo seu desempenho quando comparado ao da máquina.

Os métodos de Contornos Deformáveis de maneira geral apresentam um bom desempenho na tarefa de reconhecimento, pois o contorno chega ao objeto desejado (quando o problema de convergência consegue ser superado). Entretanto, deixam a desejar na tarefa de delineação devido a preocupações como manter a suavidade do contorno, dificuldades de mudanças topológicas e de segmentar indentações e protusões do objeto (entre outras dificuldades apresentadas no Capítulo 2), o resultado final acaba não sendo tão satisfatório, a menos que o usuário interfira.

Por outro lado, quando se trabalha com métodos interativos, como o *Live-Wire*, o usuário tem uma participação indispensável na tarefa de reconhecimento e, no caso deste método, o delineamento é bem superior ao de contornos deformáveis. Entretanto, o

resultado de todo o processo pode ser bem demorado, dependendo do número de fatias a serem segmentadas e da própria habilidade com a técnica em questão. O ideal seria um método de segmentação totalmente automático garantindo o resultado desejado.

Este capítulo apresenta o método de **Contornos Deformáveis usando Segmentos de Custo Mínimo** (CDSCM). Este método de segmentação é praticamente automático na tarefa de reconhecimento e totalmente automático na delimitação da borda. A participação do usuário só acontece ao fornecer alguns pontos iniciais em torno do objeto de interesse na primeira fatia de uma dada sequência e em eventuais correções.

Como na maioria dos métodos de Contornos Deformáveis, o método apresentado utiliza um funcional baseado na atuação de Forças Internas e Externas para realizar a deformação do contorno. A novidade está no uso de segmentos de custo mínimo com orientação de borda, o que permite ao usuário especificar as características da borda que deseja delimitar, tentando evitar que outras bordas próximas ao objeto, porém com orientação diferente, sejam também atratoras. Este conceito foi incorporado dos métodos *Live* (Capítulo 3).

O uso de segmentos de custo mínimo permite uma melhor convergência na finalização do contorno em cada fatia, o que é um problema típico de contornos deformáveis, tais como *snakes*. Para tanto, utiliza-se o algoritmo de segmentos de custo mínimo entre cada par de snaxels (como no método *Live-Wire*). Com isto, procura-se não apenas gerar um segmento de reta ligando cada dois snaxels consecutivos que estejam sobre a borda, mas fazer com que todo o contorno de fato seja aderido à borda do objeto.

O capítulo é dividido nas seguintes seções: A primeira é destinada a detalhar o método de Contornos Deformáveis usando Segmentos de Custo Mínimo. Nesta parte, apresenta-se o método de contornos deformáveis utilizado, explica-se como a orientação da borda influencia na segmentação, quais as condições de parada para o contorno e como o algoritmo de custo mínimo é executado no momento de parada do contorno a fim de proporcionar uma melhor delimitação da borda. Na segunda parte, fala-se de trabalhos relacionados ao CDSCM. Por fim, apresenta-se a conclusão do capítulo.

4.1 O modelo de Contornos Deformáveis usando Segmentos de Custo Mínimo

4.1.1 Deformação do Contorno

O método apresentado neste trabalho é dividido em duas etapas. A primeira utiliza o princípio comum aos métodos de contornos deformáveis: o uso de duas Forças, uma Interna e outra Externa, que atuam sobre o contorno, levando-o a deformar-se em uma certa direção, até atingir estabilização e parar, de preferência, sobre a borda desejada. A segunda parte utiliza os pontos de mínimo de Energia do contorno e traça segmentos

de custo mínimo entre eles, fazendo o contorno final aderir e representar com precisão a borda desejada. A Força Interna adotada é baseada no método de Dual T-Snakes [GGA00], apresentado no Capítulo 2.

Dada uma cena 3D e um conjunto $V = v_1, v_2, \dots, v_n$ pontos, formando um contorno fechado ($v_1 = v_n$), o objetivo desta etapa é obter, em cada fatia (cena 2D) um contorno final através de sucessivas deformações do contorno inicial, tal que todos os pontos v_i , $1 \leq i \leq n$ estejam sobre a borda do objeto de interesse. Para tanto considera-se cada ponto do contorno um snaxel sobre o qual atuam duas forças: uma denominada **Força Interna** (F_{int}), baseada em características do próprio contorno, e outra denominada **Força Externa** (F_{ext}) que atua baseada em características da Imagem. Da ação destas duas forças surge uma **Força Resultante (FR)** que provoca a deformação (deslocamento) do contorno. De maneira mais formal, tem-se: Seja $V = \{v_1, v_2, \dots, v_n\}$, $v_1 = v_n$, $v_i = (x_i, y_i)$ um snaxel, tem-se:

$$\mathbf{FR}(v_i, t) = \mathbf{F}_{int}(v_i, t) + \mathbf{F}_{ext}(v_i, t) \quad (4.1)$$

$i = 1, 2, \dots, n$, $t = 1, 2, \dots, k$, onde k é o número máximo de iterações até a estabilização de todos os pontos. Isto implica que $FR(v_i, k) = 0$, $i = 1, 2, \dots, n$ (muito embora alguns pontos parem para valores $1 \leq k' \leq k$).

Força Interna

No método CDSCM não há uma preocupação central na minimização de Energia, como no método original da *snakes*. Isto porque o enfoque é voltado à neutralização das Forças Interna e Externa de maneira que a combinação de ambas permita a cada snaxel parar ou prosseguir seu deslocamento, de acordo com a característica do meio onde está inserido. O contorno pára quando $FR(v_i) = 0$ para todo i .

A Força Interna utilizada apresenta-se como uma **Força elástica**, onde entre cada par de snaxels vizinhos (v_i e v_{i+1}) existe uma mola com a tendência de aproximar os snaxels.

A Força Elástica (F_{el}) apresenta a seguinte formulação:

$$\mathbf{F}_{int}(v_i, t) = \alpha \mathbf{F}_{el}(v_i, t) \quad (4.2)$$

$$\mathbf{F}_{el}(v_i, t) = \mathbf{F}_{elA} + \mathbf{F}_{elB} \quad (4.3)$$

onde

$$\mathbf{F}_{elA} = v_i - v_{i-1} \quad (4.4)$$

e

$$F_{el_B} = v_i - v_{i+1} \quad (4.5)$$

Para cada snaxel v_i é calculada a distância entre ele e os snaxels adjacentes a fim de evitar que eles fiquem posicionados ou próximos ou distantes demais. Isto é assegurado através de um parâmetro de distância máxima (d_{max}) e mínima (d_{min}) que deve ser respeitado sob a seguinte situação:

$$d_{max} \leq \text{dist}(v_i, v_j) \leq d_{min} \quad (4.6)$$

onde v_j equivale a ambos snaxels adjacentes ao v_i (v_{i+1} e v_{i-1}). d_{max} e d_{min} são parâmetros fornecidos pelo usuário.

Supondo o sentido anti-horário para o contorno, a Figura 4.1 ilustra a atuação destas forças na geração da F_{int} .

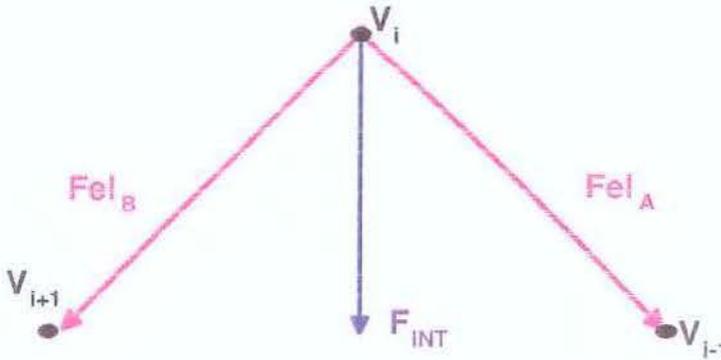


Figura 4.1: Ilustração da formação da Força Interna. Calcula-se a Força Elástica $F_{el}(A)$ (entre v_i e v_{i-1}) e $F_{el}(B)$ (entre v_i e v_{i+1}). A soma de ambas resulta na Força Interna no snaxel v_i .

Força Externa

Assim como a Força Interna, a Força Externa é calculada para cada snaxel, a cada iteração. Entretanto, a Força Externa depende da Energia da Imagem, a qual se baseia em características como o gradiente e é calculada para cada pixel da imagem.

A Energia da Imagem é calculada para cada pixel baseada na similaridade com seus 4-vizinhos q_i , $i = 1, 2, 3, 4$ (Figura 4.2).

$$E_{\text{image}}(p) = \min\{\mu(p, q_i)\} \quad (4.7)$$

onde

$$\mu(p, q_i) = M * \exp\left(-\frac{|f(p) - f(q_i)|^2}{\sigma^2}\right) \quad (4.8)$$

sendo $\mu(p, q_i)$ a afinidade de p com q_i , M o maior valor de afinidade para um pixel na imagem, $f(p)$ é a imagem filtrada por um filtro gaussiano e σ é um desvio padrão.

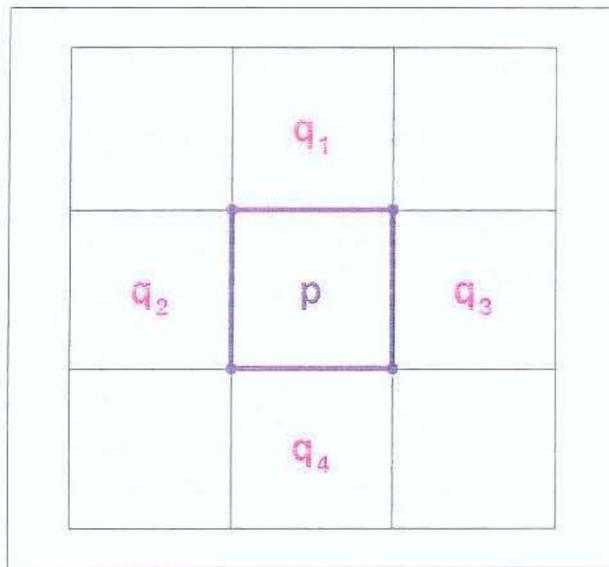


Figura 4.2: Ilustração de um pixel i e seus 4-vizinhos (pixeis q_1 , q_2 , q_3 e q_4)

Para obter-se $E_{\text{image}}(v_i)$, onde v_i é um snaxel, utiliza-se o valor de $E_{\text{image}}(p)$ onde p é um pixel que contém v_i .

Uma vez calculada a Energia da Imagem para todos os pixels que a compõem, é feita uma limiarização na Energia da Imagem, a fim de eliminar informações desnecessárias e preservar apenas as características de borda que interessem (fracas ou fortes).

Desta forma, dado um limiar máximo e um mínimo (valores que podem ser alterados pelo usuário), se $\text{limiar} - \text{minimo} \leq E_{\text{image}}(v_i) \leq \text{limiar} - \text{maximo}$, então $E_{\text{image}}(v_i) = 1$, senão $E_{\text{image}}(v_i) = 0$.

Após a limiarização é feito o cálculo da Força Externa em cada snaxel. Conforme visto no Capítulo 2, (equações 2.3 e 2.4), a Energia Externa é a mesma Energia da Imagem. Sendo assim, dado um snaxel $v_i = (x_i, y_i)$, o cálculo da $F_{\text{ext}}(i)$ é obtido calculando o Gradiente sobre a $E_{\text{ext}}(v(i))$.

$$\mathbf{F}_{\text{ext}}(\mathbf{v}_i) = \nabla(\mathbf{E}_{\text{ext}}(\mathbf{v}_i)) \quad (4.9)$$

Força Resultante

De maneira geral, pode-se dizer que em cada snaxel a Força Resultante (FR) é obtida adicionando F_{ext} a F_{int} , cada uma possuindo um peso específico. Além disso, cada uma das forças deve ser normalizada antes do cálculo da FR.

A fórmula geral para o cálculo da Força Resultante é:

$$\mathbf{FR}_i(\mathbf{v}_i) = \alpha * \mathbf{F}_{\text{int}}(\mathbf{v}_i) + \beta * \mathbf{F}_{\text{ext}}(\mathbf{v}_i) \quad (4.10)$$

É interessante observar que numa imagem podem haver regiões homogêneas onde a Força Externa é praticamente nula (pois sendo seu valor dependente do gradiente da imagem, ele será mais alto em regiões onde há contraste, indicando possíveis bordas). Nesse caso, se o contorno não estiver posicionado em um ponto de mínimo, apenas a Força Interna atua para o deslocamento do contorno, a qual recebe um peso maior do que quando atua em conjunto com a Força Externa. Este comportamento é similar ao da Força Balão [CC93], que visa exatamente levar a curva para regiões de bordas.

Uma vez calculada a Força Resultante num dado snaxel, é hora de fazê-lo deslocar-se. Para tanto, as novas coordenadas de cada snaxel i são obtidas fazendo-o mover-se na direção indicada pela \mathbf{FR}_i . A equação que expressa este movimento é:

$$\begin{aligned} x'_i &= x_i + \mathbf{FR}_i(v_i, t) \\ y'_i &= y_i + \mathbf{FR}_i(v_i, t) \end{aligned} \quad (4.11)$$

A condição de parada de deslocamento para cada snaxel será descrita na seção 4.1.3.

4.1.2 A orientação da borda

Uma borda orientada é aquela que quando caminhamos sobre ela no sentido de sua orientação, o lado esquerdo estará sempre no interior (exterior) e o lado direito no exterior (interior).

Uma forma de extrair esta informação da imagem é assumir que o lado esquerdo é sempre mais escuro (claro) que o lado direito. Isto é implementado com o sinal do vetor gradiente.

Um mesmo objeto pode ser segmentado de duas formas: por dentro ou por fora. O primeiro caso será chamado segmentação da borda interior, e o segundo de segmentação da borda exterior.

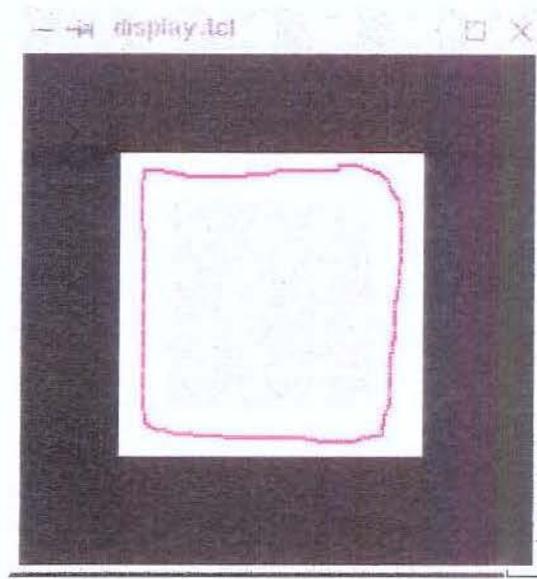


Figura 4.3: Imagem de Contorno Inicial para segmentação do Quadrado. O mesmo pode expandir a fim de segmentar o quadrado mais externo ou contrair, para segmentação do quadrado mais interno.

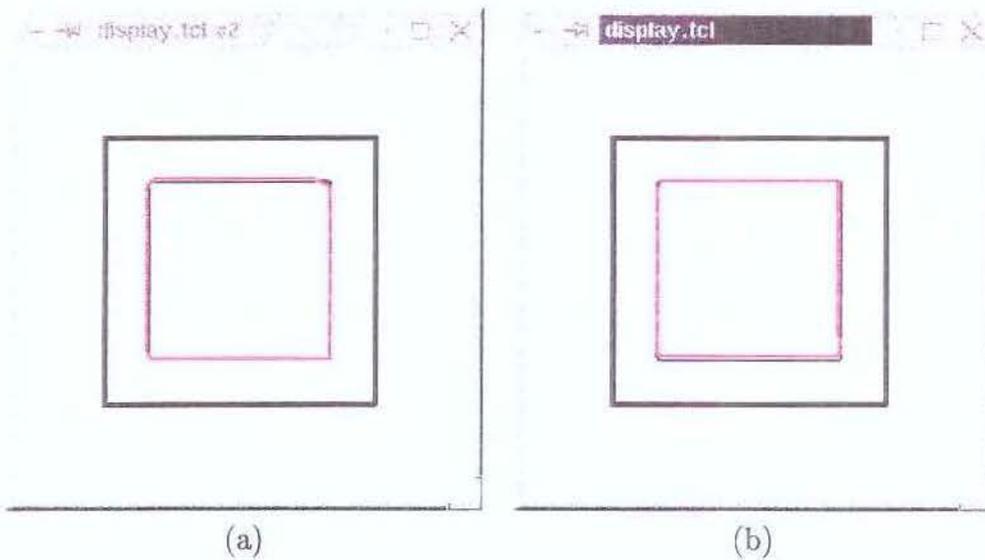


Figura 4.4: O objeto de interesse é o quadrado mais interno. (a) A segmentação foi feita pelo lado de fora do objeto. (b) A segmentação foi feita pelo lado de dentro do objeto.

Um exemplo do uso de orientação da borda pode ser visto nas Figuras 4.4(a) e 4.4(b). Observe que, a partir de um contorno inicial intermediário 4.3, entre o quadrado mais externo e o mais interno, se antes da segmentação for determinado que o contorno está externo ao objeto de interesse, pretende-se segmentar partindo do meio mais escuro para o mais claro, o objeto (quadrado mais interno) será segmentado pelo seu lado de fora. Observe o resultado da segmentação deste objeto na Figura 4.4(a).

Agora observe o contrário. Com o mesmo contorno inicial (Figura 4.3), mas informando que a borda desejada parte do meio mais escuro para o mais claro, isto fará com que o quadrado menor seja segmentado apenas pelo seu interior. O resultado desta segmentação está na Figura 4.4(b).

Na Figura 4.4, como a borda é muito fina, a diferença entre a segmentação por dentro ou fora do objeto é quase imperceptível (na imagem do gradiente a diferença é mais facilmente observada na parte inferior do objeto). Entretanto, em imagens médicas, a escolha de um dos lados pode beneficiar o resultado. Resultados em imagens tomográficas serão exibidos no Capítulo a seguir.

Não é sempre fácil segmentar um objeto tanto por sua borda Externa ou Interna. Isto porque o deslocamento de cada snaxel a cada iteração pode fazer com que ele ultrapasse a borda de interesse e não mais a encontre. Como saber qual o deslocamento necessário? Qual deve ser o máximo ou mínimo? Se o deslocamento for muito pequeno, o número de iterações necessárias pode crescer muito, pois o movimento do contorno torna-se muito lento. Por outro lado, se for muito grande, pode-se ultrapassar a borda. Poderia-se propor o uso de velocidade diferenciada, fazendo com que o contorno se movesse mais lentamente em regiões próximas à borda, e mais rapidamente em regiões homogêneas, mais ainda assim, teria-se o problema de definição do parâmetro da velocidade.

De qualquer modo, a escolha da orientação permite uma segmentação de melhor qualidade (ou o objeto é segmentado por fora ou por dentro. É mais difícil gerar um contorno final com pontos tanto por dentro quanto por fora usando a orientação). Outra vantagem para o uso de borda orientada é que esta permite evitar que outros objetos com as mesmas características de contraste, mas orientações opostas, sirvam como atratores na segmentação.

4.1.3 Condições de parada do contorno

Não se espera que todos os snaxels parem de mover-se ao mesmo tempo. Cada snaxel está envolvido num meio e numa posição com características tais que podem levar o mesmo a mover-se mais ou menos vezes. Quando um certo snaxel está sobre uma posição que satisfaz as condições de parada o mesmo deve ser mantido ali, mesmo que os demais snaxels continuem se deslocando.

Quando se fala em *condição de estabilidade* para um snaxel, no método apresentado, não significa somente que o mesmo está posicionado sobre um ponto de mínimo, mas além disso que a orientação da borda neste ponto satisfaz a condição definida pelo usuário na chamada do programa. Enquanto ambas as condições não forem satisfeitas, o snaxel deve continuar se deslocando.

Um dos problemas que se pode ter ocorre quando um snaxel está posicionado sobre a borda do objeto, mas a orientação não é a desejada. Nestes casos, os experimentos mostraram que a melhor abordagem a ser adotada é reduzir os parâmetros das forças a fim de provocar um deslocamento pequeno neste snaxel, pois a borda com orientação desejada pode não estar tão distante daquela que acabou de ser encontrada.

Um exemplo do problema mencionado pode ser visto quando o contorno está num movimento de contração, e o objeto deve ser segmentado pelo seu lado de dentro. Se o snaxel parar sobre a borda procurada, porém pelo lado de fora do objeto, o deslocamento deste snaxel deve ser pequeno a fim de simplesmente conduzi-lo para o lado interno da borda. Se o deslocamento for muito grande, o snaxel pode ser conduzido para dentro do objeto mas para fora da borda, o que o levaria cada vez mais para longe da borda desejada.

As condições de parada para o contorno são:

- (a) Todos os snaxels chegam à condição de estabilidade. Quando todos os snaxels do contorno chegarem à **condição de estabilidade**, o programa sai da função de deslocamento do contorno.
- (b) Uma porcentagem “n” de pontos chegam à condição de estabilidade. Pode acontecer de um único snaxel não encontrar a borda e continuar se deslocando, talvez até para bem longe do objeto. Pode-se evitar que isto aconteça se for definido, por exemplo, que quando noventa por cento dos snaxels já estiverem chegado à **condição de estabilidade**, todo o contorno pode parar de deslocar-se.
- (c) Foram feitas “n” iterações sobre o contorno. Quando passado um número “n” de iterações sobre o contorno (sendo “n” um parâmetro), o mesmo ainda não parou seu deslocamento, o programa pode sair desta função e informar ao usuário que a borda não foi encontrada após muitas tentativas. Isto pode ocorrer devido a problemas como má inicialização do contorno ou mesmo se os parâmetros utilizados nas forças forem tão altos (ou baixos) que os snaxels estejam se deslocando em passos grandes demais (ou muito pequenos) ultrapassando a borda (ou nem mesmo chegando até ela, mesmo após muitas iterações). Isto requer que os parâmetros sejam reavaliados, a fim de verificar se é necessário reduzi-los ou ampliá-los, o que é algo difícil de definir na prática por estar relacionado a cada objeto a ser segmentado e como tipo e qualidade da imagem com que se trabalha.

4.1.4 Segmentos de Custo Mínimo

Após a parada do contorno, supõe-se, no melhor caso, que a borda do objeto tenha sido encontrada. Sendo assim, cabe agora procurar proporcionar uma melhor delimitação do objeto, pois o que se tem até agora são alguns pontos sobre a borda, mas para gerar um contorno que de fato represente o objeto é necessário interligar estes pontos com segmentos que de fato delimitem o objeto, e não simplesmente com um segmento de reta ou interpolando com pontos entre os snaxels já parados sobre a borda. No método utilizado neste trabalho, esta delimitação é concluída com o uso do algoritmo de caminhos mínimos usado em [FUM99], já apresentado no Capítulo sobre os métodos *live*.

A primeira etapa deste processo consiste em passar como parâmetro ao algoritmo somente os snaxels que chegaram à condição de estabilidade, localizando-se sobre um ponto de mínimo. Deve-se tomar este cuidado pois não obrigatoriamente todos os snaxels o estão.

Em seguida, o algoritmo é executado procurando o caminho de custo mínimo entre cada par de snaxels consecutivos. Como o algoritmo de segmentos de custo mínimo trabalha com arestas de custo, a estratégia adotada para calcular estes custos é o seguinte:

$$C(p, q) = M - M * \exp\left(\frac{-(|f(p) - f(q)| - m)^2}{\sigma^2}\right) \quad (4.12)$$

onde q é um dos 4-vizinhos de p ; M é o nível de contraste da borda desejada; e $C(p, q)$ é o custo da aresta que divide p e q em um grafo não-direcionado. Para trabalhar com orientação da borda, basta retirar o módulo da $|f(p) - f(q)|$. Custos baixos só aparecerão para arestas com contrastes próximo ao valor de m .

Neste instante não existe mais a preocupação em manter uma distância mínima ou máxima entre os snaxels. O algoritmo acaba gerando um novo conjunto de snaxels, de maneira que ao final a distância entre os que formam o contorno seja igual a um (1).

Como o contorno final de uma fatia é aproveitado como contorno inicial na fatia seguinte, a função que verifica distâncias entre os snaxels é executada ao início da execução do programa para cada fatia. Isto é necessário porque experimentos mostraram que a essa distância (igual a um), tanto o desempenho do algoritmo de contornos deformáveis fica comprometido como a qualidade final do contorno que este gera.

Segmentação da próxima fatia

Antes que um contorno gerado para um fatia seja aproveitado na seguinte, o mesmo sofre uma certa expansão ou uma contração.

O contorno será expandido caso, na primeira fatia, o mesmo tenha sido inicializado fora do objeto de interesse. E o contorno será contraído caso, na primeira fatia, o mesmo

tenha sido inicializado dentro do objeto de interesse.

Esta alteração no contorno se deve ao fato de que, a cada fatia, o objeto também sofre alterações. Se o contorno for aproveitado tal qual foi gerado na fatia anterior, pode ocorrer de, na fatia atual, ele estar posicionado dentro do objeto de interesse (pois o objeto em geral apresenta alterações no formato a cada fatia), quando para o programa a informação recebida inicialmente era que o contorno estaria sempre fora do objeto. Tal situação levaria o contorno a se deslocar ainda mais para o interior do objeto, e não encontraria a borda desejada. A fim de evitar este problema, o contorno gerado é quase sempre deslocado a uma distância k , que pode ser positiva ou negativa, de acordo com o tipo de deformação que se deseje causar, procurando assim manter a característica inicial: se foi dito que o contorno estaria fora do objeto, isto deve valer para todas as fatias.

Se por um lado esta solução resolve um problema, por outro, gera um novo desafio: a escolha do valor para este novo parâmetro. Novamente isto dependerá tanto do objeto a ser segmentado quanto do tipo e da qualidade da imagem com a qual se trabalha. Um parâmetro com valor muito alto pode levar o contorno para próximo de outras bordas que possuam características semelhantes às do objeto procurado, o que levaria alguns snaxels a se estabilizarem em locais que não são os desejados. Por outro lado, se o deslocamento for muito pequeno o contorno ainda poderia ficar no interior do objeto quando era esperado que ele fosse conduzido para fora.

Testes foram feitos na segmentação do quadrado com e sem o uso do algoritmo de caminhos mínimos, a fim de comparar os resultados. As Figuras 4.5(a) e 4.5(b) representam a segmentação do quadrado mais interno. As Figuras 4.6(a) e 4.6(b) ilustram os resultados obtidos para o quadrado externo.

4.2 Trabalhos Relacionados

A idéia de combinar recursos utilizados por vários métodos num único não é original e tem demonstrado ser crescente. Como visto previamente, a Dual-T-Snakes [GGA00] partiu da combinação da T-Snakes [MT95, MT97, McI97] com o método Dual [GN97]. Mais recentemente, foram publicados resultados utilizando um método denominado *United Snakes* [LMT99a, LMT99b] que utiliza Contornos Deformáveis e uma variação do método *Live-wire* [MB98] na tarefa de segmentação de imagens.

A *United Snakes* é uma ferramenta que reúne diversas variações de *snakes*, incluindo diferenças finitas, B-Spline e *snake* polinomial Hermite dentro de um *framework* matemático com formulação de elemento finito genérica, que possibilita ao usuário escolher qual a função de forma que deseja trabalhar.

Faz-se aqui necessário ressaltar que o método *live-wire* utilizado na *United Snakes* não é o mesmo apresentado no Capítulo 3. Embora a adaptação dada por Mortensen e

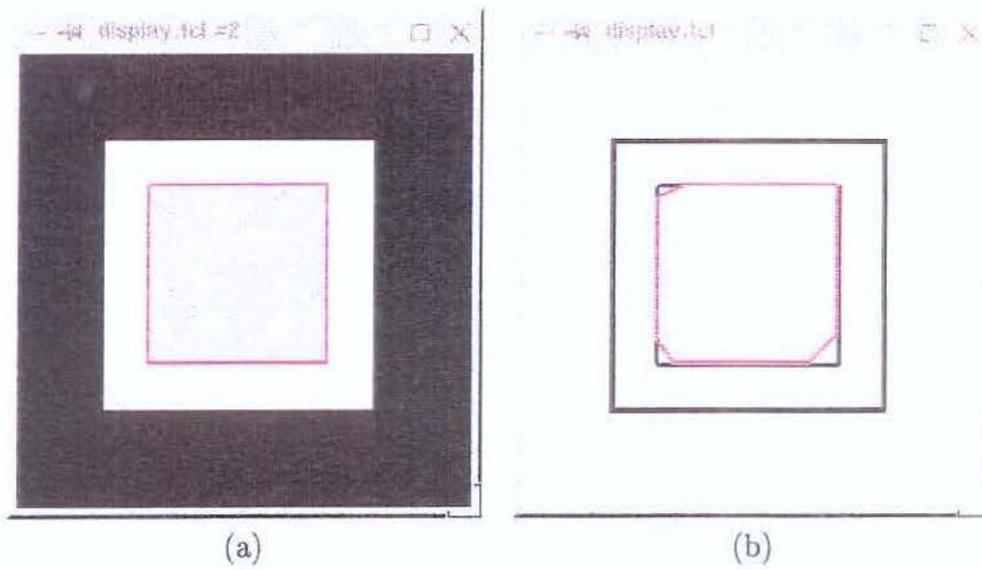


Figura 4.5: (a) Resultado da Segmentação do Quadrado Interno utilizando o algoritmo de caminhos mínimos. (b) Resultado da Segmentação do Quadrado Interno sem utilizar o algoritmo de caminhos mínimos.

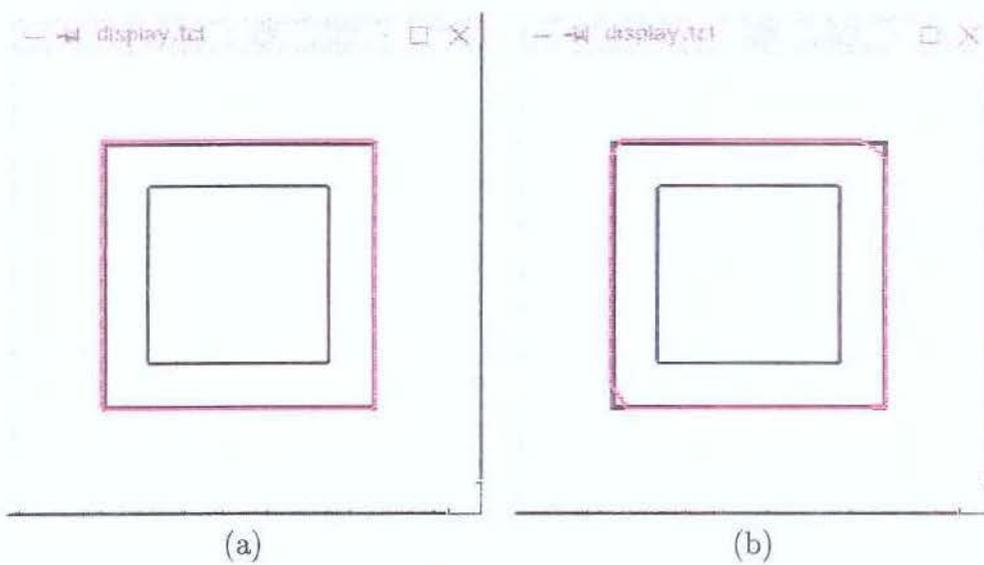


Figura 4.6: (a) Resultado da Segmentação do Quadrado Externo utilizando o algoritmo de caminhos mínimos. (b) Resultado da Segmentação do Quadrado Externo sem utilizar o algoritmo de caminhos mínimos.

Barrett [MB98] utilize o mesmo princípio apresentado em Falcão[Fal96, FUM99], ou seja, dado um ponto inicial é traçado um segmento até o novo ponto identificado pelo usuário, repetindo este processo até que toda a borda do objeto seja segmentada, as funções de custo utilizadas por Mortensen são diferentes daquelas utilizadas por Falcão. Liang utiliza em seu método *United Snakes* a versão adaptada de Mortensen, a qual apresenta vários problemas de aderência a bordas fracas. O que não ocorre em [FUM99].

Para processar uma segmentação, o método se divide nas seguintes etapas:

- (a) O usuário inicia a segmentação usando o *Live wire*.
- (b) O contorno gerado pelo *live-wire* passa a ser a *snake*, e os pontos demarcados pelo usuário durante o processo com o *live-wire* são chamados de pontos de restrição, que servem como identificadores da borda para a *snake*.
- (c) O usuário seleciona a função que deseja utilizar para mover a *snake* de acordo com o tipo de borda a ser segmentada.
- (d) Se a *snake* ficar presa em mínimos locais, o usuário pode utilizar mecanismos como o da *spring* e do *volcano*, utilizados no método original da *Snakes* [KWT87].

Esta forma de combinação do *live-wire* com a *snake* é chamada de “estática”, uma vez que o resultado do *live wire* é utilizado para inicializar a *snake* e o processo de segmentação continua usando apenas controles sobre a *snake*. Outra forma de integração é a “dinâmica”: cada vez que um ponto é marcado sobre a borda gerando mais um segmento de *live wire*, o caminho entre o último ponto marcado e o ponto livre gera uma *snake* aberta. Quando este ponto livre é marcado sobre a borda, esta *snake* aberta é unida à *snake* construída a partir dos segmentos de *live-wire* anteriores (caso existam). Liang afirma [LMT99b] que, utilizando a abordagem dinâmica, uma vez que a *snake* é automaticamente posta em movimento, o usuário pode utilizar o mecanismo de *spring* [KWT87] para corrigir qualquer problema ao longo do contorno.

Pode-se afirmar que a intersecção entre a *United Snakes* e o método de Contornos Deformáveis utilizando Segmentos de Custo Mínimo não é vazia, mas também não é total. A principal área comum a ambos os métodos é:

- Ambos são métodos que lidam com segmentação de imagens partindo de resultados obtidos através dos métodos *Snakes* (e suas variações) e *Live wire*.

Dentre as diferenças observadas, encontram-se:

- (a) A função de custo que é utilizada na geração da Energia da Imagem pela *United Snakes* [LMT99b] não é a mesma utilizada pelo CDSCM.

- (b) O algoritmo de busca de caminhos mínimos que é executado na *United Snakes* é apenas no início do processo de geração do contorno, enquanto no CDSCM este é executado na conclusão da delineação de cada fatia a ser segmentada.
- (c) Liang [LMT99b] afirma que o método *Live wire* no qual o *United Snakes* se baseia não apresenta bons resultados quando há falhas na borda da imagem ou excesso de ruído, enquanto que uma das principais motivações de atrelar ao CDSCM o algoritmo de caminhos mínimos utilizado pelo *live-wire* estava exatamente no fato de ele vir apresentando resultados muito satisfatórios exatamente nestas condições adversas [FUM99].

Quanto aos métodos *Live* de maneira geral, os pontos em comum entre eles e o CDSCM são:

- (a) O princípio de borda orientada.
- (b) São métodos baseados na abordagem de segmentação por segmentos de borda de custo mínimo.
- (c) A tarefa de delineação é realizada de forma automática.

4.3 Conclusão

Neste capítulo foi apresentado o método de Contornos Deformáveis usando Segmentos de Custo Mínimo. Este método utiliza princípios originados a partir dos métodos Dual-T-Snakes [GGA00] e dos métodos *Live* [Fal96, FUM99].

Do primeiro, aproveitou-se o mesmo conceito de *snake*: um conjunto de n pontos, chamados *snaxels*, que compõem um contorno fechado, sendo aplicado sobre cada *snaxel*, duas forças: uma Interna e outra Externa, visando posicioná-lo sobre a borda do objeto de interesse. Nestes métodos não há preocupação em minimizar a Energia, e ambos utilizam a mesma forma de geração da Força Interna.

Do segundo aproveitou-se a idéia de trabalhar com o conceito de borda orientada e do algoritmo de segmentos de custo mínimo na delineação da borda. O usuário identifica através de um parâmetro específico a orientação da borda desejada para ser segmentada, procurando minimizar o efeito atrator de outros mínimos locais que possuam algumas características diferentes.

As condições de parada estabelecidas para o contorno foram definidas a partir de vários experimentos com o próprio método CDSCM, bem como os valores aplicados aos diversos parâmetros utilizados (como os pesos aplicado às Forças Interna e Externa e sob que condições uma força deveria atuar mais que outra). Foram os diversos experimentos

também que determinaram os valores para a função de custo utilizada na geração da Energia Externa.

Capítulo 5

Resultados

Neste capítulo são apresentados os resultados obtidos neste trabalho utilizando o método de Contornos Deformáveis com Segmentos de Custo Mínimo na segmentação de imagens tomográficas do corpo humano. Os resultados são analisados e discutidos com respeito ao desempenho do método na tarefa de minimizar o envolvimento do usuário na segmentação.

O método proposto visa apenas a segmentação de objetos que definem uma única borda externa por fatia. Portanto, o método não trata mudanças topológicas. A estratégia utilizada requer um contorno inicial especificado pelo usuário na primeira fatia, o qual deve se deformar até aderir à borda externa do objeto nesta e em todas outras fatias subsequentes. Observe que o usuário também pode segmentar a borda na primeira fatia com o *Live-Wire* e o resto do processo ser executado na segunda fatia em diante. Uma falha neste processo em uma dada fatia, requer a intervenção do usuário através da reinitialização do contorno na mesma. Obviamente existem outras alternativas mais eficientes de intervenção não exploradas neste trabalho. As etapas envolvidas neste processo de segmentação são descritas abaixo:

(a) Definição de parâmetros:

O usuário especifica um intervalo da seqüência de imagens tomográficas onde ocorrerá a segmentação; um contorno inicial sobre a primeira fatia deste intervalo; a orientação da borda desejada; e a posição do contorno em relação ao objeto (se ele está fora ou dentro do objeto).

(b) Pré-processamento:

Esta etapa consiste de uma suavização da imagem original (filtro Gaussiano) para reduzir ruídos, cálculo da Energia Externa e verificação com possível correção das distâncias mínimas e máximas entre os snaxels.

(c) Deformação do contorno:

Esta etapa envolve o cálculo da Força Resultante (equação 4.10) e consequente deslocamento de cada snaxel. Este processo é repetido iterativamente até a parada do contorno nas proximidades da borda do objeto.

(d) Deformação por segmentos de custo mínimo:

Após a parada, a delineação da borda desejada é concluída calculando-se os segmentos de custo mínimo entre snaxels que estão sobre a borda.

(e) Reinicialização na próxima fatia:

Esta etapa envolve a especificação automática do contorno inicial sobre a próxima fatia.

A seguir são discutidas e analisadas as tarefas envolvidas em cada uma das etapas acima e os resultados obtidos.

5.1 Definição de parâmetros

Dada uma sequência de imagens tomográficas do joelho, composta por vinte (20) fatias, um contorno inicial é especificado nas proximidades de um objeto de interesse. As Figuras 5.1(a), 5.1(b), 5.1(c), e 5.1(d) são exemplos de contornos iniciais em torno de diferentes objetos. Estes objetos serão identificados ao longo deste capítulo pelos seguintes nomes: osso, pele, tecido mole 1 e tecido mole 2.

As distâncias mínima e máxima permitidas entre os snaxels e os valores para a tarefa de limiarização da imagem são alguns dos principais parâmetros envolvidos nesta etapa. A definição de cada um destes valores é um grande desafio, dependendo do objeto a ser segmentado.

No processo de definição das distâncias, partiu-se dos valores: distância máxima igual a seis pixels e distância mínima igual a três pixels. Após sucessivos testes, observou-se que para imagens onde o objeto a ser segmentado não é tão pequeno (embora se reconheça que a definição de pequeno ou grande é um tanto subjetiva), como na segmentação do osso e da pele (Figura 5.1(a) e (b)), os valores distância máxima 20 pixels e distância mínima 10 pixels apresentavam resultados muito bons (exemplos nas seções 5.3 e 5.4). Quando aplicadas distâncias muito inferiores (como os valores inicialmente propostos) o processo para completar a segmentação tornava-se muito lento e os resultados não eram satisfatórios. A probabilidade de acontecer intersecção entre os snaxels também cresce à medida que a distância diminua.

Mas para segmentação de objetos menores, como as apresentadas nas Figuras 5.1(c) e (d), os melhores resultados obtidos foram com distâncias inferiores, como distância

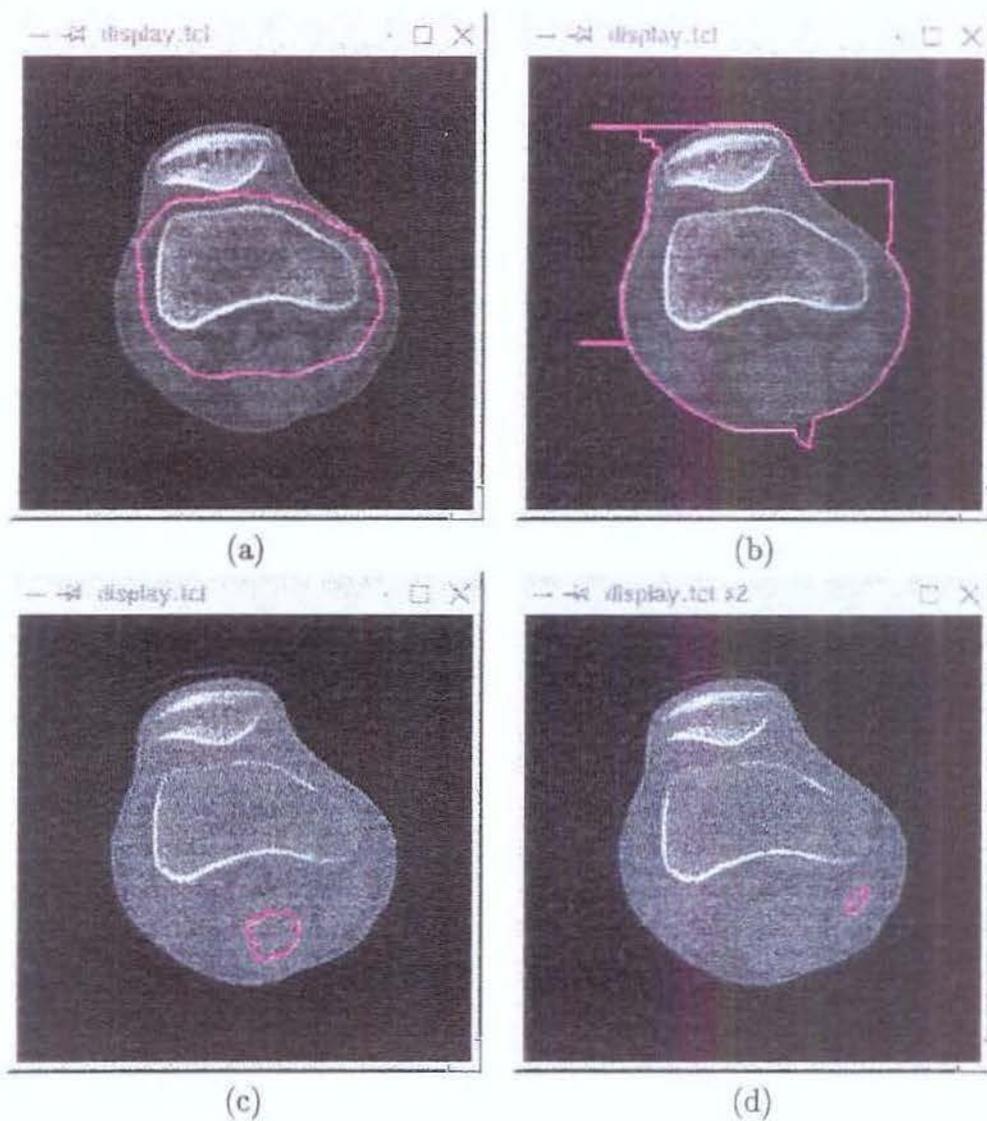


Figura 5.1: (a) Contorno inicial para segmentação do osso; (b) Contorno inicial para segmentação da pele; (c) Contorno inicial para segmentação do tecido mole 1; (d) Contorno inicial para segmentação do tecido mole 2.

máxima 6 pixels e distância mínima 3 pixels ou ainda com distância máxima 5 pixels e mínima igual a 2,5 pixels. Estes podem ser vistos na seção 5.3 e 5.4.

Um exemplo de problema devido a má escolha dos parâmetros de distância e orientação, pode ser visto na Figura 5.2. As distâncias neste caso são 10 e 20 pixels e a orientação admite que o lado de fora é mais escuro do que o de dentro (a intenção é segmentar o objeto de tecido mole pelo seu lado de fora. O contorno inicial é o mesmo apresentado na Figura 5.1(d)).

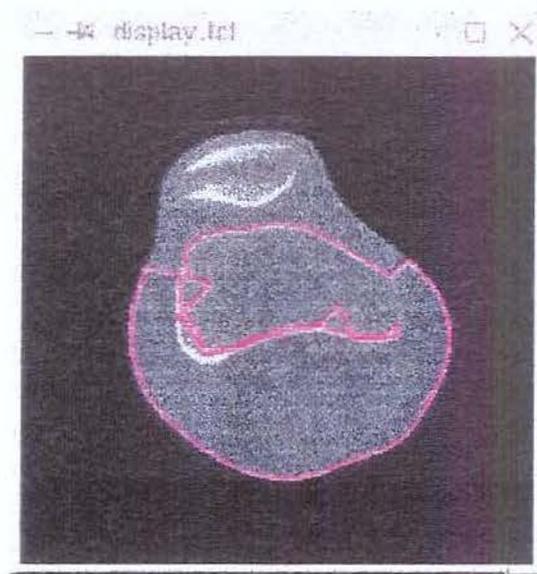


Figura 5.2: Contorno final resultante. Os erros ocorrem devido a má escolha dos parâmetros. O contorno inicial foi o mesmo utilizado no tecido mole 2 (e este era o objeto de interesse). O contorno pára em pontos de borda, porém não os do objeto desejado.

5.2 Pré-processamento

Para que a segmentação seja melhor sucedida, é necessário suavizar a imagem com um filtro Gaussiano. Em seguida, são calculados os valores da Energia da Imagem (Figura 5.3) e, de acordo com os valores de limiar pré-definidos na etapa anterior, feito uma limiarização na imagem. Esta limiarização é necessária a fim de retirar da imagem o maior número possível de informações não necessárias. Para a segmentação de bordas mais fortes, como a pele e osso, os limiares máximo e mínimo adotados foram: 120 e 0 (cento e vinte e zero). Enquanto para bordas mais fracas, como o tecido mole do joelho, os valores dos limiares eram respectivamente 200 e 100 (duzentos e cem). As imagens geradas podem ser vistas nas Figuras 5.4 e 5.5.

Conforme equação 4.8 (capítulo 4), o valor adotado para a afinidade máxima entre dois pixels foi 256 (i.e., o valor máximo de um pixel na imagem) e para o desvio padrão o valor utilizado nos testes foi igual a 100.

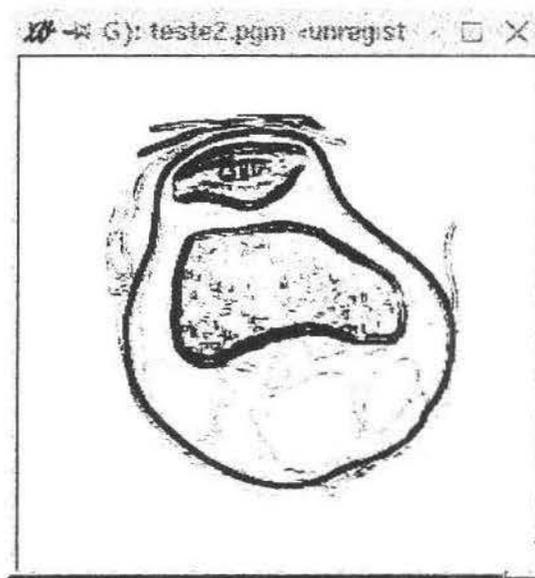


Figura 5.3: Energia da Imagem.

5.3 Deformação do contorno

A deformação do contorno está diretamente ligada às ações da Força Resultante. Para geração desta força, a definição dos valores para as constantes que atuam sobre as Forças Interna e Externa é também um desafio.

No início do processo, as constantes α e β possuem valores fixos. A Força Externa ganha um peso maior a fim de ter uma influência maior no deslocamento do snaxel, procurando conduzi-lo para mais próximo da borda de interesse. Entretanto, fatores como: a localização do snaxel, a orientação do gradiente e, especialmente, o tamanho do objeto desejado, alteram estes valores.

Os critérios utilizados, portanto, para definição de cada parâmetro e conseqüentemente para deslocamento dos snaxels são:

- (a) Se snaxel estiver posicionado em um ponto de mínimo e a orientação da borda satisfaz a condição desejada, nenhuma força é aplicada. Este snaxel pára de mover-se.

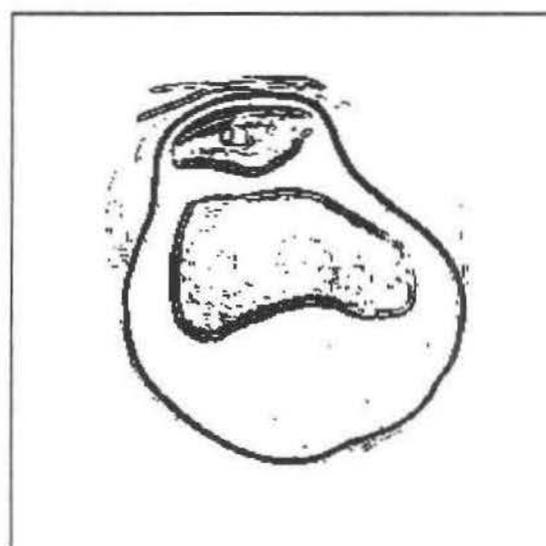


Figura 5.4: Limiarização da Energia da Imagem do joelho para bordas fortes.



Figura 5.5: Limiarização da Energia da Imagem do joelho para bordas fracas.

- (b) Se o snaxel está posicionado sobre um ponto de mínimo mas a orientação da borda é contrária à desejada, um pequeno deslocamento é provocado, a fim de retirar o snaxel do falso mínimo. O objetivo é conduzir o objeto para uma posição ainda na borda que satisfaça a condição de orientação procurada. Neste caso apenas a Força Interna é aplicada (numa intensidade muito pequena) a fim de provocar este deslocamento. A Força Externa não é utilizada pois sua direção neste ponto é contrária à desejada. Se o deslocamento do snaxel for muito grande o snaxel pode se afastar da borda.
- (c) Se o snaxel não está sobre um ponto de mínimo mas a direção do gradiente satisfaz a orientação desejada, as Forças Interna e Externa atuam com a mesma intensidade no deslocamento do snaxel, pois a Força Interna provoca a expansão (ou contração) do contorno na direção da Força Externa.
- (d) Se o snaxel não está sobre um ponto de mínimo e a direção do gradiente é contrária à desejada, apenas a Força Interna atua, pois isto indica que a região não é de borda.

Como dito anteriormente, a definição dos valores das constantes que atuam sobre as forças depende muito do tipo de borda que se deseja e da imagem na qual se trabalha. Desta forma, os valores utilizados para α e β (ver equação 4.10, Capítulo 4), são: para objetos menores (como os de tecido mole) α igual a 0.15 e 0.5, e para β os valores 0.3 e 0.5. Enquanto para objetos maiores (como osso e pele), α e β iguais a 1 ou igual a 0.5. Quando ambas as forças devem ser aplicadas com a mesma intensidade, ambas as constantes recebem o mesmo valor; se apenas a Força Interna deve atuar, β recebe valor 0 e α recebe um valor mais alto. Se Força Externa deve ter uma influência maior no deslocamento do snaxel, β recebe um valor maior que α .

Os exemplos exibidos nesta seção mostram o desempenho do modelo de Contornos Deformáveis utilizado, sem o uso de segmentos de custo mínimo.

Utilizando o contorno inicial 5.1(a) um primeiro resultado obtido pode ser visto na Figura 5.6(a). Mudando a orientação da borda obteve-se a segmentação pelo lado de fora do objeto (Figura 5.6(b)).

Na segmentação da pele, (usando como contorno inicial, o mesmo exibido na Figura 5.1(b)) obteve-se o resultado ilustrado na Figura 5.7.

Os resultados obtidos na segmentação do tecido mole estão na Figura 5.8.

Quando o contorno estabiliza não significa que uma borda foi encontrada (como já foi mencionado na seção 4.1.3 no capítulo anterior), ou que o contorno convergiu para o objeto desejado. A Figura 5.2 mostra um exemplo onde o contorno parou sobre as bordas de diversos objetos e nenhuma é a do objeto de interesse. O objeto desejado era o mesmo identificado na Figura 5.1(c). Outro problema detectado ao fim do processo é

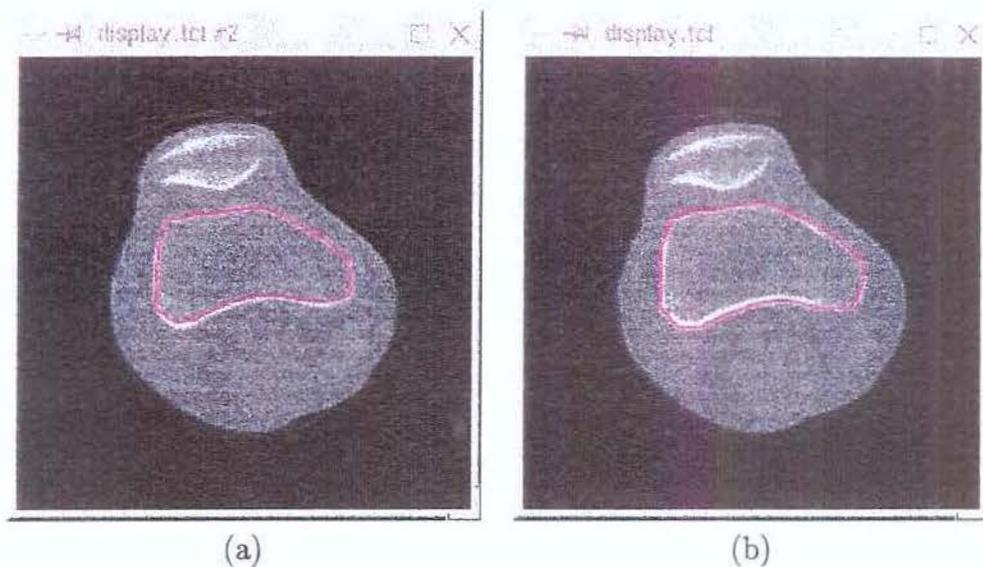


Figura 5.6: Contorno resultante da segmentação do osso, sem o uso de segmentos de custo mínimo (a) Pelo lado de dentro da borda. (b) Pelo lado de fora da borda.

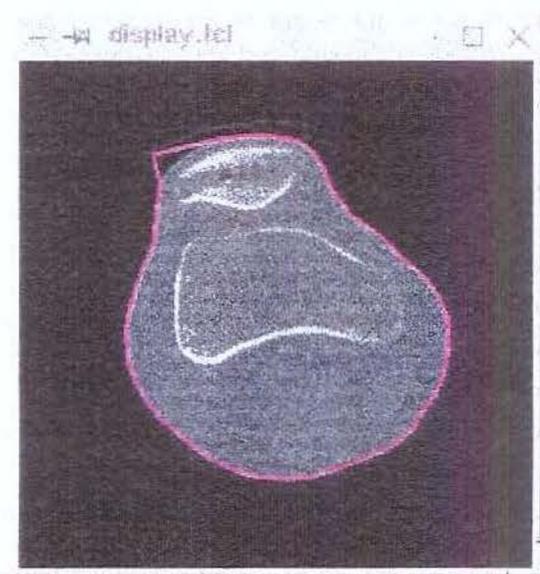


Figura 5.7: Contorno resultante da segmentação da pele que envolve o joelho, pelo lado de fora da borda, sem o uso de segmentos de custo mínimo.

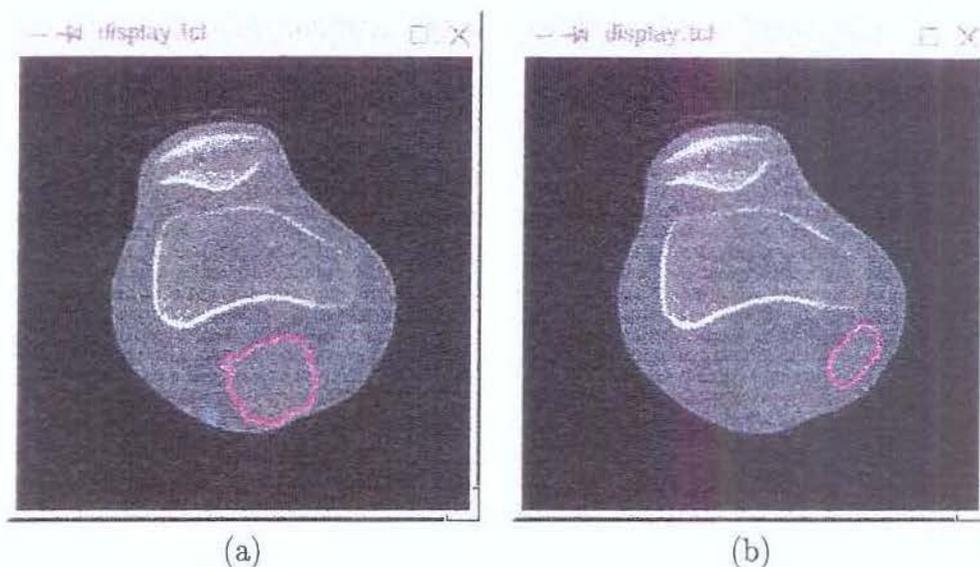


Figura 5.8: Contorno resultante da segmentação do tecido mole sem o uso de segmentos de custo mínimo.

o de delimitação: ainda que todos os snaxels parem sobre a borda desejada, é necessário criar um caminho entre eles de maneira que a borda seja redesenhada, pois os snaxels estão distantes uns dos outros. O simples fato de criar segmentos entre eles, não resulta no contorno do objeto. A fim de resolver este problema foi utilizado segmentos de custo mínimo.

5.4 Deformação por segmentos de custo mínimo

Utilizando os contornos apresentados na seção anterior como ponto de partida para execução do algoritmo de custo mínimo, os resultados obtidos foram:

Para a pele, o resultado final pode ser visto na Figura 5.9.

Na segmentação do osso os resultados foram melhores quando a borda foi segmentada pelo lado de fora do objeto (Figura 5.10(b)). Pelo lado interno, as falhas do contorno gerado na etapa anterior foram realçadas (isto ocorre porque o segmento custo mínimo gera um contorno cujos snaxels possuem distância igual a 1, e isto faz com que os erros da conversão sejam realçados)(Figura 5.10(a)). Detalhe: em ambos os casos, o contorno inicial era o mesmo (Figura 5.1(a)), apenas a orientação da borda desejada foi diferente.

Para segmentação de partes compostas por tecido mole, o resultado obtido utilizando segmentos de custo mínimo pode ser vista nas Figuras 5.11 e 5.12.

Como pode ser observado, os detalhes do local onde o contorno se encontra são realçados.

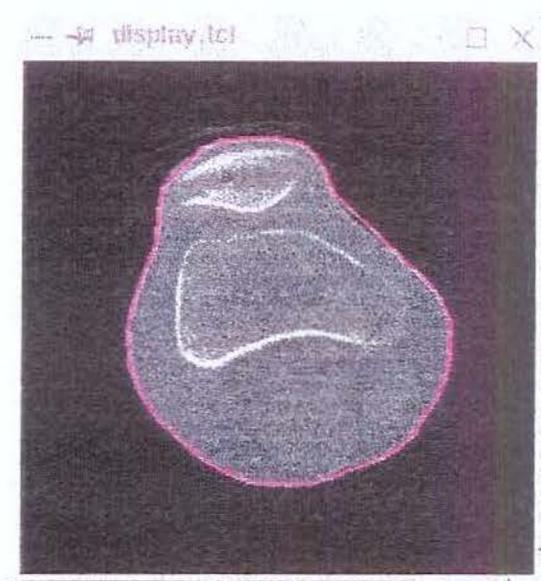


Figura 5.9: Contorno resultante da segmentação da pele que envolve o joelho, pelo lado de fora da borda, utilizando segmentos de custo mínimo para concluir a segmentação.

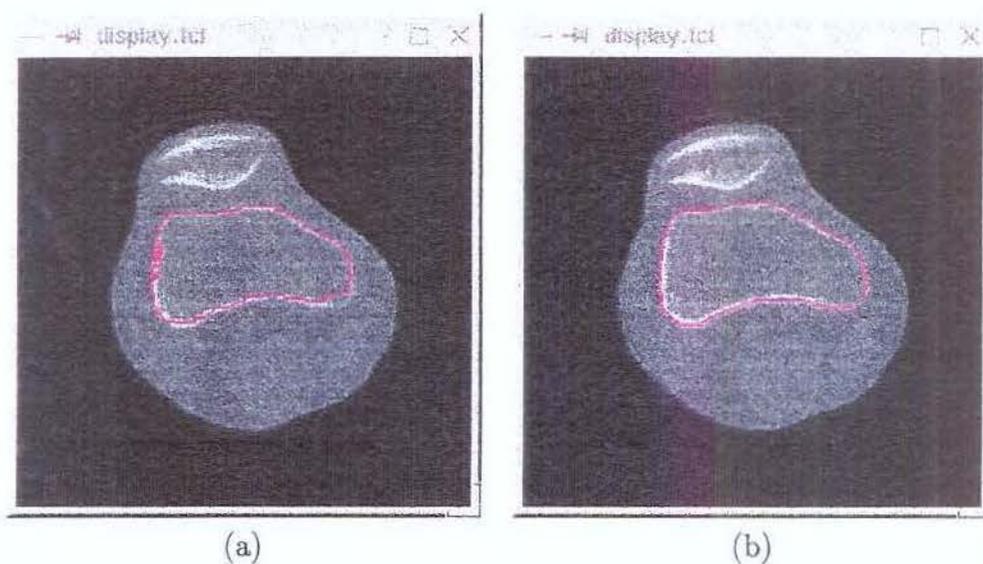


Figura 5.10: Contorno resultante da segmentação do osso, usando segmentos de custo mínimo. (a) Pelo lado de dentro da borda. (b) Pelo lado de fora da borda.

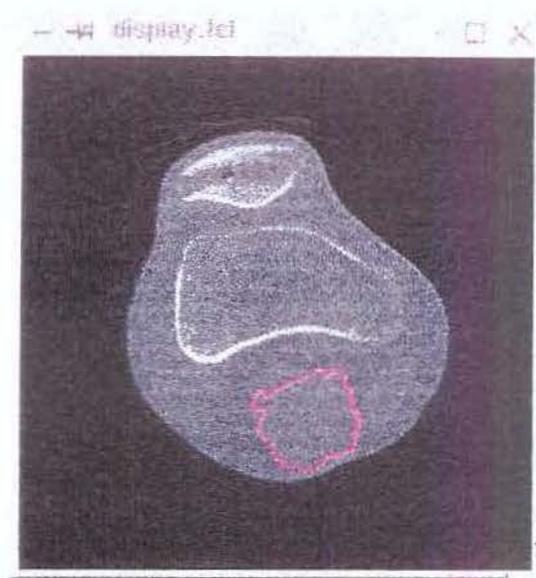


Figura 5.11: Resultado da segmentação de um objeto de tecido mole na imagem do joelho, com o uso de segmentos de custo mínimo.

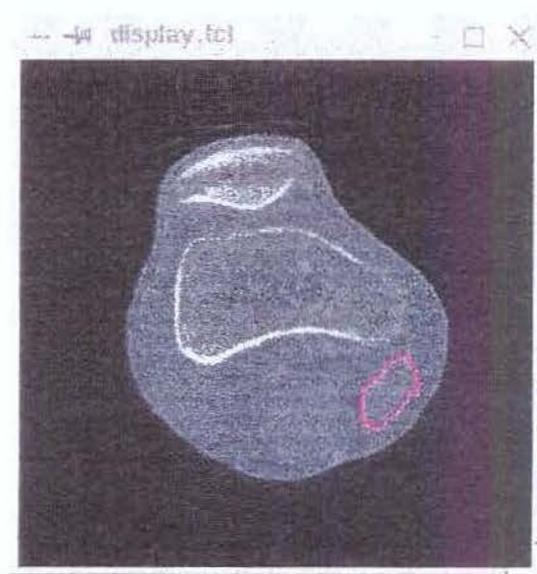


Figura 5.12: Resultado da segmentação de um segundo objeto de tecido mole na imagem do joelho, com o uso de segmentos de custo mínimo.

5.5 Reinicialização na próxima fatia

Após concluída a segmentação numa fatia, o contorno obtido deve ser usado para estimar o contorno inicial na fatia seguinte.

Usando apenas contornos deformáveis, a maior sequência obtida gerada automaticamente consiste de cinco fatias, usando a pele como objeto de interesse. Os resultados obtidos podem ser vistos na sequência 5.13.

Usando segmentos de custo mínimo, foi possível gerar todo o contorno para 20 fatias sequenciais com resultado ótimo em cada uma delas. O mais interessante foi que este resultado foi possível sem qualquer intervenção do usuário durante o processo (exceto gerando o contorno inicial na primeira fatia desta sequência), tornando esta segmentação totalmente automática. Algumas fatias deste processo podem ser vistas na Figura 5.14.

Observa-se que não apenas as características da imagem são relevantes na definição dos parâmetros a serem segmentados, mas o objeto de interesse possui peculiaridades que definem os valores dos parâmetros a serem utilizados. Além dos resultados ilustrados da imagem do joelho, outras sequências foram testadas. Nestas, as dificuldades de encontrar a combinação mais adequada de valores para os parâmetros foram confirmadas, pois os valores utilizados na imagem do joelho não eram sempre os mais apropriados em outras sequências. Isto fica ainda mais claro na sequência a seguir. A imagem tomográfica utilizada é de um pé, mas para melhor realçar os detalhes da imagem, a sequência a seguir é ilustrada com a Energia da Imagem (Figura 5.15).

A Figura 5.16(a) mostra a Energia da Imagem limiarizada com os valores 200 e 100 (máximo e mínimo, respectivamente), para segmentação do calcâneo. O contorno inicial utilizado aparece dentro do objeto de interesse na Figura 5.16(b).

Usando apenas contornos deformáveis sem segmentos de custo mínimo, e todos os mesmos parâmetros utilizados na segmentação de bordas fortes do joelho (exceto os valores do limiar), o resultado obtido para segmentação do calcâneo pode ser visto na Figura 5.17(a). Aplicando segmentos de custo mínimo, o resultado obtido foi 5.17(b) (Observe como o erro se propaga nas fatias seguintes - Figuras 5.17(c) e (d)).

Na tentativa de utilizar distâncias menores (com os mesmos valores adotados para o tecido mole do joelho) na segmentação do talus, os resultados não foram satisfatórios. Snaxels acabam se cruzando, como mostra a Figura 5.18(a). Ou, se utilizado limiares maiores, o contorno se expande demasiadamente, ultrapassando os limites do objeto de interesse (Figura 5.18(b)).

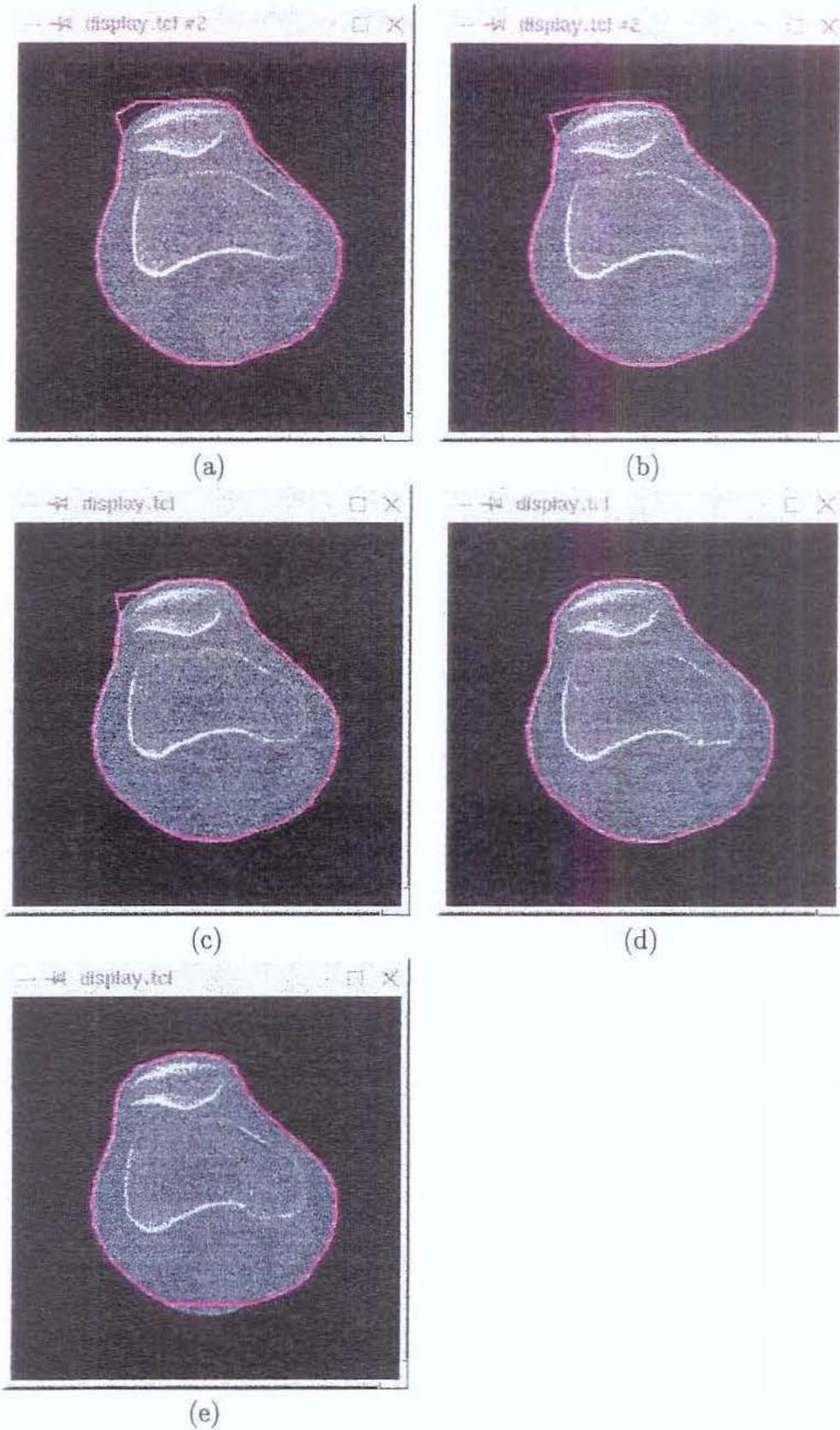


Figura 5.13: Resultado final da segmentação pele usando contornos deformáveis sem segmentos de custo mínimo. A maior sequência de imagens que foi possível segmentar sem uma nova intervenção do usuário foi de 5 fatias: (a) primeira, (b) segunda, (c) terceira, (d) quarta e (e) quinta.

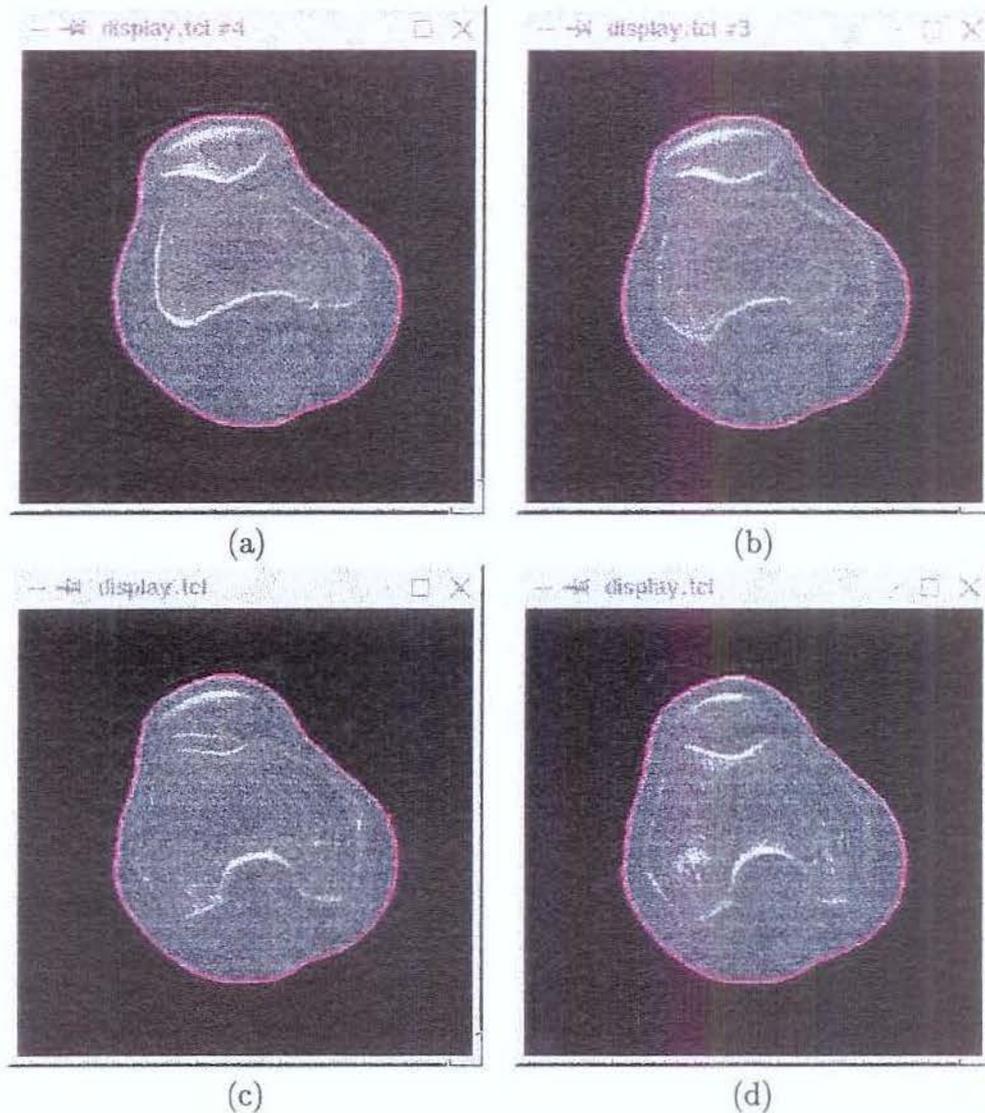


Figura 5.14: Usando segmentos de custo mínimo foi possível gerar toda a sequência de 20 fatias com apenas uma intervenção do usuário na segmentação da pele. A Figura ilustra cinco fatias desta sequência, as quais são: (a) quarta; (b) oitava; (c) décima sexta e (d) vigésima. O contorno final de uma fatia era exatamente o contorno inicial da seguinte.



Figura 5.15: Energia da imagem do pé.

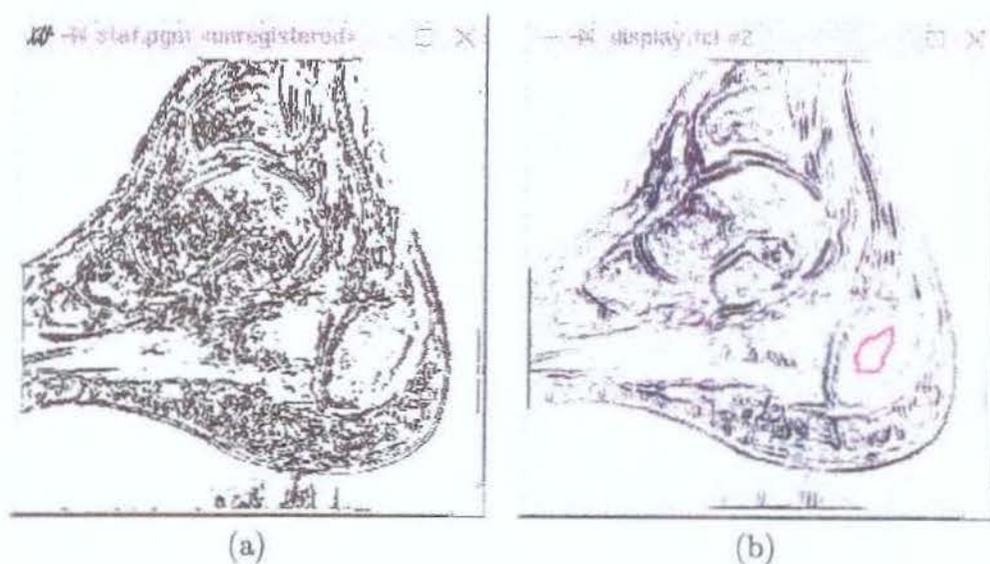


Figura 5.16: (a) Limiarização da Energia da Imagem com valores 100 e 200 para segmentação do talus; (b) Contorno inicial para segmentação do calcâneo.

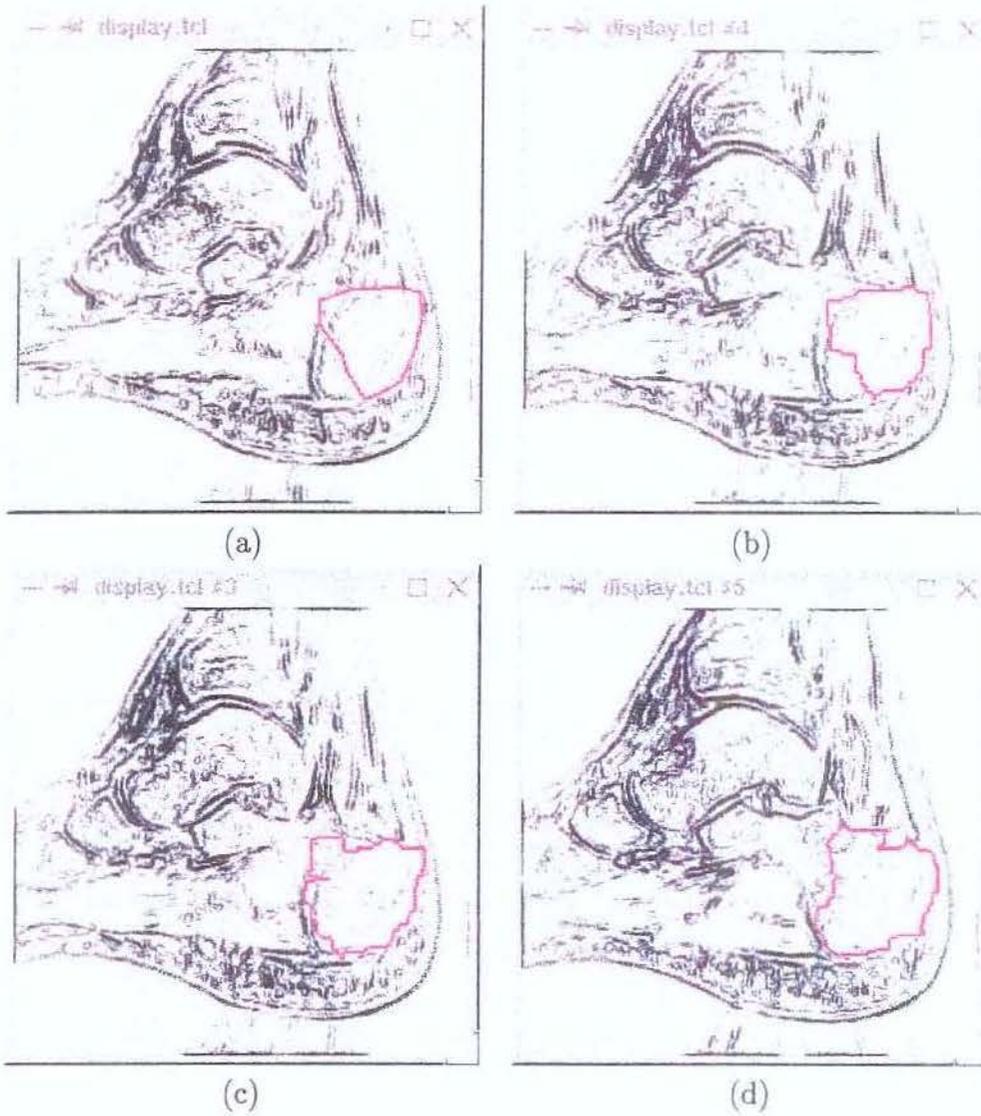


Figura 5.17: (a) Contorno final da segmentação do calcâneo usando apenas contornos deformáveis; (b) Contorno final usando segmentos de custo mínimo. As Figuras (c) e (d) mostram a propagação do erro nas fatias seguintes.

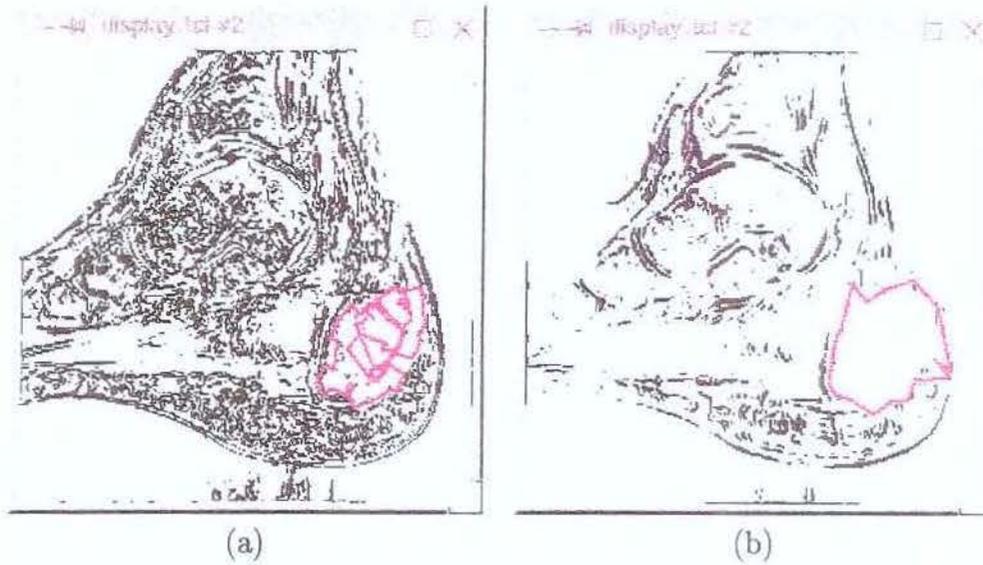


Figura 5.18: (a) Contorno final da segmentação do calcâneo usando as distâncias 3 e 6 pixels (máxima e mínima, respectivamente); (b) Contorno final usando valores 0 e 100 para a limiarização da Energia da Imagem.

5.6 Conclusão

Os experimentos mostraram que o uso de uma segmentação totalmente automática ainda é um desafio que precisa ser superado, mas que tem dado seus passos iniciais neste sentido.

Os resultados obtidos usando o método apresentado não foram sempre satisfatórios. As dificuldades maiores encontram-se na definição de parâmetros que permitam resultados favoráveis em todas as fatias. Um exemplo disto está na própria limiarização da Energia da Imagem, pois nem sempre o valor inicial adotado era o mais adequado para todas as outras fatias.

Foram feitos testes usando uma constante de expansão (contração) no contorno final de cada fatia, a fim de procurar conduzi-lo para fora (ou dentro) do objeto na fatia seguinte. No caso do tecido mole, esta constante foi fundamental para evitar uma segmentação fatia por fatia. Mas, ainda assim o melhor resultado obtido foi a segmentação imediata de duas fatias. Na terceira, o usuário já precisou intervir. As Figuras 5.8 e 5.11 mostram resultados para a primeira fatia de uma sequência. Usando esta constante de contração, o contorno inicial na fatia seguinte passou a ser 5.19(a) e o resultado final pode ser visto na Figura 5.19(b), na fatia seguinte, porém o resultado já não foi satisfatório, como pode ser visto na Figura 5.19(c).

Sendo assim, observa-se o quanto a intervenção do usuário ainda é fundamental para garantir resultados sempre ótimos. Com o método proposto, o que se pode ganhar é a minimização desta participação no mínimo pela metade, se duas fatias forem segmentadas

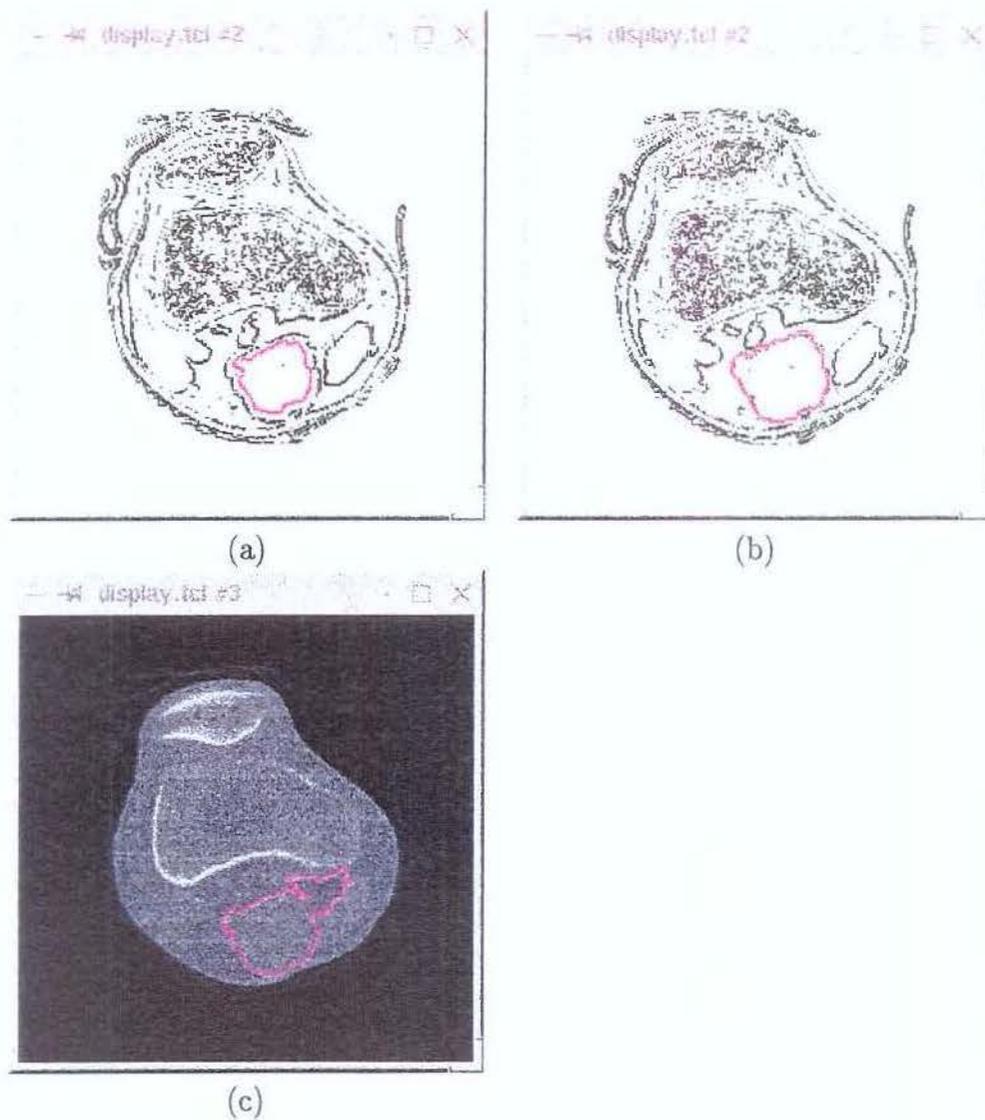


Figura 5.19: Segmentação do tecido mole. (a) Contorno inicial para a segunda fatia de uma sequência de imagens, usando uma constante de contração. (b) Contorno final obtido na segunda fatia. (c) Contorno final obtido na terceira fatia.

em sequência, dado um contorno inicial, ou fatia por fatia, o que se iguala ao método *Live-Wire*. Mas, o resultado obtido com a segmentação da pele do joelho, mostrou já ser possível reduzir esta participação ao máximo, o que já é um ganho muito significativo.

Capítulo 6

Conclusão

Neste trabalho apresentou-se um método para segmentação de imagens baseado em modelos de Contornos Deformáveis e em paradigmas que utilizam segmentos de custo mínimo. Esta combinação visou originar um método que fosse o mais automático possível nas tarefas de reconhecimento e delineação e garantisse resultados satisfatórios. Neste capítulo são apresentados os principais aspectos que originaram este método, suas vantagens e dificuldades, além da proposta de trabalhos futuros.

Segmentação de imagens divide-se basicamente em duas etapas: Reconhecimento e Delineação. A primeira, tem sido feita com melhor resultado pelo ser humano do que por algoritmos de computador, enquanto ocorre o inverso na delineação. Métodos de Contornos Deformáveis têm se aperfeiçoado na etapa de reconhecimento, de maneira a procurar reduzir a participação do usuário, o que torna o processo mais rápido e conseqüentemente mais útil em aplicações clínicas. Entretanto, dificuldades como estabilização do contorno e convergência têm sido os pontos fracos nesta abordagem.

Métodos que utilizam segmentos de custo mínimo, como o *Live-Wire* [Fal96, FUM99] têm se aperfeiçoado na tarefa de delineação, permitindo resultados ótimos mesmo na presença de ruídos e na ausência de informações. Entretanto, estes métodos podem requerer a intervenção do usuário a cada fatia, o que pode ocasionar um processamento mais demorado.

Procurando reunir as vantagens das abordagens acima, o método de Contornos Deformáveis com Segmentos de Custo Mínimo combina os princípios básicos de ambas as abordagens, ou seja, o uso de duas forças, uma Interna e outra Externa, responsáveis pelo deslocamento do contorno inicial até o objeto de interesse. Uma vez posicionado próximo à borda, segmentos de custo mínimo são responsáveis por realizarem a delineação da borda encontrada. Numa sequência de imagens, o contorno final gerado na primeira fatia é utilizado como contorno inicial na fatia seguinte. A tarefa do usuário fica mais restrita a verificar os resultados obtidos fatia por fatia, e ao perceber que o resultado não satisfaz

numa dada fatia, um novo contorno inicial é gerado para esta e o processo automático prossegue nas fatias seguintes.

As dificuldades encontradas com esta nova abordagem foram diversas: a primeira delas é a definição dos parâmetros, que variam de acordo com o objeto de interesse. Encontrar valores adequados para as distâncias entre os snaxels (pontos do contorno), definir o deslocamento máximo permitido a um snaxel de acordo com sua localização, valores para a operação de limiarização a fim de eliminar informações desnecessárias de acordo com o objeto de interesse, foram alguns destes desafios.

Outra dificuldade foi definir a condição de parada do contorno. Para resolver o problema a solução adotada foi: não basta um ponto estar posicionado sobre um ponto de borda, mas esta borda tem que estar na orientação especificada pelo usuário. Tal proposta levou a diferentes constantes que atuavam sobre as Forças Interna e Externa de acordo com a região onde o snaxel estava localizado, em cada instante da iteração. Estes parâmetros também variavam de objeto para objeto.

Aproveitar o resultado da fatia anterior na fatia seguinte também não foi uma prática sempre bem sucedida. Algumas vezes, o usuário informava que o contorno inicial estava por fora do objeto. Entretanto, ao ser gerado o contorno final numa fatia, esta poderia ficar no interior do objeto na fatia seguinte. Para resolver este problema, experimentou-se utilizar uma constante de expansão do contorno ao final de cada fatia, fazendo com que o contorno estivesse sempre externo (interno) ao objeto de interesse (de acordo com a informação inicial dada pelo usuário). Porém esta estratégia trouxe outras dificuldades, como: definir o valor deste parâmetro de expansão, de maneira que o contorno não se expandisse tanto de maneira a ficar longe do objeto de interesse (e próximo a outros pontos atratores) nem se deslocasse tão pouco que não fosse o suficiente para colocá-lo externo ao objeto desejado.

Como vantagens, observou-se que usando segmentos de custo mínimo o contorno convergia e delineava a borda de maneira superior aos testes feitos apenas com o método de contornos deformáveis adotado. O melhor sucesso obtido foi visto numa sequência de vinte fatias onde, com apenas um contorno inicial, todas as fatias foram segmentadas com sucesso.

A aplicação do conceito de borda orientada foi outra contribuição aos métodos de contornos deformáveis. Experimentos mostraram que sem o uso da orientação havia maior instabilidade no contorno. Além do que esta característica faz com que outros objetos que poderiam servir como atratores, deixem de ser.

Como trabalhos futuros, existe a proposta de realizar diversos outros estudos nos valores dos parâmetros a fim de melhorar a qualidade dos resultados e minimizar ainda mais a participação do usuário. Outra sugestão é que a inicialização do contorno seja feita usando um método de segmentação, e não usando o traçado totalmente manual, e.g., usar

o *live-wire* na primeira fatia e utilizar o resultado no processo das fatias seguintes. Além disso, pode-se fazer testes usando outras funções de custo para gerar a Energia da Imagem ou mesmo outras funções para gerar a Força Interna e Externa. Outra proposta é utilizar a Força Balão [CC93] e verificar, por exemplo, se isto agilizaria o processo de convergência da curva. Pode-se ainda, procurar ganhos maiores de estabilidade e convergência, unindo o método proposto ao de Giraldi [GGA00], por terem sido estes os pontos principais tratados no Dual T-Snakes.

É importante ressaltar que o uso de métodos híbridos tem sido uma crescente tendência na computação de imagens médicas [MT96]. Os resultados obtidos mostram que reunir pontos fortes de métodos distintos não é uma tarefa simples, tão pouco garantem sempre sucesso, mas ainda é uma experiência que vale a pena ser investigada, pois a pesquisa na área de segmentação de imagens médicas ainda apresenta muitos desafios a serem superados, e são estes investimentos que garantirão uma maior usabilidade deste processo na área clínica de muitos hospitais.

Bibliografia

- [AMO93] R.K. Ahuja, T.L. Magnanti, and J.B. Orlin. *Network Flows: Theory, Algorithms and Applications*. Prentice-Hall, Englewood Cliffs, NJ, 1993.
- [BB82] D.H Ballard and C.M. Brown, editors. *Computer Vision*. Prentice-Hall, 1982.
- [Boi88] J.D. Boissonnat. Shape reconstruction from planar cross sections. *Computer Vision, Graphics, and Image Processing*, pages 1–29, 1988.
- [Bri93] J.F. Brinkley. A flexible, generic model for anatomic shape: application to interactive two-dimensional medical image segmentation and mathing. *Computers on Biomedical Research*, 26:121–142, 1993.
- [BY93] A. Black and A. Yuille, editors. *Active Vision*. MIT Press, 1993.
- [CC90] L.D. Cohen and I. Cohen. A finite element method applied to new active contour models and 3d reconstruction from cross sections. *Proc. Third International Conference on Computer Vision*, pages 587–591, 1990.
- [CC92] L.D. Cohen and I. Cohen. Using deformable surfaces to segment 3d images and infer differential structures. *CVGIP: Image Understanding*, 56(2):242–263, Setember 1992.
- [CC93] Laurent D. Cohen and Isaac Cohen. Finite-element methods for active contour models and balloons for 2-d and 3-d images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(11), November 1993.
- [CCK95] Vikram Chalana, Wendy Costa, and Yongmin Kim. Integrating region growing and edge detection using regularization. *In Proceedings of the SPIE Conference on Medical Imaging*, 1995.
- [CLR91] T. Cormen, C. Leiserson, and R. Rivest. *Introduction to Algorithms*. MIT Press, New York, NY, 1991.

- [Coh91] L.D. Cohen. On active contour models and ballons. *In Computer Vision, Graphics, and Image Processing: Image Understanding*, 53(2):211–218, 1991.
- [Dij59] E.W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1:269–271, 1959.
- [Fal96] Alexandre X. Falcão. *Paradigmas de Segmentação de Imagem Guiada pelo Usuário: Live-Wire, Live-Lane e 3D-Live-Wire*. PhD thesis, Faculdade de Engenharia Elétrica - Universidade Estadual de Campinas, Campinas, SP, Brazil, December 1996.
- [FHW81] M.A. Fischler, J.A. Henenbaum, and H.C. Wolf. Detection of roads and linear structure in low resolution aerial imagery using a multi-source knowledge integration technique. *Computer Graphics and Image Processing*, 15(3):201–223, 1981.
- [FU97] Alexandre X. Falcão and J.K. Udupa. Segmentation of 3d objects using live wire. *Proceedings of SPIE on Medical Imaging, Newport Beach, CA*, pages 228–239, February 1997.
- [Fuk90] K. Fukunaga. *Introduction to Statistical Pattern Recognition*. Academic Press, New York, NY, 1990.
- [FUM99] Alexandre X. Falcão, J.K. Udupa, and F. K. Miyazawa. An ultra-fast user-steered segmentation paradigm: Live-wire-on-the-fly. *Proceedings of SPIE on Medical Imaging, San Diego, CA*, 1999.
- [GGA00] G.A. Giraldi, L.M. Gonçalves, and A.F. Oliveira. Dual topologically adaptable snakes. In *Proceedings of the Fifth Joint Conference on Information Sciences (JCIS'2000, Vol. 2) - Third International Conference on Computer Vision, Pattern Recognition, and Image Processing*, pages 103–106, February, 2000.
- [GN97] S.R. Gunn and M.S. Nixon. A robust snake implementation; a dual active contour. *IEEE Trans. Pattern Anal. Mach. Intell.*, 19(1):63–68, January 1997.
- [GO99] G.A. Giraldi and A.F. Oliveira. Snake models. Technical report, Universidade Federal do Rio de Janeiro, DEp. Eng. Sistemas e Computação, 1999. unpublished.
- [Gun96] Steve R. Gunn. *Dual Active Contour Models for Image Feature Extraction*. PhD thesis, Faculty of Engineering and Applied Science, Department of Electronics and Computer Science., May 1996.

- [Her92] G. Herman. Discrete multidimensional jordan surfaces. *gmip*, 54:507–515, 1992.
- [HUS96] B. E. Hirsch, J. K. Udupa, and S. Samarasekera. A new method of studying joint kinematics from 3d reconstructions of mri data. *Journal of the American Podiatric Medical Association*, 86(1):4–15, 1996.
- [Kec98] Rainer Keck. Reinitialization for level set methods. Diploma Thesis, June 1998.
- [KWT87] Michael Kass, Andrew Witkin, and Demetri Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1:321–331, 1987.
- [Lai94] Kok Fung Lai. *Deformable Contours: Modeling, Extraction, Detection and Classification*. PhD thesis, University of Wisconsin-Madison, 1994.
- [LL93] F. Leymarie and M.D. Levine. Tracking deformable objects in the plane using an active contour model. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 15:617–633, 1993.
- [LMT99a] J. Liang, T. McInerney, and D. Terzopoulos. Interactive medical image segmentation with united snakes. *MICCAI*, pages 116–127, 1999.
- [LMT99b] J. Liang, T. McInerney, and D. Terzopoulos. United snakes. *ICCV*, pages 993–940, 1999.
- [LV95] Steven Lobregt and Max. A. Viergever. A discrete dynamic contour model. *IEEE Transactions on Medical Imaging*, 14(1), March 1995.
- [Mac95] Blair Mackiewich. Active contour models (snakes). URL : <http://www.cs.sfu.ca/stella/papers/blairthesis/main/node29.htm>, 1995.
- [Mar71] A. Martelli. An application of heuristic search methods to edge and contour detection. *Communications of the ACM*, 19:73–83, 1971.
- [Mar72] A. Martelli. Edge detection using heuristic search methods. *Computer Graphics and Image Processing*, 1(2):169–182, 1972.
- [MB98] E.N. Mortensen and W.A. Barrett. Interactive segmentation with intelligent scissors. *Graphical Models and Image Processing*, 60:349–384, 1998.
- [McI97] T.J. McInerney. *Topologically Adaptable Deformable Models for Medical Image Analysis*. PhD thesis, Department of Computer Science, University of Toronto, 1997.

- [Mon71] U. Montanari. On the optimal detection of curves in noisy pictures. *Communications of the ACM*, 14(5):335–345, 1971.
- [MSV95] R. Malladi, J.A. Sethian, and B.C. Vemuri. Shape modeling with front propagation: A level set approach. *T-PAMI*, 17:158–175, 1995.
- [MT95] T. McInerney and D. Terzopoulos. Topologically adaptable snakes. In *Proc. Of the Fifth Int. Conf. On Computer Vision (ICCV'95), Cambridge, MA, USA*, pages 840–845, June 1995.
- [MT96] Tim McInerney and Demetri Terzopoulos. Deformable models in medical image analysis: a survey. *Medical Image Analysis*, 1(2):91–108, 1996.
- [MT97] T. McInerney and D. Terzopoulos. Medical image segmentation using topological adaptable surfaces. In *Proc. CVRMed'97*, March 1997.
- [NFK94] W. Nuenschwaander, P. Fua, and O Kubler. Initializing snakes. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 658–663, 1994.
- [Nil80] N.J. Nilsson. *Principles of Artificial Intelligence*. Morgan Kauffman, Palo Alto, CA, 1980.
- [OS88] S. Osher and J.A. Sethian. Fronts propagating with curvature dependent speed: Algorithms based on hamilton-jacobi formulation. *Journal of Computational Physics*, 79:12–49, 1988.
- [OS97] S.D. Olabarriaga and A.W.M Smeulders. Setting the mind for intelligent interactive segmentation: Overview, requirements, and framework. *Lecture Notes on Computer Science*, 1230:417–422, 1997.
- [PPGC84] D. Pope, D. Parker, D. Gustafson, and P. Clayton. Dynamic search algorithms in left ventricular border recognition and analysis of coronary arteries. *Computers in Cardiology. IEEE*, 9:71–75, 1984.
- [QT96] Hong Qin and D. Terzopoulos. D-nurbs: A physics-based framework for geometric design. *IEEE Trans. on Pattern Analysis and Machine Intel.*, 2(1):85–96, 1996.
- [TK95] Huseyin Tek and Benjamin B. Kinia. Volumetric segmentation of medical images by three-dimensional bubbles. *Brown University*, June 1995.
- [UH91] J. K. Udupa and G. Herman. *3D Imaging in Medicine*. CRC Press, Boca Raton, FL, 1991.