

T-SNAKES DUAIS E INICIALIZAÇÃO DE MODELOS DEFORMÁVEIS

Gilson Antonio Giraldi

TESE SUBMETIDA AO CORPO DOCENTE DA COORDENAÇÃO DOS PROGRAMAS DE PÓS-GRADUAÇÃO DE ENGENHARIA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE DOUTOR EM CIÊNCIAS EM ENGENHARIA DE SISTEMAS E COMPUTAÇÃO.

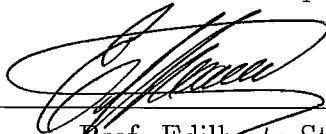
Aprovada por:



Prof. Antonio Alberto Fernandes de Oliveira, D.Sc.



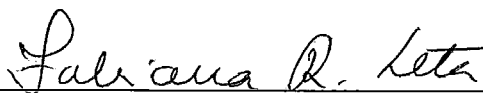
Prof. Cláudio Esperança, Ph.D.



Prof. Edilberto Strauss, Ph.D.



Prof. Aura Conci, Ph.D.



Prof. Fabiana Rodrigues Leta, Ph.D.

RIO DE JANEIRO, RJ - BRASIL
NOVEMBRO DE 2000

GIRALDI, GILSON ANTONIO

T-Snakes Duais e Inicialização de Modelo Deformáveis [Rio de Janeiro] 2000

XII, 104 p., 29,7 cm, (COPPE/UFRJ, D.Sc., Engenharia de Sistemas e Computação, 1999)

Tese – Universidade Federal do Rio de Janeiro, COPPE

1 – Modelos de Contorno Ativo

2 – Snakes Duais

3 – Snakes Topológicas - T-Snakes

4 – Inicialização de Modelos Deformáveis

I. COPPE/UFRJ II. Título (série)

Dedicatória:

À Thereza, Gabriel e André.

Agradecimentos:

Aos professores, colegas e ex-colegas do LCG: Cláudio Esperança, Ronaldo Marinho, José Brito, José Maria, Fernando Wagner, Antonio Lopes, Paulo Sérgio, Italo Matias, Mara Rios, Nelma Ribeiro, Walter, Rogério, Luiz Marcos e outros, pelo apoio e ajuda que tive de todos durante estes últimos anos,

A todos os Professores que tive, desde quem me ensinou a ler e escrever, até meu orientador, Professor Antonio Oliveira, pela formação profissional e humana que me passaram.

A minha família, particularmente meu Pai, Claudio Giraldi, e minha Mãe, Maria Jovina, pela vida que me deram.

A meus Tios Therezinha e Jose Marotto, que me acolheram no momento mais difícil de minha vida.

Ao CNPQ, pelo apoio financeiro.

Finalmente, agradeço À Deus por tudo.

Resumo da Tese apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Doutor em Ciências (D.Sc.)

T-SNAKES DUAIS E INICIALIZAÇÃO DE MODELOS DEFORMÁVEIS

Gilson Antonio Giraldi

Novembro/2000

Orientador: Antonio Alberto Fernandes Oliveira

Programa: Engenharia de Sistemas e Computação

Neste trabalho exploramos o modelo das T-Snakes para propor um *Modelo Dual de Contornos Ativos Generalizado*: um *contorno externo* envolvendo os objetos de interesse contrai e se divide enquanto que *contornos internos* aos objetos se expandem em direção às bordas procuradas.

Como resultado, conseguimos uma nova metodologia que é dual e topológica ao mesmo tempo, daí denominá-la *Dual-T-Snakes*, ou *T-Snakes Duais*.

A inicialização é um ponto crucial para a eficiência de modelos deformáveis em geral. Tendo em vista esta questão, desenvolvemos um novo método para inicializar modelos deformáveis em geral baseado em propriedades da topologia dos objetos de interesse e da escala espacial dos mesmos. O método tem como base os elementos das T-Snakes (Triangulação, Função Característica, e modelo discreto de snakes). As aplicações abordadas serão constituídas por exemplos em imagens médicas obtidas por microscopia (citologia), ressonância magnética nuclear e imagens de raio X.

Abstract of Thesis presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Doctor of Science (D.Sc.)

DUAL T-SNAKES AND INITIALIZATION OF DEFORMABLE MODELS

Gilson Antonio Giraldi

November/2000

Advisor: Antonio Alberto Fernandes Oliveira

Department: Computing and Systems Engineering

In this thesis we propose a *generalized dual active contour framework*: one T-Snake contracts and splits from outside the target(s) and the other one(s) expand(s) from inside during the process of seeking the objects boundaries in a scene.

This new approach - Dual-T-Snakes - maintains the topological capabilities of the T-Snakes and the ability of avoiding local minima of the Dual ACM.

Also we develop in this work a method to initialize deformable models which is based on properties related to the topology and spatial scale of the objects in 2D or 3D scenes. We assume some topological and scale properties for the objects of interest. From these constraints we propose a method which first defines a triangulation of the image domain. After that, we take a subsampling of the image field over the triangulation grid nodes. This field is thresholded, generating a binary one namely "Object Characteristic Function", from which a rough approximation of the boundary geometry is extracted.

Sumário

1	Introdução	1
2	Detecção de Pontos de Borda	12
2.1	Descrição Matemática	14
2.2	Bordas no Espaço de Escalas	17
3	Modelo Original de Contorno Ativo	25
3.1	Modelo Original de Snakes	26
3.1.1	Energia da Snake	27
3.2	Generalização do Modelo Original via Mecânica Lagrangeana	30
3.2.1	Princípio de Hamilton	31
3.2.2	Formulação Geral das Snakes	32
3.2.3	Discretização do Modelo	35
3.2.4	Análise Numérica	36
3.3	Limitações do Modelo Original	39
3.4	Modelo de Balloon	40
3.4.1	Instabilidade devido a Forças de Imagem	41
3.4.2	Discretização no Espaço	42
3.4.3	Inicialização	42
3.4.4	Equações do Balloon	44
4	Métodos Matemáticos	46
4.1	Programação Dinâmica	46
4.1.1	Programação Dinâmica no caso Discreto	47
4.2	Mudanças Topológicas	51
4.3	Fundamentos Matemáticos das T-Snakes	51
4.3.1	Decomposição do Domínio	52
4.3.2	Aproximação Linear por Partes	53
4.4	Teoria de Invariância	55
5	Modelos Alternativos de Interesse	58
5.1	Modelos Usando Invariância	58

5.1.1	Modelo Dual	59
5.1.2	Metodologia do Dual	63
5.1.3	AI-Snakes: Modelo de Forma Ativo	63
5.2	Modelo Dual Baseado em Programação Dinâmica	71
5.3	Métodos envolvendo mudanças Topológicas	72
5.3.1	Método dos Conjuntos de Níveis	72
5.3.2	T-Snakes	76
6	Dual-T-Snakes	85
6.1	Descrição do Dual-T-Snakes	86
6.2	Metodologia para Segmentação	90
6.3	Vantagens do Dual-T-Snakes	91
6.4	Desvantagens do Dual-T-Snakes	92
7	Implementação do Dual-T-Snakes	94
7.1	Escolha de Parâmetros	94
7.2	Passo Máximo	96
7.3	Força Externa e Força Normal	97
7.4	Suavização e Forças Internas	100
7.5	Resolução da Triangulação	102
7.6	Condição de Entropia	103
7.7	Pré-Processamento da Imagem	105
7.8	Evitando Mínimos Locais	106
7.9	Detalhes da Implementação do Dual-T-Snakes	107
8	Inicialização de Modelos Deformáveis	111
8.1	Trabalhos Relacionados	111
8.2	Método Multi-resolução	113
8.3	Método para Inicialização de Snakes	114
9	Resultados Experimentais	119
9.1	Imagens Sintéticas	120
9.1.1	Circunferência em Ruído	120
9.1.2	Variando Ruído	124
9.1.3	Background com Ruído e Artefatos	131
9.1.4	Mudanças Topológicas	131
9.1.5	Conexões Fuzzy e Dual-T-Snakes	145
9.2	Imagens Médicas	149
9.2.1	Ressonância Magnética da Cabeça	149
9.2.2	Célula do Sangue	152
9.2.3	Imagens com Várias Células	155

9.2.4	Imagens de Micrografia Eletrônica	160
9.2.5	Micrografia Eletrônica de Célula de Gato	164
9.2.6	Homem Visível	167
9.2.7	Raio-X do Tórax	169
10	Conclusões e Perspectivas	172
10.1	Perspectivas Futuras	174
A	Programação Dinâmica no Contínuo	177
B	Grupos de Lie e Invariância	181
B.1	Grupos de Lie	182
B.1.1	Espaços Topológicos	182
B.1.2	Variedades Diferenciáveis	182
B.1.3	Definição de Grupo	183
B.1.4	Definição de Grupos de Lie	184
B.1.5	Grupo de Transformações Locais	184
B.1.6	Órbitas	185
B.1.7	Campos Vetoriais em Variedades	185
B.1.8	Fluxos e Geradores Infinitesimais	187
B.1.9	Teoria de Invariantes	188
B.2	Análise Dimensional	190
B.3	Aplicações	193
C	Variedades Lineares por Partes e T-Superfícies	199
C.1	Definições Básicas	199
C.2	Aproximando Variedades Diferenciáveis Definidas Implicitamente	200
C.3	T-Superfícies	202
D	Análise de Convexidade para Snakes	205
D.1	Análise de Convexidade do Modelo Original	207
D.1.1	Análise de Convexidade	208
D.1.2	Hessiana da Energia Externa	209
D.2	Snakes em Dimensão Finita	212
D.2.1	Análise Numérica	214
D.2.2	Análise Qualitativa	215
D.2.3	Convexidade e Atratores	219
D.3	Discussão	221

E Método da Conexões Fuzzy	222
E.1 Subconjuntos Fuzzy	222
E.2 Método da Conexões Fuzzy	223
E.3 Algoritmo	225
E.4 Dual-T-Snakes e Conexões Fuzzy	226
E.5 Função de Afinidade	227
Referências Bibliográficas	227

Lista de Figuras

2.1	Modelos unidimensionais para pontos de borda.	12
2.2	Sinal irregular f com ruído.	13
2.3	(a)Imagem original. (b)Realce das bordas via limiar para o gradiente.	14
2.4	(a)Degrau da figura 2.1.a suavizado.(b)Primeira derivada.(c)Segunda derivada.(d)Terceira derivada	19
2.5	Tracking no espaço de escalas.	21
2.6	Célula de Gato: (a)Imagem de Células original.(b)Imagem suavizada por gaussiana.	21
2.7	(a)Célula de Gato: (a)Imagem de Células original.(b)Imagem obtida por difusão anisotrópica.	23
3.1	(a)Imagem original. (b)Imagem filtrada.	25
3.2	(a)Força descontínua provocando oscilações. (b)Interpolação evita instabilidades melhorando convergência.	42
3.3	Gradiente não nulo apenas em uma vizinhança do ponto de borda.	43
3.4	(a)Segunda derivada. (b)Derivada de quarta ordem	44
4.1	Curvas limitando região contendo borda.	49
4.2	Discretização da região de interesse.	49
5.1	Modelo local de forma.	59
5.2	Perfil de energia.	61
5.3	Configurações possíveis para vértices vizinhos.	67
5.4	Comparação entre a snake e o protótipo alinhado.	69
5.5	Programação Dinâmica para contornos fechados.	72
5.6	Método dos Conjuntos de Níveis.	73
5.7	Snake projetada sobre uma triangulação do tipo Coxeter-Freudenthal.	77
5.8	Merge entre duas curvas. Nós marcados são aqueles onde χ tem valor 1.	77
5.9	Polígono contendo nó interior (queimado).	80
5.10	Subespaços usados para classificação de nós.	81
5.11	Possíveis casos para classificação.	82
5.12	Triangulação do tipo J_1	84

6.1	Inicialização do Dual-T-Snake.	87
6.2	Proximidade entre duas snakes medida pela triangulação.	88
6.3	Split da Dual Snakes inicial.	89
6.4	Dual-T-Snake result for figure 6.1. A resolução da grade é 5×5 e os snaxels são os pontos em branco.	90
6.5	Evolução de curva na direção normal pode causar auto-intersecções.	92
7.1	Forças internas para as Snakes do Dual-T-Snakes.	95
7.2	Forças internas com parâmetros positivos (esquerda) e negativos (direita).	95
7.3	Com passo muito alto, a snake pode avançar o degrau perdendo a borda.	97
7.4	(a)Forças de imagem e normal para as T-Snakes. (b)Forças de imagem e normal para o Balloon.	97
7.5	Snake antes e depois da reparametrização.	98
7.6	Comportamento da força normal sobre a snake podendo gerar oscilações.	99
7.7	As forças internas nulas devido alinhamento e distribuição uniforme de snaxels.	100
7.8	Distribuição não-uniforme de snaxels devido reparametrização.	102
7.9	Snaxels muito próximos após reparametrização.	104
7.10	Snake avançou para região onde função característica não muda de sinal.	104
7.11	Imagem contendo objeto com interior complicado mas background simples.	105
7.12	Imagem contendo objeto imerso em background com ruído.	106
7.13	Fluxograma descrevendo as principais partes da implementação do Dual-T-Snakes.	110
8.1	Representação do esquema de multi-resolução usado.	112
8.2	(a)Imagem original. (b)Malha e imagem truncada pelo limiar.	113
8.3	(a)Função Característica. (b)Aproximação obtida.	114
8.4	(a) Célula de mielina. (b)Variedade Linear gerada. (c)Resultado final do método.	115
8.5	(a) Células de lisossomos. (b)Aproximações lineares por partes obtidas para resolução 5×5	115
8.6	(a) Escala de separação entre objetos mais fina que a escala interna. (b) Aproximações lineares por partes obtidas com resolução 5×5	116
8.7	Resultado obtido com resolução 3×3 , a partir da imagem 8.6.b	117

8.8	Resultado final dos passos (1)-(5) para as imagens das Figuras 8.5.a e 8.7. As resoluções usadas são 5×5 para o resultado (a) e 3×3 para (b).	118
9.1	Círcunferência em meio com Ruído.	121
9.2	Inicialização do Dual-T-Snakes.	121
9.3	Evolução do Dual-T-Snakes para a Figura 9.2: (a)Snake externa antes do ponto de congelamento. (b)Mínimo local da snake externa. (c)Snake externa congelada novamente. (d)Snake interna inicia movimento. (e)-(f)Posições intermediárias da snake interna.	123
9.4	Solução do Dual-T-Snakes correspondente a Figura 9.2.	124
9.5	Evolução do Dual-T-Snakes para força normal usual: (a)Interação 3. (b)Posição intermediária (c)Mínimo local da snake externa (interação 7). (d)Mínimo local da snake interna. (e)Posição intermediária. (f)Solução final.	125
9.6	Dual-T-Snakes em malha 5×5 : (a)Inicialização. (b)Interação 21. (c)Interação 31. (d)Solução final do Dual-T-Snakes.	126
9.7	(a)Solução do Viterbi para Figura 9.4.f. (b)Solução do Viterbi para Figura 9.6.d.	127
9.8	Inicialização para os exemplos desta seção.	127
9.9	Dual-T-Snakes: Solução e solução projetada na malha 5×5	129
9.10	Instante $t = 168$ do Dual-T-Snakes para força normal dada por (5.54).	130
9.11	(a)-(d) Soluções do Viterbi para os exemplos das Figuras 9.9.	132
9.12	(a) Elipse em meio com Ruído e artefatos. (b) Inicialização do Dual-T-Snakes	133
9.13	Evolução do Dual-T-Snakes para a Figura 9.12.b: (a)Interação 10. (b)Posição intermediária ($t = 30$) (c)Mínimo local da snake externa ($t = 41$).(d)Inicio do movimento da snake interna ($t = 42$). (e)Mínimo local da interna ($t = 152$). (f)Solução final.	134
9.14	Dual-T-Snakes em malha 5×5 : (a)Interação 12. (b)Posição intermediária ($t = 21$) (c)Posição em $t = 31$. (d)Mínimo local da snake interna ($t = 90$). (e)Mínimo local da interna ($t = 191$). (f)Solução final.	135
9.15	(a)-(b)Solução do Viterbi para as Figuras 9.13.f e 9.14.f, respectivamente.	135
9.16	(a) Figura contendo três objetos de interesse. (b) Detetor de bordas aplicado.	136
9.17	(a)Imagem a ser processada. (b)Malha usada (3×3). (c)Inicialização do Dual-T-Snakes: uma snake externa e uma snake interna a cada objeto.	137

9.18	Evolução do Dual-T-Snakes para a Figura 9.17: (a) Interação 15. (b) Interação 25 (snake externa congelada). (c) Interação 35 (snake interna ao retângulo congelada). (d) Mínimo local da snake interior ao objeto triangular. (e) Snake externa sofre split. (f) Dual Snakes formados após split evoluem. (g) Novo split da snake externa. (h) Temos agora três dual snakes independentes. (i) Instantes finais do método.	138
9.19	Solução do Dual-T-Snakes para a evolução da Figura 9.18.	139
9.20	Evolução do Dual-T-Snakes para Figura 9.18.a usando malha (5×5): (a) $t = 10$. (b) Pouco antes do split (interação 58). (c) Split na interação 60. (d) Novo split (interação 65). (e) Interação 98 (mínimo local da snake interna ao objeto triangular). (f) Interação 120. (g) Snake interior ao retângulo congelada. (h) Snake do objeto superior sendo processada. (i) Solução final.	139
9.21	Dual-T-Snakes: Resultado insatisfatório para objeto superior.	140
9.22	Dual-T-Snakes: Solução e sua projeção na malha 3×3	141
9.23	Dual-T-Snakes: Solução e sua projeção na malha 5×5	142
9.24	(a) Solução do Dual-T-Snake suavizada. (b) Espaço de busca correspondente.	144
9.25	(a)-(d) Soluções do Viterbi para os exemplos das Figuras 9.22.	146
9.26	Objetos imersos em meio com ruído exponencial.	147
9.27	Campo σ_0 obtido pelo método das conexões fuzzy para a Figura 9.26.	148
9.28	Solução do Dual-T-Snakes para normal definida segundo o campo σ_0 mostrado na Figure 9.27.	148
9.29	(a) Espaço de busca gerado a partir de off-sets. (b) Solução final.	148
9.30	(a) Imagem original. (b) Triangulação com resolução 5×5 e Função Característica do Objeto.	150
9.31	Inicialização e solução do Dual-T-Snakes para fatias 15, 55 e 60.	151
9.32	Solução obtida por programação dinâmica para a fatia de número 15.	152
9.33	Células do sangue.	153
9.34	Imagem da Figura 9.33 filtrada.	153
9.35	(a) Inicialização do Dual-T-Snakes. (b) Solução final do Dual-T-Snakes.	154
9.36	Solução final do Viterbi correspondente à Figura 9.33.	154
9.37	Células da coluna cervical.	155
9.38	Aproximações lineares obtidas para malha 55 e limiar $T = 150.0$ aplicado sobre a imagem da Figura 9.37, sem usar limites para área.	156
9.39	Resultado obtido para malha 1×1 e limiar $T = 150.0$, usando limites inferior e superior para area dados por 75.0 e 500.0, respectivamente.	156

9.40	Resultado obtido para células da coluna usando malha 1×1 e limites inferior e superior para área dados por 75.0 e 500.0, respectivamente.	157
9.41	(a)Células de tecido animal (porco). (b)Aplicação do limiar $T = 105.0$ para a intensidade de imagem. (c)Variedade lineares por partes obtidas por (1)-(4). (d) Solução final.	158
9.42	Resultado da Figura 9.41.d e malha 5×5	158
9.43	Imagem contendo objeto amorfo.	159
9.44	Inicialização obtida para a Figura 9.43 para malha 9×9	159
9.45	(a) Inicialização para malha com resolução da imagem. (b) Curva inicial e Offset.	160
9.46	(a) Espaço de busca definido pela solução inicial e seu offset. (b) Resultado do Viterbi.	161
9.47	Estrutura interior (<i>nucléolos</i>) de uma célula.	162
9.48	Segmentação da imagem da Figura 9.47: (a)Solução do Dual-T-Snakes para malha 15×15 . (b) Solução correspondente do Viterbi.	162
9.49	(a) Offsets para inicializar Dual-T-Snakes. (b) Solução do Dual-T-Snakes para malha 5×5 . (b)	163
9.50	Solução final para imagem 9.47.	163
9.51	Imagem original.	164
9.52	(a)Célula de Gato: Imagem filtrada por suavização direcional. (b) Resultado da aplicação do limiar $T = 80.0$	165
9.53	Célula de Gato: (a)Inicialização do Dual-T-Snakes. (b) Solução parcial do Dual-T-Snakes.	165
9.54	Célula de Gato: (a)Crescimento de região baseado na triangulação. (b) Re-inicializando o Dual-T-Snakes para a célula da esquerda.	166
9.55	Solução final do Dual-T-Snakes juntamente com crescimento de região para células de Gato.	167
9.56	Solução final para células de Gato.	167
9.57	(a)-(b) Imagens do Projeto Homem Visível.	168
9.58	Homem Visível: (a)Variedades lineares obtidas. (b) Resultado obtido.	168
9.59	Homem Visível: (a)Variedades lineares obtidas. (b) Inicialização final para Figura 9.57.b.	169
9.60	Radiografia do tórax.	170
9.61	Radiografia do tórax: (a)Imagem filtrada. (b) Resultado da aplicação do limiar $T = 180.0$	171
9.62	Radiografia do tórax: (a)Inicialização do Dual-T-Snakes. (b) Espaço de busca obtido.	171
B.1	Esquema com principais etapas para aplicação da Teoria de Lie em visão computacional.	195

C.1	Subdividindo um cubo (célula) para obter triangulação tipo CF.	202
C.2	Polítopos para triangulação tipo CF.	202
C.3	Formação dos espaços usados para algoritmo de classificação em 3D. . .	203
C.4	(a)Elemento triangular se moveu para nova posição queimando nó da triangulação. (b)Poliedro formado pelo elemento triangular na posições nova e antiga. (c) Partição do espaço.	204
D.1	Sistema de coordenadas locais no centro de curvatura da isolinha de potencial passando por um ponto q	210
D.2	Sistemas dinâmicos hiperbólico 3D.	218
D.3	Foco instável.	218

Capítulo 1

Introdução

Desde sua introdução por Kass et al. (Kass *et al.*, 1988), os Modelos de Contorno Ativo (*Snakes*) vêm se estabelecendo como um importante recurso para a solução de problemas em processamento de imagens digitais, particularmente para segmentação de imagens e *tracking* (Black & Yuille, 1993).

A segmentação é um passo fundamental para a análise e identificação das características relevantes em uma imagem (Jain, 1989). Segmentar uma imagem significa separá-la em regiões com propriedades *comuns* (cor, textura, etc) as quais correspondem a objetos ou partes de objetos na imagem, ou mesmo ao *background* (Jones & Metaxas, June, 1998).

De maneira geral, os métodos para segmentação fazem uso de limiares (*thresholds*) e hipóteses iniciais para a imagem tais como tipos de texturas, distribuição do histograma, etc (Zhu & Yuille, 1996). Grosseiramente, estes métodos podem ser classificados em quatro grupos (Zhu & Yuille, 1996): Filtragem local (Canny, 1983), contornos ativos (Black & Yuille, 1993), crescimento e agrupamento de regiões (Adams & Bischof, 1994) e otimização de funções de energia definidas globalmente sobre a imagem (Bayes/Minimum Description Length (MDL) (Mumford & Shah, 1989)).

Os métodos baseados em filtragem local são particularmente úteis para o realce de características de interesse (bordas, por exemplo (Canny, 1983)) e para a detecção de regiões com uma mesma textura, como é o caso típico do filtro gaussiano (Jain, 1989).

Nos métodos que usam crescimento e agrupamento de regiões procura-se dividir o domínio da imagem em subregiões, cada uma destas satisfazendo algum critério de *homogeneidade*. Os objetos *segmentados* por estas técnicas apresentam em geral bordas irregulares podendo ocorrer pequenos buracos no interior dos mesmos os quais deverão ser preenchidos em uma etapa de pós-processamento.

As técnicas envolvendo critérios globais (Bayes/MDL (Zhu & Yuille, 1996; Mumford & Shah, 1989)) procuram segmentar a imagem usando funções de energia definidas globalmente sobre a mesma. Neste sentido, têm a vantagem de tratar a imagem como um todo, mas, por outro lado, apresentam a dificuldade extra de

encontrar o mínimo global que fornece a solução desejada.

Os Modelos de Contornos Ativos, por sua vez, na sua formulação original proposta por Kass et al. (Kass *et al.*, 1988), fazem uso de informações locais sobre um contorno, podendo incorporar também informações a priori sobre a imagem como um todo ou sobre os segmentos procurados. São métodos usados para extração das bordas dos objetos da cena, sendo em geral aplicados conjuntamente com técnicas de filtragem usadas na detecção de pontos de bordas (Canny, 1983; Cohen, 1991). Uma dificuldade com estes métodos é a necessidade de uma boa estimativa inicial do contorno procurado para garantir o bom funcionamento do método.

Apesar disto, esses modelos têm a vantagem de facilitar a incorporação de informações conhecidas a priori sobre os contornos (topologia, suavidade, etc) bem como facilitar a interação com o usuário quando esta se torna necessária. Estas vantagens vêm motivando a aplicação destes modelos em diferentes contextos (Black & Yuille, 1993) bem como sua agregação a outras técnicas gerando modelos mistos, como pode ser verificado em (Zhu & Yuille, 1996).

As técnicas de contornos ativos (*snakes*) constituem o enfoque deste trabalho. Além da extração de bordas dos objetos de interesse, as *snakes* também são utilizadas para *tracking* de objetos em seqüências de imagens (Leymarie & Levine, 1993; Peterfreund, 1999; Black & Yuille, 1993), embora esta última aplicação não seja tema deste trabalho.

Snakes são um tipo particular de *modelos deformáveis* (Black & Yuille, 1993). Em geral, o termo modelo deformável em processamento de imagens refere-se a uma curva/superfície inicial (*template*) a qual se deforma sob a ação de "forças" internas (intrínsecas à geometria da curva) e externas (derivadas da imagem) atingindo o equilíbrio sobre a fronteira do objeto.

Neste trabalho, daremos enfoque especial às *snakes* em 2D, embora a extensão destes métodos para superfícies pode (em geral) ser feita sem grande dificuldade (Black & Yuille, 1993; Cohen & Cohen, 1993).

Na versão clássica (Kass *et al.*, 1988) do método de *snakes* o *template* deformável é uma curva contínua, parametrizada, e o sistema de forças (internas e externas) provem das equações de Euler-Lagrange associadas a um funcional de energia definido sobre a curva, o qual é basicamente constituído por dois termos: um termo de energia interna que incorpora restrições geométricas estabelecidas a priori para a solução (suavidade, por exemplo) e um termo de energia externa definido a partir de elementos de interesse na imagem (pontos de bordas, texturas, etc). O objetivo é que o sistema de forças possa "puxar" a curva (*snake*) em direção a uma borda ficando em equilíbrio (estável) sobre a mesma (mínimo da energia).

As equações de Euler-Lagrange para as *snakes* não possuem em geral solução analítica exigindo a utilização de métodos numéricos para obter uma solução apro-

ximada que tenha um grau suficiente de precisão. Para isto, podem ser empregados os métodos de Diferenças Finitas (Leymarie & Levine, 1993) ou de Elementos Finitos (Cohen & Cohen, 1993). O primeiro, tem a vantagem de ser simples enquanto que o segundo pode ser mais robusto.

Os modelos de snake onde a curva é representada parametricamente são denominados *modelos paramétricos* tendo como ponto de partida comum o trabalho de Kass et al. (Kass et al. , 1988). Mais adiante discutiremos os *modelos implícitos* (ver (W.J. Niessen & Viergever, 1998) para uma revisão do assunto), que constituem uma alternativa ao modelo original.

O modelo de Kass et al. (Kass et al. , 1988) possui várias limitações.

Primeiramente, a energia é em geral não convexa implicando numa alta sensibilidade à inicialização. Relacionado com este problema está também a força normal determinada a partir das equações de Euler-Lagrange. Esta força atua no sentido de colapsar contornos fechados em um ponto ou contornos abertos em um segmento de reta o que em geral é um efeito indesejado pela dificuldade de controle desta força devido sua dependência com a geometria da curva (Gunn & Nixon, 1997).

A não-convexidade da energia está diretamente relacionada com a escolha dos parâmetros do modelo. Estes parâmetros vão determinar a importância relativa de cada termo do funcional de energia. A alta sensibilidade do resultado com relação aos parâmetros se constitui em uma dificuldade inerente ao modelo de snakes (Leymarie & Levine, 1993; Gunn & Nixon, 1997), uma vez que não existe um procedimento sistemático para determiná-los.

Outro problema do modelo original é sua limitação à topologia de um segmento de reta ou de uma circunferência não sendo possível sua variação durante o processo através de operações de *split/merge*.

Neste trabalho faremos uma abordagem dos métodos de contorno ativo vinculada às soluções empregadas para estas limitações do modelo original.

Neste sentido, a não-convexidade da energia das snakes tem lugar central nesta apresentação. Este tema é analisado com rigor por Prince et al. (Davatzikos & Prince, 1999) no contexto das snakes abertas. Nesta referência, os autores partem de uma versão discreta do modelo original e analisam a energia correspondente. A energia se torna então uma função de $N - 2$ pontos, onde N é o número de pontos da curva discretizada (contando os pontos extremos fixos) e assim tem-se um problema de otimização não-linear. Neste caso, a análise de convexidade reduz-se a análise dos autovalores de sua Hessiana. Procura-se então estabelecer condições para que estes autovalores sejam estritamente positivos (energia estritamente convexa).

Uma forma de contornar o problema da não convexidade é imergindo o modelo original em um contexto dinâmico, onde a curva possui também uma aceleração que lhe permite abandonar o domínio de atração de um mínimo local (Black & Yuille,

1993). Esta generalização do método pode ser obtida diretamente via formulação Lagrangeana (Goldstein, 1981). Apesar da vantagem mencionada acima, para esta técnica ser realmente eficiente para a extração de bordas, necessita-se de um delicado *trade-off* entre os parâmetros envolvidos o que se constitui em uma dificuldade extra (Giraldi & Oliveira, 1999).

Lembramos que a formulação Lagrangeana tem a vantagem de ser independente do sistema de coordenadas utilizado. Ela é usada em diferentes trabalhos dos quais citamos em particular (Curwen & Blake, 1993; Black & Yuille, 1993) onde é tratado o problema do *tracking* de objetos.

Outra possibilidade para diminuir os efeitos da não convexidade da energia é usar métodos multi-escala. Basicamente, o modelo é resolvido para uma resolução mais baixa (escala grossa) da imagem e sua solução nesta resolução é usada como condição inicial para a escala seguinte (Leymarie & Levine, 1993). À medida que o método avança para escalas mais finas espera-se ficar mais próximo do mínimo (global) desejado. Esta metodologia tem o custo computacional extra de se montar a representação multi-escala da imagem. Além disso, caso a seqüência de escalas utilizada não seja adequada a convergência para o contorno procurado pode não ocorrer.

O problema da não-convexidade da energia está diretamente relacionado com a força normal interna que surge naturalmente das equações de Euler-Lagrange. Para diminuir os efeitos indesejáveis desta força Cohen propôs em (Cohen, 1991) a adição de uma força interna (normal) extra, juntamente com uma normalização da força externa (força de imagem).

Gang Xu et al. (Xu *et al.* , 1994) sugerem também a adição de forças internas extras mas de uma forma mais sistemática. Neste caso, a curva é aproximada localmente pelo círculo definido pelo centro e raio de curvatura e a partir desta aproximação é obtida uma relação envolvendo os parâmetros das forças internas. Esta relação fornece uma estimativa eficiente para a força normal interna. Este método tem a vantagem adicional de reduzir a sensibilidade do modelo aos parâmetros internos o que alivia o problema da escolha de parâmetros. Apesar disto, o método torna a curva menos sensível a contornos mal definidos (*subjetivos*). Além disso, tem o custo adicional da determinação da aproximação local da curva e sua extensão para o caso 3D não é óbvia.

Outra maneira de remover os efeitos indesejados da força normal de contração do modelo de snakes é através do conceito de *invariância* por transformações, o qual é bem conhecido em visão computacional (Gool *et al.* , 1995).

Esta idéia vem sendo aplicada em snakes no contexto de contornos fechados e consiste primeiramente na utilização de uma energia interna que possua propriedades adicionais de invariância além das usuais por translação e rotação.

Esta energia interna pode ser obtida através de um modelo de forma vinculado a um *protótipo* (Ip & Shen, 1998) ou simplesmente derivado de elementos que tenham as propriedades de invariância desejadas (Ip & Shen, 1998; Gunn & Nixon, 1997). Com isto, evita-se o colapso da curva, uma vez que a snake tem agora a tendência natural de adquirir a forma pré-estabelecida no modelo. Isto inclusive limita a maneira como a snake se deforma durante sua evolução.

Em (Ip & Shen, 1998) a invariância é por transformações afins sobre a curva (translação e transformações lineares não-singulares). Neste método é desenvolvida uma noção de curvatura afim-invariante empregada como elemento de suavização da solução e uma energia de modelo que vincula a forma da snake a um *protótipo*. Para este vínculo ser implementado é necessário uma correspondência entre os pontos do protótipo e da snake, o que aliás é uma característica comum dos *modelos de forma ativa*, tais como (Ip & Shen, 1998).

Este método é apropriado para segmentação de cenas onde objetos com formas previamente conhecidas foram deformados por transformações afins ou que possam ser aproximadas por uma seqüência de transformações deste tipo (projeções, por exemplo).

O método Dual (Gunn & Nixon, 1997) é outro exemplo do emprego do conceito de invariância para snakes. Este método consiste basicamente de um contorno que expande a partir do interior do objeto e de um outro contorno que contrai a partir do exterior. É estabelecida uma correspondência entre os dois contornos o que permite que os mesmos sejam interligados para obter-se uma força adicional (*driving force*) a qual é usada para retirar o contorno de mínimos locais garantindo mais robustez.

A necessidade de correspondência entre os pontos das snakes impõe restrições à forma das mesmas o que obriga a utilização de um modelo de forma. mas neste caso sem a necessidade de vínculo a um protótipo. Este modelo de forma implica naturalmente na invariância por transformações de escala uma vez que a escala (tamanho) dos objetos de interesse não é supostamente conhecida a priori.

O método dual tem a vantagem de evitar mínimos locais pela ação da força adicional, constituindo-se em uma alternativa eficiente para o problema da não convexidade da energia da snake.

Em (Gunn & Nixon, 1997) encontramos a formulação do modelo dual baseada em uma equação de evolução (método variacional). Em (Gunn, 1996a) encontramos uma outra formulação baseada em programação dinâmica. Estes modelos serão discutidos em detalhes durante este trabalho.

A utilização de programação dinâmica no contexto de snakes é apresentada em Amini et al. (Amini *et al.*, 1990) e se constitui em uma alternativa ao método variacional (Equações de Euler-Lagrange).

Classicamente, a idéia da programação dinâmica é usar o *Princípio de Optima-*

lidade (Dreyfus, 1965) para escrever o problema de otimização original como um problema de busca num espaço de curvas admissíveis e a partir daí desenvolver as equações básicas correspondentes (Dreyfus, 1965). Se pressuposto um modelo discreto para as curvas este problema de busca corresponderá a um processo multi-estágio onde o número de estágios é igual ao número de pontos da snake (*snaxels*)(Amini *et al.* , 1990).

As principais vantagens da programação dinâmica com relação aos métodos variacionais são a segurança de atingir o mínimo global e a robustez uma vez que neste caso não precisamos calcular derivadas de ordem superior da imagem (Amini *et al.* , 1990). Contudo, os métodos usando programação dinâmica são limitados pela necessidade de armazenamento em memória da ordem de $O(NM^2)$ e complexidade computacional da ordem de $O(NM^3)$ onde N é o número de snaxels e M é o tamanho da vizinhança em torno de cada snaxel, totalizando um espaço de busca discreto com NM pontos para cada interação.

Dentre os métodos de snakes que usam programação dinâmica aqueles que utilizam o algoritmo *Viterbi* (Gunn, 1996a; Bamford & Lovell, 1995) são de particular interesse para este trabalho. A idéia central neste caso é limitar a borda procurada por duas curvas, uma interior e outra exterior ao objeto correspondente, para então criar um espaço de busca que seja mais eficiente. Este método tem a vantagem de diminuir o esforço computacional, pela redução do espaço de busca agora limitado à região entre as curvas. No entanto, esta vantagem só se torna efetiva se pudermos determinar de forma automática os contornos internos e externos. Este é o ponto que limita em muito o campo de aplicação desta metodologia.

As limitações topológicas do modelo original são tratadas no contexto dos modelos paramétricos basicamente por procedimentos compostos por duas etapas: 1) determinação da necessidade de mudanças topológicas; 2) um método (*cirúrgico*) para efetuá-las.

Por mudanças topológica entendemos o fato de uma snake (fechada) se subdividir em duas (ou mais), ou várias snakes se juntarem em uma única.

Em (Durikovic *et al.* , 1995) encontramos um método deste tipo usado para dividir um contorno fechado em dois outros. Nesta referência, isto é implementado primeiramente construindo-se um histograma da intensidade da força de imagem ao longo da snake com o objetivo de identificar uma região apropriada para cortar a snake (região com contraste mais fraco). A seguir, o método procura dois pontos nesta região para serem os extremos do segmento que cortará a curva em duas partes. Este método tem a desvantagem de não tratar o problema do merge entre contornos e sua extensão para superfícies não é óbvia.

Uma metodologia mais geral para incorporar mudanças topológicas às snakes é o modelo das T-snakes (McInerney & Terzopoulos, 1995; McInerney, 1997).

A idéia básica deste método é imergir o modelo de snakes em uma decomposição simplicial (triangulação) do domínio de interesse permitindo a utilização de resultados em métodos de continuação numérica para a execução eficiente e robusta de mudanças topológicas (Allgower & Georg, 1990).

Neste método, os snaxels são determinados a partir das intersecções da snake com a triangulação. Este processo corresponde a uma reparametrização da curva (snake) baseada na subdivisão do espaço no qual a mesma está imersa. Tal procedimento (extrínseco) tem a desvantagem de perturbar a snake (particularmente nas proximidades das bordas desejadas) e tornar o método muito sensível à triangulação usada. Esta reparametrização tem também a desvantagem de tornar a energia interna da snake não-invariante por rotação e translação da curva, uma vez que sua discretização depende da sua posição no espaço.

Estas limitações da T-snake forçam que a evolução da mesma seja definida via informações estritamente locais, não envolvendo nenhum processo de minimização explícito (McInerney, 1997).

Apesar das dificuldades citadas, uma vez projetada a snake, é possível distinguir eficientemente os *nós* (vértices dos triângulos) internos e externos à mesma criando-se uma *função característica* que tem valor 1 para nós interiores e 0 para os exteriores. Esta função é o elemento central do método uma vez que dispondo de um algoritmo suficientemente rápido para atualizá-la, as mudanças topológicas da(s) snake(s) se reduzem a encontrar os triângulos de borda correspondentes (triângulos onde a função característica muda seu sinal). Este procedimento simples para a implementação de mudanças topológicas é o grande benefício obtido a partir da reparametrização descrita acima (McInerney, 1997).

Dentre as vantagens deste método estão a sua generalidade, permitindo tanto agrupamento quanto particionamento da(s) snake(s), e a possibilidade de estendê-lo naturalmente para superfícies uma vez que seus elementos básicos (triangulação, reparametrização, função característica) são eficientemente estendíveis ao caso 3D (McInerney, 1997). O método das T-snake será objeto de particular atenção neste trabalho.

As T-snakes tem como um dos métodos inspiradores as formulações implícitas do modelo de snakes, particularmente o modelo dos Conjuntos de Níveis (*Level-Sets*) devido a Sethian et al. (Malladi *et al.*, 1995).

Nesta formulação, a snake é definida pela intersecção de uma superfície bidimensional Ψ com o plano xy (*level-set*). Desta forma, a evolução da snake está vinculada à evolução da superfície (e vice-versa) o que fica precisamente determinado pelas equações do tipo *Hamilton – Jacobi* para a função $\Psi(x, y, t)$.

Neste contexto, as mudanças topológicas ocorrem naturalmente uma vez que a intersecção da superfície com um plano pode mudar a sua topologia de forma

contínua e suave desde que $\Psi(x, y, t)$ permaneça uma função durante sua evolução. Além das habilidades topológicas, acrescenta-se o fato dos métodos numéricos utilizados para evoluir $\Psi(x, y, t)$ terem boa estabilidade e da extensão para superfícies bidimensionais ser direta (Malladi *et al.*, 1995).

No entanto, o uso explícito de uma dimensão extra torna este método menos intuitivo que os modelos paramétricos, trazendo dificuldades para a incorporação de informações adicionais (de forma, por exemplo), ou vínculos tais como pontos atratores (Kass *et al.*, 1988), dificultando inclusive a interação com o usuário caso esta se torne necessária.

Neste ponto, o método das T-Snakes é mais vantajoso uma vez que sua formulação paramétrica facilita o estabelecimento de *vínculos* pelo usuário.

No método das T-Snakes usa-se uma força normal para guiar a snake em direção às bordas desejadas. O sentido desta força é definido em função de um limiar para a intensidade da imagem. Uma forma de evitar mínimos locais é variar este limiar dentro de um intervalo estabelecido por estatísticas da imagem. Além disto, o método de reparametrização usado permite identificar facilmente quando duas curvas estão próximas. Estas observações foram o ponto de partida para a idéia de colocar o método Dual e as T-Snakes em um modelo unificado, por nós denominado Dual-T-Snakes (Giraldi & Oliveira, 2000; Giraldi *et al.*, 2000d; Giraldi *et al.*, 2000c).

Resumidamente, a metodologia consiste em explorar o modelo das T-Snakes para propor um *Modelo Dual de Contornos Ativos Generalizado: um contorno externo, envolvendo os objetos de interesse, contrai e se divide enquanto que contornos internos aos objetos se expandem em direção às bordas procuradas.*

Como resultado, conseguimos uma nova metodologia que é dual e topológica ao mesmo tempo, daí denominá-la *Dual-T-Snakes*.

Para fazer a snake externa contrair e as snakes internas expandirem em direção às bordas dos objetos nós usamos também forças normais do tipo daquela usada na T-Snake.

No Dual-T-Snakes, o limiar usado para definir o sentido da força normal é interpretado como uma noção de *afinidade* a qual é usada para identificar quais snaxels podem receber força extra e assim retirar uma snake de um mínimo local de forma eficiente.

A força extra tem direção normal, não dependendo portando de correspondência entre os snaxels, como é o caso do Dual original (Gunn & Nixon, 1997). As auto-intersecções que podem ocorrer em função da evolução das curvas segundo sua normais são naturalmente tratadas pela metodologia das T-Snakes.

Para que o método seja eficiente é necessário definir elementos que tornem a evolução da snake externa e das internas interdependente. Neste trabalho isto é

implementado usando-se a *energia de imagem (normalizada) sobre as snakes* e a noção de *afinidade* citada acima.

Do ponto de vista da metodologia Dual apresentada originalmente em (Gunn & Nixon, 1997), a primeira vantagem do Dual-T-Snakes a ser ressaltada é que este não necessita de "matching" entre as snakes internas e externas. Assim, podemos prescindir de um modelo de forma a priori e deixar a forma das snakes livre durante a evolução do método. Acrescentando-se a isto as habilidades topológicas do Dual-T-Snakes (herança das T-Snake), pode-se dizer que o novo método generaliza o Dual original, tanto para formas mais gerais quanto para topologias mais gerais.

Além disso, o Dual-T-Snakes abre a possibilidade efetiva de estender a metodologia dual para 3D, via as T-Superfícies uma vez que não precisamos de correspondência entre os snaxels ou mesmo de um modelo de forma pré-estabelecido.

Do ponto de vista das T-Snakes, o Dual-T-Snakes trouxe efetivamente mais robustez quanto aos mínimos locais da energia de imagem pelo balanço da energia entre as snakes internas e externas e a noção de proximidade entre as curvas dada pela triangulação. Agora, o critério de parada baseado na proximidade (local) entre as snakes é um indicador mais seguro de que o mínimo global foi atingido. Para retirar uma T-Snake de um mínimo local podemos usar tanto força normal extra como aplicar o método de variação do limiar T proposto inicialmente nas T-Snakes (McInerney, 1997; McInerney & Terzopoulos, 1999).

Um derivado interessante do Dual-T-Snakes é a redução do espaço de busca, podendo assim ser utilizado para inicializar o Dual baseado em programação dinâmica, vencendo com isto a dificuldade com a inicialização deste último, discutida acima.

Apesar das vantagens citadas, o Dual-T-Snakes tem uma desvantagem clara com relação à inicialização, uma vez que precisamos de uma snake interna para cada objeto de interesse, além da snake externa envolvendo os mesmos. Se feito pelo usuário, este processo pode ser tedioso, principalmente se temos vários objetos de interesse. Além de evitar esta dificuldade inicial, um método para inicialização automática poderia também diminuir o custo computacional do Dual-T-Snakes.

A inicialização é de fato um ponto crucial para a eficiência de modelos deformáveis em geral.

Tendo em vista estas questões, nós desenvolvemos um novo método para inicializar modelos deformáveis em geral baseado em propriedades da topologia dos objetos de interesse e da escala espacial dos mesmos. O método tem como base os elementos das T-Snakes (Triangulação, Função Característica, e modelo discreto de snakes) e assim compõe com o Dual-T-Snakes uma metodologia unificada para segmentação e extração de bordas.

A filosofia do método segue de perto a idéia básica de alguns métodos não-paramétricos de multi-resolução usados em segmentação de imagens baseados em

modelos de Pirâmides e Quadrees (Burt *et al.* , 1981; Bamford & Lovell, n.d.; Jolion & Montanvert, 1992). Estes são estruturas de dados próprias para representar uma imagem em diferentes níveis de resolução.

A idéia básica destes métodos é que à medida que a resolução vai decrescendo pequenos artefatos no background tornam-se menos significantes em relação aos objetos de interesse. Assim, pode ser mais simples *detectar* os objetos no nível mais baixo para então fazer o tracking nos diferentes níveis de resolução.

Neste processo, é possível *delinear* as bordas em uma resolução mais baixa e reestimá-las quando aumentamos a resolução (Jolion & Montanvert, 1992; Bamford & Lovell, n.d.).

Nosso trabalho descrito em (Giraldi *et al.* , 2000e; Giraldi *et al.* , 2000a) usa uma metodologia que parte de uma resolução mais baixa, definida através de um conhecimento a priori da escala dos objetos. O método é adaptativo no sentido que a resolução somente é aumentada nas regiões onde for considerado necessário.

Para aumentar a resolução, nós simplesmente refinamos a malha mais grossa e amostramos a imagem sobre os nós correspondentes à nova malha. Assim, a estrutura subjacente tem a forma de um refinamento adaptativo (*local uniform nested refinement*) (Berger & Olinger, 1984). Um esquema semelhante foi explorado no contexto dos Métodos de Conjuntos de Níveis (Sethian, 1996). Contudo, nós não conhecemos referência de aplicação deste esquema para segmentação.

Uma limitação deste método de inicialização é a necessidade de uma *propriedade de escala local*. Esta propriedade basicamente garante que podemos reduzir a resolução da imagem sem mudar a topologia dos objetos. Contudo, existem situações onde os padrões de intensidade dos objetos de interesse não admitem tal propriedade.

Uma alternativa para estes casos seria definir um campo escalar sobre a imagem o qual caracterizasse melhor os objetos de interesse (ou suas bordas). Além disso, seria interessante que a definição deste novo campo pudesse aproveitar a intervenção do usuário na inicialização do Dual-T-Snakes.

Um método que vem ao encontro destas necessidades é o método das conexões fuzzy (*fuzzy connectedness*) (Udupa & Samarasekera, 1996).

Neste método, parte-se do argumento de que a noção de "objeto" em imagens digitais deve ser formalmente definida via conceitos *fuzzy* a fim de tratar a subjetividade inerente ao ato de *delinear* a borda de um objeto em uma cena (Udupa & Samarasekera, 1996; Herman & Carvalho, n.d.). O método é implementado através de um operador de afinidade que precisa de pontos sementes para ser definido. Estes pontos serão usados para definir as snakes internas inicializadas pelo usuário no Dual-T-Snakes.

O resultado do método das conexões fuzzy é um campo escalar que pode ser

usado na definição dos objetos. Este campo será usado para determinar o sentido da força normal no Dual-T-Snakes. Temos assim uma perfeita simbiose entre o método de extração de bordas e o método de segmentação fuzzy, o que motivou a utilização deste último neste trabalho.

Nos capítulos seguintes, discutiremos com precisão as limitações apontadas inicialmente para em seguida apresentar os modelos alternativos de interesse para este trabalho chegando por fim ao Dual-T-Snakes. Daremos atenção especial aos modelo das T-snakes (McInerney, 1997) e ao modelo Dual (Gunn, 1996a), os quais constituem os pilares do nosso trabalho. O conceito de invariância também receberá particular atenção. O texto que segue está organizado da seguinte forma.

O Capítulo 2 apresenta uma caracterização de pontos de borda baseada em conceitos de geometria Riemanniana, a qual pode ser usada para imagens multivariadas (coloridas, por exemplo). Em seguida discutimos esta caracterização no contexto dos espaços de escala. O objetivo deste Capítulo é introduzir os elementos básicos envolvidos na definição da energia externa das snakes.

Partindo dos elementos introduzidos no Capítulo 2, apresentamos no Capítulo 3 o modelo original das snakes e sua generalização via Formulação Lagrangeana. São discutidos também aspectos numéricos do mesmo, bem como suas limitações básicas.

No capítulo 4 apresentamos os elementos matemáticos usados para resolver as limitações explicitadas no Capítulo 3: Programação Dinâmica, Invariância e Variedades Implícitas.

No Capítulo 5 faremos uso dos elementos discutidos até então para apresentar as os modelos alternativos de interesse.

Em seguida, apresentaremos no Capítulo 6 o Dual-T-Snakes, principal contribuição deste trabalho. O método para inicialização de modelos deformáveis que desenvolvemos é apresentado no Capítulos 8.

Reservamos o Capítulo 7 para discutir detalhes da implementação do Dual-T-Snakes. Nosso objetivo com isto é facilitar a discussão dos resultados experimentais feita no Capítulos 9. Finalizamos apresentando propostas para trabalhos futuros e conclusões no Capítulos 10.

Os Apêndices A, B, C e D são complementares deste trabalho e sua leitura não é pré-requisito para a compreensão do mesmo.

O Apêndice A desenvolve aspectos da programação dinâmica no contínuo completando assim a discussão da seção 4.1 que trata do tema apenas no caso discreto.

O Apêndice B desenvolve aspectos da Teoria de Invariância no contexto dos Grupos de Lie. Finalizamos este apêndice indicando possíveis direções para explorar a Teoria de Lie no contexto de modelos de snake envolvendo invariância por transformações, tais como (Ip & Shen, 1998).

O Apêndice C generaliza os conceitos algébricos da seção 4.2 para variedades de dimensão qualquer, estabelecendo assim a base teórica para a generalização do Dual-T-Snakes para 3D.

Os conceitos relativos à análise de convexidade do método de snakes são detalhados no Apêndice D, onde estendemos o estudo apresentado em (Davatzikos & Prince, 1999) para modelos dinâmicos de snakes. Os resultados apresentados são parte do nosso trabalho (Giraldi & Oliveira, 1999) e também constituem uma contribuição desta tese.

O método das conexões fuzzy é descrito no Apêndice E e sua conexão com o Dual-T-Snakes é desenvolvida.

Capítulo 2

Detecção de Pontos de Borda

Neste trabalho, estamos interessados na extração de bordas de objetos em imagens digitais. Neste sentido, o problema inicial é a determinação dos pontos que podem pertencer às bordas (genericamente denominados *pontos de bordas (edges)*), os quais são caracterizados por uma abrupta variação da imagem (Jain, 1989) nas suas vizinhanças.

Neste capítulo faremos uma abordagem conceitual e formal para o problema de detecção de pontos de borda em imagens bidimensionais sem o objetivo de desenvolver técnicas específicas no tema. A apresentação que segue tem por finalidade principal colocar o problema da detecção de pontos de borda dentro do contexto das técnicas de snakes.

Seja então a Figura 2.1 onde mostramos os modelos (unidimensionais) usuais para os tipos de pontos de borda comumente encontrados em processamento de imagem (Canny, 1983).

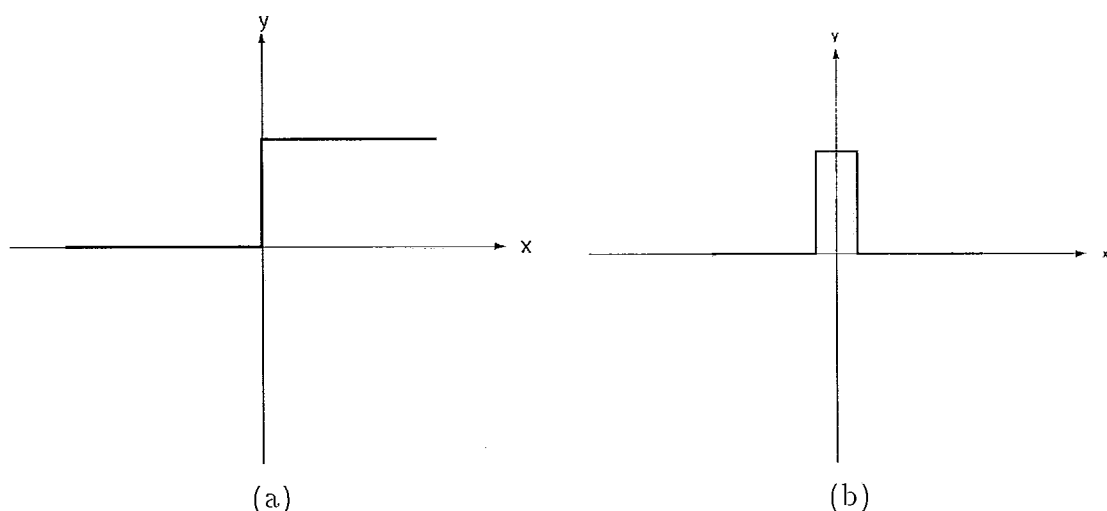


Figura 2.1: Modelos unidimensionais para pontos de borda.

Estas figuras representam modelos ideais. As funções (sinais) que obtemos na

prática são em geral irregulares e apresentam ruídos.

A Figura 2.2 é um exemplo de uma situação mais próxima da prática. Nesta figura vemos um sinal com um degrau acentuado entre $x = 90$ e $x = 100$.

Uma possibilidade para caracterizar este comportamento é tomar os extremos locais de df/dx , ou seja, pontos de inflexão.

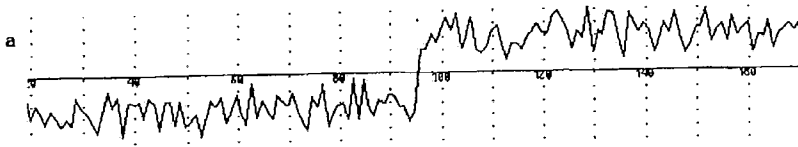


Figura 2.2: Sinal irregular f com ruído.

No entanto, na Figura 2.2 existem vários pontos de inflexão que não representam degraus. Assim, métodos que simplesmente localizam extremos locais da derivada fornecem falsos pontos de borda, inclusive nas proximidades do degrau procurado, o que implica em mais de uma resposta para um mesmo ponto.

Estas considerações se estendem para o caso 2D. Neste caso, dada uma imagem, precisamos considerar também uma orientação além da taxa de variação da imagem em um dado ponto.

Uma referência fundamental no contexto de detecção dos pontos de borda é o trabalho de Canny (Canny, 1983). Partindo de considerações como as que fizemos acima, Canny estabelece inicialmente três critérios para avaliação do desempenho de um detetor: (a) baixa probabilidade de marcar *falsos* pontos de borda ou de perder pontos de borda; (b) os pontos detectados devem estar próximos do centro da borda (ponto de inflexão no caso 1D) correspondente; (c) fornecer resposta única para cada ponto de borda da imagem. Em seguida, estes critérios são desenvolvidos quantitativamente para enfim obter-se um detetor ótimo via métodos variacionais.

Segundo Canny (Canny, 1983), podemos classificar os métodos para detecção de pontos de bordas de acordo com as seguintes características: a) utilização de uma superfície que aproxima a imagem; b) métodos que estimam derivadas, usualmente primeira ou segunda derivadas da imagem; c) utilização de técnicas no domínio da frequência.

Dentre estes métodos, aqueles do item (b) serão os enfocados nesta seção pela sua relação direta com o método de snakes (Black & Yuille, 1993).

Uma prática comum neste caso, é utilizar um limiar de corte para o valor da derivada (gradiente) com o objetivo apenas de *realçar* as regiões onde se encontram as bordas dos objetos de interesse (Jain, 1989), como mostra a figura 2.3. O mesmo

efeito pode ser obtido via *zero-crossings* da segunda derivada (máximos locais da primeira derivada) (Jain, 1989).

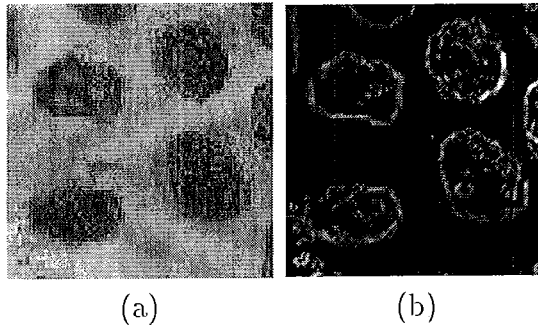


Figura 2.3: (a)Imagem original. (b)Realce das bordas via limiar para o gradiente.

A seguir, daremos uma formulação geral para a caracterização de pontos de borda baseada nas derivadas da imagem, ou mais precisamente, na Primeira Forma Fundamental associada à mesma.

2.1 Descrição Matemática

Partindo da idéia de que pontos de borda são pontos com alta variação da imagem (*alto contraste*), o primeiro passo será aplicar um modelo matemático que traduza este conceito quantitativamente.

Uma possibilidade neste sentido é usar a Primeira Forma Fundamental (Sapiro, 1997) associada à imagem e usar o fato de que os autovalores (autovetores) da mesma em um dado ponto fornecem as taxas (direções) de máxima e mínima variação da imagem no ponto, respectivamente (Dubrovin *et al.*, 1984).

Seja então

$$\Phi(u_1, u_2) : D \subset \mathbb{R}^2 \rightarrow \mathbb{R}^m, \quad (2.1)$$

uma imagem multi-variada com m componentes:

$$\Phi_i(u_1, u_2) : D \subset \mathbb{R}^2 \rightarrow \mathbb{R}, i = 1, \dots, m. \quad (2.2)$$

Como exemplo de imagens multi-variadas podemos citar imagens coloridas codificadas num sistema usual como o *RGB* ou *HLS* e imagens multi-canal obtidas por satélites. Sejam agora dois pontos $P = (u_1^0, u_2^0)$ e $Q = (u_1^1, u_2^1)$ e a diferença:

$$\Delta\Phi = \Phi(P) - \Phi(Q).$$

da imagem calculada nestes pontos.

Se a distância $d(P, Q)$ for pequena, esta diferença pode ser aproximada pela diferencial

$$d\Phi = \sum_{i=1}^{i=2} \frac{\partial\Phi}{\partial u^i} du^i,$$

O vetor $d\Phi$ tem dimensão m e sua norma Euclideana quadrada é dada por:

$$d\Phi^2 = \sum_{i=1}^{i=2} \sum_{j=1}^{j=2} \left\langle \frac{\partial\Phi}{\partial u^i}, \frac{\partial\Phi}{\partial u^j} \right\rangle du_i du_j. \quad (2.3)$$

onde \langle, \rangle indica o produto interno usual. Esta forma quadrática é denominada a Primeira Forma Fundamental da superfície correspondente (Dubrovin *et al.*, 1984).

Relembrando conceitos fundamentais em geometria Riemanniana, ressaltamos primeiramente que os coeficientes g_{ij} dados por:

$$g_{ij} = \left\langle \frac{\partial\Phi}{\partial u^i}, \frac{\partial\Phi}{\partial u^j} \right\rangle,$$

formam uma matriz simétrica $[g_{ij}]$, definida-positiva, a qual fornece uma métrica (tensor simétrico de segunda ordem) sobre a superfície correspondente à imagem. Assim podemos calcular normas de vetores tangentes e distância entre pontos sobre a superfície sem qualquer referência ao espaço ambiente (Dubrovin *et al.*, 1984).

Usando a notação padrão em geometria Riemanniana, nós denotamos:

$$d\Phi^2 = \sum_{i=1}^{i=2} \sum_{j=1}^{j=2} g_{ij} du_i du_j = \begin{bmatrix} du_1 \\ du_2 \end{bmatrix}^T \begin{bmatrix} g_{11} & g_{12} \\ g_{21} & g_{22} \end{bmatrix} \begin{bmatrix} du_1 \\ du_2 \end{bmatrix}. \quad (2.4)$$

Para um vetor unitário $\vec{v} = (\cos\theta, \text{sen}\theta)$, $d\Phi^2(\vec{v})$ indica a taxa de variação da imagem na direção de \vec{v} , a qual é dada por:

$$d\Phi^2(\vec{v}) = \begin{bmatrix} \cos\theta \\ \text{sen}\theta \end{bmatrix}^T \begin{bmatrix} g_{11} & g_{12} \\ g_{21} & g_{22} \end{bmatrix} \begin{bmatrix} \cos\theta \\ \text{sen}\theta \end{bmatrix} \quad (2.5)$$

É um fato conhecido (Dubrovin *et al.*, 1984) que os extremos da forma quadrática (2.4) são obtidos nas direções dos autovetores do tensor métrico $[g_{ij}]$, e os valores máximo e mínimo correspondentes são os autovalores associados. Os autovalores são dados por:

$$\lambda_{\pm} = \frac{g_{11} + g_{22} \pm \sqrt{(g_{11} - g_{22})^2 + 4g_{12}^2}}{2}. \quad (2.6)$$

Os autovetores associados (ortogonais entre si) serão denotados por $(\cos \theta_{\pm}, \text{sen} \theta_{\pm})$ com as direções θ_{\pm} dados respectivamente pelas seguintes expressões:

$$\theta_+ = \frac{1}{2} \arctan \left(\frac{2g_{12}}{g_{11} - g_{22}} \right), \quad (2.7)$$

$$\theta_- = \theta_+ + \frac{\pi}{2}. \quad (2.8)$$

Observe que λ_{\pm} são valores não negativos uma vez que $[g_{ij}]$ é positiva definida. O autovetor θ_+ fornece a direção de máxima variação da imagem em um ponto e o autovetor θ_- fornece a direção de mínima variação da imagem neste mesmo ponto. Os autovalores correspondentes fornecem as respectivas taxas de máxima e mínima variação no ponto dado.

Por exemplo, para uma imagem em níveis de cinza ($m = 1$) teremos:

$$\lambda_+ = \|\nabla\Phi\|^2, \lambda_- = 0, \text{ onde } (\cos \theta_+, \text{sen} \theta_+) \parallel \nabla\Phi, \quad (2.9)$$

recuperando assim a caracterização usual de pontos de borda para imagens deste tipo (Sapiro, 1997).

Contudo, se $m > 1$, o valor de λ_- pode não ser identicamente nulo. Uma primeira caracterização para pontos de borda neste caso, análoga a selecionar $\|\nabla\Phi\|^2$ no caso de $m = 1$, seria uma função do tipo $f = f(\lambda_+ - \lambda_-)$. Por exemplo, possíveis escolhas para f seriam (Sapiro, 1997):

$$\begin{aligned} f &= (\lambda_+ - \lambda_-)^{1/p}, \quad p > 0 \\ &\text{ou,} \\ f &= \sqrt{\lambda_+}. \end{aligned}$$

Analogamente ao caso $m = 1$, os pontos de borda seriam aqueles para os quais $|f(\lambda_+ - \lambda_-)|$, ultrapasse um limiar previamente estabelecido (Sapiro, 1997).

Esta caracterização de pontos de borda tem a vantagem de agregar as informações locais da imagem em duas únicas quantidades λ_{\pm} dadas pela expressão (2.6) facilitando sua aplicação. Além disso, uma vez que os coeficientes g_{ij} constituem um tensor, esta caracterização de pontos de borda é independente do sistema de coordenadas (isto é, da representação de cor) usado (Sapiro, 1997; Sapiro, 1995).

Embora simples e geral, esta metodologia tem limitações decorrentes da instabilidade numérica que pode ocorrer no cálculo de derivadas para imagens com ruídos (Terzopoulos, 1986b; Poggio *et al.*, 1985; Snyder, 1991).

A seguir analisaremos o problema da instabilidade numérica. No próximo capítulo veremos o método de snakes como uma metodologia para a extração das informações geométricas de interesse do conjunto de pontos detectado.

2.2 Bordas no Espaço de Escalas

No texto que segue, diremos que uma função $f : \mathfrak{R}^m \rightarrow \mathfrak{R}$; $f = f(u^{i_1}, u^{i_2}, \dots, u^{i_m})$ é de classe C^n se suas derivadas parciais:

$$\frac{\partial^n}{\partial u^{i_1} \partial u^{i_2} \dots \partial u^{i_m}}; \quad i_1 + i_2 + \dots + i_m = n, \quad i_1, i_2, \dots, i_m \in N, \quad (2.10)$$

são contínuas. Diremos ainda que uma função f é *suave* caso f seja de classe C^∞ e usaremos a notação:

$$N_n^m = \{(i_1, i_2, \dots, i_m) \mid i_1 + i_2 + \dots + i_m = n, \quad i_1, i_2, \dots, i_m \in N\}$$

O cálculo das derivadas que surgem em (2.3) sofre em geral de instabilidade numérica devido ruídos na imagem original (Terzopoulos, 1986b; Dolgoplova & Ivanov, 1966). Uma solução bem conhecida para este problema é filtrar (convoluir) a imagem original com um kernel s , satisfazendo as seguintes propriedades (MA & Li, 1998):

1. $s \in C^n$,
2. $s(u_1, u_2)$ é uma função par de u_1 e u_2 . Além disso:

$$\int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} s(u_1, u_2) du_1 du_2 = 1,$$

3. Para qualquer $(i, j) \in N_k^2 (k = 1, \dots, n)$

$$\int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} |u_1|^i |u_2|^j s(u_1, u_2) du_1 du_2$$

existe e é finita.

Por simplicidade, vamos supor $m = 1$ em (2.1), e assim teremos $s : \mathfrak{R}^2 \rightarrow \mathfrak{R}$ para o kernel acima.

Assim, dado um operador diferencial qualquer \mathcal{D} , aplicamos este operador a uma versão *suavizada* $\tilde{\Phi}$ de Φ dada pela convolução $\Phi * s$:

$$\tilde{\Phi} = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \Phi(v, w) s(u_1 - v, u_2 - w) dv dw, \quad (2.11)$$

obtendo-se assim:

$$\mathcal{D}\tilde{\Phi} = \mathcal{D} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \Phi(v, w) s(u_1 - v, u_2 - w) dv dw, \quad (2.12)$$

Fazendo uso das propriedades diferenciais do kernel s podemos derivar sob o sinal de integração obtendo finalmente:

$$\mathcal{D}\tilde{\Phi} = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \Phi(v, w) [\mathcal{D}s(u_1 - v, u_2 - w)] dv dw. \quad (2.13)$$

Por exemplo, se:

$$\mathcal{D} = D_n = \sum_{(i,j) \in \mathbb{N}_n^2} b_{ij} \frac{\partial^n}{\partial u_1^i \partial u_2^j}, \quad (2.14)$$

teremos a expressão (2.13) correspondente dada por:

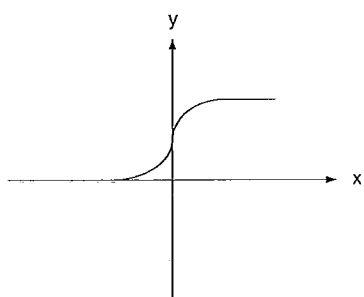
$$D_n \tilde{\Phi} = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \Phi(v, w) \left[\sum_{(i,j) \in \mathbb{N}_n^2} b_{ij} \frac{\partial^n}{\partial u_1^i \partial u_2^j} s(u_1 - v, u_2 - w) \right] dv dw. \quad (2.15)$$

Para o caso 1D representado na Figura 2.1, mostramos a seguir as funções obtidas por (2.15) para $D_n = d^n/dx^n$, $n = 0, 1, 2, 3$, onde D_0 denota o operador identidade. O kernel s pode ser a gaussiana definida logo a seguir em (2.17).

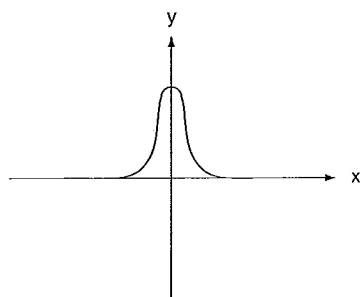
É importante observar que o processo de filtragem dado pela convolução $\Phi * s$ tem implicitamente as seguintes propriedades:

- (1) Linearidade: a convolução da soma é a soma das convoluções;
- (2) Invariância por Deslocamento (*Spatial shift invariance*): o peso de s na convolução (2.11) depende apenas do vetor diferença $(u_1 - v, u_2 - w)$.

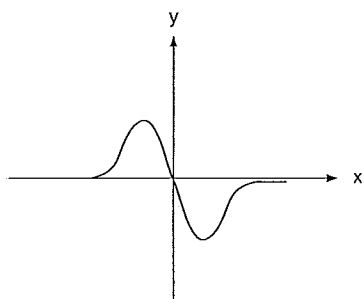
Se a estas propriedades acrescentarmos a imposição de que s tenha simetria circular (*isotropia*) bem como as Leis de Invariância por Escala da Análise Dimensional (Teorema dos π , Apêndice B) então chegamos ao resultado notável de que o *Kernel*



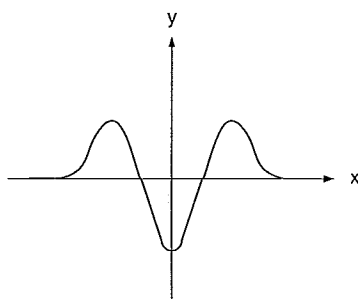
(a)



(b)



(c)



(d)

Figura 2.4: (a) Degrau da figura 2.1.a suavizado. (b) Primeira derivada. (c) Segunda derivada. (d) Terceira derivada.

Gaussiano é o único operador no espaço de escalas (Florack *et al.*, 1992). Seu desvio padrão σ pode ser identificado com escala espacial (Florack *et al.*, 1992).

A importância do kernel Gaussiano é corroborada por indícios biológicos de que a funcionalidade da retina humana pode ser parcialmente modelada por filtros Gaussianos de várias larguras ou por suas derivadas (Florack *et al.*, 1992).

A utilização do Kernel Gaussiano foi o ponto de partida dos métodos de representação multi-escala de imagens (Witkin, 1983). A idéia central destes métodos (Koenderink, 1984) se resume em imergir a imagem original em uma família de imagens derivadas $I(x, y, t)$ obtidas pela convolução da imagem original $I_0(x, y)$ com o Kernel Gaussiano:

$$I(x, y, t) = G_t * I_0, \quad (2.16)$$

onde:

$$G_\sigma(x, y) = C\sigma^{-1} \exp\left(-\frac{1}{4\sigma} (x^2 + y^2)\right) \quad (2.17)$$

é a função de Gauss.

Como indicado por Koenderink (Koenderink, 1984) e Hummel (Hummel, 1986) esta família de imagens fornece a solução fundamental da equação (do calor), tomando a imagem original como condição inicial; ou seja (Junior & Iorio, 1988):

$$\frac{\partial I(x, y, t)}{\partial t} = c\nabla^2 I(x, y, t) \quad (2.18)$$

$$I(x, y, 0) = I_0(x, y)$$

onde c é uma constante (denominada constante de difusão). Koenderink desenvolveu sua formulação a partir de dois critérios (1)-(2) acima, bem como do princípio de causalidade baixo:

Princípio de Causalidade: não podem ser criados "artefatos" quando aumentamos a escala.

Estes critérios levam naturalmente à equação de difusão.

Os métodos multi-escala procuram localizar as estruturas de interesse (pontos de borda, por exemplo) em uma escala mais grossa para em seguida fazer o *tracking* destas estruturas no espaço de escalas até a resolução desejada (Witten, 1993). Na Figura 2.5 representamos esquematicamente este processo.

Nesta figura, $\sigma_4 > \sigma_3 > \dots > \sigma_0$. Identificada uma estrutura de interesse na escala σ_4 faz-se o tracking da mesma no espaço de escalas até a escala mais fina σ_0 .

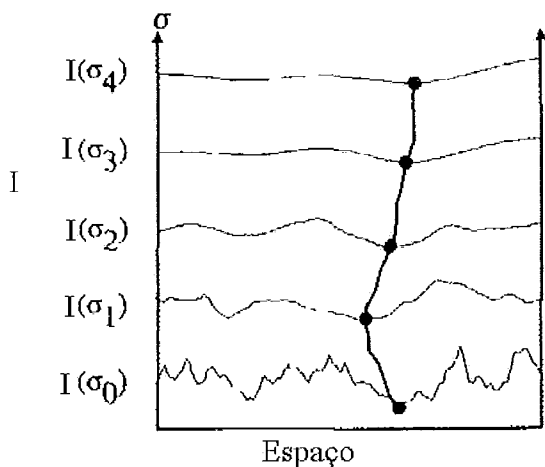
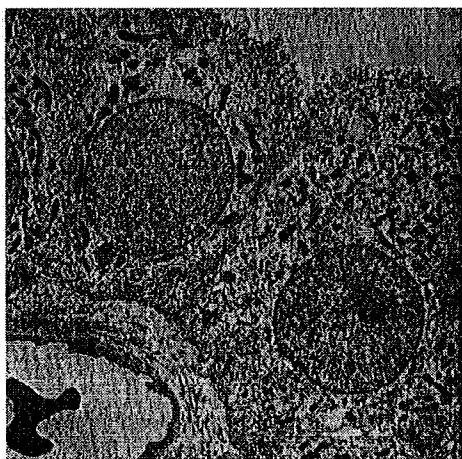


Figura 2.5: Tracking no espaço de escalas.

Esta abordagem para o espaço de escalas baseada na equação de difusão linear (2.18) apresenta o efeito mostrado na figura 2.6.a de "difundir" indiscriminadamente tanto a borda na direção do background da imagem como vice-versa. Este efeito é útil para o método de snakes discutido no próximo capítulo em função dos métodos numéricos a serem utilizados. Mas, no contexto de detecção de pontos de borda, é indesejado por aumentar a imprecisão na localização dos mesmos (Perona & Malik, 1990).



(a)



(b)

Figura 2.6: Célula de Gato: (a)Imagem de Células original.(b)Imagem suavizada por gaussiana.

Esta observação foi ponto de partida para aos métodos multi-escalas não-lineares introduzidos por (Perona & Malik, 1990). Nestes métodos, a idéia central é abandonar a linearidade pressupostas na convolução (2.11), permitindo que a constante

da equação de difusão em (2.18) possa variar no espaço e possivelmente no tempo (Alvarez *et al.*, 1992; Catta *et al.*, 1992). Tem-se agora um problema de valor inicial do tipo:

$$\frac{\partial I(x, y, t)}{\partial t} = \text{div}(c(x, y, t) \nabla I), \quad (2.19)$$

$$I(x, y, 0) = I_0(x, y), \quad (2.20)$$

onde por simplicidade vamos supor a imagem I em tons de cinza ($m = 1$ em (2.1)).

A primeira questão nesta nova abordagem é como definir $c(x, y, t)$.

Seguindo a apresentação dada em Perona-Malik (Perona & Malik, 1990), vamos supor que os pontos de borda são alinhados com o eixo y ($I_y = 0$). Assim, sem perda de generalidade, a equação de difusão não-linear (2.19) toma numa vizinhança da borda a forma:

$$\frac{\partial I(x, y, t)}{\partial t} = \frac{\partial}{\partial x}(c(x, y, t) I_x(x, y, t)). \quad (2.21)$$

Vamos escolher c como sendo uma função do gradiente de I : $c(x, y, t) = g(I_x(x, y, t))$. Definindo $\phi(I_x) \equiv g(I_x) \cdot I_x$ podemos colocar a equação acima na forma:

$$I_t = \frac{\partial I}{\partial t} = \frac{\partial}{\partial x}(\phi(I_x)) = \phi'(I_x) \cdot I_{xx}. \quad (2.22)$$

Nós estamos interessados em analisar a variação no tempo da declividade da borda: $\frac{\partial I_x}{\partial t}$. Para isto, devemos observar primeiro que se $c(x, y, t) > 0$ então a solução I de (2.19) será suave (Perona & Malik, 1990). Logo, a ordem de diferenciação poderá ser invertida; ou seja:

$$\frac{\partial}{\partial t}(I_x) = \frac{\partial}{\partial x}(I_t) = \frac{\partial}{\partial x}\left(\frac{\partial}{\partial x}\phi(I_x)\right) = \frac{\partial}{\partial x}(\phi'(I_x) \cdot I_{xx}). \quad (2.23)$$

$$\phi'' \cdot I_{xx}^2 + \phi' \cdot I_{xxx}.$$

Vamos supor que a borda seja orientada tal que $I_x > 0$ (ver Figura 2.4.a). No ponto de inflexão temos $I_{xx} = 0$, e $I_{xxx} \ll 0$ uma vez que o ponto de inflexão na Figura 2.4.a corresponde ao ponto com declividade máxima (Figura 2.4.b).

Assim, em uma vizinhança do ponto de inflexão a derivada $\partial I_x / \partial t$ tem sinal oposto a $\phi'(I_x)$. Se $\phi'(I_x) > 0$ a declividade do ponto de borda decresce com

o tempo. Caso contrario, se $\phi'(I_x) < 0$ a declividade vai aumentar com o tempo. Assim, a borda torna-se *mais fina*; ou seja, a largura do pulso na Figura 2.4.b diminui. Temos portando mais precisão na localização da borda que é o efeito desejado!

Uma extensão interessante da difusão não linear para imagens multi-variadas é usado em (Sapiro, 1995). Nesta referência usa-se a equação:

$$\frac{\partial \Phi}{\partial t} = g_{cor}(\lambda_+, \lambda_-) \frac{\partial^2 \Phi}{\partial \theta_-^2}, \quad (2.24)$$

onde λ_{\pm} e θ_- são dadas pelas expressões (2.6)-(2.8), respectivamente. A função $g_{cor}(\lambda_+, \lambda_-)$ é qualquer função decrescente da diferença $(\lambda_+ - \lambda_-)$ acrescentada com o objetivo de obter maior controle sobre a difusão local, a exemplo da função c em (2.19).

A equação (2.24) define uma difusão não linear e anisotrópica. Tal método tende a preservar as bordas uma vez que a difusão da imagem ocorre ortogonalmente à direção que as define (θ_+ em (2.7)) e, como g_{cor} é decrescente, a tendência é que a borda se torne mais fina com o tempo.

A Figura 2.7 esclarece o efeito esperado com esta definição para $c(x, y, t)$. Observe que neste caso a difusão da imagem em direção ao background (e vice-versa) é dificultada pois nos pontos de borda o autovetor θ_- em (2.8) é tangente à borda em questão. Isto dificulta a "mistura" entre objeto e background que é precisamente o efeito desejado nesta metodologia.

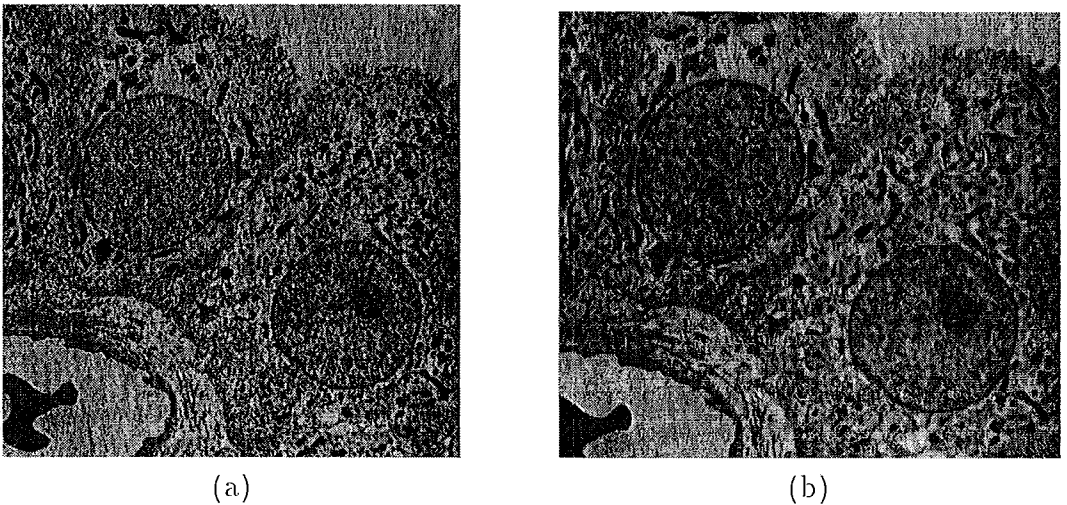


Figura 2.7: (a)Célula de Gato: (a)Imagem de Células original.(b)Imagem obtida por difusão anisotrópica.

Mais recentemente, o trabalho de (Florack *et al.*, 1995) coloca os modelos dados por (2.18) e (2.19) em uma formulação covariante através da qual define os conceitos de espaços de escala linear e não linear mais precisamente.

Do ponto de vista das snakes, a suavização Gaussiana é o procedimento comumente usado para diminuir ruídos na imagem. A difusão entre background e objetos é neste caso um efeito desejável em função dos métodos de otimização empregados. No entanto, a formulação implícita proposta em Sapiro (Sapiro, 1997) é um belo exemplo de como a difusão anisotrópica pode aparecer naturalmente também no contexto das snakes.

No próximo capítulo, nos deteremos na formulação paramétrica das snakes e na sua relação com o que foi discutido neste capítulo.

Capítulo 3

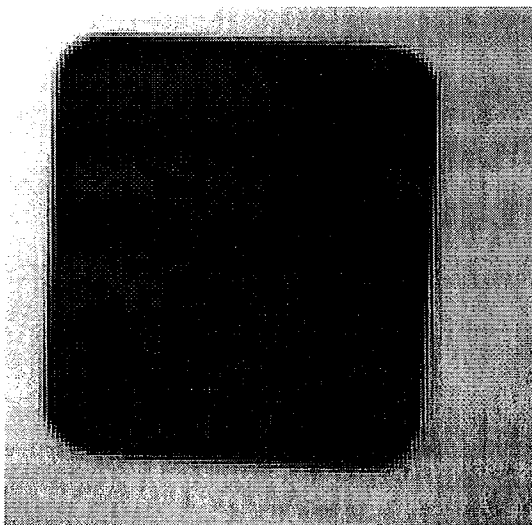
Modelo Original de Contorno Ativo

No capítulo anterior desenvolvemos uma caracterização para pontos de borda baseada na primeira forma fundamental associada à imagem.

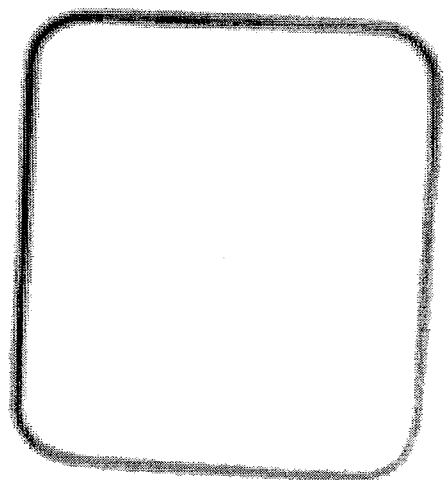
No texto que segue estaremos sempre supondo imagens em tons de cinza e assim $m = 1$ na expressão (2.1).

Pelo que foi discutido no capítulo anterior podemos realçar as bordas selecionando pontos da imagem (suavizada por uma Gaussiana) com alto gradiente. O resultado obtido nada mais é que uma filtragem passa-alta da imagem a qual realça as fronteiras dos “objetos” da cena.

A Figura 3.1.(b) a seguir mostra o resultado da aplicação desta caracterização para a imagem da Figura 3.1.(a).



(a)



(b)

Figura 3.1: (a)Imagem original. (b)Imagem filtrada.

Até este momento, não temos informação precisa da geometria das bordas dos

objetos. Por exemplo, não temos ainda resposta para a seguinte pergunta: qual a curva plana que melhor representa a borda do objeto representado na Figura 3.1.(a).

É neste contexto que o método de snakes (ou de Contorno Ativo) se insere primeiramente (Black & Yuille, 1993). Vejamos alguns detalhes do modelo original de snakes.

3.1 Modelo Original de Snakes

Como já dito na introdução, snakes são um tipo particular de modelo deformável no qual um template inicial (curva) é deformado em direção á borda desejada pela ação de forças internas e externas a ele (Kass *et al.* , 1988; Black & Yuille, 1993). O que se deseja neste processo é que a snake atinja a posição de equilíbrio apenas quando o modelo aproximar a borda desejada.

O resultado do processo fornece assim informações da forma e da posição da borda na imagem. É neste sentido que diremos que o método de snakes é uma técnica de *extração de bordas* de objetos em uma imagem.

Basicamente, este método originalmente proposto por Kass *et al.* (Kass *et al.* , 1988), parte da idéia de formular o problema da extração de bordas via a otimização de um funcional montado a partir das características de interesse (bordas, no caso) (Gunn, 1996a).

Precisamente, retomemos a definição de pontos de borda para imagens em tons de cinza:

$$\|\nabla I\|^2 > T, \tag{3.1}$$

onde T é um limiar previamente estabelecido usando-se, por exemplo, um histograma da intensidade do campo gradiente na imagem.

Observemos a Figura 3.1. A curva que melhor representa a borda desejada será aquela que melhor se ajuste aos pontos onde $\|\nabla I\|^2$ é alto. Uma maneira precisa de formular isto é dizer que estamos procurando um mínimo (global) para o funcional:

$$E_{ext}(c) = \int_c P(c(s)) ds, \tag{3.2}$$

onde:

$$P = -\|\nabla I\|^2, \tag{3.3}$$

e s é um parâmetro (não necessariamente comprimento de arco). Observemos, que o problema da extração da borda fica assim colocado como um problema de otimização

de um funcional definido em um espaço de curvas a ser estabelecido a priori (Black & Yuille, 1993).

Posto desta forma, teremos um problema numericamente instável uma vez que a otimização do funcional acima via métodos variacionais é um problema mal-condicionado (Terzopoulos, 1986b). Isto obriga a introdução de um termo de regularização (Poggio *et al.*, 1985; Snyder, 1991).

Nestas condições, o funcional de energia da snake ficará formado por um termo intrínseco, dependente apenas da geometria interna da mesma e por uma energia de imagem dada por um funcional do tipo (3.2).

Uma possibilidade para a energia da snake é dada por uma adaptação ao caso 2D do modelo de placa fina largamente usado em modelos deformáveis 3D (Black & Yuille, 1993). Foi esta a escolha feita por Kass *et al.* (Kass *et al.*, 1988) no modelo original o qual passamos a tratar agora.

3.1.1 Energia da Snake

Como dito acima, a energia da snake será composta por um termo diretamente relacionado à imagem que guarda as características de interesse da mesma e um termo de *regularização* ou de *suavização* (Poggio *et al.*, 1985; Snyder, 1991). Este último, além de garantir estabilidade numérica irá guardar características geométricas supostas a priori para a solução.

O primeiro passo para a definição precisa da energia é definir o espaço de curvas (Cohen, 1991). Seja então Ad um espaço de curvas parametrizadas por um parâmetro s , $c(s) = (x(s), y(s))$, tal que:

$$Ad = \{c : [0, 1] \rightarrow D \subset \mathbb{R}^2; c \in C^4\},$$

onde D é um domínio de interesse, podendo ser todo o domínio de definição da imagem ou apenas uma parte deste.

O grau de diferenciabilidade quatro, imposto aos elementos de Ad , é necessário uma vez que adota-se em (Kass *et al.*, 1988) a formulação variacional para o método de otimização (*Formulação Forte*). Uma outra alternativa seria utilizar a *Formulação Fraca* e então utilizar o Método dos Elementos Finitos (Cohen & Cohen, 1993) o que exigiria um menor grau de diferenciabilidade.

Em Kass *et al.* (Kass *et al.*, 1988) o funcional de energia da snake tem a forma geral dada por:

$$E(c) = E_{ext}(c) + E_{int}(c) + E_{vinculo},$$

onde E_{ext} é um funcional do tipo definido em (3.2), denominado a *energia externa* da snake:

$$E_{ext}(c) = \int_0^1 P(c(s)) ds, \quad (3.4)$$

E_{int} é o termo de regularização, denominado *energia interna*, dado por:

$$E_{int}(c) = \int_0^1 \left(\omega_1 \|c'(s)\|^2 + \omega_2 \|c''(s)\|^2 \right) ds. \quad (3.5)$$

onde ω_1 e ω_2 são parâmetros internos denominados parâmetros de elasticidade e rigidez, respectivamente, e $c'(s) = dc/ds$ e $c''(s) = d^2c/ds^2$.

O potencial externo P vai gerar um campo de forças externo à curva que deverá “puxar” a mesma em direção às bordas.

Outras possibilidades para a energia interna podem ser encontradas em (Snyder, 1991) onde encontramos uma discussão criteriosa para o problema da regularização em visão computacional.

O termo $E_{v\u00ednculo}$ corresponde a forças de vínculo que podem ser definidas a priori ou através de interação com o usuário. Não trataremos deste termo nesta discussão.

Assim, a energia da snake toma a forma:

$$E(c) = \int_0^1 \left(\omega_1 \|c'(s)\|^2 + \omega_2 \|c''(s)\|^2 \right) ds + \int_0^1 P(c(s)) ds. \quad (3.6)$$

Este funcional corresponde à energia potencial de uma corda fina imersa em um campo de forças externo definido pelo potencial P (uma variação do modelo de placa fina usado para superf\u00edcies (Black & Yuille, 1993)). Vejamos alguns detalhes deste modelo iniciando pelos par\u00e2metros do mesmo.

O efeito dos par\u00e2metros internos ω_1 e ω_2 na energia acima pode ser explicitado considerando-se as express\u00f5es para as derivadas $c'(s)$ e $c''(s)$ em (3.6). Seja ent\u00e3o o comprimento de arco α da curva c . Ent\u00e3o, as derivadas em (3.6) tomam a forma:

$$\frac{dc}{ds} = \frac{dc}{d\alpha} \frac{d\alpha}{ds}, \quad (3.7)$$

$$\frac{d^2c}{ds^2} = K \vec{N} \left(\frac{d\alpha}{ds} \right)^2 + \vec{T} \frac{d^2\alpha}{ds^2}, \quad (3.8)$$

onde \vec{T} e \vec{N} denotam o vetor tangente unit\u00e1rio e o vetor normal unit\u00e1rio, respectivamente, e K \u00e9 a curvatura. Assim, os par\u00e2metros ω_1 e ω_2 em (3.6) multiplicam, respectivamente, os seguintes termos:

$$\left\| \frac{dc}{ds} \right\|^2 = \left(\frac{d\alpha}{ds} \right)^2, \quad (3.9)$$

$$\left\| \frac{d^2c}{ds^2} \right\|^2 = \left\| K \vec{N} \left(\frac{d\alpha}{ds} \right)^2 + \vec{T} \frac{d^2\alpha}{ds^2} \right\|^2 \quad (3.10)$$

Vemos desta forma que o termo que multiplica ω_1 varia durante a deformação somente se o comprimento de arco da curva se alterar. Portanto, este termo está diretamente vinculado à elasticidade da curva (*corda*), o que justifica a denominação *parâmetro de elasticidade* para ω_1 .

O termo (3.10) inclui a curvatura e o vetor normal e, portanto, está diretamente vinculado à forma da curva. Quanto maior o fator ω_2 , mais dependente é a energia da corda de sua geometria, o que implica em maior resistência à deformação ou mais precisamente, em maior rigidez da corda. Em síntese: o coeficiente ω_1 pondera uma energia associada a uma compressão ou a um estiramento da snake enquanto que ω_2 pondera uma energia associada à curvatura.

No trabalho de Kass et al. (Kass *et al.*, 1988) usam-se as equações de Euler-Lagrange para extremizar o funcional de energia (Dubrovin *et al.*, 1984). Observe-mos que o este funcional é da forma:

$$E = \int_0^1 L(c, c', c'') ds. \quad (3.11)$$

onde:

$$c' = \frac{dc}{ds}, \quad c'' = \frac{d^2c}{ds^2}, \quad \text{e } L(c, c', c'') = \omega_1 \|c'(s)\|^2 + \omega_2 \|c''(s)\|^2 + P(c), \quad (3.12)$$

Portanto, as equações de Euler-Lagrange correspondentes têm a forma geral (Dubrovin *et al.*, 1984):

$$\frac{\partial L}{\partial x} - \frac{\partial}{\partial s} \left(\frac{\partial L}{\partial x'} \right) + \frac{\partial^2}{\partial s^2} \left(\frac{\partial^2 L}{\partial x''} \right) = 0, \quad (3.13)$$

$$\frac{\partial L}{\partial y} - \frac{\partial}{\partial s} \left(\frac{\partial L}{\partial y'} \right) + \frac{\partial^2}{\partial s^2} \left(\frac{\partial^2 L}{\partial y''} \right) = 0, \quad (3.14)$$

as quais são geralmente escritas na notação compacta:

$$\frac{\partial L}{\partial c} - \frac{\partial}{\partial s} \left(\frac{\partial L}{\partial c'} \right) + \frac{\partial^2}{\partial s^2} \left(\frac{\partial^2 L}{\partial c''} \right) = 0. \quad (3.15)$$

Para o caso da Lagrangeana L dada em (3.12) obtém-se o seguinte sistema de equações diferenciais ordinárias:

$$-2 \frac{d}{ds} \left(\omega_1 \frac{dc}{ds} \right) + 2 \frac{d^2}{ds^2} \left(\omega_2 \frac{d^2 c}{ds^2} \right) - \nabla P = 0. \quad (3.16)$$

Do ponto de vista mecânico, esta equação explicita o fato de que a *energia potencial* definida pelo funcional (3.11) terá um extremo quando as forças internas e de campo (força externa dada por ∇P) se anulam. Este equilíbrio é o ponto central da técnica e dele depende a eficiência da mesma.

Um detalhe que a expressão (3.16) acima deixa explícito é a alta dependência das forças internas com relação à forma da curva. Esta dependência pode ser crítica, particularmente nas proximidades da borda, o que motivou trabalhos como (Xu *et al.*, 1994) onde se procura substituir a componente normal das forças internas por um termo de mais fácil controle.

Outra observação importante é com relação ao potencial P dado por (3.3). Este potencial está vinculado à caracterização de pontos de borda apresentada no capítulo anterior. Outras possibilidades discutidas em Kass *et al.* (Kass *et al.*, 1988) são:

$$P_{linha} = \pm I(x, y), \text{ e } P_{zero} = (\nabla^2 (G_\sigma * I))^2, \quad (3.17)$$

onde P_{linha} geraria um campo que atrairia a snake para linhas escuras (sinal +) ou claras (sinal -) e P_{zero} atrairia a snake para zero-crossings do Laplaciano, os quais localizam os pontos de borda segundo a teoria de Marr-Hildreth (Marr & Hildreth, 1980).

É conveniente neste momento ressaltar o fato de que o método de snakes não pressupõe que a imagem I , usada na definição do potencial P em (3.3), seja o resultado de uma filtragem passa alta da image inicial. De fato, podemos usar a definição (3.3) para P com a imagem I sendo apenas uma versão suavizada da imagem original.

Vejamos agora uma generalização do modelo apresentado nesta seção.

3.2 Generalização do Modelo Original via Mecânica Lagrangeana

A idéia básica neste caso é imergir o modelo dado por (3.16) em um meio viscoso a então usar elementos da formulação Lagrangeana da mecânica clássica (Goldstein, 1981) para obter as equações de movimento para a curva (snake).

A formulação Lagrangeana é mais conveniente que a Newtoniana por conduzir a equações (Equações de Euler-Lagrange do movimento) cuja forma não depende do sistema de coordenadas usado (Goldstein, 1981). Vejamos inicialmente os elementos básicos desta teoria para, em seguida, apresentar o modelo de snakes na sua formulação mais geral.

3.2.1 Princípio de Hamilton

Nesta formulação da Mecânica Clássica, a configuração instantânea de um sistema físico é descrita pelos valores de suas coordenadas *generalizadas* q_1, q_2, \dots, q_n e corresponde a um ponto num hiperespaço Cartesiano onde os q 's formam os eixos coordenados. Este espaço $2n - \textit{dimensional}$ é denominado espaço de configuração e será denotado por \bar{E} .

À medida que o tempo t evolui, o estado do sistema muda, e portanto, o ponto que o caracteriza no hiperespaço se desloca traçando uma curva que define a evolução do sistema. Assim, a expressão *movimento do sistema* corresponde ao movimento do ponto no hiperespaço ao longo da curva correspondente a qual fica parametrizada na forma:

$$c(t) = (q_1(t), q_2(t), \dots, q_n(t)). \quad (3.18)$$

Na formulação Lagrangeana da Mecânica o comportamento de um sistema físico pode ser obtido via o *Princípio de Hamilton* (Goldstein, 1981) que estabelece o seguinte: Para sistemas mecânicos para os quais todas as forças (exceto as de vínculo) são derivadas de um potencial (generalizado), o movimento do sistema de um instante t_1 a um instante posterior t_2 é tal que a integral de linha:

$$J = \int_{t_1}^{t_2} (T - V) dt \quad (3.19)$$

onde T é a energia cinética e V a energia potencial do sistema, tem um valor extremo (máximo ou mínimo local) para a curva correspondente.

A função $T - V$ em (3.19) é denominada Lagrangeana do sistema e será denotada por L .

A extremização do funcional J leva a equações de Euler-Lagrange da forma (Goldstein, 1981):

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}_i} \right) = \frac{\partial L}{\partial q_i}, \quad i = 1, \dots, n \quad (3.20)$$

onde $\dot{q} = dq/dt$.

Neste texto, denotaremos este sistema na forma compacta:

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}} \right) = \frac{\partial L}{\partial q}, \quad (3.21)$$

onde $q = (q_1, \dots, q_n)$ é o vetor das coordenadas generalizadas do sistema.

3.2.2 Formulação Geral das Snakes

A idéia central é submeter um contorno inicial (corda fina) a uma dinâmica para a qual os extremos do funcional (3.19) sejam trajetórias no hiperespaço convergindo para os contornos de interesse.

O primeiro passo é então montar a Lagrangeana do sistema, e para tal devemos escolher coordenadas convenientes e expressar as forças que definirão a dinâmica nestas coordenadas.

Não vamos restringir a dimensão do espaço de curvas admissíveis a um valor finito. Assim, teremos um espaço de configuração com dimensão infinita do tipo Ad na seção 3.1.1 acima. Uma curva do espaço Ad é um ponto deste espaço enquanto que uma curva deformando-se no tempo é representada por uma trajetória do tipo (3.18) em Ad .

Seja então uma curva que se deforma no tempo. Seja μ a densidade linear de massa ao longo da curva e γ o coeficiente de amortecimento correspondente à ação das forças de atrito. Assim, associaremos à curva uma energia potencial E_p do tipo dado pela expressão (3.11) e uma energia cinética T que depende da velocidade; ou seja (Black & Yuille, 1993):

$$E_p(c) = \int_c \left(\omega_1 \|c'(s)\|^2 + \omega_2 \|c''(s)\|^2 \right) ds + \int_c P(c(s)) ds, \quad (3.22)$$

$$T = \mu \int_c \left\| \frac{\partial c}{\partial t} \right\|^2 ds. \quad (3.23)$$

O movimento da curva estará sujeito também a forças de atrito dadas por um potencial generalizado dependente apenas da velocidade e de γ . Em geral este potencial (U_{at}) é definido por (Goldstein, 1981):

$$\frac{\partial U_{at}}{\partial t} = -\frac{1}{2} \lambda \left\| \frac{\partial c}{\partial t} \right\|^2, \quad (3.24)$$

Então a Lagrangeana do problema pode ser escrita na forma:

$$L(c, \dot{c}, c', c'') = T - E_p - U_{at}, \quad (3.25)$$

Uma vez que temos dois parâmetros independentes s e t , o funcional J em (3.19) toma a forma:

$$J = \int_{t_1}^{t_2} \int_0^1 L ds dt \quad (3.26)$$

o que conduz às seguintes equações de Euler-Lagrange (Goldstein, 1981; Dubrovin *et al.*, 1984):

$$\frac{\partial L}{\partial x} - \frac{\partial}{\partial t} \left(\frac{\partial L}{\partial \dot{x}} \right) - \frac{\partial}{\partial s} \left(\frac{\partial L}{\partial x'} \right) + \frac{\partial}{\partial s^2} \left(\frac{\partial L}{\partial x''} \right) = 0, \quad (3.27)$$

$$\frac{\partial L}{\partial y} - \frac{\partial}{\partial t} \left(\frac{\partial L}{\partial \dot{y}} \right) - \frac{\partial}{\partial s} \left(\frac{\partial L}{\partial y'} \right) + \frac{\partial}{\partial s^2} \left(\frac{\partial L}{\partial y''} \right) = 0. \quad (3.28)$$

Usando (3.25) e retomando a notação compacta introduzida em (3.15) podemos escrever este sistema na forma:

$$\frac{\partial \tilde{L}}{\partial c} - \frac{\partial}{\partial t} \left(\frac{\partial \tilde{L}}{\partial \dot{c}} \right) - \frac{\partial}{\partial s} \left(\frac{\partial \tilde{L}}{\partial c'} \right) + \frac{\partial}{\partial s^2} \left(\frac{\partial \tilde{L}}{\partial c''} \right) = - \frac{\partial}{\partial \dot{c}} \left(\frac{\partial U_{at}}{\partial t} \right), \quad (3.29)$$

onde \tilde{L} corresponde à parte conservativa ($T - V$) de L e U_{at} à parte dissipativa da Lagrangeana do sistema, como definido em (3.24).

Ao sistema (3.29) devemos agregar condições iniciais e de contorno que garantam a unicidade da solução. Podemos também agregar condições de vínculo decorrentes por exemplo de interação com o usuário.

As condições de contorno dependem das curvas admissíveis serem abertas ou fechadas. No caso de curvas fechadas o sistema a ser resolvido pode ser da forma:

$$\mu \ddot{x} + \lambda \dot{x} - \frac{\partial}{\partial s} \left(\omega_1 \frac{\partial x}{\partial s} \right) + \frac{\partial^2}{\partial s^2} \left(\omega_2 \frac{\partial^2 x}{\partial s^2} \right) = - \frac{1}{2} \frac{\partial P}{\partial x}, \quad (3.30)$$

$$\mu \ddot{y} + \lambda \dot{y} - \frac{\partial}{\partial s} \left(\omega_1 \frac{\partial y}{\partial s} \right) + \frac{\partial^2}{\partial s^2} \left(\omega_2 \frac{\partial^2 y}{\partial s^2} \right) = - \frac{1}{2} \frac{\partial P}{\partial y}, \quad (3.31)$$

sujeito a

$$c(0, t) = c(1, t), \quad \frac{\partial c(s, 0)}{\partial t} = \vec{v}_0^{\rightarrow}(s), \quad (3.32)$$

onde a primeira condição (de contorno) explicita o fato da curva ser fechada e a segunda condição (inicial) fornece a velocidade da curva no instante inicial.

Uma propriedade útil deste novo modelo é que a snake agora tem inércia. Assim, ela pode escapar de mínimos locais o que em princípio aumenta as suas chances de chegar ao mínimo global. Contudo, a real eficiência do método depende de um delicado *trade-off* entre os parâmetros do modelo o qual dificilmente é conseguido na prática.

Em (Giraldi & Oliveira, 1999) estudamos estas dificuldades tanto qualitativa quanto quantitativamente para modelos de snake definidos em um espaços de configuração com dimensão finita. No Apêndice D desenvolvemos os elementos básicos deste trabalho.

É oportuno destacar que a substituição do espaço admissível Ad por um espaço de dimensão finita, por exemplo, através do uso de splines (Black & Yuille, 1993) ou NURBS (Qin & Terzopoulos, 1996), pode diminuir a complexidade do método gerando uma formulação elegante e com um apelo geométrico proveniente do sistema dinâmico resultante (Giraldi & Oliveira, 1999).

O sistema (3.30)-(3.32) não tem em geral solução analítica uma vez que o potencial P pode ser absolutamente genérico. Assim, torna-se necessária a utilização de métodos numéricos para obter uma aproximação da solução desejada.

Vejamos a análise dos parâmetros.

Elasticidade e Rigidez

Do ponto de vista contínuo o efeito de ω_1 e ω_2 permanece o mesmo já discutido na seção 3.1.1.

O efeito destes parâmetros no comportamento do modelo acima é analisado também em trabalhos como (Leymarie & Levine, 1993; Giraldi & Oliveira, 1999). Contudo, o desenvolvimento de procedimentos sistemáticos para estimar estes parâmetros tem recebido pouca atenção, como ressaltado por (R. Fisker, 1998).

A análise de convexidade apresentada em (Davatzikos & Prince, 1999) e reproduzida na seção D.1 é um caminho para limitar o intervalo de variação destes parâmetros numa dada aplicação.

Outra possibilidade seria permitir que estes parâmetros variem no tempo e no espaço o que pode aumentar o custo computacional do modelo (Davatzikos & Prince, 1999).

Em (Xu *et al.*, 1994) encontramos um outro método baseado na adição de um novo termo de energia tal que a força normal exercida sobre cada ponto tenha a mesma intensidade, independente da forma do contorno. O novo modelo torna-se menos sensível à escolha de parâmetros, mas por outro lado é pouco eficiente para extrair contornos subjetivos.

Neste trabalho, a metodologia assumida será considerar estes parâmetros constantes as quais podem ser atualizadas durante a evolução da snake (Leymarie & Levine, 1993).

Massa Linear e Coeficiente de Amortecimento

Estes parâmetros estão relacionados com a estabilidade numérica e a taxa de convergência, como descrito a seguir.

A situação ideal é escolher estes parâmetros de tal forma que a velocidade da snake decaia rapidamente somente quando a snake estiver próxima da borda desejada. Este comportamento é conhecido como *amortecimento crítico*, sendo bem conhecido no contexto da teoria de oscilações (Nayfeh & Mook, 1979), e significando que a parte transiente da solução contínua decai rapidamente.

No modelo descrito pelas equações (3.30)-(3.31) nós precisamos ser cuidadosos pois o mesmo é em geral não linear. Por exemplo, se simplificarmos a energia 3.4 de tal forma que a equação de movimento torne-se aquela de um oscilador harmônico forçado (Black & Yuille, 1993), nós podemos conseguir amortecimento crítico em alguns modos de vibração mas não em todos. O mesmo será verdadeiro para o modelo dado pela expressões (3.30)-(3.31), como demonstrado no Apêndice D.

Vejamos agora, o efeito destes parâmetros na convergência do método numérico.

3.2.3 Discretização do Modelo

Supondo a existência e unicidade da solução de (3.30)-(3.31), para o conjunto de condições iniciais e de contorno dadas por (3.32), vamos apresentar o método básico utilizado na literatura para discretizar o modelo de snakes (Leymarie & Levine, 1993).

A metodologia aqui é aquela padrão das técnicas para discretização de equações diferenciais por diferenças finitas. Escolhemos um passo τ no tempo e um passo h ao longo da snake. Com isto, a snake ficará discretizada em um certo número de pontos (*snaxels*) e o tempo em uma seqüência de instantes.

Dentre as possibilidades para aproximar as derivadas parciais costuma-se escolher diferenças centrais dadas por (Chapra & Canale, 1988):

$$\dot{x}(s, t - \tau) \approx \frac{1}{2\tau} [x(s, t) - x(s, t - 2\tau)], \quad (3.33)$$

$$\ddot{x}(s, t - \tau) \approx \frac{1}{\tau^2} [x(s, t) - 2x(s, t - \tau) + x(s, t - 2\tau)]. \quad (3.34)$$

$$\frac{\partial^2 x}{\partial s^2}(s, t) \approx \frac{1}{h^2} [x(s - h, t) - 2x(s, t) + x(s + h, t)]. \quad (3.35)$$

$$\frac{\partial^4 x}{\partial s^4}(s, t) \approx \frac{1}{h^4} [x(s - 2h, t) - 4x(s - h, t) + 6x(s, t) - 4x(s + h, t) + x(s + 2h, t)]. \quad (3.36)$$

Existem certos aspectos a serem considerados na escolha do tipo de diferenças finitas usado para discretizar o modelo. Por exemplo, é necessário que a matriz de rigidez correspondente seja simétrica e possua o mínimo de bandas possível (Leymarie & Levine, 1993).

Usando estas expressões, obtém-se a seguinte discretização do sistema (3.30)-(3.31):

$$\left[\left(\frac{\mu}{\tau^2} + \frac{\lambda}{2\tau} \right) I + K \right] X^n = R_X(\bullet, \bar{t} - \tau) + \left(\frac{2}{\tau^2} \right) X^{n-1} - \left(\frac{\mu}{\tau^2} I - \frac{1}{2\tau} C \right) X^{n-2}, \quad (3.37)$$

$$\left[\left(\frac{\mu}{\tau^2} + \frac{\lambda}{2\tau} \right) I + K \right] Y^n = R_Y(\bullet, \bar{t} - \tau) + \left(\frac{2}{\tau^2} \right) Y^{n-1} - \left(\frac{\mu}{\tau^2} I - \frac{1}{2\tau} C \right) Y^{n-2}, \quad (3.38)$$

onde K é a matriz simétrica proveniente da discretização dos termos envolvendo as derivadas espaciais, (X^n, Y^n) é o vetor de snaxels no instante $\bar{t} = n\tau$, e $R(\bullet, \bar{t} - \tau)$ é o campo externo sobre a snake discretizada (Leymarie & Levine, 1993).

As equações (3.37)-(3.38) podem ainda ser escritas na forma compacta abaixo:

$$\begin{aligned} AX^n &= B_X^{n-1, n-2}, \\ AY^n &= B_Y^{n-1, n-2}, \end{aligned} \quad (3.39)$$

onde :

$$A = \left[\left(\frac{\mu}{\tau^2} + \frac{\lambda}{2\tau} \right) I + K \right], \quad (3.40)$$

e

$$B_X^{n-1, n-2} = R(\bullet, \bar{t} - \tau) + \left(\frac{2}{\tau^2}\right) X^{n-1} - \left(\frac{\mu}{\tau^2} I - \frac{1}{2\tau} C\right) X^{n-2}, \quad (3.41)$$

$$B_Y^{n-1, n-2} = R(\bullet, \bar{t} - \tau) + \left(\frac{2}{\tau^2}\right) Y^{n-1} - \left(\frac{\mu}{\tau^2} I - \frac{1}{2\tau} C\right) Y^{n-2}. \quad (3.42)$$

3.2.4 Análise Numérica

Vamos agora estudar as propriedades do método de discretização acima. Para fornecer resultados coerentes, este esquema numérico deve ter as propriedades de consistência, estabilidade e convergência (Hirsch, 1988).

A consistência expressa o fato de que a equação discretizada (3.39) deve tender à equação diferencial correspondente quando $\tau \rightarrow 0$ e $h \rightarrow 0$. Esta condição é diretamente verificada pelas expressões (3.33)-(3.36).

Quanto a estabilidade do esquema numérico usado devemos ressaltar que estaremos supondo um limite superior finito para a intensidade do campo externo.

Assim, a análise da estabilidade do sistema (3.39) pode ser feita estudando-se o condicionamento da matriz A , seguindo a metodologia apresentada em (Leymarie & Levine, 1993). Se A for bem condicionada então teremos uma garantia de estabilidade para o esquema numérico da seção anterior (Golub & Loan, 1985; Hirsch, 1988).

O número de condição de uma matriz A , não singular, na norma 2, é dado por (Golub & Loan, 1985):

$$\kappa(A) = \frac{\lambda_{\max}(A)}{\lambda_{\min}(A)}, \quad (3.43)$$

onde $\lambda_{\min}(A)$ e $\lambda_{\max}(A)$ denotam os autovalores de A com maior e menor valor absoluto, respectivamente. Pela análise de sensibilidade de sistemas lineares sabemos que quanto mais próximo $\kappa(A)$ estiver da unidade mais bem condicionada será a matriz A . Conseqüentemente, o sistema linear (3.39) terá melhor estabilidade (Golub & Loan, 1985).

Lembrando que A tem a forma $(\beta I + K)$ nós podemos escrever:

$$\kappa(A) = \kappa(\beta I + K) = \frac{\lambda_{\max}(K) + \beta}{\lambda_{\min}(K) + \beta}. \quad (3.44)$$

Supondo snakes fechadas, é oportuno ressaltar que a matriz K possui neste caso um autovetor dado por $(1, 1, \dots, 1)^T$ cujo autovalor é nulo (Leymarie & Levine, 1993). Este autovetor corresponde à translações da snake, as quais não alteram a

geometria intrínseca da mesma, deixando portando as forças internas inalteradas (daí o autovalor associado ser nulo). Com esta observação podemos simplificar esta expressão para:

$$\kappa(A) = \frac{\lambda_{\max}(K)}{\beta} + 1. \quad (3.45)$$

Por outro lado, $\lambda_{\max}(K)$ pode ser limitado pela relação:

$$\lambda_{\max}(K) \leq \max_j \left(\sum_i |K_{ij}| \right). \quad (3.46)$$

De (3.45)-(3.46) vem que valores altos de β podem melhorar o condicionamento da matriz A e conseqüentemente a estabilidade do sistema linear (3.39). Uma vez que:

$$\beta = \left(\frac{\mu}{\tau^2} + \frac{\lambda}{2\tau} \right), \quad (3.47)$$

fica claro que podemos aumentar esta constante aumentando os valores de μ e λ ou diminuindo o passo no tempo τ . Se diminuirmos muito o passo no tempo teremos um resultado mais preciso, mas isto implicará num esforço computacional extra que pode tornar a performance do algoritmo insatisfatória.

Por outro lado o aumento indiscriminado de λ implica em uma perda de energia cinética muito rápida o que pode levar a uma solução distante da borda procurada.

O resultado acima deixa claro também a necessidade de escolher o passo no tempo com cuidado pois o mau condicionamento do sistema linear decorrente de um passo alto pode implicar em uma solução muito distante da solução real do sistema linear.

De qualquer modo, o resultado acima é suficiente para garantir estabilidade pois este mostra que o erro entre a solução exata do sistema linear e aquela encontrada computacionalmente para cada interação pode ser controlado pela escolha de parâmetros adequados. Fica portanto demonstrada a estabilidade do esquema numérico.

Quanto ao parâmetro h , sua escolha é muito dependente da aplicação pois este deve ser pequeno suficiente para que a curva possa se ajustar ao contorno de interesse.

Quanto à convergência do método numérico; isto é, a garantia de que a solução numericamente computada para o sistema linear converge para a sua solução exata quando $\tau \rightarrow 0$ e $h \rightarrow 0$, basta usar o Teorema de Equivalência de Lax (Hirsch, 1988)

para obtê-la, como consequência direta da consistência e estabilidade discutidas acima.

Outro aspecto importante são os critérios de parada para o processo iterativo. Encontramos duas propostas na literatura (Leymarie & Levine, 1993), ambas partindo do fato das bordas corresponderem a vales da energia potencial (3.22) nos quais a curva deve ficar confinada. Na primeira, interrompe-se o processo iterativo quando:

$$\|(X^n, Y^n) - (X^{n-1}, Y^{n-1})\| < \varepsilon, \quad (3.48)$$

para um ε previamente estabelecido.

No segundo critério, postula-se que a energia de campo média E_l sobre a curva teria taxa de variação pequena próximo da borda; ou seja, que:

$$E_l(t) = \frac{\int_0^1 E_{ext} ds}{\int_0^1 \left\| \frac{dc}{ds} \right\| ds}, \quad (3.49)$$

tenha pequena variação quando a curva c está próxima da borda procurada. Assim, um critério de parada seria:

$$|\Delta E_l| = |E_l(t) - E_l(t - \tau)| < \varepsilon, \quad (3.50)$$

para uma tolerância ε previamente estabelecida.

Ambos os critérios dependem de baixa velocidade da curva para serem satisfeitos. Suponha que o campo externo não varie de forma abrupta nas proximidades da borda, como resultado de uma suavização Gaussiana num esquema multi-escala (Leymarie & Levine, 1993). Então, o segundo critério seria menos sensível a oscilações da curva em torno do ponto de equilíbrio por envolver uma média com relação ao comprimento de arco da curva.

Entretanto, se a borda define um vale de potencial muito estreito, pequenas variações da posição da snake nas proximidades da borda podem levar a mudanças grandes no numerador de $E_l(t)$ não havendo vantagem em usar o critério (3.50).

Neste trabalho, escolhemos o critério (3.48) por ser mais simples. Os resultados que obtivemos até agora com este critério são satisfatórios

3.3 Limitações do Modelo Original

Apesar de suas vantagens para extração de contornos, o modelo original de snakes apresenta várias limitações discutidas em (McInerney, 1997; Gunn & Nixon, 1997;

Xu *et al.* , 1994).

Primeiramente, o funcional de energia (3.22) é em geral não convexo e assim, os métodos de otimização baseados em informações locais (tais como os métodos variacionais descritos nas seções 3.1.1 e 3.2) não oferecem garantia de que a solução encontrada seja o mínimo global (Gunn & Nixon, 1997). Este problema implica que a solução é sensível à inicialização do método, seja para o modelo original dado por (3.16) ou para sua generalização (3.29).

Ao problema da não-convexidade da energia acrescenta-se as dificuldades com a componente normal das forças internas a qual pode ser obtida a partir dos dois primeiros termos da expressão (3.16) (Gunn & Nixon, 1997). Esta força tem o efeito de suavizar a solução final mas tende a contrair a curva inicial (suposta fechada) fazendo com que a mesma colapse em um ponto caso as forças de campo não sejam suficientes para equilibrá-la. Tal comportamento traz dificuldades para a escolha dos parâmetros do modelo.

Esse comportamento da força normal interna está diretamente relacionado com a não invariância da energia (3.22) por transformações de escala (Gunn & Nixon, 1997; Ip & Shen, 1998). Isto fica explícito na equação (3.9) a qual mostra que a energia da snake varia com a mudança do comprimento de arco. Assim, por exemplo, a energia interna de uma circunferência ($(s \rightarrow r(\cos(s), \sin(s)))$) diminui quando seu raio diminui.

Tal comportamento pode não ser desejável quando possuímos informações a priori sobre a geometria dos objetos de interesse. Por exemplo, quando sabemos que os objetos de uma cena são deformações de um protótipo, não é em geral interessante que escalas menores tenham prioridade. Em resumo, a energia interna da snake deve ser modificada para admitir invariância por outras transformações além da rotação e translação.

Uma outra limitação inerente ao modelo da seção 3.1 é a sua dificuldade de tratar com mudanças da topologia da curva inicial; isto é, splits da curva inicial (McInerney & Terzopoulos, 1995; McInerney, 1997). A menos que acrescentemos outros recursos ao modelo, a evolução da curva baseada na minimização do funcional (3.22) impõe que a variedade correspondente á curva permaneça conexa durante a sua evolução; isto é, a curva não pode sofrer subdivisões (*splits*).

Neste trabalho estamos particularmente interessados em soluções para os problemas dos mínimos locais e limitações topológicas do modelo original. As questões referentes à invariância são também discutidas pela sua relação com outros temas abordados neste trabalho.

A seguir desenvolvemos o modelo de Balloon que foi uma primeira abordagem para o problema da não convexidade da energia.

3.4 Modelo de Balloon

Se considerarmos $\mu = 0$ e $\gamma = 1$ na equação (3.23), encontraremos o seguinte sistema:

$$\dot{x} - \omega_1 \frac{\partial^2 x}{\partial s^2} + \omega_2 \frac{\partial^4 x}{\partial s^4} = -\frac{1}{2} \frac{\partial P}{\partial x}, \quad (3.51)$$

$$\dot{y} - \omega_1 \frac{\partial^2 y}{\partial s^2} + \omega_2 \frac{\partial^4 y}{\partial s^4} = -\frac{1}{2} \frac{\partial P}{\partial y}, \quad (3.52)$$

Se agora usarmos um método de discretização equivalente àquele usado na seção 3.2.3, teremos o seguinte sistema para resolver:

$$[I + \tau K] X^n = X^{n-1} + \tau R_X(\bullet, \bar{t} - \tau), \quad (3.53)$$

$$[I + \tau K] Y^n = Y^{n-1} + \tau R_Y(\bullet, \bar{t} - \tau), \quad (3.54)$$

Em (Cohen, 1991) esta formulação é discutida e suas dificuldades básicas analisadas. As soluções apresentadas deram origem ao modelo de *Balloon*, que apresentaremos abaixo por sua relação com o nosso trabalho.

3.4.1 Instabilidade devido a Forças de Imagem

Vamos examinar o efeito da força de imagem $R(\bullet, \bar{t} - \tau)$, como definido anteriormente.

Na seção (3.2.3) vimos que quanto menor o passo no tempo τ melhor a estabilidade numérica do modelo (3.39). Lembremos também que toda a análise de estabilidade estava dependente da suposição de que o vetor $R(\bullet, \bar{t} - \tau)$ é limitado. As componentes deste vetor, denotadas por $R_i(\bullet, \bar{t} - \tau)$, são dadas pelo gradiente:

$$R_{X_i}(\bullet, \bar{t} - \tau) = \frac{\partial P}{\partial x} (v_i^{n-1}), \quad (3.55)$$

$$R_{Y_i}(\bullet, \bar{t} - \tau) = \frac{\partial P}{\partial y} (v_i^{n-1}), \quad (3.56)$$

onde v_i^{n-1} é o snaxel i , da snake discretizada, no instante $\bar{t} - \tau$ e P é o potencial definido em (3.3).

Embora supondo $R(\bullet, \bar{t} - \tau)$ limitado, se a componente $R_i(\bullet, \bar{t} - \tau)$ é muito grande o ponto v_i^{t-1} pode ser deslocado de um valor muito alto fazendo com que a snake passe além da borda desejada em uma vizinhança de v_i^{t-1} . Este comportamento pode causar oscilações e instabilidade no movimento da snake diminuindo a performance do método ou mesmo fazendo com que a snake estabilize em um mínimo diferente daquele desejado.

Se nós escolhermos τ pequeno suficiente tal que o movimento $\tau F(v^{t-1})$ nunca seja grande, por exemplo, nunca seja maior que a distância entre dois pixel vizinhos, então este problema seria evitado. Contudo, procedendo desta forma, gradientes com menor intensidade atrairiam pouco a curva devido a atenuação decorrente da multiplicação por τ .

Desta forma, em lugar de atuar no passo τ de discretização do tempo, a solução apresentada em (Cohen, 1991) é normalizar a força externa segundo:

$$\begin{aligned} F &= -\frac{\nabla P}{\|\nabla P\|}, \text{ se } \|\nabla P\| > T \\ F &= 0, \text{ caso contrário.} \end{aligned} \quad (3.57)$$

onde T é um limiar necessário para evitar problemas numéricos decorrentes da divisão por valores próximos de zero.

Assim, o deslocamento da snake fica controlado, evitando-se passos muito altos devido ao campo externo.

Um problema desta metodologia é que pequenos e grandes gradientes de imagem terão a mesma influência sobre a curva. Isto tornar o modelo mais sensível a mínimos locais, particularmente para imagens com muitos artefatos. Uma forma de diminuir este problema é aplicar previamente um detetor de bordas à imagem original e, em seguida, aplicar o modelo de snakes para extrair a borda desejada. Contudo, isto não resolve totalmente o problema, como será discutido a seguir.

3.4.2 Discretização no Espaço

A força $F = \nabla P$ é conhecida apenas nos nós da malha correspondente à imagem, e portanto, podem existir zero-crossings fora do grid correspondente. Isto conduz a oscilações entre pixels vizinhos a um ponto de borda (ver Figura 3.2) o que pode impedir que a snake atinja o equilíbrio.

Este problema é resolvido por interpolação bilinear para estimar F em pontos não pertencentes à malha original. Assim teremos uma definição contínua de F evitando o problema citado (ver 3.2.b).

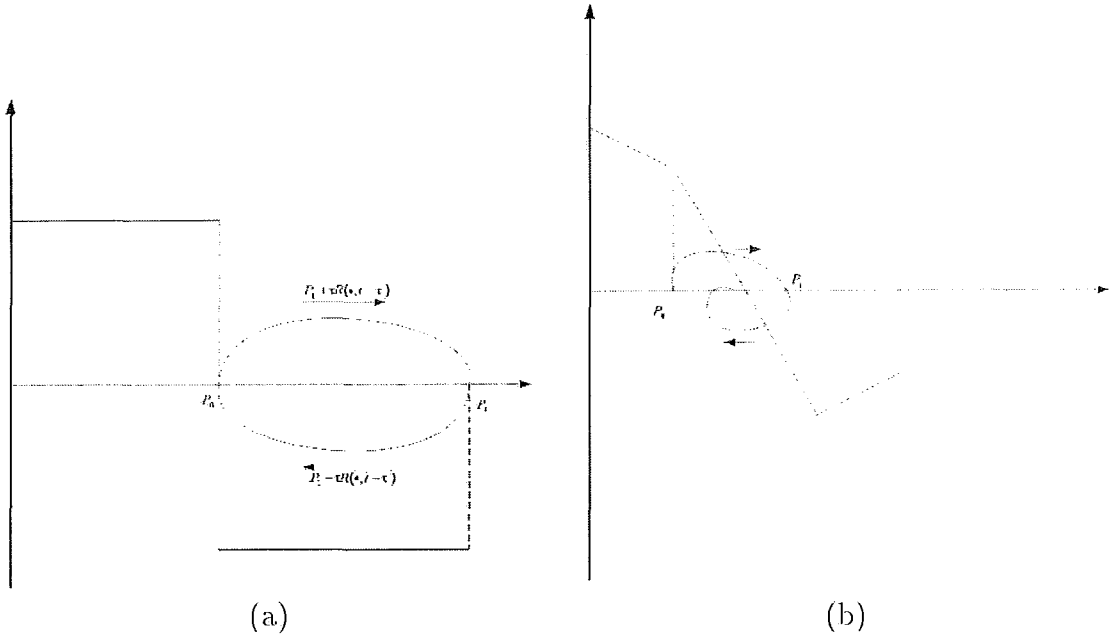


Figura 3.2: (a)Força descontínua provocando oscilações. (b)Interpolação evita instabilidades melhorando convergência.

3.4.3 Inicialização

A escolha da curva inicial para o modelo correspondente ao sistema (3.51)-(3.52) deve levar em conta dois problemas básicos:

- a) Se a curva não está próxima o suficiente de uma borda, ela não será atraída por ela,
- b) Se a curva não está submetida a forças externas, ela colapsa porque a energia interna dada pela expressão (3.5) tem como mínimo global um segmento de reta (snake aberta) ou um ponto (snakes fechadas).

O comportamento do item (a) é facilmente compreendido observando-se que o alcance da força de imagem é em geral estritamente local. A Figura 3.3, que representa um modelo unidimensional para pontos de borda (Canny, 1983), deixa claro este problema.

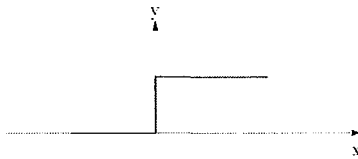


Figura 3.3: Gradiente não nulo apenas em uma vizinhança do ponto de borda.

Quanto ao item (b), uma observação simples esclarece tal comportamento. Su-

ponha que tomemos a seguinte circunferência para inicializar o modelo:

$$c(s, t) = r(\cos(s, t), \text{sen}(s, t)) \quad (3.58)$$

onde r é o raio da circunferência. Vamos supor também s seja o parâmetro de arco para simplificar o resultado. Então, substituindo esta expressão nas equações (3.51)-(3.52), encontraremos após um simples cálculo das derivadas o seguinte resultado:

$$\frac{\partial c}{\partial t} = -(\omega_1 + \omega_2) \{r(\cos(s, t), \text{sen}(s, t))\} - \frac{1}{2} \nabla P \quad (3.59)$$

ou seja, as forças internas possuem uma resultante normal à circunferência e dirigida para o centro da mesma (vetor entre chaves tem a direção do raio). Se a força externa tem pouca intensidade, esta resultante interna faz a snake colapsar em um ponto.

É interessante também analisarmos o significado geométrico das aproximações por diferenças finitas das derivadas espaciais de segunda e quarta ordem dadas em (3.35)-(3.36). Seja então uma snake $c = c(s)$ discretizada segundo a seqüência $s_i = ih; i = 0, 1, \dots, (N - 1)$. As Figuras 3.4(a)-(b) mostram duas vizinhanças do ponto $c(s_i) = P_i$ em uma snake discretizada. Neste caso, as expressões (3.35)-(3.36) podem ser escritas na seguinte forma:

$$\frac{dc}{ds}(s_i) = \frac{2}{h^2} \left[\frac{1}{2} (P_{i+1} + P_{i-1}) + P_i \right] = \frac{2}{h^2} V_1, \quad (3.60)$$

$$\frac{d^4c}{ds^4}(s_i) = \frac{2}{h^4} \left\{ \left[\frac{1}{2} (P_{i+2} + P_{i-2}) + P_i \right] - 4 \left[\frac{1}{2} (P_{i+1} + P_{i-1}) + P_i \right] \right\} = (3.61)$$

$$\frac{2}{h^4} (V_2 - 4V_1). \quad (3.62)$$

onde os vetores V_1 e V_2 estão representados nas Figuras 3.4.a,b, sendo dados pelas expressões entre colchetes das equações anteriores.

Estas expressões serão utilizadas durante este texto no momento oportuno.

3.4.4 Equações do Balloon

Para resolver as dificuldades (a),(b) da seção anterior, a solução proposta em (Cohen, 1991) é adicionar uma força normal extra ao modelo a fim de fazer a snake convergir para aborda em regiões onde o campo externo seja nulo.

O modelo de *Balloon* é dado por:

$$\frac{\partial c}{\partial t} - \omega_1 \frac{\partial^2 c}{\partial s^2} + \omega_2 \frac{\partial^4 c}{\partial s^4} = \kappa_1 F - \kappa_2 \vec{n}, \quad (3.63)$$

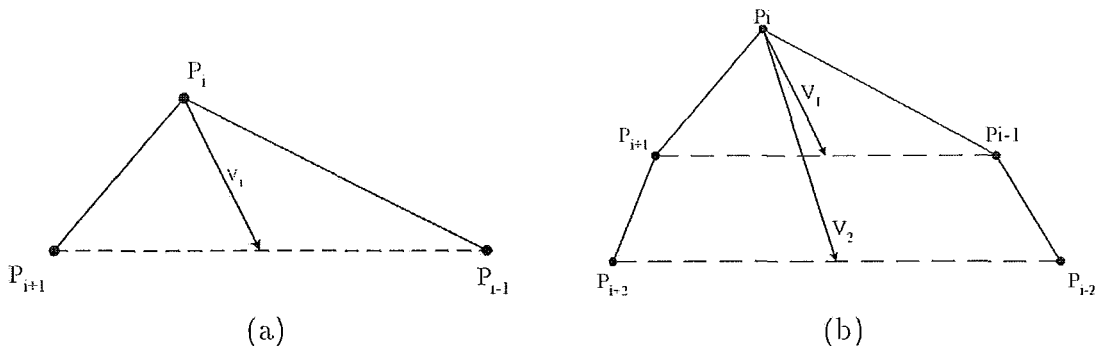


Figura 3.4: (a)Segunda derivada. (b)Derivada de quarta ordem .

onde F é o campo externo normalizado segundo a expressão (3.57), \vec{n} é a normal (unitária) à snake e κ_1, κ_2 são fatores de escala. Agora a curva se comporta como um "balão", daí o nome do modelo, o qual se expande até atingir o equilíbrio pela ação das forças de imagem. O modelo discreto obtido por um esquema de discretização idêntico àquele da seção 3.2.3 é dado pelo seguinte sistema linear:

$$[I + \tau K] X^n = X^{n-1} + \tau [R_X(\bullet, \bar{t} - \tau) - N_X(\bullet, \bar{t} - \tau)], \quad (3.64)$$

$$[I + \tau K] Y^n = Y^{n-1} + \tau [R_Y(\bullet, \bar{t} - \tau) - N_Y(\bullet, \bar{t} - \tau)], \quad (3.65)$$

onde $N(\bullet, \bar{t} - \tau) = (N_X, N_Y)$ é a matriz correspondente à nova força normal sobre a snake.

O novo modelo tem quatro parâmetros. Em (Cohen, 1991) encontra-se a sugestão de que κ_1 e κ_2 sejam da mesma ordem, com κ_1 ligeiramente maior que κ_2 . Assim, quando a snake atinge a borda, o campo externo poderá "segurar" a snake evitando que a mesma ultrapasse a borda.

Quanto à rigidez e elasticidade, é interessante que os coeficientes não nulos da matriz de rigidez K tenham a mesma ordem de magnitude. Em (Cohen, 1991) obteve-se bons resultados quando estes parâmetros são da ordem de h^2 para ω_1 e da ordem de h^4 para ω_2 .

Capítulo 4

Métodos Matemáticos

Neste capítulo discutimos conceitos matemáticos usados em modelos alternativos que procuram superar as limitações do modelo original apresentadas na seção 3.3. A apresentação dos modelos propriamente ditos é deixada para o próximo capítulo.

4.1 Programação Dinâmica

Em Amini et al. (Amini *et al.*, 1990) encontramos uma alternativa ao método variacional (Euler-Lagrange) baseada em programação dinâmica. Nesta seção vamos discutir os principais elementos desta proposta realçando suas vantagens e desvantagens em relação ao método variacional para a otimização da energia das snakes.

Os métodos variacionais vêm sendo usados para a solução de vários problemas em visão computacional, tais como cálculo do fluxo óptico e shape from shading (Black & Yuille, 1993). As limitações deste método no contexto destes problemas assim como no contexto das snakes, se originam no alto grau de diferenciabilidade necessária, gerando instabilidade numérica na presença de ruídos, e dificuldades para garantir se a solução encontrada é o ótimo global (Amini *et al.*, 1990).

A programação dinâmica, por outro lado, tem a vantagem de garantir a globalidade da solução encontrada sendo também numericamente mais estável que o método variacional. Contudo, a programação dinâmica tem limitações em função do montante de memória ($O(NM^2)$) necessário e da complexidade computacional ($O(NM^3)$), onde N é o número de snaxels e M é o número de pontos vizinhos a cada snaxel (gerando um *espaço de busca discreto* com NM pontos).

Neste capítulo iniciaremos com a apresentação dos princípios básicos da programação dinâmica no caso discreto (Dreyfus, 1965). O caso contínuo foi deixado para o Apêndice A, não sendo sua leitura pré-requisito para discussão que segue.

4.1.1 Programação Dinâmica no caso Discreto

Neste capítulo, formalizamos conceitos referentes a certos problemas de decisão do ponto de vista da Programação Dinâmica.

Os problemas a serem tratados têm a propriedade de que, a cada instante de uma seqüência finita (t_1, t_2, \dots, t_N) , uma escolha deve ser feita dentre um conjunto finito de possibilidades. Formalmente, diremos que a cada instante t_i temos um conjunto finito de M possibilidades de escolha (*estados*) constituindo o *estágio* i . Uma vez que N e M são finitos, temos um espaço de estados discreto (*espaço de busca discreto*) com NM elementos.

Os problemas que podem ser colocados desta forma são denominados *problemas de decisão finitos em multi – estágio* e serão tratados nesta seção no contexto da programação dinâmica (Dreyfus, 1965).

Nosso ponto de partida será a aplicação do *Princípio de Optimalidade* que estabelece o seguinte: *O caminho ótimo entre dois pontos é também ótimo entre quaisquer dois outros pontos sobre o mesmo.*

Através deste princípio (Dreyfus, 1965) podemos obter uma expressão recursiva a qual é básica para a aplicação de programação dinâmica em snakes.

Se mantivermos a representação das bordas dos objetos de interesse por curvas contínuas como no modelo original de snakes (seção 3.1) então precisaríamos usar conceitos de programação dinâmica contínua. Infelizmente, esta formulação recai em equações diferenciais complicadas de difícil utilização na prática (ver Apêndice A ou (Dreyfus, 1965)).

Assim, a alternativa adotada por Amini et al. (Amini *et al.*, 1990) assim como nesta apresentação será supor que as bordas procuradas podem ser representadas por um modelo discreto, e assim formular as snakes no contexto discreto em lugar do contínuo usado no capítulo anterior. Vamos supor inicialmente que as curvas são abertas com extremos fixos.

Desta forma, representaremos as curvas de interesse através de um conjunto de pontos $\{v_i = (x_i, y_i), i = 0, \dots, N + 1\}$ ligados seqüencialmente, lembrando que v_0 e v_{N+1} são fixos (extremos da curva). Como no capítulo anterior, denotaremos o espaço formado por estas curvas por Ad (*espaço admissível*).

Com isto, o funcional de energia associado tomará (ver abaixo) a forma de uma função das $2N$ variáveis correspondentes aos pontos livres $\{v_i = (x_i, y_i), i = 1, \dots, N\}$ e o problema de otimização correspondente torna-se um problema de otimização não-linear do tipo:

$$\begin{aligned} \min E(v_1, \dots, v_N), \\ Ad \end{aligned} \tag{4.1}$$

onde Ad é espaço de curvas admissíveis definido acima.

Nos próximo capítulo veremos o modelo Dual e então explicitaremos o funcional de energia E em (4.11). Por hora, basta termos em mente que este funcional será constituído por uma energia interna e outra externa, podendo ser pensado como uma forma discreta daquele usado no modelo original (seção 3.1). Assim, vamos supor que E tem a forma:

$$E(v_1, \dots, v_N) = E_{int}(v_1, \dots, v_N) + E_{ext}(v_1, \dots, v_N), \quad (4.2)$$

onde E_{int} e E_{ext} são dados, respectivamente, por:

$$E_{int}(v_1, \dots, v_N) = \alpha \sum_{i=0}^{N-1} E_{int}^i(v_i, v_{i+1}, v_{i+2}), \quad (4.3)$$

$$E_{ext}(v_1, \dots, v_N) = \beta \sum_{i=1}^N P(v_i), \quad (4.4)$$

onde $P(v) = -\|\nabla I(v)\|^2$ e α e β são parâmetros a serem estabelecidos a priori.

A expressão (4.3) significa que a energia interna associada a um snaxel v_i depende de seus vizinhos v_{i-1} e v_{i+1} além do próprio v_i .

A idéia central é formular o problema em (4.11) como um problema de decisão finito em multi-estágios. Para isto temos que definir um conjunto finito de estágios e a cada estágio associar um conjunto finito de possibilidades (escolhas). Cada escolha implica em um custo e a solução do problema é a seqüência de escolhas que ao final tem custo mínimo.

Lembrando que a função E em (4.11) depende dos N pontos livres da snake, é imediato definir o número de estágios igual a N . O número de possíveis escolhas para cada estágio vai depender de como discretizamos o domínio.

A Figura 4.1 a seguir representa um exemplo onde temos duas curvas limitando a região onde se encontra a borda procurada. A região entre as curvas é o domínio a ser considerado.

A Figura 4.2 mostra a maneira mais natural de subdividirmos este domínio aproveitando a semelhança geométrica entre a borda procurada e as curvas limites.

Nesta figura, estabelecemos uma correspondência entre os pontos das curvas limites e posteriormente discretizamos os segmentos que unem pontos correlatos em M partes. Se aumentarmos N ou M , melhoramos a precisão na localização da borda, mas aumentamos o custo computacional.

Uma vez que estamos considerando snakes abertas, precisamos escolher os pontos inicial e terminal da snake. O espaço de busca usado (Figura 4.2) impõe que estes

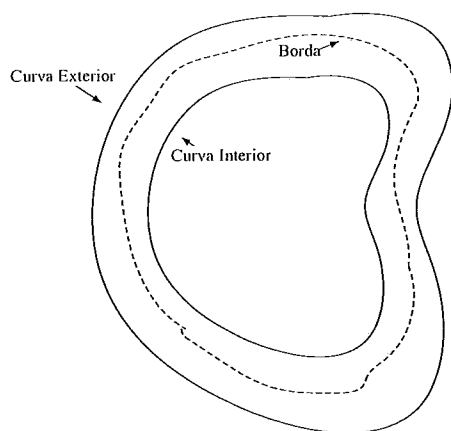


Figura 4.1: Curvas limitando região contendo borda.

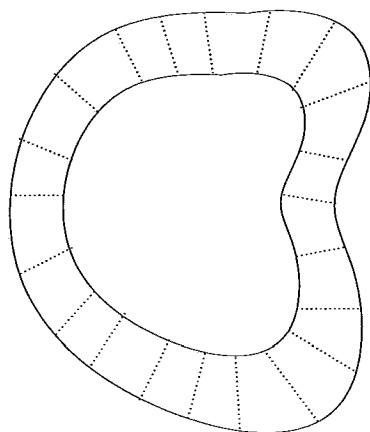


Figura 4.2: Discretização da região de interesse.

pontos pertençam a segmentos consecutivos. Como acima, denotaremos estes pontos por v_0 e v_{N+1} .

Retomando o funcional de energia em (4.2) e usando (4.3)-(4.4), vemos que a energia total da snake pode ser escrita na forma

$$E_{snake}(v) = E_0(v_0, v_1, v_2) + E_1(v_1, v_2, v_3) + \dots + E_{N-1}(v_{N-1}, v_N, v_{N+1}), \quad (4.5)$$

onde:

$$E_i(v_i, v_{i+1}, v_{i+2}) = \alpha E_{int}^i(v_i, v_{i+1}, v_{i+2}) + \beta P(v_i), \quad (4.6)$$

e $v = (v_0, v_1, \dots, v_N, v_{N+1})$.

Vamos definir uma função auxiliar denominada *função de valor ótimo*, dada por:

$$S(v_{i+2}, v_{i+1}) = \text{energia da curva ótima conectando o ponto inicial } v_0 \text{ ao ponto } v_{i+1}. \quad (4.7)$$

Pelo Principio de Optimalidade (Dreyfus, 1965; Amini *et al.*, 1990), podemos escrever $S(v_{i+2}, v_{i+1})$ na forma recursiva:

$$S(v_{i+2}, v_{i+1}) = \min_{v_i} \{S(v_{i+1}, v_i) + E_i(v_i, v_{i+1}, v_{i+2})\}. \quad (4.8)$$

Acrescentemos a esta definição a restrição:

$$S(v_1, v_0) = 0. \quad (4.9)$$

Posto desta forma, observemos que a solução do problema (4.11) é exatamente:

$$\min_{v_N} \{S(v_{N+1}, v_N)\}. \quad (4.10)$$

A expressão 4.8 é básica para a aplicação da programação dinâmica discreta em snakes. A seção 5.2 do próximo capítulo apresenta um modelo de snakes baseado nos conceitos acima descritos.

A apresentação de conceitos em programação dinâmica para o caso contínuo está feita no Apêndice A.

A seguir analisaremos questões topológicas.

4.2 Mudanças Topológicas

Neste trabalho denominamos *mudanças topológicas* ao fato de uma curva simples se dividir em duas ou mais componentes conexas (*split*) ou duas curvas simples se juntarem em uma única (*merge*).

Uma forma natural de formular mudanças topológicas seria supor que as curvas em questão são diferenciáveis e representá-las implicitamente; isto é, defini-las como um conjunto de nível de uma função $\Psi : \mathfrak{R}^2 \rightarrow \mathfrak{R}$ definida convenientemente.

Assim, mudanças topológicas ocorreriam naturalmente e suavemente, desde que função Ψ não perdesse sua suavidade. Este é o conceito fundamental das formulações contínuas e implícitas para mudanças topológicas (Malladi *et al.*, 1995).

O resultado destas propostas recai em conceitos de geometria e topologia diferenciais. O ponto de partida destas formulações é o Teorema da Função Implícita que estabelece que (Lima, 1985):

Teorema da Função Implícita: Seja $H : \mathfrak{R}^{N+K} \rightarrow \mathfrak{R}^N$ uma aplicação suave tal que $0 \in \text{range}(H)$. Então:

$$M = \{x \in \mathfrak{R}^{N+K} \mid H(x) = 0, x \text{ é um ponto regular de } H\},$$

é uma variedade diferenciável suave de dimensão K .

O método mais conhecido cuja base teórica parte deste teorema, deve-se a Malladi *et al.* (Malladi *et al.*, 1995) sendo comumente conhecido como Método dos Conjuntos de Níveis (*Level Sets*). Este método será descrito resumidamente no próximo capítulo.

Apesar das habilidades topológicas, o uso explícito de uma dimensão extra torna este método menos intuitivo que os modelos paramétricos, trazendo dificuldades para a incorporação de informações adicionais (de forma, por exemplo), ou vínculos definidos a priori ou criados pelo usuário durante a evolução do método.

Uma alternativa aos modelos implícitos é o método da T-Snakes. Este último é um modelo paramétrico capaz de lidar com mudanças topológicas tanto para 2D quanto 3D. Este método é fundamental para nosso trabalho e será apresentado em detalhes no próximo capítulo. A seguir, discutimos sua base teórica.

4.3 Fundamentos Matemáticos das T-Snakes

O método das T-Snakes (McInerney & Terzopoulos, 1995; McInerney, 1997) aborda as questões topológicas dentro do contexto da topologia combinatória a fim de obter uma formulação paramétrica com habilidades topológicas para as snakes.

Seus elementos básicos são uma decomposição simplicial (triangulação) do domínio,

um modelo discreto de snakes e uma *função característica*, definida nos nós da triangulação, que distingue o interior do exterior das snakes.

O par triangulação/função característica é a chave para a incorporação eficiente de mudanças topológicas em um modelo paramétrico de snakes; o que ficará mais claro após a apresentação que segue.

4.3.1 Decomposição do Domínio

Dado um subconjunto fechado $D \subset \mathbb{R}^n$ existem dois métodos básicos para decomposição de domínio: não-simpliciais e simpliciais.

Os métodos não-simpliciais são baseados em decomposições celulares do espaço e não podem ser usados para representar superfícies ou contornos sem a utilização de métodos para retirar antigüidades. Caso típico de um método que utiliza este tipo de decomposição do espaço é o *Marching Cubes* (Lorenson & Cline, 1987).

Métodos usando decomposição simplicial do domínio não apresentam ambigüidades para a geração de aproximações poligonais para curvas e superfícies (Allgower & Georg, 1990).

Vejam alguns conceitos fundamentais nesta área (Allgower & Georg, 1990).

Um conjunto de pontos $\{v_1, v_2, \dots, v_{k+1}\} \subset \mathbb{R}^n$ é dito "afim-independente" se as diferenças $v_2 - v_1, v_3 - v_1, \dots, v_{k+1} - v_1$ formam um conjunto de vetores linearmente independente no \mathbb{R}^n .

O fecho convexo $\left\{v = \sum_{i=1}^{k+1} \alpha_i v_i \mid \alpha_i \geq 0, i = 1, \dots, k+1 \text{ e } \sum_{i=1}^{k+1} \alpha_i = 1\right\}$, com vértices dados por $k+1$ pontos afim-independentes $\{v_1, v_2, \dots, v_{k+1}\}$ do \mathbb{R}^n , é chamado um *k-simplex* e será denotado por $[v_1, v_2, \dots, v_{k+1}]$.

Seja $\sigma = [v_1, v_2, \dots, v_{k+1}]$ um *k-simplex* e seja $1 \leq l \leq k+1$. Se $\{w_1, w_2, \dots, w_{l+1}\}$ é um subconjunto de vértices de σ , nós chamamos o *l-simplex* $\tau = [w_1, w_2, \dots, w_{l+1}]$ uma *face l-dimensional* de σ , ou simplesmente uma *l-face*. De particular interesse são:

- (1) 0 – faces as quais correspondem a vértice de σ ;
- (2) 1 – faces as quais são também chamadas *arestas* de σ ;
- (3) $(k - 1)$ – faces as quais são chamadas *faces* de σ .

Seja Γ uma família não-vazia de $(n + 1)$ – simplices em \mathbb{R}^{n+1} . Nós chamamos Γ uma triangulação do \mathbb{R}^{n+1} se:

a.

$$\bigcup_{\sigma \in \Gamma} \sigma = \mathbb{R}^{n+1}, \tag{4.11}$$

b. a intersecção $\sigma_1 \cap \sigma_2$ de dois simplices quaisquer $\sigma_1, \sigma_2 \in \Gamma$ é vazia ou uma face comum aos simplices;

c. a família Γ é localmente finita, isto é, qualquer subconjunto compacto de \mathbb{R}^{n+1} corta um número finito de simplices $\sigma \in \Gamma$.

Seja Γ uma triangulação do \mathfrak{R}^{n+1} , e seja $0 \leq k \leq n + 1$. Nós denotamos por

$$\Gamma^k = \{ \tau \mid \tau \text{ é uma } k\text{-face de algum simplex } \sigma \in \Gamma \},$$

a família de todas as k -faces em Γ .

Para \mathfrak{R}^2 um 2-simplex é um triângulo. Para o \mathfrak{R}^3 um 3-simplex é um tetraedro. Seguindo a nomenclatura padrão, nós chamamos *nó* a um vértice de um simplex e a coleção de nós e arestas de Γ chamamos *malha* (identificada com Γ^1). A triangulação Γ é identificado com Γ^{n+1} ($\Gamma = \Gamma^{n+1}$).

Dois simplices $\sigma_1, \sigma_2 \in \Gamma$ são ditos adjacentes se eles têm uma face em comum.

Dado um simplex, o procedimento usado para obter os adjacentes a este é denominado *pivotiamento* e depende do tipo de triangulação usado. O lema a seguir prepara sua introdução.

Seja Γ uma triangulação do \mathfrak{R}^{n+1} . Considere um simplex $\sigma \in \Gamma$ e seja τ uma face de σ . Então, existe um único simplex $\bar{\sigma} \in \Gamma$ tal que:

- (1) $\bar{\sigma} \neq \sigma$;
- (2) τ uma face de $\bar{\sigma}$;

Prova: Ver (Allgower & Georg, 1990).

Denotemos por *del* v_i a operação de eliminação de um vértice v_i .

Seja $\sigma = [v_1, v_2, \dots, v_{n+2}]$ um $(n + 1)$ -simplex de uma triangulação de \mathfrak{R}^{n+1} , e seja $\tau = [v_1, \dots, \text{del}v_i, \dots, v_{n+2}]$ a face de σ oposta ao vértice v_i . Pelo lema anterior, deve existir um único nó \bar{v}_i diferente de v_i e tal que $\bar{\sigma} = [v_1, \dots, \bar{v}_i, \dots, v_{n+2}] \in \Gamma$. A passagem de σ para $\bar{\sigma}$ é chamada *pivotiamento*. Nós dizemos que o vértice v_i de σ é *pivotiado* no v_i , e que o simplex $\bar{\sigma}$ é *pivotiado* no simplex σ através da face τ .

4.3.2 Aproximação Linear por Partes

Seja $H : \mathfrak{R}^{n+1} \rightarrow \mathfrak{R}^n$ uma aplicação.

Dada uma triangulação Γ de \mathfrak{R}^{n+1} , nós desejamos aproximar as componentes de $H^{-1}(0)$ usando apenas os valores de H nos nós da triangulação. Não necessitamos de nenhuma suposição sobre suavidade ou continuidade sobre H , a menos que desejemos estudar erros de truncamento para a aproximação linear por partes a ser definida. Vejamos algumas definições:

Seja $H : \mathfrak{R}^{n+1} \rightarrow \mathfrak{R}^n$ uma aplicação e seja Γ uma triangulação de \mathfrak{R}^{n+1} , e seja $\tau = [v_1, v_2, \dots, v_{k+1}] \in \Gamma^k$ uma k -face de Γ onde $0 \leq k \leq n + 1$.

a. Por $H_\tau : \tau \rightarrow \mathfrak{R}^n$ denotamos a única aplicação afim que coincide com H nos vértices v_i de τ , isto é:

$$H_\tau(v) = \sum_{i=1}^{k+1} \alpha_i H(v_i), \text{ para } v = \sum_{i=1}^{k+1} \alpha_i v_i,$$

b. Seja $\text{tn}g(\tau)$ (*espaço tangente de τ*) o espaço linear que contém todas as diferenças $w_1 - w_2$ com $w_1, w_2 \in \tau$. Então a Jacobiana de H é a aplicação linear $H'_\tau : \text{tn}g(\tau) \rightarrow \mathbb{R}^n$ obtido por $H'_\tau(w_1 - w_2) = H_\tau(w_1) - H_\tau(w_2)$ para $w_1, w_2 \in \tau$. Uma vez que H_τ é afim, esta definição não é ambígua, isto é: $H_\tau(w_1) - H_\tau(w_2) = H_\tau(\tilde{w}_1) - H_\tau(\tilde{w}_2)$ para $w_1 - w_2 = \tilde{w}_1 - \tilde{w}_2$.

c. Finalmente, a aproximação linear por partes de H é obtida pela união:

$$H_\Gamma = \bigcup_{\sigma \in \Gamma} H_\sigma,$$

isto é, $H_\Gamma(v) = H_\sigma(v)$ para todo $v \in \sigma$ e $\sigma \in \Gamma$. Não existe ambigüidade nesta definição uma vez que pode ser facilmente demonstrado que $H_{\sigma_1}(v) = H_{\sigma_2}(v)$ para $v \in \sigma_1 \cap \sigma_2$.

Um ponto $x \in \mathbb{R}^{n+1}$ é chamado um ponto regular de H_Γ se e somente se:

(a) x não pertence a qualquer k -face $\tau \in \Gamma^k$ tal que $k < n$;

(b) H'_τ tem posto máximo para todo $\sigma \in \Gamma^n \cup \Gamma^{n+1}$ tal que $x \in \sigma$.

Um valor $y \in \mathbb{R}^n$ é chamado um *valor regular* de H_Γ se todos os pontos de $H_\Gamma^{-1}(y)$ são regulares. Se um ponto não é regular será dito *singular*. Analogamente, se um valor não é regular ele será dito *singular*.

Perturbação: Seja o vetor:

$$\vec{\varepsilon} = \begin{pmatrix} \varepsilon^1 \\ \cdot \\ \cdot \\ \cdot \\ \varepsilon^n \end{pmatrix}$$

onde $\varepsilon > 0$ e $\varepsilon^i > 0, i = 1, \dots, n$, são potências de ε . Denominaremos este vetor um ε -vetor.

Proposição. Para qualquer subconjunto compacto $C \subset \mathbb{R}^{n+1}$ existem no máximo um número finito de $\vec{\varepsilon}$ tal que $C \cap H_\Gamma^{-1}(\vec{\varepsilon})$ contém um ponto singular de H_Γ . Conseqüentemente, $\vec{\varepsilon}$ é um valor regular de H_Γ para quase todo $\vec{\varepsilon}$.

Prova: Ver (Allgower & Georg, 1990).

Nós chamamos um n -simplex $\tau \in \Gamma^n$ *completamente rotulado* (*completely labeled*) se e somente se ele contém soluções v_ε da equação $H_\Gamma(v) = \vec{\varepsilon}$ para qualquer $\varepsilon > 0$ suficientemente pequeno.

Um $(n+1)$ -simplex $\sigma \in \Gamma$ é chamado *transverso ou de borda* (com respeito a H) se ele contém uma n -face completamente rotulada.

(Princípio Door-In-Door-Out): Um $(n+1)$ -simplex tem nenhum ou exatamente duas n -faces completamente rotuladas.

Assim, podemos finalmente enunciar o seguinte algoritmo, com o qual é possível traçar qualquer componente conexa de $H_T^{-1}(0)$,

Algoritmo de Continuação (*Door-In-Door-Out*)

Inicialização

. $\sigma_0 \in \Gamma$ triângulo de borda (simplex inicial)

. τ_0 uma n -face completamente rotulada de σ_0 ;

Repetir para $n = 1, 2, \dots$

.encontrar $\sigma_n \in \Gamma$, $\sigma_n \neq \sigma_{n-1}$ tal que $\tau_{n-1} = \sigma_n \cap \sigma_{n-1}$;

.encontrar a n -face completamente rotulada τ_n de σ_n tal que $\tau_n \neq \tau_{n-1}$;

.parar quando retornar ao triângulo inicial $\sigma_0 \in \Gamma$.

A inicialização do algoritmo depende da escolha do simplex inicial σ_0 . No caso das snakes a escolha deste simplex é feita através dos pontos da própria snake. No próximo capítulo veremos isto em detalhes.

O algoritmo acima gera uma seqüência

$$\sigma_0 \supset \tau_0 \subset \sigma_1 \supset \tau_1 \subset \sigma_2 \supset \tau_2 \subset \sigma_3 \dots \tag{4.12}$$

de triângulos de borda, $(n + 1) - \text{simplices } \sigma_i$, cada um destes contendo duas faces completamente rotuladas τ_{i-1}, τ_i . Uma vez que nenhum $(n + 1) - \text{simplices}$ pode ter mais que duas n -faces completamente rotuladas, a seguinte classificação é facilmente verificada:

(a) Infinita: Ambas as seqüências infinitas $\sigma_0, \sigma_1, \dots$ e τ_0, τ_1, \dots geradas são tais que $\sigma_i \neq \sigma_{i+1}$ e $\tau_i \neq \tau_{i+1}, i = 0, 1, \dots$

(b) Cíclica: Existe um inteiro finito $\tilde{n} > 0$ tal que $\sigma_0 = \sigma_{\tilde{n}}$ e $\tau_0 = \tau_{\tilde{n}}$. A relação cíclica $\sigma_i = \sigma_{\tilde{n}+i}$ e $\tau_i = \tau_{\tilde{n}+i}$ vale para $i = 0, 1, \dots$

Uma prova formal desta proposição é obtida via teoria de grafos (Allgower & Georg, 1990).

Uma vez que neste trabalho estamos interessados particularmente em snakes $2D$ a apresentação acima fornece os elementos necessários aos próximos capítulos.

Contudo, reservamos o Apêndice C para a generalização dos conceitos acima para variedades de dimensão qualquer, uma vez que estes serão necessários para a extensão do nosso trabalho para $3D$.

4.4 Teoria de Invariância

Esta teoria é particularmente útil em situações em que, conhecidas as grandezas fundamentais (x_1, x_2, \dots, x_n) envolvidas em um problema, desejamos determinar uma quantidade $F(x_1, x_2, \dots, x_n)$ que se mantenha inalterada (*invariante*) quando as grandezas fundamentais são transformadas.

No caso das snakes, por exemplo, é razoável que sua energia interna não mude quando aplicamos operações de translação ou rotação sobre a curva correspondente. Tal propriedade implica que a energia interna seja independente da posição da snake na imagem o que é uma propriedade geralmente assumida para termos de regularização (Snyder, 1991).

Em visão computacional, particularmente em problema envolvendo calibração da câmera, a utilidade do conceito de invariância aparece mais naturalmente (Gool *et al.*, 1995).

Em processamento de imagem, alguns resultados envolvendo invariância por transformações de escala estão na base das representações multi-escala de imagens (Florack *et al.*, 1992).

Nestes exemplos, a aplicação do conceito de invariância depende da definição precisa das variáveis (x_1, x_2, \dots, x_n) de interesse e a definição do "grupo" de transformações que vai agir sobre estas variáveis, ou mais precisamente, sobre o espaço que estas definem. Feito isto, temos condições de colocar a seguinte questão:

Existem grandezas (quantidades) $F = F(x_1, x_2, \dots, x_n)$ que não mudam seu valor quando aplicamos as transformações de interesse sobre as grandezas fundamentais (x_1, x_2, \dots, x_n) ?

Esta pergunta nos leva diretamente ao problema de identificação de invariantes, o qual pode ser posto rigorosamente no contexto da teoria dos grupos de Lie (Olver, 1986).

De maneira mais precisa, a busca de invariantes está diretamente relacionada à derivada de Lie correspondente, a qual expressa a variação de uma função real ou de um campo (vetorial, por exemplo) sob a ação de um grupo de transformações (Olver, 1986).

Assim, o ponto de partida da teoria de invariância de Lie é um grupo de transformações que atuam em uma variedade diferenciável (ou em um espaço vetorial). A derivada de Lie correspondente fornece uma condição necessária e suficiente para que uma função com valores reais definida na variedade seja invariante pela ação do grupo de transformações. Fazendo uso de um sistema de coordenadas local, teremos um sistema de equações diferenciais parciais, cujas soluções fornecerão os invariantes procurados (Olver, 1986).

Tal formalismo vem sendo estudado no contexto de problemas em visão computacional (Gool *et al.*, 1995) procurando-se estabelecer sua utilidade prática. Uma primeira dificuldade neste sentido, é que os problemas em visão computacional não se apresentam diretamente na forma de espaços e seus grupos de transformações, o que exige um estudo cuidadoso do problema para apresentá-lo de maneira apropriada para a aplicação da teoria.

No contexto das snakes, não encontramos referências da aplicação da teoria de

Lie na busca de novos modelos. Em geral as pesquisas têm sido conduzidas de maneira mais intuitiva e menos formal (Gunn & Nixon, 1997; Ip & Shen, 1998).

Contudo, uma vez que os conceitos presentes na idéia de invariância têm lugar comum na teoria de Lie, dedicamos o Apêndice B para sumarizar as idéias principais desta teoria no contexto de interesse. No final deste Apêndice apresentamos possíveis aplicações da teoria de Lie no contexto de snakes.

O leitor pode saltar a leitura do Apêndice B sem perda para a compreensão do texto que segue.

Capítulo 5

Modelos Alternativos de Interesse

Neste capítulo descrevemos alguns modelos baseados nos elementos discutidos no capítulo anterior que procuram corrigir as limitações inerentes ao modelo original de snakes.

Dentre os métodos apresentados, o modelo Dual e das T-Snakes têm relação direta com nosso trabalho. O modelo das AI-Snakes não está diretamente relacionado com nosso método, mas constitui-se em um exemplo elegante onde o conceito de invariância tem de fato um papel central. O método dos Conjuntos de Níveis é um dos modelos que inspirou o desenvolvimento das T-Snakes, particularmente no que diz respeito à idéia de uma função que distingue o interior do exterior da snake e da política usada para atualizá-la. Por esta razão será apresentado neste capítulo.

5.1 Modelos Usando Invariância

O conceito de *invariância* por transformações, largamente empregado em visão computacional (Gool *et al.*, 1995), vem sendo usado no contexto dos modelos de contornos ativos discretos para remover os efeitos indesejados da componente normal da força interna ou para facilitar a incorporação de informações a priori (Gunn & Nixon, 1997; Ip & Shen, 1998).

Esta idéia vem sendo aplicada para contornos ativos fechados e consiste primeiramente na utilização de uma energia interna que possua propriedades adicionais de invariância além das usuais por translação e rotação.

Esta energia interna pode ser obtida através de um modelo de forma vinculado a um *protótipo* (Ip & Shen, 1998) ou simplesmente derivada de elementos que tenham as propriedades de invariância desejadas (Ip & Shen, 1998; Gunn & Nixon, 1997). Com isto, evita-se o colapso da curva, uma vez que a snake tem agora a tendência natural de adquirir a forma pré-estabelecida no modelo, limitando inclusive a maneira como a snake se deforma durante sua evolução. Ao mesmo tempo, a energia interna assim definida deve fornecer o termo de regularização (ou suavização) necessário.

5.1.1 Modelo Dual

O método Dual (Gunn & Nixon, 1997; Gunn, 1996a) é um exemplo do emprego do conceito de invariância para evitar as dificuldades com a força normal. Neste caso, as transformações de interesse são as transformações rígidas (translação e rotação) e transformações de escala. Assim, um snaxel ocupando uma posição V é transformado para uma posição U segundo:

$$U = s \begin{pmatrix} \cos \theta & -\text{sen} \theta \\ \text{sen} \theta & \cos \theta \end{pmatrix} V + \begin{pmatrix} a \\ b \end{pmatrix} = AV + B \quad (5.1)$$

onde s é o parâmetro de escala, θ é o ângulo de rotação e $(a, b)^T$ é o vetor de translação.

Este método consiste de um contorno que expande a partir do interior do objeto e de um outro contorno que contrai a partir do exterior.

É estabelecida uma correspondência entre os dois contornos o que permite que os mesmos sejam interligados para obter-se uma força adicional (*driving force*) a qual é usada para retirar os contornos de mínimos locais garantindo mais robustez.

Vejamos inicialmente a definição da energia interna invariante pelas transformações em questão.

Energia Interna

A necessidade de uma correspondência (*matching*) entre os pontos dos contorno interno e externo obriga que sejam impostas restrições para a forma das snakes pois do contrário a correspondência entre pontos das duas curvas pode se tornar complicada. Esta restrição de forma é incorporada ao modelo via um modelo de forma local que passaremos a descrever.

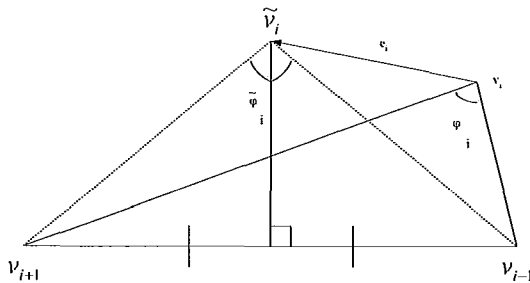


Figura 5.1: Modelo local de forma.

Na Figura 5.1, o vetor \vec{e}_i é tal que

$$\|v_{i-1} - \tilde{v}_i\| = \|v_{i+1} - \tilde{v}_i\|, \quad (5.2)$$

isto é:

$$\vec{e}_i = \frac{1}{2}(v_{i-1} + v_{i+1}) - v_i + \frac{1}{2}\theta_i R(v_{i-1} - v_{i+1}), \quad (5.3)$$

onde R é uma matriz de rotação de 90° e θ_i está relacionado ao ângulo interno $\tilde{\varphi}_i$ em v_i por:

$$\theta_i = \cot\left(\frac{\tilde{\varphi}_i}{2}\right). \quad (5.4)$$

Com isto, o modelo de forma é estabelecido em (Gunn & Nixon, 1997) pela seguinte expressão:

$$E_{int} = \frac{1}{2} \sum_{i=0}^{N-1} \left(\frac{\|\vec{e}_i\|}{h} \right)^2. \quad (5.5)$$

Observe que E_{int} tem um mínimo global quando $\vec{e}_i = 0$, $i = 0, 1, \dots, N - 1$. Usando (5.2)-(5.4) pode ser mostrado que isto acontece quando:

$$\varphi_i = \pi(N - 2)/N, i = 0, 1, \dots, N - 1, \quad (5.6)$$

os quais são ângulos internos do polígono regular com vértices dados pelos pontos v_i (Gunn & Nixon, 1997). Além disto, a energia (5.5) é invariante por transformações de rotação, translação e escala como desejado (Gunn & Nixon, 1997).

A energia total do modelo Dual é dada por:

$$E = \frac{\lambda}{N} E_{int}(v_i) + (1 - \lambda) E_{ext}(v_i), \quad (5.7)$$

onde E_{ext} é uma forma discreta de (3.4):

$$E_{ext} = \sum_{i=0}^{N-1} P(v_i), \quad (5.8)$$

com potencial externo P dado pela expressão:

$$P(v_i) = -\|\nabla I\|^2. \quad (5.9)$$

O fator λ em (5.7) é um parâmetro de suavização tomado entre 0 e 1 (Gunn & Nixon, 1997).

Com a energia interna dada por (5.7) os problemas com a força normal interna são evitados uma vez que a curva tem agora a tendência natural a obter a forma de um polígono regular em vez de colapsar para um ponto.

Tal tendência tem também a propriedade de limitar a forma da curva durante a sua evolução. Em (Gunn & Nixon, 1997) os autores tomam vantagem desta limitação da forma das snake para estabelecer uma correspondência entre os pontos das mesmas. Tal correspondência é usada para definir uma força adicional usada para retirar uma snake de um mínimo local.

Vejamos esta força em detalhes.

Força Adicional e Mínimos Locais

Para compreendermos a utilidade desta força consideremos a Figura 5.2:

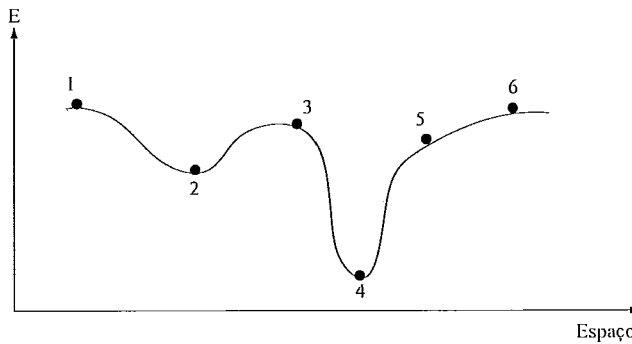


Figura 5.2: Perfil de energia.

Nesta figura temos um perfil de energia (aqui representada em 1D por simplicidade) da snake e vamos supor a mesma em uma configuração inicial com energia dada pela altura da posição 1.

Esta figura permite ilustrar os principais problemas associados aos métodos de minimização baseados em informações locais (steepest descent, Euler Lagrange, etc). Para qualquer um destes métodos, o resultado final partindo da posição 1 será uma configuração com energia dada pela posição 2, embora o resultado desejado seja o correspondente ao ponto 4.

A dificuldade aqui está no fato que estes métodos de otimização, pela natureza local dos mesmos, não podem *saber* que na posição 4 há um vale mais profundo que em 2.

Aqui é possível apresentar a principal contribuição do método Dual: usar duas snakes e com isto estabelecer um balanço entre posição e energia, a fim de ter garantias da globalidade da solução encontrada.

Assim, se inicializarmos uma snake na posição 1 e outra na posição 6 da Figura 5.2, quando a primeira estabilizar seu movimento na posição 2 poderemos evoluir a segunda snake e quando esta atingir o repouso comparar a posição e a energia das snakes. Neste caso, a segunda snake atingirá a posição 4 indicando diretamente que a posição 2 era um mínimo local mas não global.

Tal análise é ilustrativa, mas ao mesmo tempo grosseira se lembrarmos que a energia (5.7) está definida em espaço com dimensão $2N$, onde N é o número de snaxels.

Neste caso, podemos ter duas snakes parcialmente ajustadas ao contorno, mas em posições de equilíbrio (mínimos local da energia). Certamente, uma vez que as snakes têm o mesmo número de snaxels, podemos identificar que as mesmas estão distantes e que portanto ao menos uma está em um mínimo local. Escolhida uma snake para evoluir via algum critério a ser especificado, o próximo problema é como saber automaticamente qual snaxel deve ser movimentado?

Para abordar esta questão propõe-se em (Gunn & Nixon, 1997; Gunn, 1996a) um algoritmo guloso: movimenta-se inicialmente os snaxels que aumentam menos a energia. Rigorosamente, perde-se a garantia do ótimo com este procedimento. Na prática, advogam os autores em (Gunn & Nixon, 1997; Gunn, 1996a), que os resultados têm sido satisfatórios. A utilização de algoritmos gulosos em snakes foi estudada também em (Bulpitt & Efford, 1996).

Para movimentar os snaxels de uma snake retirando-a de um mínimo local é necessário usar uma força extra a qual denominamos *Força Adicional*. O efeito desta força pode ser ilustrado pela figura 5.2. Uma vez concluído que 2 não é ótimo global, esta força extra teria a finalidade de *puxar* a snake para fora do vale correspondente ao ponto 2 até que a mesma atinja a posição 3 quando então esta terá condições de evoluir naturalmente em direção ao ponto 4.

A força adicional é dada por:

$$F_{adicional} = g(t) \frac{u_i - v_i^t}{\|u_i - v_i^t\|}, \quad (5.10)$$

onde v_i^t é um snaxel do contorno sendo processado no instante t , u_i é o contorno em repouso neste instante e $g(t)$ é a intensidade da força.

O critério de parada adotado em (Gunn & Nixon, 1997) é do tipo dado pela expressão (3.48):

$$\max_i \|v_i^{t+1} - v_i^t\| < \delta, \quad (5.11)$$

onde δ é um parâmetro pré-estabelecido.

A equação de evolução do método Dual é finalmente dada por:

$$v_i^{t+1} = v_i^t + \frac{1}{2} \left(\lambda \frac{e_i}{h} + (1 - \lambda) F_i \right) + g(t) \frac{u_i - v_i^t}{\|u_i - v_i^t\|}, \quad (5.12)$$

onde F_i é dada pelo gradiente do potencial externo.

O método Dual aqui descrito é também denominado *Contorno Ativo Dual Baseado em Evolução* para diferenciar do Dual baseado em programação dinâmica que apresentaremos a seguir.

O algoritmo correspondente ao método Dual descrito nesta seção pode ser resumido da seguinte forma.

Algoritmo

Inicialização: Interação com o usuário para definir snakes interna e externa,

Passo 1: Selecionar a snake com energia (5.7) mais elevada,

Passo 2: Se o movimento desta snake satisfizer a condição de terminação aumentar a força de arrasto (5.10) até que o movimento da snake ultrapasse o parâmetro δ em (5.11),

Passo 3: Quando a energia da snake começar a decrescer, remover a força de arrasto e evoluir a snake até seu equilíbrio,

Passo 4: Se as snakes ocupam a mesma posição, parar a execução. Caso contrário voltar ao passo 1.

Para comparar a posição das snakes no passo 4 acima é usado uma norma no espaço dos Descritores de Fourier das curvas; isto é, dados dois contornos u e v a distância entre eles é definida por:

$$d(u, v) = \left[\sum_{n=-M}^M |\mu_n - \nu_n|^2 \right]^{\frac{1}{2}},$$

onde μ_n e ν_n são os Descritores de Fourier para as curvas u e v , respectivamente (Jain, 1989).

5.1.2 Metodologia do Dual

O método Dual evolucionário tem sua metodologia fundamentada em três pontos:

- 1) Balanço entre a energia da snake interna e da externa para escolher a snake a ser processada,
- 2) Critério para decidir quais snaxel de uma snake estabilizada pode ser movido,
- 3) Critério final de parada: snakes próximas.

Destes itens, o mais crítico é o segundo. Isto porque, se este critério não for bem definido, porções de uma snake já acomodadas sobre a borda desejada podem ser retiradas desta posição, o que implica em perda de eficiência e performance.

5.1.3 AI-Snakes: Modelo de Forma Ativo

O modelo Dual da seção 5.1.1 incorpora um modelo de forma local.

Nesta seção, analisaremos um caso onde o modelo de forma é global. Nosso ponto de partida será o conhecimento a priori de um *protótipo* do objeto, e o fato de que a borda procurada é uma deformação deste protótipo.

Assim, é natural que o modelo de contorno ativo incorpore tanto a fidelidade ao protótipo quanto seja invariante pelas transformações que geram as possíveis deformações do objeto.

Em (Ip & Shen, 1998) a invariância é por transformações afins sobre a curva (translação e transformações lineares não-singulares). Neste método é desenvolvida uma noção de curvatura afin-invariante empregada como elemento de suavização da solução e uma energia de modelo que vincula a forma da snake ao *protótipo* em questão.

Para que o vínculo entre o protótipo e a snake seja estabelecido é necessário uma correspondência entre os pontos do protótipo e da snake, o que aliás é uma característica comum dos *modelos de forma ativa* (Ip & Shen, 1998).

Este método é apropriado para segmentação de cenas onde objetos com formas previamente conhecidas foram deformados por transformações afins ou que possam ser aproximadas por uma seqüência de transformações deste tipo.

Seja, por exemplo, uma seqüência de fotos de um mesmo objeto tiradas de diferentes posições. As transformações no caso são *projetivas* as quais podem ser aproximadas por uma seqüência de transformações afins. O objeto fotografado (mais exatamente, sua fronteira) nos fornece o protótipo.

Neste caso, a aplicação da noção de invariância tem a finalidade de manter fidelidade ao *protótipo* e às transformações correspondentes. Ao mesmo tempo a energia interna deverá garantir também a suavidade necessária para a solução final.

Iniciemos pela análise da noção de curvatura inerente ao modelo original de snakes (seção 3.1).

Curvatura

Inicialmente, vejamos porque a curvatura gaussiana não é invariante por transformações afins. Vamos retomar a descrição continua para curvas.

Seja então uma curva suave

$$c(\tau) = (x(\tau), y(\tau)), \quad \tau \in \mathcal{R}.$$

Neste caso, a curvatura gaussiana $k(\tau)$ é dada por:

$$k(\tau) = \frac{\dot{x}(\tau)\ddot{y}(\tau) - \ddot{x}(\tau)\dot{y}(\tau)}{\left[(\dot{x}(\tau))^2 + (\dot{y}(\tau))^2\right]^{3/2}} \quad (5.13)$$

onde:

$$\dot{x}(\tau) = \frac{dx}{d\tau}, \ddot{x}(\tau) = \frac{d^2x}{d\tau^2},$$

idem para y .

Seja s o comprimento de arco correspondente à curva c . Então nós teremos:

$$ds^2 = (dx)^2 + (dy)^2,$$

Assim a expressão (5.13) pode ser reescrita como:

$$k(\tau) = \frac{\begin{vmatrix} \dot{x}(\tau) & \ddot{x}(\tau) \\ \dot{y}(\tau) & \ddot{y}(\tau) \end{vmatrix} \cdot (d\tau)^3}{(ds)^3} = \frac{ang}{dist} \quad (5.14)$$

onde:

$$ang = \begin{vmatrix} \dot{x}(\tau) & \ddot{x}(\tau) \\ \dot{y}(\tau) & \ddot{y}(\tau) \end{vmatrix} \cdot (d\tau)^3 \text{ e } dist = (ds)^3.$$

Vejamos como ang e $dist$ se comportam sob transformações afins. Seja então:

$$\begin{pmatrix} u(\tau) \\ v(\tau) \\ 1 \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & b_1 \\ a_{21} & a_{22} & b_2 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x(\tau) \\ y(\tau) \\ 1 \end{pmatrix} \quad (5.15)$$

a curva c transformada pela transformação afim definida pela matriz quadrada de ordem 3 do lado direito desta igualdade. Então, um cálculo direto mostra que:

$$ang = \begin{vmatrix} \dot{u}(\tau) & \ddot{u}(\tau) \\ \dot{v}(\tau) & \ddot{v}(\tau) \end{vmatrix} \cdot (d\tau)^3,$$

é transformado da seguinte forma:

$$ang = \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} \begin{vmatrix} \dot{x}(\tau) & \ddot{x}(\tau) \\ \dot{y}(\tau) & \ddot{y}(\tau) \end{vmatrix} (d\tau)^3 = (a_{11}a_{22} - a_{12}a_{21}) \cdot ang. \quad (5.16)$$

Esta expressão mostra que o valor de ang é simplesmente alterado por um fator de escala linear dado pelo determinante da matriz que define a transformação. Para $dist$ teremos:

$$\begin{pmatrix} du \\ dv \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \cdot \begin{pmatrix} dx \\ dy \end{pmatrix}, \quad (5.17)$$

e assim teremos:

$$dist = ((du)^2 + (dv)^2)^{3/2} = ((a_{11}dx + a_{12}dy)^2 + (a_{21}dx + a_{22}dy)^2)^{3/2}, \quad (5.18)$$

o que mostra que o valor de $dist$ é alterado de forma não linear por transformações afins e assim $dist$ não pode ser tornado afim-invariante exceto em situações especiais. Logo, fica demonstrado que a curvatura Gaussiana não é afim-invariante, e conseqüentemente o modelo original de snakes (seção 3.1) também não (veja que a curvatura Gaussiana aparece explicitamente na equação (3.8)).

Área obtida da curvatura é afim-invariante

Vamos mostrar agora que uma versão discreta para $(1/2) ang$ é a área de um triângulo definido por três vértices da curva c . Seja então as aproximações das derivadas em (5.13) por diferenças finitas:

$$\begin{aligned} \dot{x}(\tau) &\doteq \frac{x(\tau) - x(\tau - \Delta\tau)}{\Delta\tau}, \\ \dot{y}(\tau) &\doteq \frac{y(\tau) - y(\tau - \Delta\tau)}{\Delta\tau}, \\ \ddot{x}(\tau) &\doteq \frac{x(\tau) - 2x(\tau - \Delta\tau) + x(\tau - 2\Delta\tau)}{(\Delta\tau)^2}, \\ \ddot{y}(\tau) &\doteq \frac{y(\tau) - 2y(\tau - \Delta\tau) + y(\tau - 2\Delta\tau)}{(\Delta\tau)^2}. \end{aligned} \quad (5.19)$$

Usando estas aproximações podemos escrever o numerador da expressão (5.14) da seguinte forma:

$$\frac{1}{2}ang \doteq \frac{1}{2} \begin{vmatrix} x(\tau - 2\Delta\tau) & x(\tau - \Delta\tau) & x(\tau) \\ y(\tau - 2\Delta\tau) & y(\tau - \Delta\tau) & y(\tau) \\ 1 & 1 & 1 \end{vmatrix}, \quad (5.20)$$

o qual é a área do triângulo formado pelos vértices:

$$(x(\tau - 2\Delta\tau), y(\tau - 2\Delta\tau)), (x(\tau - \Delta\tau), y(\tau - \Delta\tau)), (x(\tau), y(\tau)).$$

Vejamos como se comporta a área dada por (5.20) sob transformações afins. Neste caso teremos:

$$area_{\tau_1 \tau_2 \tau_3} = \frac{1}{2} \begin{vmatrix} x(\tau_1) & x(\tau_2) & x(\tau_3) \\ y(\tau_1) & y(\tau_2) & y(\tau_3) \\ 1 & 1 & 1 \end{vmatrix}, \quad (5.21)$$

Denotemos por $area_{\tau_1 \tau_2 \tau_3}^{transf}$ esta área transformada de acordo com a expressão (5.15); isto é:

$$area_{\tau_1 \tau_2 \tau_3}^{transf} = \frac{1}{2} \begin{vmatrix} u(\tau_1) & u(\tau_2) & u(\tau_3) \\ v(\tau_1) & v(\tau_2) & v(\tau_3) \\ 1 & 1 & 1 \end{vmatrix}, \quad (5.22)$$

Expressando agora 5.22 acima em função da área original $area_{\tau_1 \tau_2 \tau_3}$ obteremos:

$$area_{\tau_1 \tau_2 \tau_3}^{transf} = \frac{1}{2} \left| \begin{pmatrix} a_{11} & a_{12} & b_1 \\ a_{21} & a_{22} & b_2 \\ 0 & 0 & 1 \end{pmatrix} \right| \left| \begin{vmatrix} x(\tau_1) & x(\tau_2) & x(\tau_3) \\ y(\tau_1) & y(\tau_2) & y(\tau_3) \\ 1 & 1 & 1 \end{vmatrix} \right| = (a_{11}a_{22} - a_{12}a_{21}) area_{\tau_1 \tau_2 \tau_3}. \quad (5.23)$$

Esta expressão mostra que a área de qualquer triângulo, quando apropriadamente normalizada, é invariante por transformações afins. Com isto estamos em condições de definir uma energia interna invariante por estas transformações.

Energia Interna

A energia interna da AI-snake é dada pela seguinte expressão:

$$E_{int} = E_{suav} + E_{modelo}, \quad (5.24)$$

onde E_{suav} e E_{modelo} são dados como segue.

Para definir o primeiro termo retomemos o resultado (5.23) e através dele definamos uma noção de curvatura baseada em área e que seja afim-invariante.

Algumas considerações iniciais devem ser feitas. Seja inicialmente a Figura 5.3.

Uma vez que tencionamos usar área para definir curvatura e por sua vez o termo de suavização E_{suav} em (5.24) devemos estar conscientes que minimizar este termo de energia implicará em minimizar área e portanto devemos nos precaver quanto às situações na Figura 5.3.(c). Felizmente, estas possibilidades ficam descartadas uma vez que o efeito dos outros termos de energia e da direção de busca mostrada na Figura 5.3.(a) impedem que a mesma ocorra (Ip & Shen, 1998).

Baseados nestas idéias o termo de suavização será definido por:

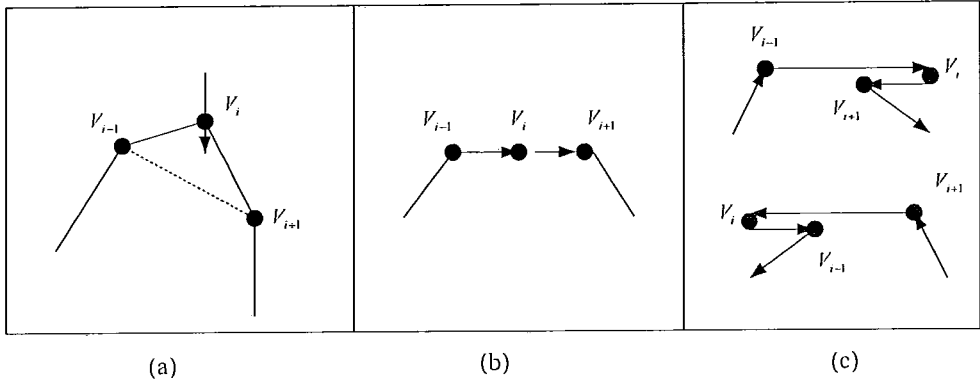


Figura 5.3: Configurações possíveis para vértices vizinhos.

$$E_{suav} = \sum_{i=0}^{N-1} e_{suav}(v_i), \quad (5.25)$$

onde:

$$e_{suav}(v_i) = \alpha \frac{|area_i|}{AREA_{now}} \frac{AREA_{modelo}}{\frac{1}{2}d^2}, \quad (5.26)$$

sendo α um parâmetro, $AREA_{now}$ a área da snake na presente interação, a qual pode ser calculada por:

$$AREA_{now} = \frac{1}{2} \sum_{i=0}^{N-1} \begin{vmatrix} u_i & u_{i+1} & u_{cent} \\ v_i & v_{i+1} & v_{cent} \\ 1 & 1 & 1 \end{vmatrix}$$

onde (u_{cent}, v_{cent}) são as coordenadas do centróide da snake dadas por:

$$\begin{pmatrix} u_{cent} \\ v_{cent} \end{pmatrix} = \frac{1}{N} \sum_{i=0}^{N-1} \begin{pmatrix} u_i \\ v_i \end{pmatrix}. \quad (5.27)$$

$AREA_{modelo}$ é a área do protótipo (modelo) e d é uma constante representando a média das distâncias entre os snaxels no protótipo. Pelo resultado (5.23) segue diretamente que E_{suav} é afim-invariante.

O termo E_{modelo} em (5.24) é definido para garantir a fidelidade ao protótipo em questão. Este protótipo, denominado $c_{protótipo}$, é definido por uma poligonal fechada dada pelo conjunto de pontos $\{W_i = (x_i, y_i) | i = 0, \dots, N-1\}$.

A snake c_{snake} pode ser vista como uma versão deformada e afim-transformada do protótipo $c_{protótipo}$. Uma vez determinado a correspondência entre os pontos de

$c_{protótipo}$ e os pontos de c_{snake} é possível comparar o protótipo transformado com a snake atual (Figura 5.4 a seguir).

O protótipo transformado é denominado $c_{alinhado}$ e é obtido pela expressão:

$$\begin{pmatrix} x_i^{alinhado} \\ y_i^{alinhado} \end{pmatrix} = A \begin{pmatrix} x_i \\ y_i \end{pmatrix}, \quad (5.28)$$

onde a matriz A representa uma transformação afim.

A idéia central é obter uma energia de modelo que aumente caso a *forma* da snake se afaste daquela do protótipo transformado segundo a expressão (5.28).

Uma vez que a área de um triângulo pode ser convenientemente normalizada para torna-se afim-invariante define-se a energia E_{modelo} baseada também em área.

Da Figura 5.4 nós vemos que o processo de tornar V_i próximo de $W_i^{alinhado}$ é equivalente a fazer as áreas totais dos triângulos definidos pelos pontos $(W_{i-1}^{alinhado}, V_i, W_i^{alinhado})$ e $(W_i^{alinhado}, V_i, W_{i+1}^{alinhado})$ próximo de zero. A energia E_{modelo} pode portanto ser definida por:

$$E_{modelo} = \sum_{i=0}^{N-1} e_{modelo}(V_i), \quad (5.29)$$

tal que:

$$e_{modelo}(V_i) = \beta \frac{\left| S_{W_{i-1}^{alinhado}, V_i, W_i^{alinhado}} \right| + \left| S_{W_i^{alinhado}, V_i, W_{i+1}^{alinhado}} \right|}{AREANow} \frac{AREAModelo}{d^2}, \quad (5.30)$$

onde $S_{W_{i-1}^{alinhado}, V_i, W_i^{alinhado}}$ e $S_{W_i^{alinhado}, V_i, W_{i+1}^{alinhado}}$ são as áreas dos triângulos definidos pelos pontos $(W_{i-1}^{alinhado}, V_i, W_i^{alinhado})$ e $(W_i^{alinhado}, V_i, W_{i+1}^{alinhado})$ respectivamente e β é um parâmetro de suavização (o valor para d e $AREAModelo$ foram definidos acima).

Energia Externa

Uma vez que nos propomos manter uma certa fidelidade entre a snake e o protótipo devemos definir a energia externa em conformidade com esta proposta. Isto pode ser feito usando-se informações direcionais do gradiente. Para cada snake, a energia externa será definida pelo produto interno do vetor normal à snake com a direção do gradiente, pesado pela magnitude do gradiente em questão. Assim teremos:

$$E_{ext} = \sum_{i=0}^{N-1} e_{ext}(v_i), \quad (5.31)$$

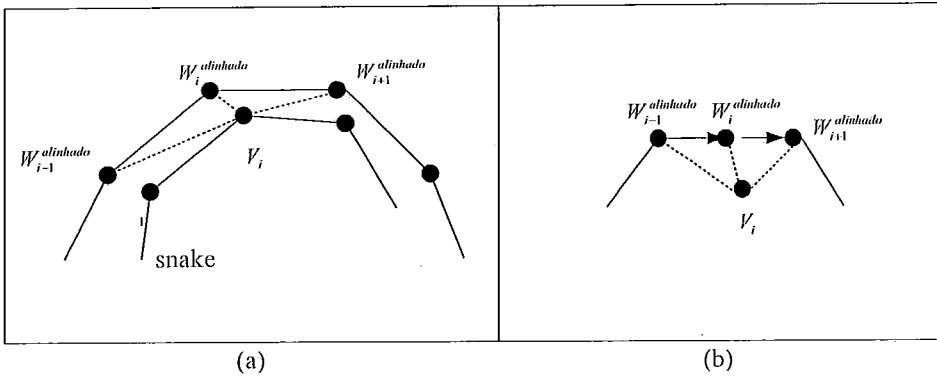


Figura 5.4: Comparação entre a snake e o protótipo alinhado.

tal que:

$$e_{ext}(v_i) = \gamma (1 - |\nabla I(v_i)| |n(v_i) \cdot h(v_i)|), \quad (5.32)$$

onde $n(v_i)$ é a normal à snake no ponto v_i tomada com sentido para o interior da snake.

Energia Total

A energia total usada no AI-Snake é finalmente dada por:

$$E_{snake} = \alpha \sum_{i=0}^{N-1} \frac{|area_i|}{AREA_{now}} \frac{AREA_{modelo}}{\frac{1}{2}d^2} + \beta \sum_{i=0}^{N-1} \frac{|S_{W_{i-1}^{alinhado}, v_i, W_i^{alinhado}}| + |S_{W_i^{alinhado}, v_i, W_{i+1}^{alinhado}}|}{AREA_{now}} \frac{AREA_{modelo}}{d^2} + \gamma \sum_{i=0}^{N-1} (1 - |\nabla I(v_i)| |n(v_i) \cdot h(v_i)|), \quad (5.33)$$

onde α , β e γ são os parâmetros do modelo, a serem estabelecidos a priori.

O problema de correspondência entre a snake e protótipo é tratado em detalhes em (Ip & Shen, 1998). Não abordaremos este problema nesta apresentação.

Uma vez determinado esta correspondência, é imediato obter a transformação afim que melhor relaciona as duas curvas.

Sem perda de generalidade, vamos supor que o ponto $W_{i+j_0} = (u_{i+j_0}, v_{i+j_0})$ do protótipo corresponde ao ponto $v_i = (x_i, y_i)$ da snake. Então, procuremos uma matriz A que minimize o erro ε_A dado por:

$$\varepsilon_A = \|Q' - AQ\|^2, \quad (5.34)$$

onde A é a matriz que representa a transformação afim procurada e:

$$Q = \begin{pmatrix} x_1 & x_2 & x_N \\ y_1 & y_2 & y_N \\ 1 & 1 & 1 \end{pmatrix}, \quad (5.35)$$

$$Q' = \begin{pmatrix} u_{1+j_0} & u_{2+j_0} & u_{N+j_0} \\ v_{1+j_0} & v_{2+j_0} & v_{N+j_0} \\ 1 & 1 & 1 \end{pmatrix}, \quad (5.36)$$

são as matrizes dos pontos da snake e do protótipo respectivamente. A solução de (5.34) fornece a transformação afim que melhor relaciona a snake e o protótipo segundo o critério de mínimos quadrados. Esta solução é dada por:

$$A = (Q'Q^T) (QQ^T)^{-1}.$$

5.2 Modelo Dual Baseado em Programação Dinâmica

Na seção 5.1.1 descrevemos um método Dual baseado na equação de evolução (5.12). Nesta seção discutiremos uma versão do método Dual desenvolvida em (Gunn, 1996b) a qual é baseada nos conceitos de programação dinâmica discreta apresentados na seção 4.1 do capítulo anterior.

Consideremos inicialmente uma situação como aquela da Figura 4.1 onde supomos que a borda de interesse está no interior da região limitada pelas duas curvas destacadas.

Procedendo exatamente como na seção 4.1 vamos montar o espaço de busca estabelecendo inicialmente uma correspondência entre N pontos em cada curva e em seguida conectando estes pontos com segmentos de retas discretizados em M pontos (Figura 4.2). O espaço de busca final contém NM pontos.

O algoritmo de busca que passamos a descrever é denominado Viterbi (Gunn, 1996b; Bamford & Lovell, 1995). Este problema tem as características dos *problemas de decisão finitos em multi-estágio* abordados na seção 4.1. A única dificuldade é que procuramos agora contornos fechados e não abertos como foi o caso da discussão no capítulo anterior.

Para lidar com contornos fechados a solução adotada é usar duas buscas (Gunn, 1996b). Na primeira, escolhem-se dois segmentos adjacentes no espaço de busca e

então faz-se uma busca (equação (4.8)) para obter o contorno aberto ótimo para estes segmentos (Figura 5.5.(a)).

Em seguida, escolhem-se dois pontos no meio da primeira solução para serem os pontos inicial e final para a próxima busca. Aplica-se a expressão recursiva (4.8) novamente para então obter a solução final (Figura 5.5.(b)).

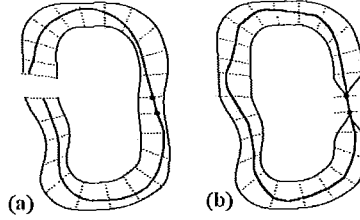


Figura 5.5: Programação Dinâmica para contornos fechados.

O funcional de energia para os contornos abertos é do tipo daquele apresentado em (4.2) e tem a seguinte forma (Gunn, 1996b):

$$E_{snake} = \sum_{i=0}^{N-3} E_i \quad (5.37)$$

onde:

$$E_i = \alpha E_{int}(v_i, v_{i+1}, v_{i+2}) + \beta E_{ext}(v_{i+1}) + \lambda E_{linha}(v_{i+1}), \quad (5.38)$$

com E_{int} , E_{ext} e E_{linha} dados por:

$$E_{int}(v_i, v_{i+1}, v_{i+2}) = \left(\frac{v_{i+2} - 2v_{i+1} + v_i}{\|v_{i+2} - v_i\|} \right)^2, \quad (5.39)$$

$$E_{ext}(v_i) = -\|\nabla I(v_i)\|^2, \quad (5.40)$$

$$E_{linha}(v_{i+1}) = \pm I(v_{i+1}). \quad (5.41)$$

Como já dito acima, a relação de recorrência correspondente a ser usada tem a forma clássica dada por (4.8).

5.3 Métodos envolvendo mudanças Topológicas

Analisaremos nesta seção dois modelo alternativo que têm a finalidade de corrigir as limitações topológicas do modelo original.

5.3.1 Método dos Conjuntos de Níveis

A idéia central nesta formulação, chamada *Conjunto de Níveis (Level Sets)* (Malladi *et al.*, 1995), é representar a snake como o conjunto de nível $\{x \in \mathbb{R}^2 | G(x) = 0\}$ de uma função $G : \mathbb{R}^2 \rightarrow \mathbb{R}$.

Esta idéia tem sua origem nos trabalhos envolvendo propagação de interfaces em física de materiais onde uma frente (modificação do meio) se desloca alterando o mesmo (Sethian, 1988). Por este motivo a terminologia usual desta formulação (implícita) para contornos ativos usa o termo *frente* em lugar de snake.

Sejam então uma função G , a qual assumiremos agora dependente do espaço e também do tempo:

$$G : \mathbb{R}^2 \times \mathbb{R}^+ \rightarrow \mathbb{R}, \quad (5.42)$$

tal que a frente em questão, no instante $t = 0$ seja dada por uma curva c :

$$c(s, t = 0) = (x | G(x, t = 0) = 0). \quad (5.43)$$

O próximo passo será encontra a partir destes elementos uma formulação Eule-riana para o movimento da frente c .

Seguindo o trabalho de Setian (Malladi *et al.*, 1995) vamos supor que a frente se propaga ao longo da sua direção normal com velocidade \vec{F} , onde \vec{F} pode ser uma função de vários elementos incluindo curvatura, direção normal, etc.

A Figura 5.6 ilustra o problema para o caso onde a função G tem a forma inicial de um parabolóide e a curva definida pela expressão (5.43) tem a forma de uma circunferência.

Nestas condições, necessitamos de uma equação para a evolução da função $G(x, t)$ a qual contem o movimento de $c(s, t)$ como o conjunto de nível dado por:

$$c(s, t) = (x | G(x, t) = 0). \quad (5.44)$$

Seja então $x(t)$, $t \in \mathbb{R}^+$ a trajetória de um ponto da frente que se propaga. Isto é, $x(t = 0)$ é um ponto na frente inicial $c(s, t = 0)$ e $x_t = \vec{F}(x, t)$ com o vetor x_t normal à frente no ponto $x(t)$. Uma vez que a função G é sempre zero sobre a frente, nós devemos ter:

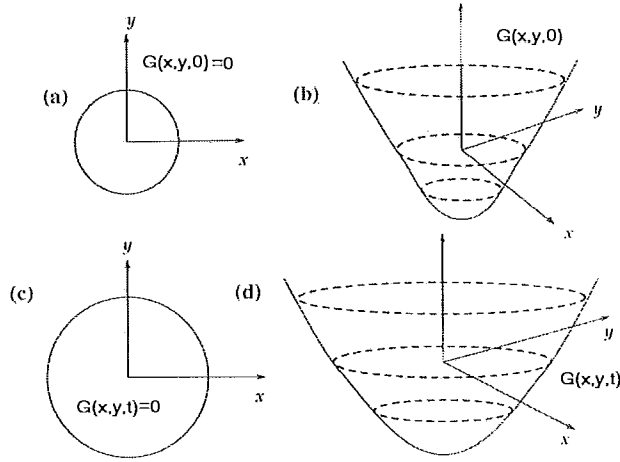


Figura 5.6: Método dos Conjuntos de Níveis.

$$G(x(t), t) = 0. \quad (5.45)$$

Supondo G diferenciável tendo zero como um valor regular, o Teorema da Função Implícita nos permite derivar (5.45) com relação ao tempo para obter:

$$G_t + \sum_{i=1}^N G_{x_i} x_{i_t} = 0, \quad (5.46)$$

onde x_i é a i th componente de x . Lembrando que o somatório na última expressão é exatamente o produto escalar entre o gradiente (com relação a x) de G e o vetor velocidade podemos escrever:

$$\sum_{i=1}^N G_{x_i} x_{i_t} = \nabla G \cdot \vec{F} = \|\nabla G\| \|\vec{F}\|, \quad (5.47)$$

obtemos finalmente a equação procurada:

$$G_t + F \|\nabla G\| = 0, \quad (5.48)$$

onde $F = \|\vec{F}\|$.

A equação (5.48) tem como condição inicial a função $G(x, t = 0)$ a qual deve ser definida a priori. Para certas escolhas da função de velocidade F esta equação

assume a forma de uma equação de Hamilton-Jacobi, bem conhecidas em mecânica clássica (Goldstein, 1981).

O condição inicial $G(x, t = 0)$ pode ser definida naturalmente pela expressão

$$G(x, t = 0) = \pm d, \tag{5.49}$$

onde d é a distância de x a $c(s, t = 0)$, e o sinal indica se o ponto é interior (-) ou exterior (+) à curva inicial.

Existem quatro vantagens desta formulação Euleriana baseada na equação de Hamilton-Jacobi acima. A primeira provém do fato de que a função $G(x, t)$ sempre permanece uma função diferenciável desde que F seja suave (Malladi *et al.*, 1995). Contudo, o conjunto de nível $\{G(x, t) = 0\}$, e portanto a frente $c(s, t)$ pode mudar a topologia, dividindo-se ou agregando-se, durante a evolução da função $G(x, t)$ (Malladi *et al.*, 1995), incorporando operações topológicas naturalmente.

A segunda vantagem diz respeito à análise numérica. Se $G(x, t)$ permanece uma função diferenciável durante sua evolução, nós podemos usar diferenças finitas para discretizar a equação (5.48) tanto no tempo quanto no espaço.

Contudo, a técnica correta para aproximar as derivadas espaciais na equação (5.48) vem de uma condição de entropia apropriada para frentes propagando-se, a qual é discutida em detalhes em (Sethian, 1996). Uma condição de entropia comumente usada é a seguinte: *uma vez que um ponto foi queimado, ele permanece queimado*; isto é, se um ponto se tornou interior à frente então ele permanecerá interior durante toda a evolução da mesma (Malladi *et al.*, 1995).

Para obter um esquema numérico para discretizar a equação (5.48) que respeite esta condição de entropia usa-se técnicas em *leis de conservação hiperbólicas*. Em (Sethian, 1996) encontramos uma discussão detalhada a respeito.

A terceira vantagem desta formulação implícita é que a geometria intrínseca da frente pode ser facilmente determinada a partir da função G . Por exemplo o vetor normal à frente e a curvatura da mesma são facilmente obtidos pelas expressões:

$$\vec{n} = \nabla G,$$

$$K = \nabla \cdot \left(\frac{\nabla G}{\|\nabla G\|} \right),$$

respectivamente, onde o gradiente e o operador divergente ($\nabla \cdot$) são todos calculados em relação a x .

Por fim a última vantagem é que não existe diferença significativa em seguir frentes em 2D ou 3D. Para isto basta estender as estruturas de dados e os operadores

diferenciais. Assim, superfícies em propagação são facilmente formuladas através de uma equação análoga a (5.48).

Extração de Bordas

Nesta seção, nós descrevemos como a formulação acima pode ser usada para extração de bordas.

Primeiramente, notemos que a frente é uma curva que evolui no tempo. Uma vez que a idéia é extrair bordas em uma dada imagem, a frente deve ser forçada a parar na vizinhança da borda desejada, analogamente ao que apresentamos para as snakes paramétricas já discutidas. Assim, o problema está em como definir adequadamente a função velocidade F na equação (5.48). Em (Sethian, 1996) encontramos uma discussão detalhada a respeito.

Uma possibilidade interessante para F é discutida em (Sapiro, 1997) sendo dada pela expressão a seguir:

$$\begin{aligned} \frac{\partial G}{\partial t} &= \|\nabla G\| \nabla \cdot \left(g_{cinza}(I) \frac{\nabla G}{\|\nabla G\|} \right) \\ &= g_{cinza}(I) \|\nabla G\| \nabla \cdot \left(\frac{\nabla G}{\|\nabla G\|} \right) + \nabla g_{cinza}(I) \cdot \nabla G, \end{aligned} \quad (5.50)$$

onde $g_{cinza}(r) \rightarrow 0$ se $r \rightarrow +\infty$.

Nesta expressão, o segundo termo da segunda igualdade dirige a frente para as fronteiras dos objetos, isto é, para os vales de g_{cinza} . O primeiro termo incorpora explicitamente a curvatura (veja equação 5.3.1) tendo o papel de suavizar o resultado final (Malladi *et al.*, 1995; Sethian, 1996).

5.3.2 T-Snakes

O método dos conjuntos de níveis da seção anterior tem em sua base teórica elementos de geometria e topologia diferencial (geodésicas, Teorema da Função Implícita, etc).

O método das T-Snakes (McInerney & Terzopoulos, 1995; McInerney, 1997) que passaremos a apresentar agora aborda as questões topológicas num outro contexto. A idéia fundamental é converter o problema para o contexto da Topologia Combinatória a fim de obter uma formulação paramétrica com habilidades topológicas para as snakes.

No caso dos conjuntos de níveis inicializamos o método definindo a função G a qual tem a snake inicial como conjunto de nível e zero como um valor regular.

No presente caso, vamos construir uma função linear por partes χ definida sobre uma triangulação do plano e cuja solução para $\chi(x) = 0$ corresponda à snake.

Tendo como base o Princípio Door-In-Door-Out da seção 4.3, poderemos aplicar o algoritmo do final da seção 4.3.2 (Algoritmo Door-In-Door-Out) para obter cada uma das componentes conexas da snake resolvendo assim as mudanças topológicas (splits e merges).

A função linear por partes em questão é denominada *função característica* em (McInerney & Terzopoulos, 1995; McInerney, 1997) podendo ser definida por:

$$\chi : D \subset \mathbb{R}^2 \rightarrow \{0,1\} \tag{5.51}$$

tal que: $\chi(p) = 1$ se $p \in O$ e $\chi(p) = 0$, caso contrário, onde p é um nó da malha.

Para construir esta função o método usado consiste primeiramente em projetar as curvas (snakes) sobre a triangulação usada como mostra a Figura 5.7 a seguir.

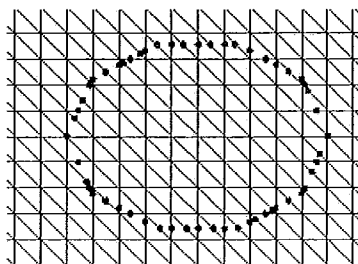


Figura 5.7: Snake projetada sobre uma triangulação do tipo Coxeter-Freudenthal.

Os pontos assim obtidos serão usados posteriormente para definir os snaxels da snake. Feito esta projeção a determinação de χ pode ser feita por um algoritmo do tipo scan line (Rogers, 1985). A Figura 5.8 é um exemplo de como a função χ e o algoritmo da seção 4.3 fornecem naturalmente o merge entre duas curvas.

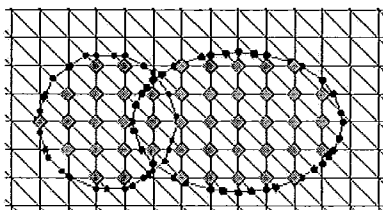


Figura 5.8: Merge entre duas curvas. Nós marcados são aqueles onde χ tem valor 1.

Nesta figura, os nós marcados são aqueles onde a função característica tem valor 1 e os demais onde esta função tem valor 0. Aplicando o algoritmo Door-In-Door-Out, obteremos ao final a seqüência dos triângulos de borda. O merge entre as curvas pode então ser representado por uma poligonal fechada contida na região dada pela união destes triângulos.

Observe que não precisamos determinar precisamente a solução de $\chi(x) = 0$. O que precisamos é a segurança de que a seqüência aresta-triângulo-aresta obtida pelo algoritmo é cíclica e finita, e assim, que o método é consistente. Estas propriedades foram apresentadas no final da seção 4.3 e são fundamentais para o método das T-Snakes.

Além da triangulação do espaço (no caso, o \mathbb{R}^2), da projeção da snake e da função característica, um outro componente do método das T-Snakes é um modelo discreto de contorno ativo, o qual passaremos a definir agora.

Modelo de Snake

A T-Snake, assim como o contorno ativo Dual descrito acima, é definida por um conjunto de N pontos (snaxels) cujas posições $\{v_i = (x_i, y_i), i = 0, \dots, N - 1\}$, são conectadas em seqüência formando uma curva fechada.

Cada par de pontos consecutivos v_i, v_{i+1} é chamado um "elemento do modelo".

Os pontos da snake são supostos ligados por molas as quais são definidas por um parâmetro de elasticidade a_i e um comprimento natural l_i , isto é, a snake resiste a expansão ou compressão somente quando a distância entre snaxels $\|r(t)\| = \|v_{i+1} - v_i\|$ é maior ou menor que l_i , respectivamente.

Assim, dada a deformação $e_i = \|r_i(t)\| - l_i$, nós definimos a força de tensão correspondente pela expressão:

$$\alpha_i = a_i e_i r_i(t) - a_{i-1} e_{i-1} r_{i-1}(t). \quad (5.52)$$

Uma vez que o conjunto de snaxels e molas não permanece constante, o comprimento de repouso das molas no instante t é definido em termos dos comprimentos das molas no instante anterior $t - \Delta t$. Isto fornece ao modelo o comportamento de um material com viscoelasticidade não nula (McInerney, 1997).

Em adição à força (5.52) é conveniente definir uma força de rigidez para minimizar as curvaturas locais da snake garantindo suavidade para a solução. Esta força é definida da seguinte forma:

$$\beta_i = b_i \left(v_i - \frac{1}{2} (v_{i-1} + v_{i+1}) \right), \quad (5.53)$$

a qual atua no sentido de minimizar a distância entre um ponto v_i e o centróide do segmento definido pelos seus vizinhos.

O modelo tem também uma força normal:

$$F_i = sig(v_i) kn_i, \quad (5.54)$$

onde $sig(v_i) = -1$ se $I(v_i) < T$ e $sig(v_i) = 1$ caso contrário (T é um limiar para a intensidade de imagem previamente determinado).

Na expressão 5.54, n_i é a normal ao vértice v_i e k é fator que define a intensidade da força. Esta é uma força tipo balloon (seção 3.4), sendo usada para puxar a snake em direção às bordas dos objetos procurando evitar que a snake pare em regiões onde o campo externo é nulo.

Outra possibilidade para definir o sinal da força normal é usar estatísticas da imagem para definir a função $sig(v_i)$:

$$sig(v_i) = \begin{cases} 1 & \text{se } |I(v_i) - \mu| \leq \sigma, \\ 0, & \text{caso contrário,} \end{cases}$$

onde μ e σ são a média e variância da imagem, respectivamente.

As forças dadas (5.52)-(5.54) são forças internas.

As forças externas são definidas em função das características de interesse na imagem, no caso, bordas dos objetos da cena. Uma possibilidade é definir o campo externo através do gradiente do potencial usual:

$$P = - \|\nabla I\|^2, \quad (5.55)$$

ou de uma normalização do tipo daquela usada no Balloon (seção 3.4):

$$f = \begin{cases} 0 & \text{se } \nabla P < T_F, \\ \lambda \frac{\nabla P}{|\nabla P|} & \text{se } \nabla P > T_F, \end{cases}$$

onde T_F é um limiar previamente estabelecido e λ é um fator de escala para a força.

A equação de evolução para a T-Snake é finalmente dada por:

$$v_i^{(t+\Delta t)} = v_i^t + \frac{\Delta t}{\gamma} (\alpha_i^t + \beta_i^t + F_i^t + f_i^t), \quad (5.56)$$

Durante a evolução da T-Snake alguns nós da malha tornam-se interiores à mesma (supondo expansão). Estes nós são denominados *nós queimados*, em analogia ao método dos conjuntos de níveis da seção 5.3.1, o qual inspirou o presente método (McInerney, 1997).

Para evitar os problemas conhecidos para evolução de curvas na direção normal (desenvolvimento singularidades) o método das T-Snakes adota uma condição de *entropia* do tipo da usada no método dos conjuntos de níveis (McInerney, 1997): *uma vez que um nó é queimado, ele permanecerá queimado.*

Esta condição tem também a finalidade de permitir a definição de um critério de parada eficiente (daí o nome entropia). Neste sentido, primeiramente define-se uma *temperatura* para cada snaxel a qual é o número de deformações que o triângulo em que o elemento de modelo correspondente se encontra permaneceu como um triângulo de borda.

Uma T-Snake é considerada em equilíbrio quando a temperatura de todos os seus snaxels ultrapassar um limiar denominado *ponto de congelamento* (*freezing point*). Este limiar é estabelecido empiricamente.

A condição de entropia acima torna esta definição mais eficiente como critério de parada, pois do contrário uma snake poderia oscilar, expandindo e contraindo, e assim o número de interações necessárias para atingir o equilíbrio poderia ser muito alto. No entanto, esta condição limita o movimento da snake o que pode trazer dificuldades como veremos a seguir.

Uma vez que a T-Snake atingiu o equilíbrio, a malha correspondente à divisão simplicial pode ser descartada e a snake evoluir como uma snake discreta segundo a equação (5.56)

Atualização da Função Característica

Ao final de cada passo de evolução da snake, devemos atualizar a função característica definida em (5.51). Isto é equivalente a determinar o conjunto de nós da malha que foram queimados durante a evolução da snake. Passaremos agora a descrever o algoritmo correspondente (McInerney, 1997).

Cada elemento de modelo (par de snaxels consecutivos) pode ter passado sobre nenhum apenas um ou vários nós durante sua evolução. Inicialmente, para cada elemento de modelo nós formamos um polígono usando a atual posição e a posição anterior. Este polígono permite facilmente determinar os nós que podem ter sido queimados (Figura 5.9).

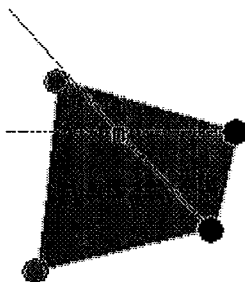


Figura 5.9: Polígono contendo nó interior (queimado).

Para cada nó no interior do polígono nós particionamos a imagem em 4 subes-

paços através das semi-retas indicadas na Figura 5.10:

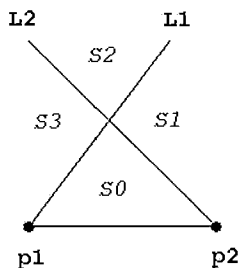


Figura 5.10: Subespaços usados para classificação de nós.

Estas linhas são denominadas $L1$ e $L2$ na figura acima e são formadas ligando o snaxel p_1 com o nó v e o snaxel p_2 com o nó v , respectivamente.

Durante um passo de evolução, p_1 e p_2 são movidos para novas posições p_{1n} e p_{2n} respectivamente. O passo máximo de cada snaxel é restringido a um valor muito menor que a dimensão do domínio da imagem evitando assim comportamentos degenerados. Com isto, podemos assumir que p_{1n} e p_{2n} podem estar em qualquer um dos 4 subespaços da Figura 5.10. Portanto, temos um total de 16 possibilidades a considerar mostradas nas Figuras 5.11.(1)-5.11.(16).

Os quatro pontos p_1 , p_2 , p_{1n} e p_{2n} formam o polígono fechado Q que pode ser convexo ou não (Figura 5.11). Vejamos duas definições importantes (McInerney, 1997).

(1). Um ponto p é dito interior ao polígono Q se um raio partindo de p intercepta exatamente uma aresta de Q ou exatamente 3 aresta de Q , ou se p pertence à poligonal definida pelos vértices de Q .

(2). Um nó da malha v é dito "queimado" se ele for interior ao polígono Q .

De acordo com estas definições, vemos que os casos (1), (2), (7) da Figuras 5.11 classificam v como queimado enquanto que os casos (3)-(6) e (8)-(12) classificariam v como não queimado. Assim, a simples classificação de p_{1n} e p_{2n} em um dos subespaços da figura 5.10 fornece imediatamente se v foi queimado ou não para a maioria dos casos. Os casos com ambigüidade (casos (13)-(16)) necessitam de um teste adicional (McInerney, 1997).

Algoritmo das T-Snakes

Finalmente, podemos apresentar o algoritmo das T-Snakes como segue (McInerney, 1997).

Algoritmo:

Inicialização: Projetar as snakes iniciais sobre a malha

Até que o ponto de congelamento seja atingido:

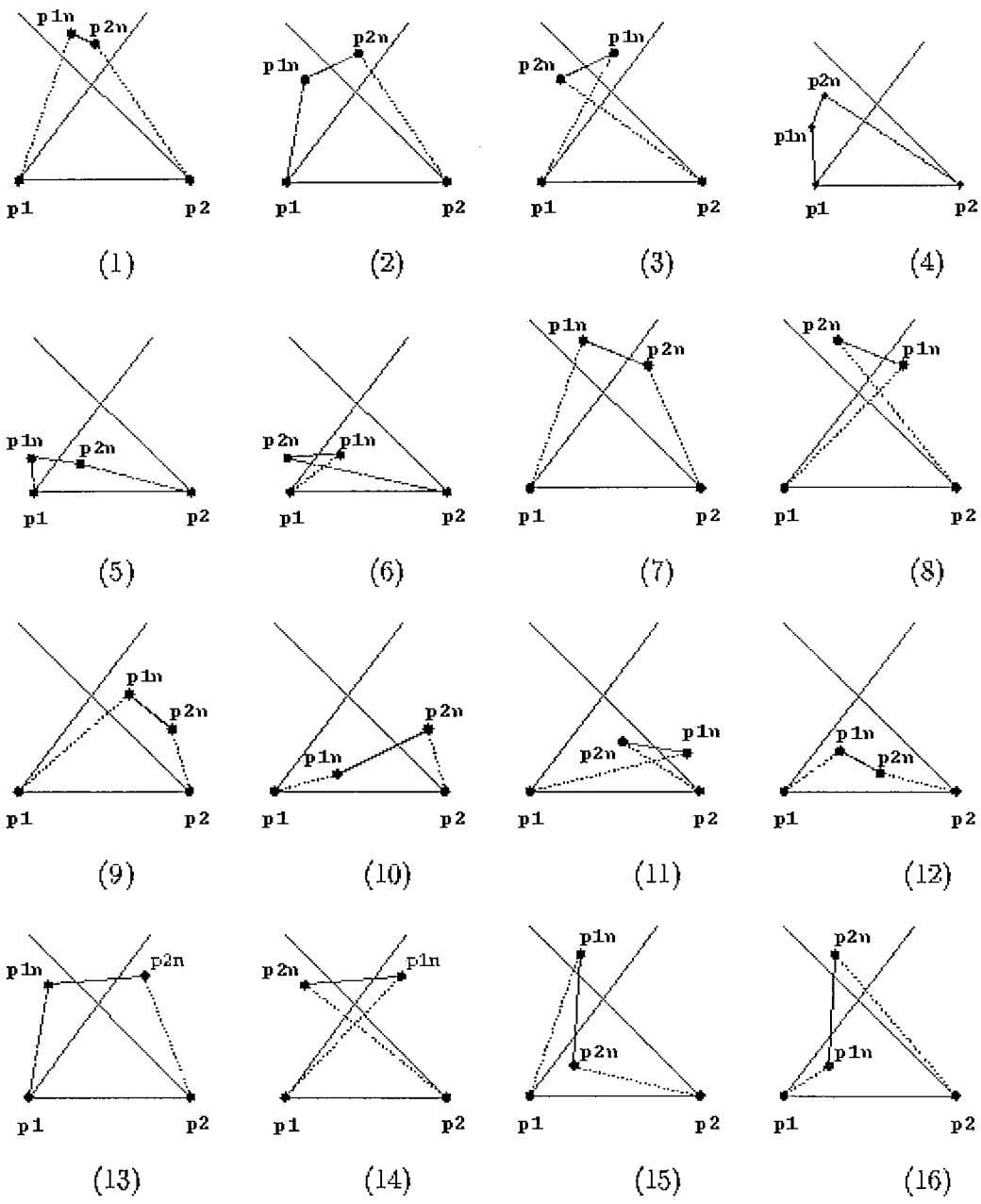


Figura 5.11: Possíveis casos para classificação.

1. Computar as forças internas e externas e atualizar a posição dos snaxels de acordo com a equação (5.56).
2. Computar as interseções entre a malha e os elementos de modelo.
3. Atualizar a função característica definida pela equação (5.51).
4. Usando a função característica determinar o correspondente conjunto de triângulos de borda usando o algoritmo Door-In-Door-Out.
5. Para cada aresta da sequência de arestas obtida escolher um ponto dentre os calculados no item 2 para ser um novo snaxel. Se não houver nenhum, criar um ponto na aresta em questão. Descartar todas as outras interseções.
6. Atualizar temperatura dos snaxels.

Apesar de suas habilidade topológicas, o método das T-Snakes tem várias limitações. Uma vez que este método é central para nosso trabalho faremos a seguir uma discussão destas limitações.

Limitações das T-Snakes

Em (McInerney, 1997) algumas limitações da T-Snake são discutidas e algumas soluções propostas.

O primeiro ponto é a restrição à evolução imposta pela condição de entropia usada: "se um nó é queimado ele permanece queimado".

Esta política é vantajosa para mudanças topológicas mas tem a desvantagem de limitar a evolução da snake em apenas expansão ou apenas contração, mas não ambos, o que pode dificultar a interação com o usuário caso esta se torne necessária.

Uma possível solução para este problema seria reverter a direção de evolução ocasionalmente por um pequeno intervalo de tempo (McInerney, 1997).

A resolução da malha controla o grau da flexibilidade geométrica da T-Snake. Se o objeto de interesse contém cavidades profundas ou protuberâncias finas, a resolução da malha deverá ser fina suficiente para permitir que a T-Snake se acomode à borda desejada.

Contudo, uma vez que estamos usando triangulações regulares tal procedimento implica em um número excessivo de snaxels em regiões com baixa curvatura trazendo problemas de performance.

A utilização de métodos adaptativos (Berger, 1986) seria uma solução caso pudemos saber automaticamente onde a malha necessita ser refinada, o que na prática é difícil. Na seção ?? indicaremos uma possível solução para este problema baseada em uma metodologia que parte de uma malha mais grossa, refinando-a até a resolução desejada.

Outra limitação da T-Snake relacionada com nosso trabalho é o fato do método de reparametrização da snake (projeção sobre a malha) ser extrínseco, isto é, depender não somente da forma do contorno mas também da maneira como o espaço

foi dividido (decomposição simplicial) bem como da posição da snake neste espaço.

Este ponto é menos problemático para a T-Snake uma vez que sua evolução não é baseada explicitamente na minimização de um funcional de energia e adotamos uma condição de entropia. Apesar disto, tal reparametrização perturba a snake. Em última análise, este procedimento de reparametrização é o motivo pelo qual a T-Snake não é em geral um bom método de finalização exigindo que, após sua estabilização, a malha seja descartada e a snake evolua independente da triangulação.

Esta reparametrização implicaria também na não-invariância da energia interna por transformações de translação e rotação uma vez que os snaxels dependem da posição da snake no espaço.

Outro aspecto relacionado à decomposição do domínio se refere ao viés, na direção da diagonal secundária, de uma triangulação como a da Figura 5.7 . Isto provém da observação de que a snake terá menos snaxels em regiões aproximadamente paralelas à diagonal, o que vai se refletir nas forças internas correspondentes (expressões (5.52)-(5.54)).

Uma forma de evitar este problema seria usar uma triangulação do tipo J_1 como na Figura 5.12 abaixo.

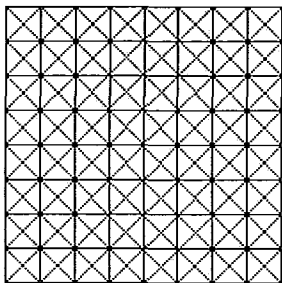


Figura 5.12: Triangulação do tipo J_1 .

Capítulo 6

Dual-T-Snakes

Apesar das habilidades topológicas, o método das T-Snakes também é afetado pelos problemas causados pela não-convexidade da energia de imagem. A utilização de estatísticas para redefinir o sentido da força normal (expressão 5.55) diminui a sensibilidade do método com relação a mínimos locais mas não resolve o problema. A dificuldade está em determinar automaticamente se uma T-Snake congelada atingiu ou não a borda desejada.

Uma forma de diminuir as dificuldades com a não convexidade é através de uma inicialização eficiente (próxima da borda), como destacamos em (Giraldi *et al.* , 2000e; Giraldi *et al.* , 2000a). Quanto mais próxima estiver a solução inicial do mínimo global, mais facilmente o método pode atingi-lo.

No Capítulo 8 nós apresentamos uma contribuição neste sentido, toda ela baseada na metodologia das T-Snakes e no conceito de *propriedade de escala local* ali definido.

Embora útil, o que será demonstrado no Capítulo 9, a necessidade de uma escala local para os objetos de interesse, o que basicamente significa que os padrões de intensidade definem regiões conexas da imagem, limita a classe de aplicações deste método.

Métodos multi-escala tais como (Leymarie & Levine, 1993) podem ser outra possibilidade para tratar a não-convexidade da energia, uma vez que a passagem de uma resolução mais grosseira para outra mais fina perturba a snake facilitando com que a mesma evolua em direção à região desejada. Outras abordagens encontradas na literatura para a inicialização automática de modelos deformáveis em geral fazem uso de Transformadas de Hough (Lai & Chin, 1995), redes neurais e momentos (Williams *et al.* , 1997).

O método dual da seção 5.1.1 é certamente o mais robusto com relação a mínimos locais, dentre os modelos apresentados no Capítulo 5. Contudo, sua versão evolucionária depende do estabelecimento a priori de modelos de forma (ainda que locais). Além disso, a necessidade de correspondência entre os pontos das snakes é outro fator limitante do método. Por outro lado, na sua versão baseada em programação dinâmica (seção 5.2), o método dual apresenta a dificuldade clara com relação à

inicialização da snake interna e externa a fim de que o espaço de busca possa ser criado eficientemente. Contudo, este método é o mais robusto pelas propriedades de otimização global da programação dinâmica (seção 4.1).

Assim, nossa motivação inicial para o desenvolver o método descrito a seguir foi *generalizar o método Dual Evolucionário*, de maneira a prescindir de um modelo de forma pré-estabelecido e evitando a necessidade de correspondência entre as snakes interna e externa.

O ponto inicial do novo método veio da observação de que a triangulação das T-Snakes fornece uma forma simples de verificar proximidade de curvas. Ao mesmo tempo, a força normal nas T-Snakes pode ser usada no lugar da força adicional. A definição daquela pode ser usada na decisão de qual snaxel movimentar em uma snake congelada.

Resumindo, a proposta foi colocar o método Dual e as T-Snakes em um modelo unificado. Como resultado, conseguimos uma nova metodologia que é dual e topológica ao mesmo tempo, daí denominá-la *Dual-T-Snakes*. Além disto, podemos agora generalizar o Dual Evolucionário para 3D, como indicaremos a seguir.

Um derivado interessante do Dual-T-Snakes é a redução do espaço de busca, podendo assim ser utilizado para inicializar o Dual Baseado em Programação Dinâmica, vencendo com isto a dificuldade com a inicialização deste último citada acima.

A próxima seção descreve o novo método em detalhes para o caso bidimensional. Indicamos também como o Dual-T-Snakes pode ser estendido para 3D. A seguir, mostramos como usar o resultado do método para inicializar um algoritmo de Viterbi descrito na seção 4.1. Finalmente, apontamos as vantagens e desvantagens do novo método.

6.1 Descrição do Dual-T-Snakes

Nesta seção apresentamos o método Dual-T-Snakes (Giraldi *et al.* , 2000d; Giraldi *et al.* , 2000c), o qual é uma contribuição deste trabalho.

A idéia básica desta nova metodologia é explorar o modelo da T-Snakes para propor um *Modelo Dual de Contornos Ativos Generalizado*: um *contorno externo*, envolvendo os objetos de interesse, contrai e se divide enquanto que *contornos internos* aos objetos se expandem em direção às bordas procuradas.(Figura 6.1).

Uma vez que a posição inicial das snakes será em geral longe das bordas desejadas, sofrendo inclusive mudanças topológicas durante sua evolução, não é interessante usarmos processos de minimização explícita para evoluir as snakes. Assim a escolha do modelo de snakes da seção 5.3.2 é imediata.

Para fazer a snake externa contrair e as snakes internas expandirem em direção às bordas dos objetos nós usamos forças normais do tipo daquela apresentada na

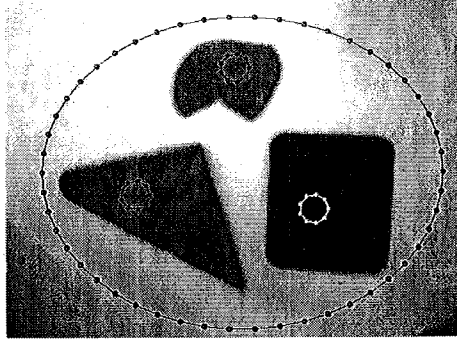


Figura 6.1: Inicialização do Dual-T-Snake.

equação 5.54.

Além disso, é necessário definir elementos que tornem a evolução da snake externa e das internas interdependente a fim de tornar o método unificado. Neste trabalho isto é implementado usando-se a *energia de imagem (normalizada) sobre as snakes* e uma noção de *afinidade*.

Especificamente, nós usamos duas definições para energia de imagem sobre as snakes: uma para o contorno externo ((E_{outer})) e outra para o conjunto de contornos internos envolvidos por este ((E_{inner})):

$$E_{outer} = \sum_{i=0}^{N-1} (-\|\nabla I(v_i)\|^2) / N, \quad (6.1)$$

$$E_{inner} = \frac{1}{m} \left(\sum_{k=0}^m \left(\sum_{i=0}^{N_k-1} (-\|\nabla I(v_i)\|^2) / N_k \right) \right), \quad (6.2)$$

onde m é o número de curvas internas, N_k é o número de snaxels da snake interna k , N é o número de snaxels da snake externa, I é a intensidade de imagem, suposta em tons de cinza, a menos de menção explícita ao contrário.

Se:

$$E_{inner} > E_{outer} \quad (6.3)$$

então uma curva interna deve ser evoluída. Para tal, nós primeiro usamos um *operador de afinidade* para estimar os pixels da imagem mais prováveis de pertencer às bordas dos objetos de interesse. Com base neste operador nós podemos associar a cada snaxel um valor indicando sua proximidade da borda. O campo sobre a snake assim obtido é então binarizado pela comparação com um limiar, assumindo valor zero para snaxels próximos da borda e 1 caso contrário.

Portanto, se a expressão (6.3) é verdadeira, a snake interna com maior número de snaxels com afinidade não nula é escolhida.

Por outro lado, se :

$$E_{outer} > E_{inner} \quad (6.4)$$

então a snake externa será evoluída, caso sua função de afinidade não seja identicamente nula.

Uma vantagem de usarmos um operador de afinidade é simplificarmos a imagem reduzindo assim as dificuldades para a escolha de parâmetros para o modelo.

Dentre as possibilidades para definir o operador de afinidade podemos citar a utilização de um limiar T para a intensidade de imagem, estatísticas, e métodos fuzzy (Apêndice E). Esta última possibilidade é interessante por tirar mais vantagem da interação com o usuário na inicialização do Dual-T-Snakes (posicionamento das snakes internas e externas).

O próximo ponto é avaliarmos a similaridade entre dois contornos. A idéia é tomarmos vantagem da metodologia de subdivisão do espaço da T-Snake: a diferença entre as Funções Características da curva externa e da curva interna (denominada *Characteristic-Diff*) fornece uma noção (local) da proximidade entre as snakes, como mostra a Figura 6.2.

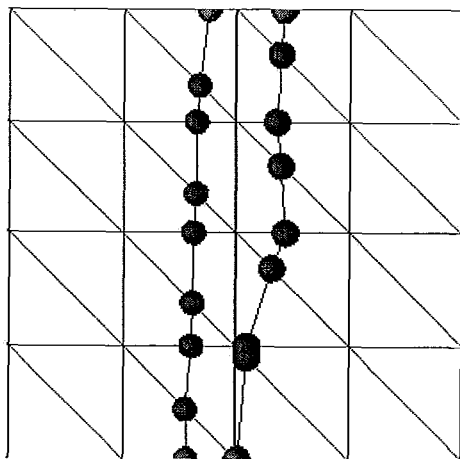


Figura 6.2: Proximidade entre duas snakes medida pela triangulação.

Por exemplo, no caso da triangulação CF desta figura, nós poderíamos interromper o movimento de todos os snaxels de uma snake interna que estão no interior de um triângulo σ se algum dos vértices $v \in \sigma$ tem as seguintes propriedades:

Propriedade (a). Todos os 6 triângulos adjacentes a v têm um vértice onde *Characteristic-Diff* = 0;

Propriedade (b). Um destes triângulos é cortado pelo contorno externo correspondente

Contudo, a política de interromper o movimento de regiões da snake pode não ser eficiente por introduzir *defeitos* sobre o resultado final (Figura 6.4). Na implementação do método há apenas um *flag* indicando que as condições (a)-(b) acima foram verificadas ($flag = 0$, neste caso). A execução do método termina quando todos os *flags* são nulos.

Uma vez que a condição de entropia da seção 2.1 é adotada, nós podemos associar um ponto interno a cada snake interior. Quando a snake externa sofre um split em duas ou mais snakes estes pontos podem ser usados para identificar qual snake interna está envolvida por uma snake externa gerada na operação de split.

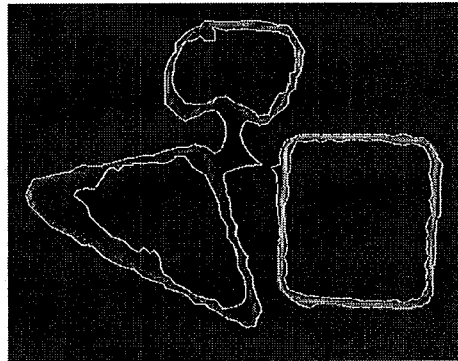


Figura 6.3: Split da Dual Snakes inicial.

O ponto de congelamento é usado para indicar quando uma T-Snake está em uma posição de equilíbrio.

É importante ressaltar que o balanço entre energia/afinidade, juntamente com a noção de proximidade entre as snakes descrita acima, permitem evitar mínimos locais. De fato, se uma T-Snake foi congelada nós podemos aumentar a força normal para os snaxels com afinidade não nula e $flag \neq 0$.

No algoritmo a seguir denominamos uma *Dual Snake* a uma lista de T-Snakes onde a primeira é uma snake externa e as demais são snakes internas a esta. O método pode ser resumido como segue:

Dual-T-Snakes:

1. Colocar todas as dual snakes em uma fila.
2. Até que esta fila fique vazia fazer:
 - 2.1. Retirar uma dual snake da fila.
 - 2.2. Usar as energias das expressões (6.1) and (6.2) e a função de afinidade para encontrar a snake a ser processada.
 - 2.3. Se todos os snaxels da snake escolhida estão congelados aumentar a força normal naqueles com afinidade zero e $flag \neq 0$ até que a energia da snake

comece a decrescer.

2.4. Remover a força adicional e deixar a T-Snake evoluir normalmente

2.5 Se a snake sendo processada sofrer mudança topológica, montar as dual snakes resultantes e retornar ao passo 2.1.

2.6 Quando a snake estive congelada, analisar se a mesma está próxima de uma snake de outro tipo (interna/externa). Neste caso, remover a dual snake correspondente da fila.

2.7. Caso contrário, montar a dual snake(s) resultante e voltar ao passo 2.2.

Quando uma snake externa sofre um split, cada parte conexas gerada, juntamente com as snakes em seu interior, formarão uma nova dual snake.

Cada dual snake é completamente independente das outras o que possibilita implementações em arquiteturas paralelas. A Figura 6.4 mostra o resultado correspondente à Figura 6.1.

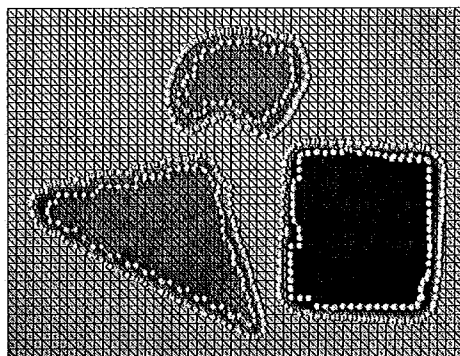


Figura 6.4: Dual-T-Snake result for figure 6.1. A resolução da grade é 5×5 e os snaxels são os pontos em branco.

Para estender o método desta seção para 3D basta usar o modelo das T-Superfícies (Apêndice C) em lugar das T-Snakes. Os demais elementos do modelo correspondente - Dual-T-Superfície - permaneceriam os mesmos do caso 2D descrito acima.

Embora não tenhamos feito testes para o caso 3D, acreditamos que o mesmo possa se constituir em uma nova contribuição para a área, como será ressaltado no Capítulo 10.

6.2 Metodologia para Segmentação

Uma vez completa a execução do Dual-T-Snakes, podemos tirar vantagem da redução do espaço de busca e aplicar um modelo de snakes baseado em programação dinâmica para obter a borda limitada por cada dual snake.

A idéia é aproveitar as habilidades topológicas e a robustez com relação aos mínimos locais do Dual-T-Snakes para inicializar o algoritmo de Viterbi (seção 5.2),

uma vez que cada borda está limitada por uma dual snake (Figura 6.4). Assim, conseguimos vencer a limitação com relação à inicialização do Dual Baseado em Programação Dinâmica (Gunn, 1996b), aproveitando com isso as propriedades de otimização global deste último.

O resultado é uma metodologia robusta e eficiente para segmentação e extração de bordas.

Apesar destas vantagens, alguns cuidados devem ser tomados.

Primeiramente, na definição do espaço de busca do Viterbi, não podemos ter intersecção entre segmentos. Além disto, devido a irregularidades na solução do Dual-T-Snakes, podemos ter regiões com densidade de pontos mais baixa que outras.

Uma primeira possibilidade para evitar estes problemas seria utilizar métodos globais para estabelecer correspondência entre curvas (Geiger *et al.*, 1995). Contudo, isto implicaria em um custo extra ao modelo.

Uma observação oportuna vem do fato de que nós precisamos realizar duas buscas para obter o resultado final (seção 5.2). Assim, podemos inicialmente fazer um matching entre as curvas usando testes locais e de pouco custo computacional para evitar os defeitos mencionados. O espaço de busca assim obtido serviria para a primeira busca.

Em seguida, usamos o resultado obtido para redefinir o espaço de busca. Para isto basta tomar dois *offsets* para a solução encontrada na primeira busca, estabelecer um matching entre eles e criar um espaço de busca como anteriormente. Finalmente, aplicamos novamente a programação dinâmica (expressão (4.8)) para obter o resultado desejado.

Outra possibilidade seria usar a solução da primeira busca, discretizá-la regularmente e aplicar a relação de recorrência dada pela expressão (4.8) usando uma janela de busca centrada em cada ponto da curva discreta obtida, tal como em (Amini *et al.*, 1990).

Resumindo, o método das Dual-T-Snakes juntamente com o Viterbi formam uma metodologia para extração de bordas composta por quatro etapas:

- (a) Inicialização das snakes internas e externas pelo usuário;
- (b) Computação da função de afinidade
- (c) Aplicação do Dual-T-Snakes para limitar região de interesse
- (d) Extração das bordas desejadas pelo algoritmo de Viterbi

O método herda as habilidades topológicas das T-Snakes o que permite sua aplicação em casos onde a imagem consiste de vários objetos imersos em um background.

6.3 Vantagens do Dual-T-Snakes

Do ponto de vista da metodologia dual apresentada originalmente em (Gunn & Nixon, 1997; Gunn, 1996a), a primeira vantagem do Dual-T-Snakes é que este não necessita de "matching" entre as snakes internas e externas.

Uma vez que não necessitamos de correspondência entre as snakes, não precisamos limitar a forma da snake durante sua evolução, e portanto podemos prescindir de um modelo de forma estabelecido a priori.

No Dual original a força adicional não pode ser definida na direção normal à snake porque isto propicia a ocorrência de auto-intersecções na curva durante sua evolução, como mostra a Figura 6.5.

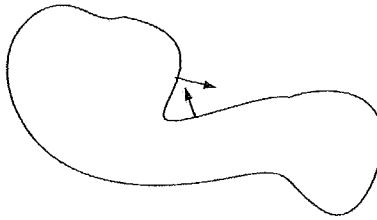


Figura 6.5: Evolução de curva na direção normal pode causar auto-intersecções.

No Dual-T-Snakes pudemos definir a força adicional na direção normal porque o processo de reparametrização das T-Snakes resolve auto-intersecções naturalmente.

Ao mesmo tempo, a imersão do dual na contexto das T-Snakes permitiu aplicar esta metodologia para imagens com vários objetos de interesse, aumentando seu campo de aplicações.

Portanto, o Dual-T-Snakes generaliza o Dual original, tanto para formas mais gerais quanto para topologias mais gerais.

Além disso, existe agora a real possibilidade de estender a metodologia dual para 3D via as T-Superfícies uma vez que não precisamos de correspondência entre os snaxels ou mesmo de um modelo de forma pré-estabelecido.

Do ponto de vista das T-Snakes, o Dual-T-Snakes trouxe efetivamente mais robustez quanto aos mínimos locais da energia de imagem, pelo balanço da energia e pela noção de proximidade entre as snakes internas e externas. Agora, o critério de parada baseado na similaridade entre as curvas é um indicador mais seguro de que o mínimo global foi atingido se comparado com o método de variação do limiar T usado na definição da força normal dada a partir da expressão (5.55)

Finalmente, a redução do espaço de busca permite evitar os problemas causados pelas necessidades de armazenamento em memória e processamento da programação dinâmica.

6.4 Desvantagens do Dual-T-Snakes

Uma questão importante a ser considerada é o passo máximo permitido para uma T-Snake durante uma interação da expressão (5.56). Como veremos no Capítulo 7, este passo deve ser limitado a no máximo a distância entre dois pixels vizinhos por interação, a fim de que a definição das energias de imagem dadas em (6.1)-(6.2) sejam eficientes.

Se comparado com a T-Snakes, isto implica em perda de performance uma vez que o passo de uma T-Snake pode ser maior que este valor, caso não tenhamos imagens com muito ruído.

Além disto, as limitações das T-Snakes apontadas na seção 5.3.2 vão se aplicar naturalmente ao Dual-T-Snakes.

Um ponto mais cuidadoso se refere à escolha de parâmetros do novo método, uma vez que temos movimentos distintos para as snakes interna e externa.

Basicamente, o problema está no fato de que as forças de suavização definidas por (5.52)-(5.54) possuem uma componente normal que (de maneira geral) tende a contrair a snake, o que pode ser favorável para a snake externa (que precisa contrair) mas pode ser desfavorável para a interna (que precisa expandir).

Nos Capítulos 7 e 9 discutiremos estas questões em detalhes e as soluções encontradas.

Por fim, a inicialização do Dual-T-Snakes depende da interação com o usuário, principalmente para o posicionamento das T-Snakes internas. O método de inicialização do Capítulo 8 a seguir é uma possível solução para este problema que pode ser aplicada sob certas condições. Este método pode diminuir o custo computacional do Dual-T-Snakes, uma vez que a curva inicial obtida está em geral próxima do objeto de interesse.

Capítulo 7

Implementação do Dual-T-Snakes

Na implementação do Dual-T-Snakes alguns detalhes merecem especial atenção. Além disso, alguns aspectos das T-Snakes foram revistos de acordo com as características específicas do novo método.

Neste Capítulo procuramos analisar com certo detalhe estes aspectos, particularmente no que diz respeito aos parâmetros e às forças envolvidas.

O modelo de Balloon (Cohen, 1991), desenvolvido na seção 3.4, será sempre uma diretriz para a apresentação que segue. Isto se deve principalmente pela utilização de uma força normal extra do tipo daquela usada no Balloon.

No final desta discussão apresentamos um fluxograma que fornece um *over-view* do algoritmo.

7.1 Escolha de Parâmetros

O primeiro ponto a ser analisado se refere à escolha dos parâmetros. Uma vez que o modelo básico é dado pelas T-Snakes, nós temos a tensão elástica (a_i), o comprimento natural (l_i), rigidez (b_i), o fator de escala k para a força normal e o coeficiente λ da força de imagem. Este ponto é central no Dual-T-Snakes porque precisamos escolher parâmetros para duas snakes: a interna e a externa. A Figura 7.1 ilustra bem as dificuldades inerentes a este fato.

Nesta figura representamos um pequeno trecho das snakes. Os vetores correspondentes à tensão elástica (α_i) e à rigidez (β_i) estão desenhados de acordo com a análise da seção 3.4.3. Estamos supondo que o comprimento natural l_i seja nulo e que a região tem pouco contraste, portanto, com força externa desprezível.

Assim, as snakes só poderão caminhar em direção à borda caso as resultantes intrínsecas sejam orientadas no sentido da mesma.

Contudo, observemos que as forças internas possuem resultante apontada para a borda no caso da snake externa, mas apontando no sentido contrário para a snake interna.

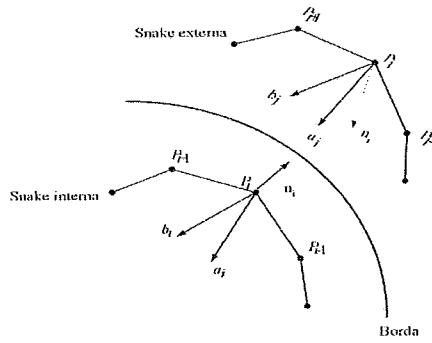


Figura 7.1: Forças internas para as Snakes do Dual-T-Snakes.

A dificuldade maior decorrente deste comportamento está na escolha do fator de escala para a força normal. Este fator precisa ser alto suficiente para que a T-Snake interna se movimente no sentido desejado, mas não pode aumentar demasiadamente a resultante das forças internas da outra snake, pois, se isto ocorrer, poderemos ter dificuldades para estabilizar esta snake próximo da borda desejada.

A forma mais direta para resolver esta dificuldade seria usar parâmetros distintos para as snakes internas e externas. Contudo, isto implica em calibrar duas snakes, o que é indesejável, visto que não temos um procedimento sistemático para a escolha de parâmetros em modelos deformáveis em geral (R. Fisker, 1998).

Uma outra proposta seria tomarmos como negativos os parâmetros a_i, b_i . Com isto, mudaríamos o sentido das forças internas (Figura 7.2). No entanto, esta escolha não tem se mostrado eficiente na prática e o papel de suavização das forças internas neste caso torna-se pouco claro (Figura 7.2).

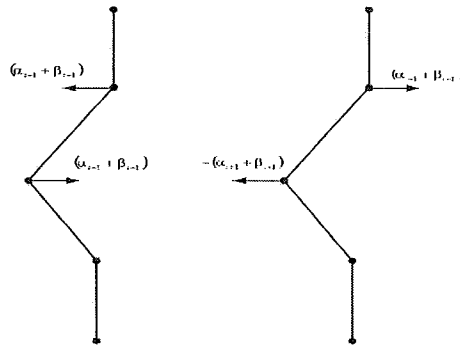


Figura 7.2: Forças internas com parâmetros positivos (esquerda) e negativos (direita).

A força de tensão dada por (5.52) tem como uma de suas finalidades uniformizar a distribuição dos snaxels ao longo da snake ao tentar evitar que os mesmos se aproximem (ou se distanciem) além do comprimento natural l_i usado na definição da força de mola (equação 5.52).

No entanto, a reparametrização das T-Snakes é feita partindo-se da projeção da snake sobre a triangulação. Logo uma snake com distribuição uniforme de pontos pode passar a ter uma distribuição pouco uniforme após a reparametrização, como mostra a Figura 7.8 a seguir.

Considerando estas observações, nós optamos primeiramente por diminuir o número de parâmetros, facilitando assim sua escolha, em detrimento (possivelmente) de uma maior suavidade do resultado final do Dual-T-Snakes. Neste sentido, usamos apenas rigidez, força normal e força de imagem, colocando nulos todos os outros parâmetros.

Mesmo assim, devemos ficar atentos para o problema indicado na Figura 7.1, onde vemos as forças de rigidez e normal apontando ambas para a borda no caso da snake externa mas apontando em direções distintas no caso da interna. As definições da força normal serão a chave para tratar esta dificuldade, como veremos a seguir.

O modelo final a ser resolvido numericamente tem a forma:

$$v_i^{(t+\Delta t)} = v_i^t + \frac{\Delta t}{\gamma} (\beta_i^t + F_i^t + f_i^t), \quad (7.1)$$

o qual corresponde ao modelo de balloons dado pela expressão (3.63), com $\omega_2 = 0$, discretizado por diferenças finitas usando-se um método idêntico àquele da seção 3.2.3. A análise numérica portanto fornece a mesma conclusão obtida na seção 3.2.3; ou seja, quanto menor o intervalo de tempo τ maior a estabilidade numérica.

Neste trabalho, a determinação exata dos parâmetros envolvidos foi feita de forma empírica. Estes são considerados como constantes e o modelo é submetido a uma etapa de calibração na qual fazemos uma série de simulações computacionais a fim de encontrar valores ótimos para estas constantes.

7.2 Passo Máximo

As energias de imagem do Dual-T-Snakes dadas pelas expressões (6.1) e (6.2) fazem sentido somente se não permitirmos passos altos da snake em uma interação. A Figura 7.3 ajuda esclarecer este fato. Observe que, uma vez que o potencial externo depende do contraste da imagem, nos patamares indicados na figura a energia de imagem é nula, e assim se o passo for alto, passando de um patamar para o outro em uma interação, não haverá variação da energia.

Uma forma de evitar esta dificuldade seria usar energias que dependem da própria intensidade imagem (e não do seu contraste), uma vez que a força normal necessita de um limiar para ser definida. Contudo, aumentar o passo de interação torna o movimento da snake mais insável, principalmente para imagens com ruídos.

Assim, tomamos a iniciativa de estabelecer um passo máximo da ordem da distância entre dois pixels da imagem (*uma unidade*). Desta forma, a evolução das

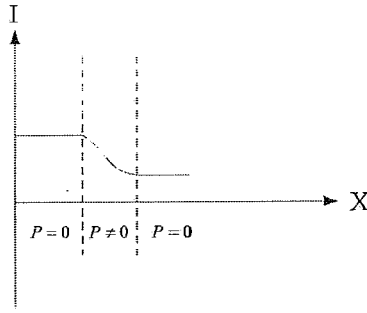


Figura 7.3: Com passo muito alto, a snake pode avançar o degrau perdendo a borda.

snakes se faz de forma mais controlada. Ao mesmo tempo, melhoramos a estabilidade numérica e consideramos apropriadamente as informações da imagem evitando o problema representado na Figura 7.3.

7.3 Força Externa e Força Normal

Um ponto importante a ser discutido é a política adotada para estas duas forças no Dual-T-Snakes.

Assim como no Balloon, usamos a força externa normalizada segundo a expressão (3.57). Esta escolha tem mostrado bons resultados facilitando o controle da snake durante sua evolução.

Exatamente como no caso do Balloon, a força normal nas T-Snakes é usada para fazer as snakes convergirem para as bordas de interesse. Contudo, na vizinhança de uma borda, as T-Snakes e o Balloon funcionam distintamente. A Figura 7.4 representa como a força de imagem e a força normal atuam na vizinhança de uma borda nestes dois modelos.

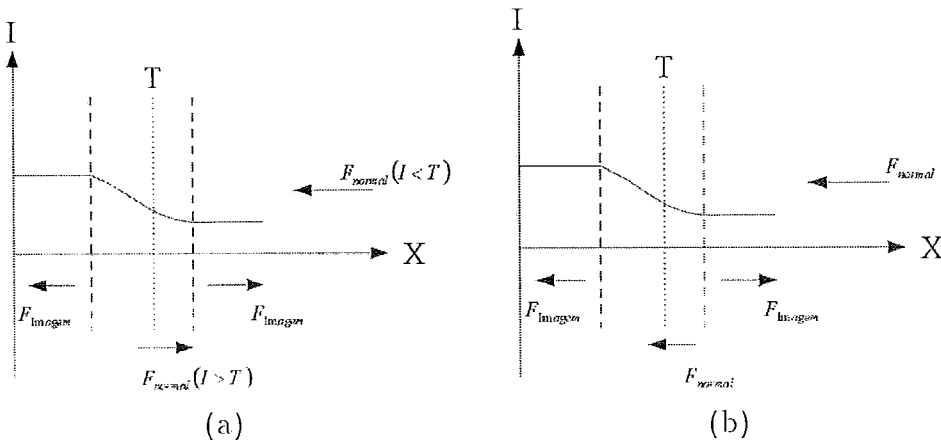


Figura 7.4: (a)Forças de imagem e normal para as T-Snakes. (b)Forças de imagem e normal para o Balloon.

No caso do Balloon não temos inversão da força normal após o limiar T . Por este motivo, os coeficientes κ_1 e κ_2 devem ser da mesma ordem de grandeza a fim de que tenhamos um equilíbrio das forças correspondentes antes da força de imagem inverter seu sentido devido a forma do potencial P definido em (3.3). Este equilíbrio implica em deslocamentos pequenos da snake na vizinhança da borda e portanto é percebido usando-se o critério de parada dado pela expressão (3.48).

Este procedimento não pode ser aplicado para as T-Snakes devido seu método de reparametrização. A Figura 7.5 mostra os snaxels acomodados sobre a borda após uma evolução dada pela equação 7.1 e a T-Snake final após a reparametrização. Esta *perturbação* da snake impede que um critério de parada baseado em deslocamentos pequenos seja utilizado, o que em nossa opinião motivou a idéia do ponto de congelamento usada em (McInerney, 1997; McInerney & Terzopoulos, 1999). Isto fornece um critério de parada eficiente, mas, do ponto de vista do Dual-T-Snakes, traz um problema: o número de interações pode ser muito alto caso as snakes fiquem congeladas várias vezes durante sua evolução.

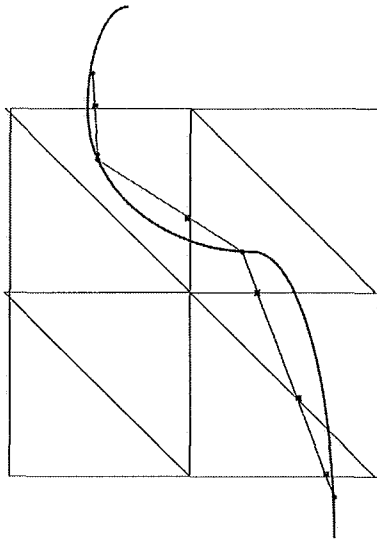


Figura 7.5: Snake antes e depois da reparametrização.

Portanto, se aplicarmos o modelo de Balloons ao Dual-T-Snakes poderemos ter uma perda significativa de performance, uma vez que o Balloon é muito sensível a mínimos locais da imagem. Isto é verificado principalmente para imagens com muito ruído e artefatos no background, que constituem exatamente a principal classe de aplicações do Dual-T-Snakes.

Mesmo a definição (5.54) para a força normal tem suas limitações para imagens deste tipo. Nestes casos, a inversão da força normal usada na definição (5.54) pode conduzir a muitas oscilações da snake decorrentes da não homogeneidade da imagem. A Figura 7.6 ilustra tal situação.

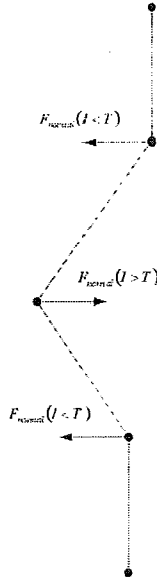


Figura 7.6: Comportamento da força normal sobre a snake podendo gerar oscilações.

Neste caso, poderemos ter oscilação dos snaxels representados o que pode deixar a snake *preza* na região indicada. Se este comportamento se repete constantemente, o número de vezes que a snake atinge o ponto de congelamento aumenta muito.

Considerando estes fatos, a solução que encontramos foi redefinir a força normal como segue:

$$F_i(v_i) = Val(v_i) kn_i, \quad (7.2)$$

onde:

$$Val(v_i) = 1 \text{ se } I(v_i) < T, \quad (7.3)$$

$$Val(v_i) = 0 \text{ caso contrário}, \quad (7.4)$$

sendo n_i a normal unitária no vértice v_i , k o fator que define a intensidade da força e T o limiar para a imagem I .

Desta forma, não teremos a tendência a oscilar, como na Figura 7.6. Por outro lado, o efeito de *puxar* a snake para a borda é mantido, uma vez que os snaxels com força normal não nula avançam a snake, podendo retirar os demais de mínimos locais decorrentes de irregularidades na imagem. Certamente ainda podemos ter um congelamento da snake fora da região desejada, contudo o número de vezes que isto acontece é sensivelmente reduzido.

Os testes mostraram que a definição E.12 é mais eficiente para imagens com ruído, o que veremos no Capítulo 9.

Um ponto importante é que agora não precisamos manter os coeficientes da força normal e de imagem com a mesma ordem de grandeza, uma vez que próximo da borda a força normal será nula. Este é outro aspecto positivo da definição acima para a força normal uma vez que estamos aproveitando esta força apenas para aproximar a snake da região de interesse. Quando próximo da borda, a força normal é anulada pela definição E.12 e o equilíbrio fica dependendo apenas das forças elástica e externa. Evitamos assim a preocupação com o equilíbrio mais delicado entre a força normal e a força de imagem observado no Balloon (Figura 7.4).

Por outro lado, a suavidade do resultado final pode ser afetada, o que é discutido a seguir.

7.4 Suavização e Forças Internas

Um ponto importante a ser ressaltado com a relação à definição (E.12) é o aparecimento de oscilações da snake quando próximo das bordas. A Figura 7.7 é representativa deste comportamento.

Nesta figura, temos um seqüência de snaxels alinhados e espaçados uniformemente. Neste caso, as forças internas T-Snakes dadas pelas expressões das (5.52 e (5.53) são nulas (o mesmo acontecendo com as expressões discretas (3.60) e (3.61) do Balloon).

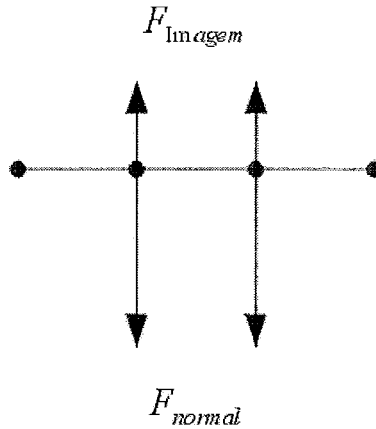


Figura 7.7: As forças internas nulas devido alinhamento e distribuição uniforme de snaxels.

Portanto, neste caso o sistema de forças sobre a snake fica resumido às forças normal e de imagem. Se a snake estiver próxima de uma borda, a definição (E.12) implica que a força normal poderá ser nula para alguns snaxels e portanto teremos apenas força de imagem atuando sobre os mesmos.

Neste caso, nos snaxels onde atua apenas a força de imagem, teremos um *salto*.

Embora limitado a no máximo uma unidade, este salto pode diminuir a suavidade da curva na região.

Este é um *defeito* da definição (E.12). Contudo, do ponto de vista do Dual-T-Snakes, este problema não é crítico uma vez que este método é usado apenas para limitarmos a região de interesse, na qual será definido o espaço de busca. A perda de suavidade indicada não tem se mostrado prejudicial para a determinação posterior do espaço de busca. Além disto, as oscilações alteram a temperatura dos snaxels fazendo com que os mesmos atinjam o ponto de congelamento. O critério de parada das T-Snakes é portanto eficiente nesta proposta.

É oportuno destacar que no método de Balloon a suavidade do resultado final é preservada primeiramente porque a força normal não é desligada nas proximidades da borda. Em segundo lugar, o fato da força normal e de imagem apresentarem intensidades da mesma ordem de grandeza impede *saltos* do tipo discutido acima quando estamos próximos da borda. Possíveis defeitos de suavidade que ocorram em uma interação, são em geral corrigidos pelas forças de tensão e rigidez as quais atuam como um filtro passa-baixa sobre a curva.

O efeito deste filtro pode ser melhor compreendido se tomarmos o sistema linear (3.64) e usarmos a inversa da matriz $[I + \tau K]$ para isolar X^n (idem para Y^n):

$$X^n = [I + \tau K]^{-1} \{ X^{n-1} + \tau [R_X(\bullet, \bar{t} - \tau) - N_X(\bullet, \bar{t} - \tau)] \}, \quad (7.5)$$

A matriz inversa $[I + \tau K]^{-1}$ produz o efeito de suavização que corta frequências mais altas da curva. Este efeito é mais complicado de ser obtido no Dual-T-Snakes.

Além das dificuldades relativas à força normal, uma outra justificativa para este problema é com relação ao termo de quarta ordem da equação (3.63). A versão discreta deste termo poderia ser acrescentada às T-Snakes, aumentando a *rigidez* das mesmas.

Contudo, maior rigidez pode implicar em mais dificuldade para expandir a snake interna, o que pode trazer dificuldades para calibrar a força normal. Acrescenta-se a isto a não uniformidade da distribuição de snaxels sobre a snake, característica do método de reparametrização usado. Isto implica, que setores lineares da snake podem apresentar snaxels distribuídos de forma não homogênea (Figura 7.8) criando irregularidades na curva durante sua evolução.

Considerando ainda que nosso objetivo com o Dual-T-Snakes é apenas limitar a região de interesse acreditamos mais interessante demonstrar a potencialidade do método sem termos de quarta ordem.

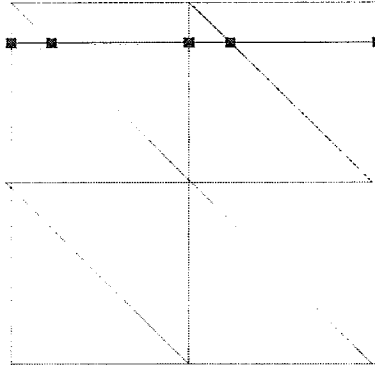


Figura 7.8: Distribuição não-uniforme de snaxels devido reparametrização.

7.5 Resolução da Triangulação

A resolução da triangulação usada está vinculada dentre outros fatores à escala dos objetos de interesse.

Uma metodologia sistemática para definir esta resolução poderia ser útil não apenas para a eficiência do método mas também para a implementação de técnicas multi-grid para T-Snakes e, conseqüentemente, para o Dual-T-Snakes também.

Tal procedimento é altamente dependente da aplicação. Por exemplo, vamos supor como em (Davatzikos & Prince, 1995) que conhecemos a priori que a curvatura dos objetos de interesse na cena tem como limite superior um valor k_0 .

Vamos então tomar a seguinte série de Fourier para a fronteira procurada:

$$x(s) = X_0 + \sum_{l=1}^{\infty} \{X_{c_l} \cos(2\pi l s) + X_{s_l} \sin(2\pi l s)\} \quad (7.6)$$

$$y(s) = Y_0 + \sum_{l=1}^{\infty} \{Y_{c_l} \cos(2\pi l s) + Y_{s_l} \sin(2\pi l s)\} \quad (7.7)$$

Seguindo (Davatzikos & Prince, 1995) vamos supor que somente uma frequência tem energia. Assim, ignorando possíveis sobreposições, translações e o ponto inicial, as funções coordenadas (7.6)-(7.7) tornam-se:

$$x(s) = \frac{1}{k} \cos(Lks), \quad (7.8)$$

$$y(s) = \frac{1}{k} \sin(Lks), \quad (7.9)$$

onde L é o comprimento total da borda. Uma vez que $k \leq k_0$, a frequência máxima na qual esta função pode existir será:

$$l_{\max} = Lk_0/2\pi. \quad (7.10)$$

Para nosso propósito, este desenvolvimento simplificado é suficiente. Pela teoria de amostragem (Jain, 1989), a frequência l_{\max} significa que o período de amostragem deve ser:

$$T_{\text{sampling}} = \frac{2\pi}{l_{\text{sampling}}} \leq \frac{2\pi}{l_{\max}} = 4\pi^2/Lk_0, \quad (7.11)$$

o que implica que o número de pontos amostrados N_{sampling} deve satisfazer:

$$N_{\text{sampling}} \geq \left\lceil \frac{L^2 k_0}{4\pi^2} \right\rceil. \quad (7.12)$$

Como consequência, um método multi-grid poderia ser implementado. Por exemplo, poderíamos partir de uma malha com resolução definida pela expressão (7.12) e, após a estabilização da T-snake, refinar a malha e evoluir a T-snake novamente, obtendo agora um ajuste mais fino.

Quando projetamos uma curva sobre a triangulação, além da não-uniformidade da distribuição de pontos obtidos, poderemos ter pontos muito próximos o que pode trazer problemas numéricos.

Para evitar esta dificuldade, optamos por estabelecer primeiramente uma distância mínima para os snaxels que evite os problemas numéricos. Após a reparametrização da snake, analisamos a distância entre cada snaxel e seu vizinho, e caso esta seja menor que o valor mínimo permitido, substituímos estes snaxels pelo nó da malha mais próximo (Figura 7.9).

Além disto, estabelecemos uma distância mínima usada no cálculo da força da expressão (5.53). Se dois snaxels vizinhos possuem distância menor que esta última, tomamos o snaxel seguinte.

Observando a expressão (5.53), vemos que, aumentar a resolução da malha do Dual-T-Snakes implica (de forma geral) em reduzir forças internas da snake, a menos que alteremos os parâmetros correspondentes.

Esta perda de rigidez da snake pode ser interessante para que a mesma avance para regiões com curvatura acentuada.

7.6 Condição de Entropia

Outra questão importante é a maneira pela qual implementamos a condição de entropia: *uma vez que um nó é queimado ele permanecerá queimado.*

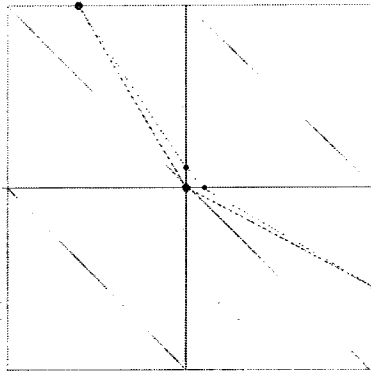


Figura 7.9: Snaxels muito próximos após reparametrização.

O ponto central aqui é a política usada para satisfazer esta condição.

O primeiro passo neste sentido é não alterar o valor da função característica em nós já queimados.

Assim, uma vez movimentada a snake segundo a equação de evolução usada, poderemos ter uma situação como a da Figura 7.10 onde a snake avançou sobre nós já queimados.

Quando projetada sobre a triangulação, os pontos correspondentes à região tracejada ficarão em triângulos onde a função característica não muda seu valor, e portanto não serão considerados durante a aplicação do algoritmo de classificação descrito na seção 5.3.2.

Neste caso, teremos que *corrigir* a snake nestas regiões, o que pode ser feito, por exemplo, substituindo-se os três snaxels indicados pelo nó apontado. Esta é a política por nós adotada.

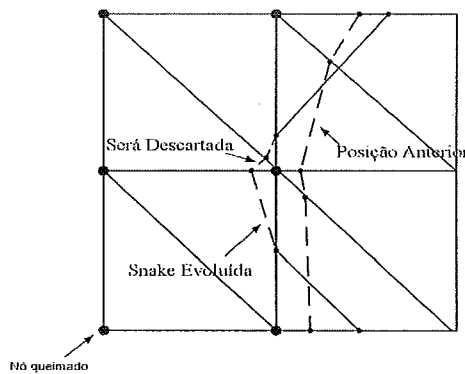


Figura 7.10: Snake avançou para região onde função característica não muda de sinal.

Uma outra possibilidade seria evitar que a snake passe por nós da triangulação já queimados usando testes locais, antes que a snake seja movimentada. Esta política

no entanto não diminui a complexidade do método e suas vantagens não são aparentes.

7.7 Pré-Processamento da Imagem

Os métodos de snakes são em geral precedidos de algum tipo de tratamento da imagem. Isto se faz necessário tanto para reduzir ruídos quanto para realçar as bordas.

No caso do Dual-T-Snakes esta questão está diretamente vinculada com o operador de afinidade.

Existe um detalhe de natureza prática a ser apontado inicialmente.

Consideremos a Figura 7.11 . Enquanto o interior do objeto é muito ruidoso, o background é simples. Assim, um método de snakes usual, com a snake inicializada no background poderia ser eficiente e apresentar melhor performance.

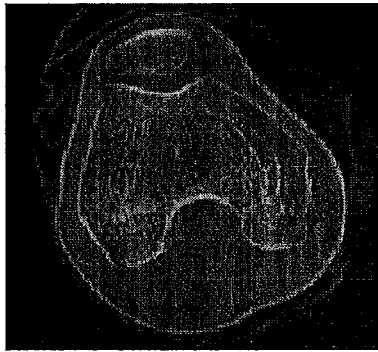


Figura 7.11: Imagem contendo objeto com interior complicado mas background simples.

A relação custo/benefício do método dual é mais vantajosa em situações como a da Figura 7.12 onde temos o interior e o exterior do objeto de interesse complicados.

Este fato é importante para definir o tipo de pré-processamento da imagem, caso este se torne necessário. A idéia é usar filtros com pouca localização no espaço a fim de criar uma *faixa* contendo a borda desejada. Esta faixa será extraída via o Dual-T-Snakes e posteriormente a programação dinâmica fará o resto.

Dentre os métodos que podem fornecer tal resultado, temos filtros passa alta (Canny, 1983), estatística para a imagem (Jain, 1989) e método das conexões fuzzy (Udupa & Samarasekera, 1996; Jones & Metaxas, June, 1998), este último sendo particularmente interessante por aproveitar a interação com o usuário na inicialização do Dual-T-Snakes (Apêndice E).

A exemplo do Balloon, fazemos interpolação bilinear da força externa F para termos uma definição contínua da mesma.

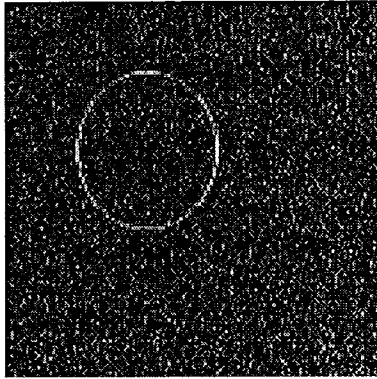


Figura 7.12: Imagem contendo objeto imerso em background com ruído.

7.8 Evitando Mínimos Locais

Uma vez detectado que uma snake congelada não está próxima da borda (critério de proximidade da seção 6.1) o passo seguinte é intervir em seu movimento fazendo com que mesma saia do ponto de congelamento.

Neste trabalho, consideramos três possibilidades:

1. Aumentar normal
2. Variar Limiar
3. Mudar resolução da malha

A primeira política não é interessante para imagens com muito ruído. Isto decorre do fato de estarmos usando a força normal dependente do limiar da imagem.

Como exemplo, consideremos novamente a Figura 7.6. Se o comportamento representado se repetir ao logo da snake então, pela definição (5.54), teremos oscilação da mesma sem retirá-la do congelamento. A definição (E.12) seria mais eficiente, uma vez que esta não inverte o sentido da normal e portanto não cria oscilações. Contudo, a sensibilidade da T-Snake com relação a mínimos locais aumenta o que pode implicar em perda de performance.

Esta política é útil em imagens com muitos artefatos no background mas pouco ruído. Nestes casos, o aumento da força normal tem o papel básico de vencer localmente a resultante das forças de suavização e de campo, tirando a snake de mínimos locais da energia de imagem.

Em imagens muito ruidosas, a variação do limiar da imagem pode ser mais eficiente uma vez que isto pode liberar snaxels presos pelo limiar adotado, perturbando a snake e permitindo que esta avance em direção à borda. Contudo, esta tem a desvantagem de exigir uma estatística da imagem. Além disto, o limiar deve ser relaxado com cuidado para não criar *falhas* muito grandes na borda, o que poder fazer com que uma snake parcialmente estabilizada avance além da mesma.

Como já comentado acima, a mudança na resolução da malha altera as forças

internas, mudando a resultante de forças sobre cada snaxel. Esta alternativa é particularmente interessante para objetos cujas bordas possuem pontos com alta curvatura.

7.9 Detalhes da Implementação do Dual-T-Snakes

Nesta seção, apresentamos detalhes da implementação que fizemos do Dual-T-Snakes. Nossa implementação é seqüencial, feita para estações IBM e usando os recursos para desenvolvimento de softwares e visualização do Data Explorer (Corporation, 1993).

O Data Explorer é um tipo particular de Ambiente Modular de Visualização (*Modular Visualization Environments: MVE*) o que grosseiramente remete às características comuns dos mesmos (Cameron, 1995):

- .Modular

- .Constituído por blocos de código (Módulos).

- .Metodologia orientada a objetos.

- .Novos módulos podem ser criados pelo usuário.

- .Ambiente

- .Tipicamente, um ambiente de programação visual.

- .Permite integração e criação interativa de programas.

- .Visualização

- .O objetivo final dos programas criados é tipicamente visualização de dados.

Estes sistemas tem como vantagens a interface com o usuário simplificada pelo ambiente de programação visual, a possibilidade de criação de novos módulos e integração destes com outros já disponíveis.

Os MVE têm como limitações a dificuldade de integração entre visualização e simulação e a dificuldade de manipulação de grandes volumes de dados.

Considerando estes fatores, optamos por utilizar um sistema do tipo MVE para o desenvolvimento da implementação do Dual-T-Snakes pois acreditamos que os recursos de programação visual, programação distribuída e modularidade destes sistemas trazem bons resultados.

Assim como no Capítulo 6, chamaremos *Dual Snake* a uma lista de T-Snakes onde a primeira é uma snake externa e as demais são snakes internas a esta.

O programa possui basicamente três estruturas: a fila de dual snakes a serem processadas (*group_dual_in*), a fila de dual snakes ancoradas (*final_result*) e uma *hash table* usada durante a execução do algoritmo para efetuar mudanças topológicas.

O Dual-T-Snakes tem complexidade $O(N_T)$, onde N_T é o número de triângulos da malha.

Cada entrada da hash table tem uma *chave* dada por um valor inteiro calculado pela expressão:

$$chave = 2 * id1 + triangle_id + 2 * counts_y * id_x, \quad (7.13)$$

onde $(id0, id1)$ são as coordenadas (inteiras) do nó esquerdo inferior da célula que contém o ponto e $triangle_id = 0$ se o ponto pertence ao triângulo inferior e $triangle_id = 1$ caso contrário (estamos supondo triangulações CF, como a da Figura 7.10).

Na inicialização do loop principal do algoritmo as snakes são projetadas sobre a triangulação, para em seguida aplicarmos um algoritmo de scan line que inicializa a função característica.

Uma vez evoluída a snake segundo a equação 7.1, esta função será atualizada segundo o algoritmo de classificação da seção 5.3.2. Uma vez projetada a snake sendo processada, seus pontos são distribuídos pela hash table para a finalização do método de reparametrização das T-Snakes.

Cada ponto da snake projetada é guardado na entrada correspondente da hash table, juntamente com outras informação, segundo a estrutura a seguir:

```
typedef struct _Pqueue{
    double x;           /* abscissa do ponto */
    double y;           /* ordenada do ponto */
    int id0;            /* abscissa da célula */
    int id1;            /* ordenada da célula */
    int curve;          /* identificador da curva no dual snake */
    int flag;           /*Ponto Ancorado: flag=0; Freezing_Point: flag=2 */
    char triangle_id;   /* 0 para triângulo inferior e 1 para superior */
    int position;       /* Índice do ponto na snake */
    char edge;          /* 0 para horizontal; 1 para diagonal; 2 para vertical */
    struct _Pqueue *next;
} Pqueue;
```

Ao final deste processo, dada uma entrada da hash table, ou ela aponta para NULL ou tem o endereço da fila que guarda os snaxels pertencente ao triângulo correspondente. Esta hash table será percorrida e ao final teremos as snakes propriamente ditas (já reparametrizadas).

Isto foi implementado da seguinte forma. Uma vez encontrada uma entrada da hash table que não seja nula, escolhemos a aresta do primeiro ponto guardado como sendo a *door-in* e a próxima como sendo a *door-out*. Com isto, sabemos exatamente para qual entrada da hash table devemos seguir. Este processo se repete até que a primeira aresta tipo *door-in* obtida seja encontrada novamente; quando então, teremos uma componente conexa da snake completa.

Uma vez que as dual snakes são independentes entre si, a sua escolha para o processamento é feita seguindo simplesmente a ordem em que aparecem na fila *group_dual_in*.

Selecionada uma duas snake, a curva a ser processada é definida segundo os critérios de energia e afinidade da seção 6.1.

Se uma snake sofreu mudança topológica, as dual snakes por ela definidas são colocadas no final do *group_dual_in* para serem processadas posteriormente.

O algoritmo possui dois loops que compõem o loop principal.

No primeiro, uma snake congelada será processada usando-se algum dos critérios (1)-(3) acima para retirar a snake do mínimo local correspondente. Uma vez feito isto, o segundo loop evolui a snake até que o próximo ponto de congelamento seja atingido ou uma mudança topológica ocorra. Neste último caso, o *group_dual_in* é atualizado com as dual snakes resultantes e volta-se ao início do loop principal. A equação de evolução é dada pela expressão 7.1.

O fluxograma a seguir esclarece como estes elementos foram organizados na implementação do Dual-T-Snakes.

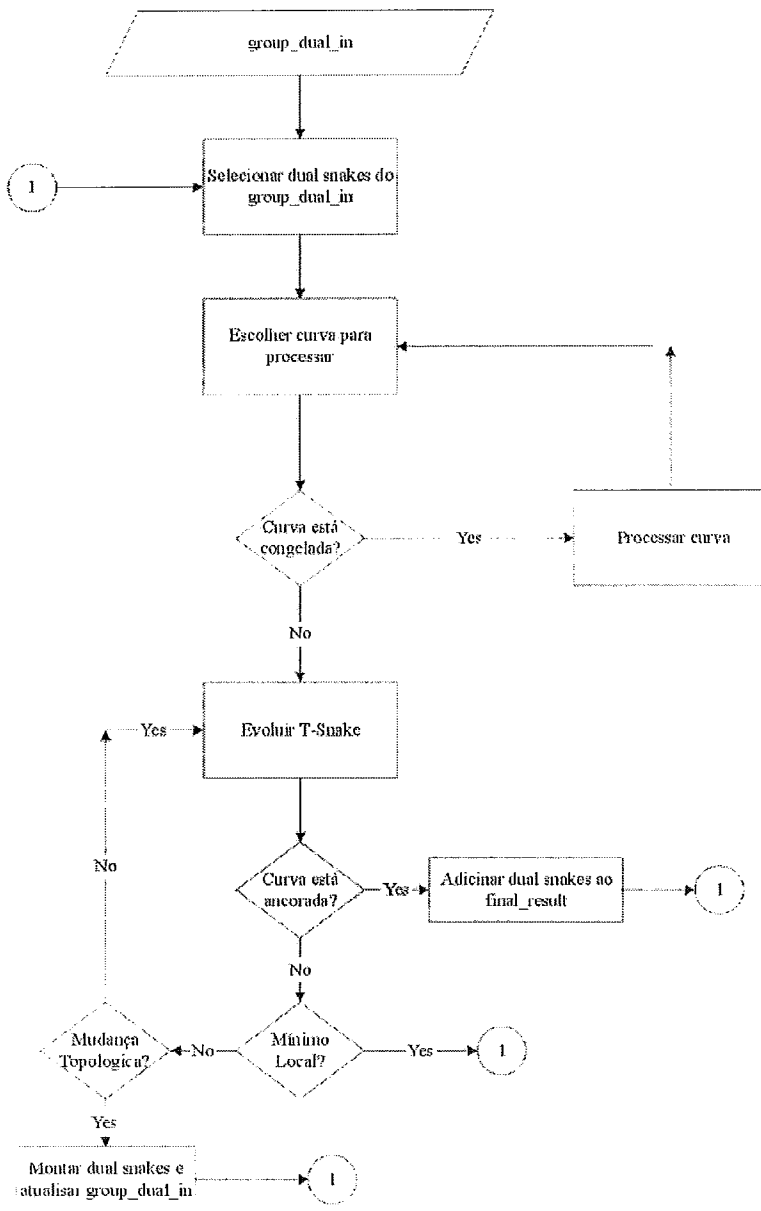


Figura 7.13: Fluxograma descrevendo as principais partes da implementação do Dual-T-Snakes.

Capítulo 8

Inicialização de Modelos Deformáveis

Um ponto crucial para a reconstrução de geometrias em imagens 2D ou 3D via modelos deformáveis, é a inicialização do modelo.

Como já ressaltado para o caso particular das snakes (Kass *et al.* , 1988), a energia do modelo é em geral não-convexa (Davatzikos & Prince, 1999; Giraldi & Oliveira, 1999) e portanto a solução final é muito dependente da posição inicial. Assim, um procedimento para obter uma primeira aproximação da borda pode melhorar a performance e a precisão do resultado final.

Em (Giraldi *et al.* , 2000e; Giraldi *et al.* , 2000a) nós apresentamos um novo método para inicializar modelos deformáveis baseado em propriedades da topologia e da escala espacial dos objetos de interesse. O método proposto adota uma metodologia de multiresolução tendo como base teórica os mesmos elementos das T-Snakes (*Triangulação, Função Característica e Snakes Discretas*), o que o torna ideal para inicializar o Dual-T-Snakes.

O enfoque dos trabalhos apresentados em (Giraldi *et al.* , 2000e; Giraldi *et al.* , 2000a) é na inicialização de snakes 2D, mas o mesmo procedimento pode ser estendido para 3D, como discutido abaixo.

8.1 Trabalhos Relacionados

Nossa metodologia de inicialização adota a filosofia básica de alguns métodos não-paramétricos multi-resolução para segmentação de imagens baseados em modelos de Pirâmides e Quadtree (Burt *et al.* , 1981; Bamford & Lovell, n.d.; Jolion & Montanvert, 1992). Estes métodos fazem uso de estruturas de dados apropriadas para representar uma imagem em diferentes níveis de resolução.

A idéia básica nestes casos, é que à medida que a resolução vai decrescendo pequenos artefatos no background tornam-se menos significativos em relação aos objetos de interesse. Assim, pode ser mais simples *detectar* os objetos no nível mais

baixo para então fazer o tracking dos mesmos nos diferentes níveis da estrutura de dados.

Desta forma, é possível *delinear* as bordas em uma resolução mais baixa e re-estimá-las quando aumentamos a resolução (Jolion & Montanvert, 1992; Bamford & Lovell, n.d.).

No trabalho descrito nesta seção, a resolução mais baixa da imagem é definida através de um conhecimento a priori da escala dos objetos, sendo adaptativo no sentido que a resolução é melhorada somente nas regiões onde for considerado necessário.

Para aumentar a resolução, nós simplesmente refinamos a malha e amostramos a imagem sobre os nós correspondentes à nova malha. Assim, a estrutura de refinamento é localmente uniforme e adaptativa (*local uniform nested refinement*) (Berger & Oliger, 1984), sendo representada na Figura 8.1. Estes métodos são bem conhecidos para solução numérica de equações diferenciais parciais, tais como aquelas do método de conjuntos de níveis da seção 5.3.1 (Sethian, 1996). Contudo, não conhecemos referência anterior sobre aplicações destes esquemas para segmentação.

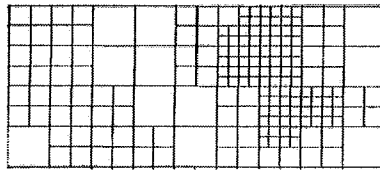


Figura 8.1: Representação do esquema de multi-resolução usado.

Métodos de multi-resolução podem ser particularmente úteis para problemas em análise de imagens e visão computacional que possam ser expressos como problemas de minimização de funções de energia. Em (Heitz *et al.* , 1994) nós encontramos um método deste tipo, chamado Relaxação Multi-escala (*Multiscale Relaxation*) o qual tem sido usado no contexto de snakes (Mignotte & Meunier, February, 2000). Neste método, o problema de otimização global é resolvido dentro de uma seqüência de subespaços do espaço original de configurações. Cada subespaço define uma função de energia (*coarse energy function*) cujos parâmetros são derivados da função objetivo original definida na resolução máxima. O problema de otimização vinculado (*constrained optimization*) é implementado em (Heitz *et al.* , 1994) usando-se uma estrutura de pirâmide.

Por outro lado, existem métodos de relaxação em malhas , tais como aquele apresentado em (Terzopoulos, 1986a). Neste caso, a resolução da imagem não é reduzida (é sempre a resolução máxima). Contudo, a resolução da malha usada para discretizar o problema de otimização vai sendo melhorada durante a execução do algoritmo.

O método descrito abaixo é multigrade/multiresolução quando segmentamos a imagem mas não quando aplicamos o modelo deformável para extrair as bordas. Este modelo (T-Snakes, seção 5.3.2) é usado no final do processo apenas. Nós não necessitamos usar métodos de relaxação multi-escala neste passo final uma vez que tomamos sempre a resolução máxima da imagem para evoluir as T-Snakes. Além disto, ao aplicarmos T-Snakes estamos supondo que a resolução da malha é suficiente para fornecermos uma aproximação inicial das bordas. Assim, nós não necessitamos usar métodos de relaxação multigrad durante a evolução das T-Snakes.

8.2 Método Multi-resolução

Nesta seção, nós estaremos interessados em aplicações onde os padrões de intensidade de um objeto O (ou do background) possam ser caracterizadas por um limiar T ou uma estatística (média μ e variância σ) da intensidade de imagem I (McInerney & Terzopoulos, 1999). Isto é:

$$p \in O \implies I(p) > T \text{ or } |I(p) - \mu| \leq k\sigma \quad (8.1)$$

onde k é uma constante definida a priori.

Se a intensidade de imagem não for apropriada ou tivermos uma imagem multi-variada (imagem colorida, com texturas, etc), qualquer outro campo escalar derivado da imagem pode ser usado (Jones & Metaxas, June, 1998; Giraldi *et al.*, 2000e).

Primeiramente, nós assumimos uma *Propriedade de Escala Local*: dado um ponto $p \in O$, existe uma bola B_p com raio $r_p > 1$, a qual contém p e pertence inteiramente ao objeto (p não é necessariamente o centro da bola). Assim, uma vez que o conjunto de pontos (pixels) em uma imagem digital é finito, existe um raio r_l tal que $r_p \leq r_l$, para todo $p \in O$.

$$r_l \geq r_p, \text{ para todo } p \in O. \quad (8.2)$$

Esta propriedade de escala local implica que nós podemos reduzir a resolução da imagem para $r_l \times r_l$ sem alterar a topologia dos objetos de interesse. Esta é a idéia básica do método, representada na Figura 8.2.

Nesta figura, nós usamos um limiar para a intensidade de imagem o qual identifica o objeto ($T < 150$). Na Figura 8.2.b nós temos uma triangulação CF com resolução 10×10 , e a imagem binária correspondente.

Agora, nós podemos definir uma função simples, a qual chamaremos *Função Característica do Objeto*, como segue:

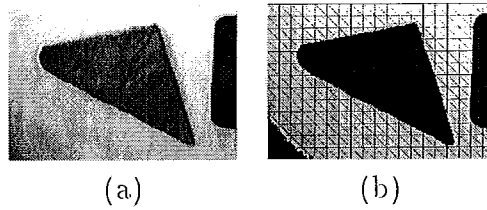


Figura 8.2: (a)Imagem original. (b)Malha e imagem truncada pelo limiar.

$$\chi : D \subset \mathbb{R}^2 \rightarrow \{0,1\} \quad (8.3)$$

onde $\chi(p) = 1$ se p satisfaz (8.1) e $\chi(p) = 0$ caso contrário, sendo p um nó da triangulação. Para o caso 3D a definição é análoga.

Podemos ir um passo além, como mostra a Figura 8.3.b, onde temos a curva gerada pela aplicação do algoritmo *Door-In-Door-Out* da seção 4.3.2. Observe que esta curva é uma aproximação (grosseira) da borda procurada.

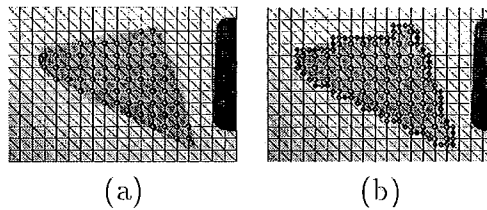


Figura 8.3: (a)Função Característica. (b)Aproximação obtida.

A idéia agora, é usar um método que possa melhorar esta aproximação bem como a sua suavidade. O método das T-Snakes é o mais indicado, como veremos a seguir.

8.3 Método para Inicialização de Snakes

A seção anterior propriamente exemplifica os passos básicos do método de inicialização para snakes 2D proposto em (Giraldi *et al.* , 2000e; Giraldi *et al.* , 2000a):

- 1)Extração de estatísticas dos objetos ou do background;
- 2)Triangulação do domínio;
- 3)Definição das Funções Características dos Objetos;
- 4)Aplicação do algoritmo *Door-In-Door-Out*;
- 5)Aplicação de T-Snakes para suavizar as aproximações obtidas.

As estatísticas (ou limiares) podem ser obtidos pelos procedimentos usuais de análise de histograma ou técnicas de reconhecimento de padrões (em (Jain *et al.* , 2000) encontramos uma revisão recente do assunto).

A aplicação eficiente do *Door-In-Door-Out* pressupõe certas restrições para a topologia das bordas dos objetos de interesse. Especificamente, supomos que estas podem ser representadas por curvas (superfícies) com as seguintes propriedades: (a) Fechadas, (b) Orientadas, (c) Conexas.

Com estas propriedades assumidas, nós não necessitamos nos preocupar com o interior das curvas fechadas assim geradas pelo passo (4). A Figure 8.4.b, a qual é o resultado obtido pela aplicação dos passos (1)-(4) à imagem mostrada na Figura 8.4.a, esclarece este fato.

Uma vez que nós não temos splits de objetos devido a propriedade local de escala, não precisamos efetuar merge de regiões. Assim, podemos usar as curvas fechadas geradas para obter uma aproximação grosseira das bordas procuradas. Este é o ponto principal deste trabalho.

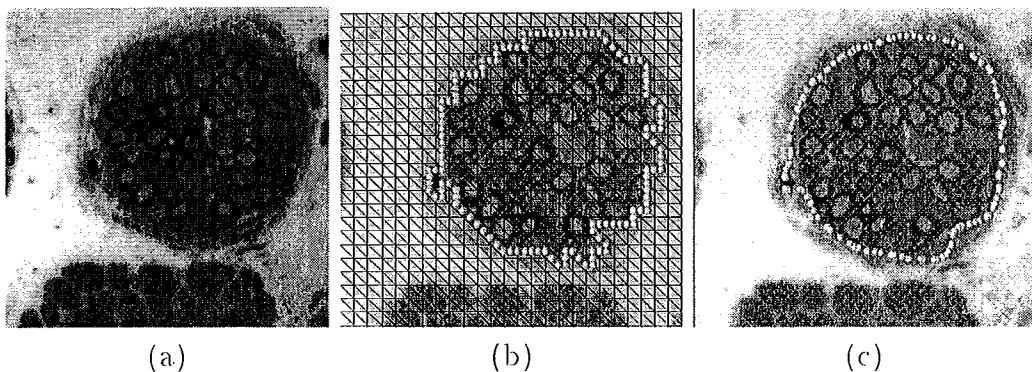


Figura 8.4: (a) Célula de mielina. (b) Variedade Linear gerada. (c) Resultado final do método.

Observe que, de acordo com a propriedade Cíclica da seção 4.3.2, o algoritmo *door-in-door-out* gera curvas fechadas ou curvas que iniciam e terminam nas fronteiras da imagem (Allgower & Georg, 1990). Assim, uma inspeção simples descarta curvas abertas. Além disto, ambas as propriedades (a) e (b) da seção 4.3.2 garantem que as curvas poligonais geradas são simples (Allgower & Georg, 1990). Logo, a propriedade (b) acima é sempre satisfeita.

As curvas restantes satisfazem (a)-(c) acima, como podemos observar na Figura 8.4.b.

Contudo, algumas destas curvas podem limitar uma região de área muito pequena devido a artefatos ou ruídos no background. Logo, elas deverão ser descartadas. A Figura 8.5 mostra um exemplo deste comportamento.

Além disto, algumas curvas podem conter mais de um objeto de interesse, como é o caso mostrado na Figura ??b.

Agora, nós podemos usar um conjunto de características pré-definidas tais como área e perímetro (Santaney *et al.*, 1994). Para cada uma destas características,

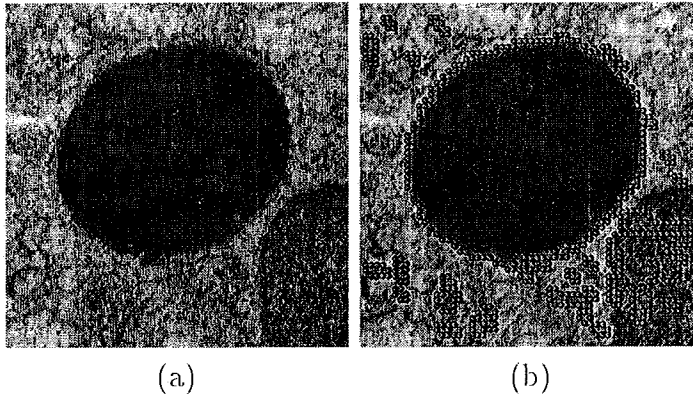


Figura 8.5: (a) Células de lisossomos. (b) Aproximações lineares por partes obtidas para resolução 5×5 .

a idéia é encontrar limites superior e inferior baseados em conhecimento a priori dos objetos de interesse. Neste trabalho nós usamos apenas área da região interior às curvas. Pela propriedade de escala local e a triangulação usada nós podemos estabelecer como limite inferior para a área o valor $4(r_l)^2$, onde r_l está definido na expressão 8.2.

A definição para o limite superior é muito dependente da aplicação. Por exemplo, para imagens médicas poderíamos usar dados anatômicos sobre as estruturas de interesse.

É importante ressaltar que o limite superior não é um ponto essencial para o método. O papel deste parâmetro é somente evitar esforço computacional para processar regiões onde as poligonais envolvem um único objeto.

Dentre as curvas resultantes, uma simples inspeção descarta aquelas que envolvem regiões cujas áreas estão abaixo do limite inferior. Quanto àquelas cujas áreas correspondentes ultrapassam o limite superior, estas devem ser processadas em uma resolução mais fina.

De fato, em imagens como a da Figura ??a, a escala *exterior* correspondente à separação entre os objetos pode ser mais fina que aquela definida pela propriedade de escala local dos mesmos. O problema correspondente se enquadra na categoria conhecida como *Problemas de Duas-Escalas*.

Uma característica comum destes problemas é que as escalas espaciais podem variar dramaticamente de uma região para outra da imagem. Isto acontece nas regiões da Figura ??b onde a segmentação ainda não está completa. Nestas regiões, a escala exterior pode ser mais fina que a escala local dos objetos. Assim, na resolução mais grossa (5×5) não conseguimos *separar* os objetos. Para corrigir este resultado, nós aumentamos a resolução (apenas) nestas regiões para melhorar o nível de detalhe.

Para malhas uniformes do tipo daquela da Figura 8.1 isto pode ser implementado

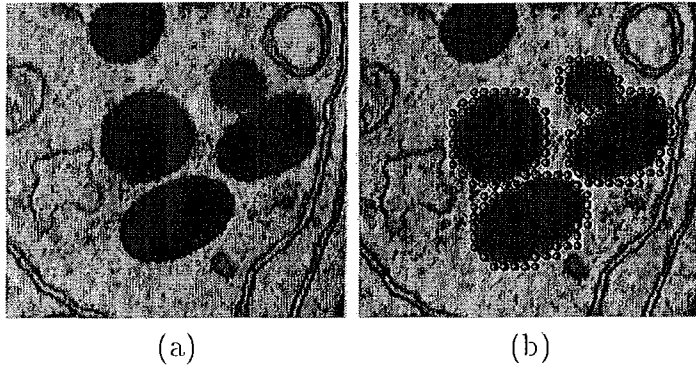


Figura 8.6: (a) Escala de separação entre objetos mais fina que a escala interna. (b) Aproximações lineares por partes obtidas com resolução 5×5 .

por um esquema de multi-resolução cuja estrutura de dados subjacente é aquela de um *Refinamento Adaptativo de Malhas* (Berger & Olinger, 1984).

Nestas estruturas cada nó no nível de refinamento l sofre split em η^n nós no nível $l+1$, onde η é o *fator de refinamento* e n é a dimensão do espaço ($\eta = 2$ e $n = 2$, no presente caso). Assim, em 2D, a estrutura de nós possui η^2 ponteiros para os filhos, η^2 ponteiros para os nós vizinhos e um ponteiro para o pai.

Com isto, podemos melhorar o resultado apresentado na Figura 8.6.b. Usando agora uma resolução 3×3 obtém-se o resultado mostrado na Figura 8.7



Figura 8.7: Resultado obtido com resolução 3×3 , a partir da imagem 8.6.b

Em alguns casos, apesar de usarmos a resolução da própria imagem, algumas curvas ainda podem limitar regiões com área maior que o limite superior, como é o caso da curva limitando as duas células do canto direito superior da Figura 8.7.

Se nós temos uma estatística como na expressão (8.1), então podemos relaxar o limiar ($T \rightarrow T \pm \Delta T$) e aplicar os passos (2)-(5) novamente, mas somente nas regiões onde for identificada tal necessidade. Contudo, no caso da Figura 8.7 este procedimento não seria eficiente pois comprometeria o resultado obtido nas regiões onde a aproximação está satisfatória.

Nestes casos, nos parece mais eficiente uma metodologia semi-automática com ferramentas interativas para corrigir o defeito observado. Nos Capítulos 9 e 10 voltaremos a esta questão.

Dentre as curvas resultantes, pode ocorrer que algumas delas ainda correspondam a pequenos artefatos, muito próximos no background. Em geral, estas curvas somente podem ser descartadas aumentando a resolução.

Após aplicar os passos (1)-(4) nós temos como resultado uma aproximação grosseira das bordas procuradas. A idéia é usar agora um modelo de snake para obter a aproximação final. O modelo das T-Snakes foi escolhido, primeiramente, porque ele pode lidar naturalmente com as auto-intersecções que podem ocorrer durante a evolução das curvas obtidas as quais podem apresentar regiões irregulares e com curvatura muito elevada (Figura 6.5).

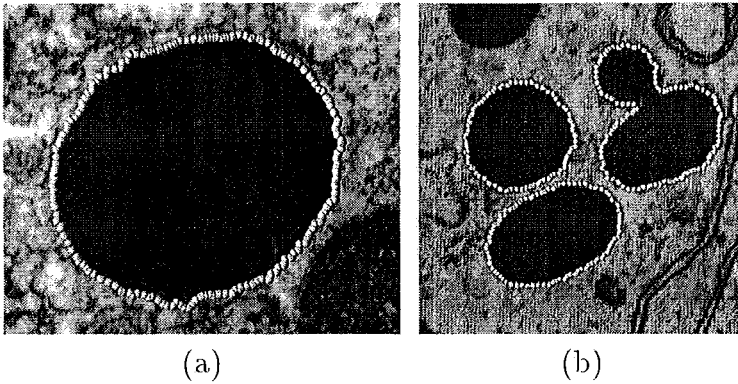


Figura 8.8: Resultado final dos passos (1)-(5) para as imagens das Figuras 8.5.a e 8.7. As resoluções usadas são 5×5 para o resultado (a) e 3×3 para (b).

Segundo, porque escolhendo T-Snakes, os passos (1)-(5) compõem uma metodologia unificada, baseada em Triangulação, Limiares e Funções Características, para inicialização de modelos deformáveis.

O algoritmo de continuação *door-in-door-out* usado tem muito em comum com métodos usados para geração de iso-superfícies em malhas hierárquicas. Contudo, estes métodos não usam as restrições de escala e topologia acima, as quais formalizam nosso *conhecimento a priori* das estruturas de interesse.

A extensão do método desta seção para 3D é direta. Para isto, basta usar as T-Superfícies (seção C.3) no passo (5) e uma extensão para 3D do algoritmo *Door-In-Door-Out* para gerar as superfícies procuradas (Variedades Lineares por Partes

bidimensionais). Procedendo desta forma, as restrições topológicas (a)-(c) da seção anterior serão automaticamente respeitadas, o que pode ser demonstrado pela teoria subjacente descrita no Apêndice C.

Capítulo 9

Resultados Experimentais

Neste capítulo apresentamos resultados experimentais para imagens 2D obtidos utilizando-se o Dual-T-Snakes juntamente com Programação Dinâmica (Viterbi), bem como com o método de inicialização do Capítulo 8. Os exemplos apresentados são constituídos basicamente por imagens sintéticas e imagens médicas.

Na discussão que segue procuramos inicialmente demonstrar a viabilidade e eficiência do Dual-T-Snakes, tanto para evitar mínimos locais quanto no que diz respeito às suas habilidades topológicas. Neste sentido, um ponto crítico é a escolha dos parâmetros das forças do modelo 7.1, e como este se comporta quando aumentamos o ruído da imagem ou mudamos a resolução da triangulação.

O desenvolvimento de procedimentos sistemáticos para a escolha de parâmetros é um campo ainda aberto no contexto de modelos deformáveis em geral, e snakes em particular, como ressaltado em (R. Fisker, 1998). Assim, nossa política para a escolha de parâmetros é empírica, baseada em tentativa e erro.

Esta dificuldade na "calibração" do método de snakes exige que os valores destes coeficientes tenham alguma "estabilidade" com relação a fatores tais como aumento de ruído na imagem e presença de artefatos. Este ponto é crucial para o Dual-T-Snakes uma vez que temos potencialmente duas snakes para calibrar.

Assim, uma vez determinado um conjunto de parâmetros eficiente para uma dada imagem, a metodologia seguida nesta apresentação será analisar o comportamento do modelo quando alteramos características da imagem tais como ruído e artefatos no background ou mudamos a resolução da malha.

Neste sentido, o primeiro ponto a ser destacado são as conseqüências da definição para força normal dada pela equação 7.3. Em imagens com ruídos, esta política melhora a estabilidade do algoritmo e conseqüentemente sua performance. Contudo, analogamente ao caso do Balloon (seção 3.4), a snake continua sensível com relação aos mínimos locais da energia de imagem; um problema que o Dual-T-Snakes tem recursos para resolver.

O limiar usado para a intensidade de imagem pode deixar "artefatos" no background ou mesmo eliminar partes da borda introduzindo subjetividade na mesma.

É interessante portanto analisar o comportamento do Dual-T-Snakes com relação a artefatos e bordas subjetivas.

Comparamos os resultados obtidos com aqueles usando o modelo das T-Snakes, particularmente com relação aos mínimos locais. Esta comparação se faz pertinente para mostrar como a metodologia dual acrescenta robustez à T-Snake sem limitar suas habilidades topológicas.

Os resultados obtidos são avaliados mediante comparação visual.

Nos casos estudados, indicamos os valores do parâmetros e o número de interações, possibilitando assim uma análise mais completa.

O resultados obtidos com o Viterbi (seção 5.2), os quais fornecem a borda desejada, são mostrados no final de cada seção.

Os resultados da seção 9.2.1 a seguir ilustram como o método da seção 8.3 pode ser usado para inicializar automaticamente o Dual-T-Snakes. As seções 9.2.3, 9.2.4 e 9.2.6 mostram resultados deste mesmo método para imagens de células e imagens extraídas do Projeto Homem Visível. As demais seções são dedicadas exclusivamente ao Dual-T-Snakes/Viterbi

Na apresentação abaixo entendemos por uma *interação* a uma execução de algum dos loops do fluxograma da Figura 7.13. De maneira equivalente, diremos que cada interação corresponde a um *instante de tempo* t da evolução da snake. Durante toda a discussão, chamamos *Snake 0* a snake externa de um dual snakes e *Snake i* , $i = 1, 2, \dots$ as internas.

9.1 Imagens Sintéticas

Nesta seção, os testes feitos têm a finalidade principal de demonstrar o comportamento do Dual-T-Snakes e do Viterbi com relação a ruídos, artefatos no background e mudança na resolução da malha.

Iniciamos com uma imagem com um único objeto imerso em um meio ruidoso. Analisamos o comportamento do Dual-T-Snakes para as duas definições da força normal (equações (7.3) e (5.54)).

Em seguida, analisaremos o comportamento do modelo (e dos parâmetros) quando aumentamos o ruído. Posteriormente, a resolução da malha é alterada.

Finalmente, usamos imagens com vários objetos, quando então as habilidades topológicas do método são demonstradas.

9.1.1 Circunferência em Ruído

A imagem a seguir mostra uma circunferência imersa em um meio ruidoso. Embora a forma geométrica seja simples, este exemplo caracteriza uma típica situação

onde o Dual-T-Snakes mostra sua potencialidade, uma vez que o interior da circunferência e seu exterior oferecem dificuldades para o movimento da snake.

Caso contrário, em imagens onde o background ou o interior do objeto são simples (pouco ruído e poucos artefatos) a possibilidade mais direta seria usar uma única snake evoluindo na região que oferece menor dificuldade ao seu movimento.

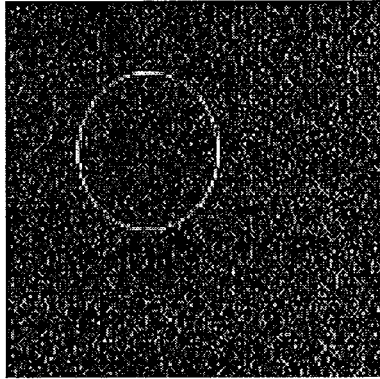


Figura 9.1: Círculo em meio com Ruído.

O imagem tem resolução (128×128) e a inicialização do Dual-T-Snakes neste caso foi feita usando-se uma malha triangular com resolução (3×3) . A Figura 9.2 mostra as snakes no instante inicial e a malha em questão.

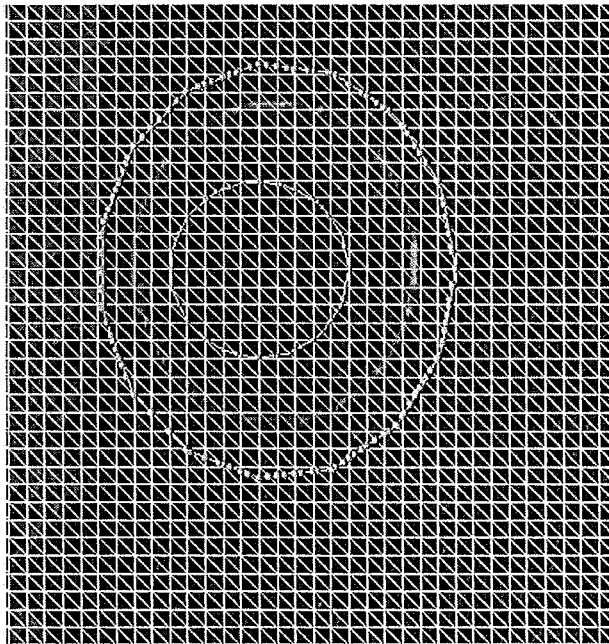


Figura 9.2: Inicialização do Dual-T-Snakes.

Os valores dos parâmetros usados no modelo 7.1 são os seguintes: $k = 200.0$, $\lambda = 50.1$, $b = 50.0$, *ponto de congelamento* = 5.

A imagem tem ruído aproximadamente gaussiano, com média 20.0. Assim, assumimos um limiar $T = 20.0$ na equação 7.3 e variamos este limiar quando necessitamos retirar uma snake de um mínimo local. O intervalo de variação depende da variância correspondente. No caso, usamos um $T + \Delta T \leq 30.0$.

A Figura 9.3 mostra posições intermediárias das snakes.

A solução obtida pelo Dual-T-Snakes está na Figura 9.4.

O número de interações para a snake interna foi igual a 22 enquanto que para a externa foi 14. Pela distância máxima entre a snake externa e a borda observada na Figura 9.2 vemos que, em média, temos que esta snake avança 9/14 pixels a cada interação, o que é considerado aceitável.

A Figura 9.5 mostra o exemplo acima usando a definição da normal usada originalmente nas T-Snakes (equação (5.54)). O número de interações aumentou para 26 no caso da snake interna mantendo-se para a externa. Apesar da mudança de performance ser pouco significativa neste caso, podemos observar que o movimento no caso anterior se faz de forma mais suave. Na próxima seção veremos um exemplo bem mais crítico deste comportamento.

É importante observar o que aconteceria caso tivéssemos apenas uma T-Snake. A Figura 9.5.c mostra a T-Snake externa congelada. A aplicação de estatísticas para retirar a snake da posição de congelamento (equação (5.55)) poderia ser útil. Contudo, como saber automaticamente se a T-Snake congelada está ou não sobre a borda desejada?

Na metodologia do Dual-T-Snakes, o algoritmo identifica primeiramente que o critério de proximidade da seção 6.1 não está satisfeito. Isto é um indicativo de que alguma das snakes não está sobre a borda desejada. A idéia agora, é que a snake com energia maior (equações (6.1)-(6.2)) esteja mais longe da borda, e portanto, deverá ser evoluída, caso sua função de afinidade não seja nula.

Na Figura 9.6 usamos a mesma imagem acima, contudo diminuimos a resolução da malha para (5×5) . Neste caso, o ponto de congelamento precisa ser alterado uma vez que a dimensão das células aumentou. Em geral, escolhemos um valor da ordem da dimensão da diagonal das células (10, no caso).

De acordo com a discussão do Capítulo 7, a mudança na resolução da malha altera as forças internas. De fato, a tendência é aumentar a força de rigidez uma vez que a distância entre os snaxel aumenta em média. Uma vez que dobramos a dimensão da célula, é razoável usar a proporção entre as dimensões das células como diretriz para redefinir a elasticidade. Assim, usamos $b = 25.0$ neste caso.

Todos os outros parâmetros foram mantidos com os valores especificados na Tabela 9.1. As Figuras 9.6.(a)-(d) mostram posições intermediárias das snakes. A solução do Dual-T-Snakes está na Figura 9.6.d. O número de interações correspondente está na Tabela 9.2. O resultado é satisfatório mais uma vez.

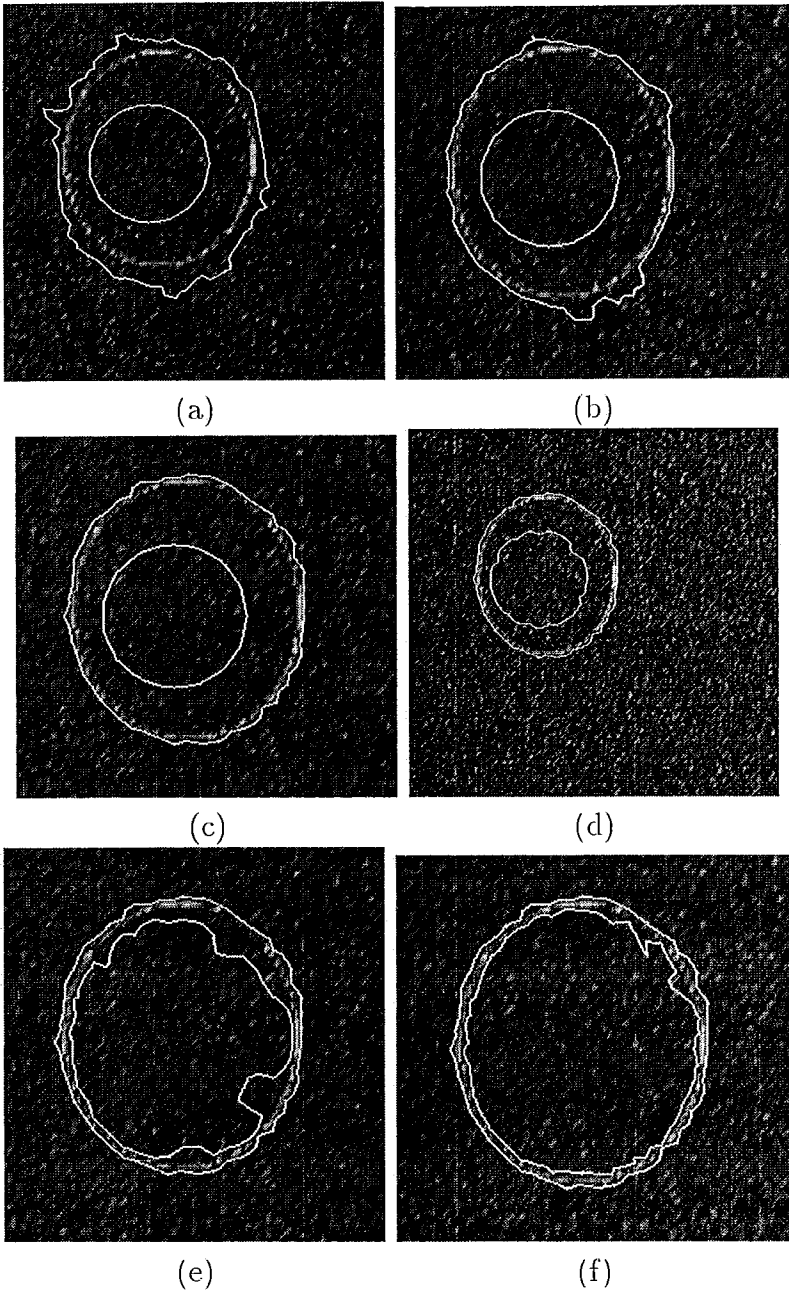


Figura 9.3: Evolução do Dual-T-Snakes para a Figura 9.2: (a)Snake externa antes do ponto de congelamento. (b)Mínimo local da snake externa. (c)Snake externa congelada novamente. (d)Snake interna inicia movimento. (e)-(f)Posições intermediárias da snake interna.

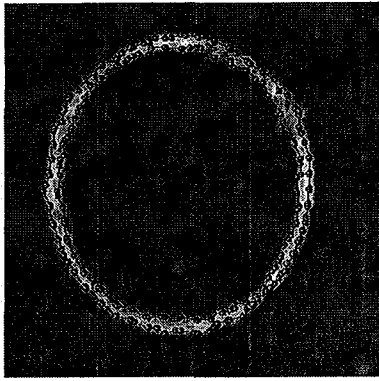


Figura 9.4: Solução do Dual-T-Snakes correspondente a Figura 9.2.

Assim como na Figura 9.4, a solução final tem boa suavidade. Número de iterações neste caso foi 21 para a externa e 39 para a interna. De maneira geral, a performance deste caso, quando comparado com o inicial (Figura 9.2) não teve mudança significativa, como pode ser observado pelas Tabelas 9.1-9.2 que resumem as estatísticas desta seção.

Imagem	k	λ	b	malha	P. Congelamento
Figura 9.3	200.0	50.1	50.0	3×3	5
Figura 9.5	200.0	50.1	50.0	3×3	5
Figura 9.6	200.0	50.1	25.0	5×5	10

Tabela 9.1: Parâmetros usados nos testes desta seção para o Dual-T-Snakes.

As Figuras 9.7 mostram as soluções obtidas pelo Viterbi para os exemplos desta seção. Os parâmetros usados nesta fase foram os seguintes: $\alpha = 1200.0$, $\beta = 0.5$ and $\lambda = 140.0$.

9.1.2 Variando Ruído

Nos exemplos que seguem partimos da imagem inicial na Figura 9.8 e variamos o ruído como indicado na Tabela 9.3. A resolução usada é sempre (5×5) nos exemplos correspondentes. Os parâmetros do Dual-T-Snakes estão também listados nesta tabela.

A Tabela 9.4 mostra o número de interações de cada snake com relação aos loops do fluxograma da Figura 7.13.

Consideremos o pior caso, Tabela 9.4.d. Pela Figura 9.8 observamos que distância máxima entre a snake interna (Snake 1) e a borda procurada é da ordem de 45 pixels. Assim, teríamos nestas regiões uma velocidade média de 45/149 pixels por interação.

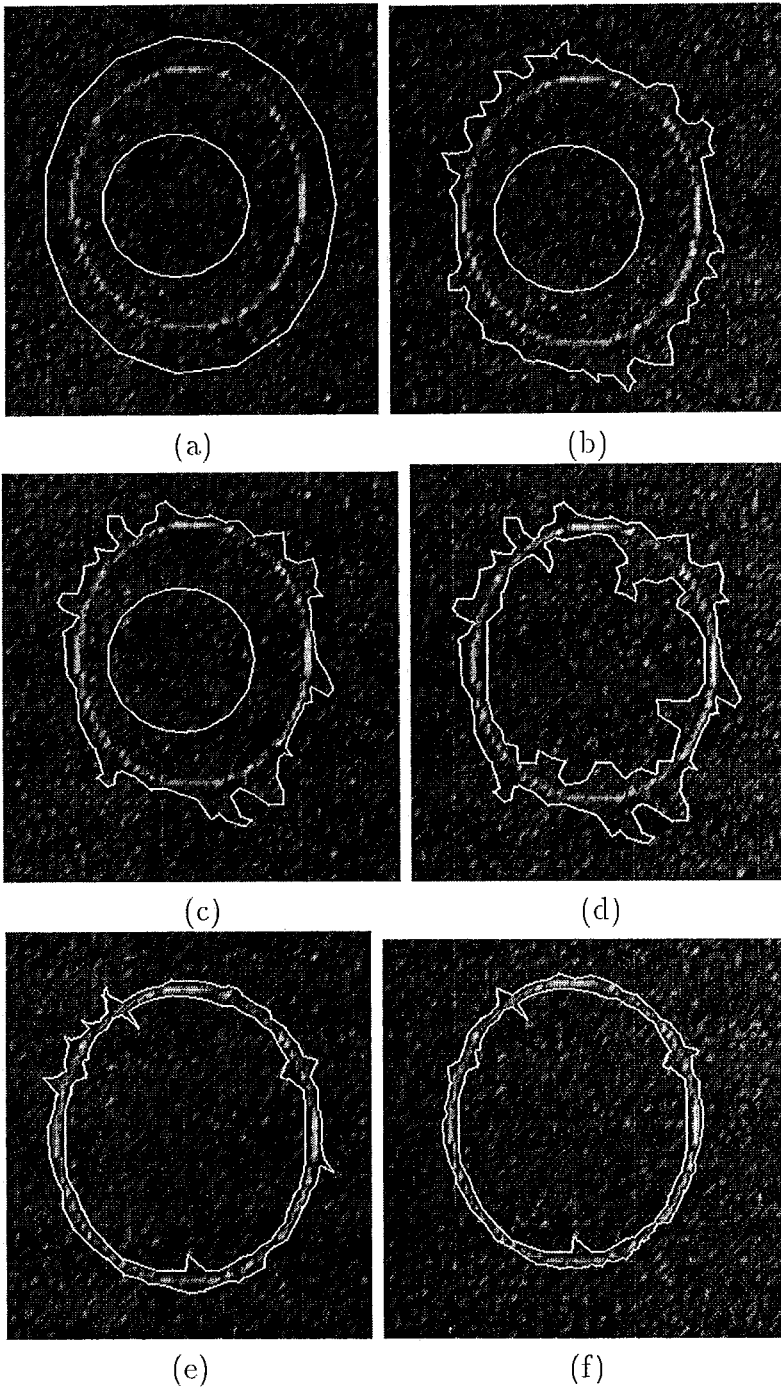
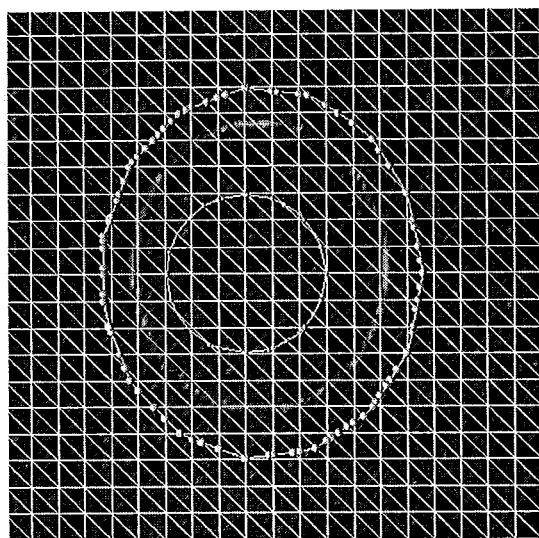
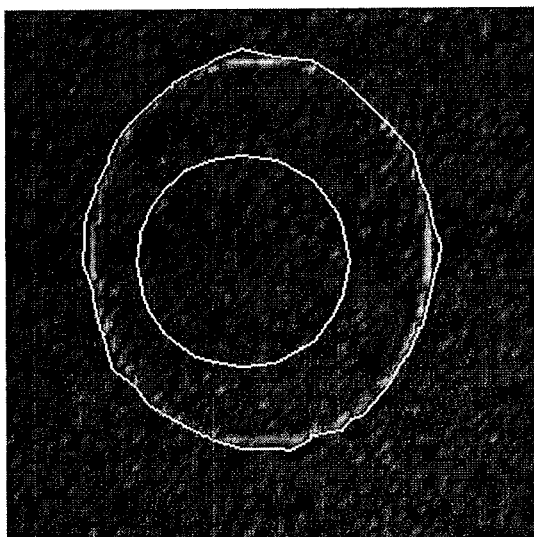


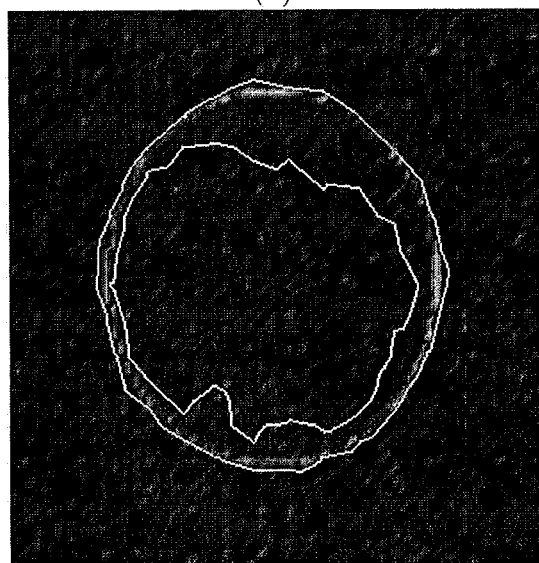
Figura 9.5: Evolução do Dual-T-Snakes para força normal usual: (a)Interação 3. (b)Posição intermediária (c)Mínimo local da snake externa (interação 7). (d)Mínimo local da snake interna. (e)Posição intermediária. (f)Solução final.



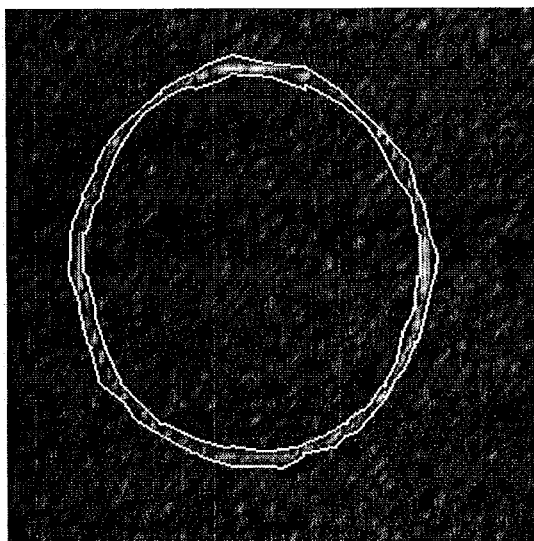
(a)



(b)

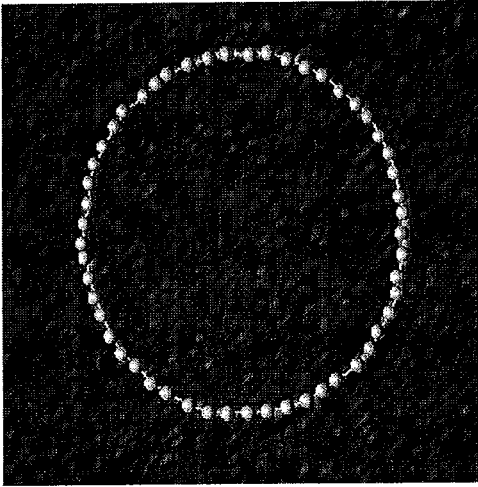


(c)

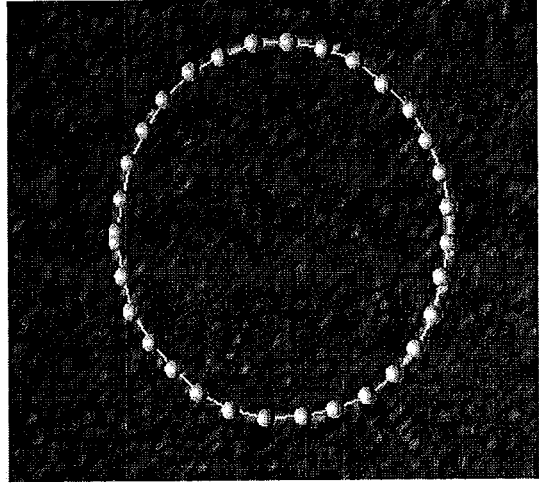


(d)

Figura 9.6: Dual-T-Snakes em malha 5×5 : (a)Inicialização. (b)Interação 21. (c)Interação 31. (d)Solução final do Dual-T-Snakes.



(a)



(b)

Figura 9.7: (a) Solução do Viterbi para Figura 9.4.f. (b) Solução do Viterbi para Figura 9.6.d.

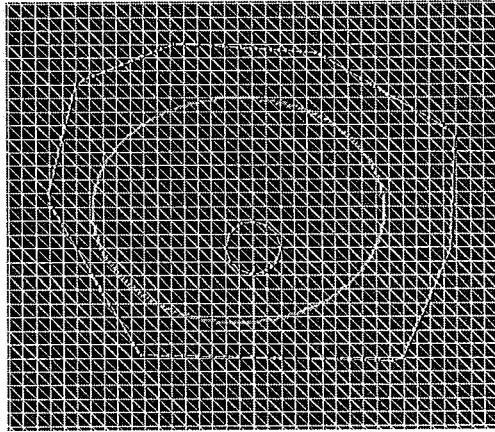


Figura 9.8: Inicialização para os exemplos desta seção.

Imagem 9.3	loop 1	loop 2
Snake 0	2	12
Snake 1	10	12

(a)

Imagem 9.5	loop 1	loop 2
Snake 0	6	8
Snake 1	10	16

(b)

Imagem 9.6	loop 1	loop 2
Snake 0	0	21
Snake 1	19	20

(c)

Tabela 9.2: (a) Performance para força normal dada pela equação 7.3. (b) Performance para força normal dada pela equação 5.54. (c) Performance para malha com resolução 5×5 .

Imagem	Média , Variância	k	λ	b	T
9.9.a	$\mu = 30.0$, $\sigma = 5.0$	200.0	50.1	50.0	30.0
9.9.c	$\mu = 30.0$, $\sigma = 10.0$	200.0	50.1	50.0	30.0
9.9.e	$\mu = 50.0$, $\sigma = 25.0$	200.0	50.1	50.0	50.0
9.9.g	$\mu = 50.0$, $\sigma = 40.0$	200.0	50.1	50.0	50.0

Tabela 9.3: Parâmetros usados nos testes desta seção para o Dual-T-Snakes.

Considerando que nesta figura temos um ruído com média 50.0 e variância 40.0, esta performance nos parece aceitável.

As Figuras 9.9 mostram os resultados do Dual-T-Snakes correspondentes. Ao lado de cada resultado mostramos também as snakes projetadas sobre a triangulação para facilitar a avaliação de cada caso. Observando estas figuras, o primeiro aspecto que chama atenção é a distância entre as snakes interna e externa em certas regiões (Figura 9.9.c, por exemplo).

Este comportamento decorre do critério de parada baseado nas propriedades (a) e (b) da seção 6.1. Ao final de cada interação, o algoritmo analisa a snake sendo processada procurando vizinhanças cortadas pelas duas snakes. Uma vez identificado que todos os snaxels das duas snakes satisfazem este critério, a execução do algoritmo é interrompida. Os defeitos observados não têm se mostrado comprometedores do resultado final do Viterbi, exceto para regiões da borda com curva alta, o que veremos a seguir.

Vejam novamente o efeito da força normal dada pela equação 7.3. Se mantido os valores dos parâmetros acima mas tomada a definição da normal dada pela equação (5.54), observamos uma diminuição considerável de performance. Enquanto

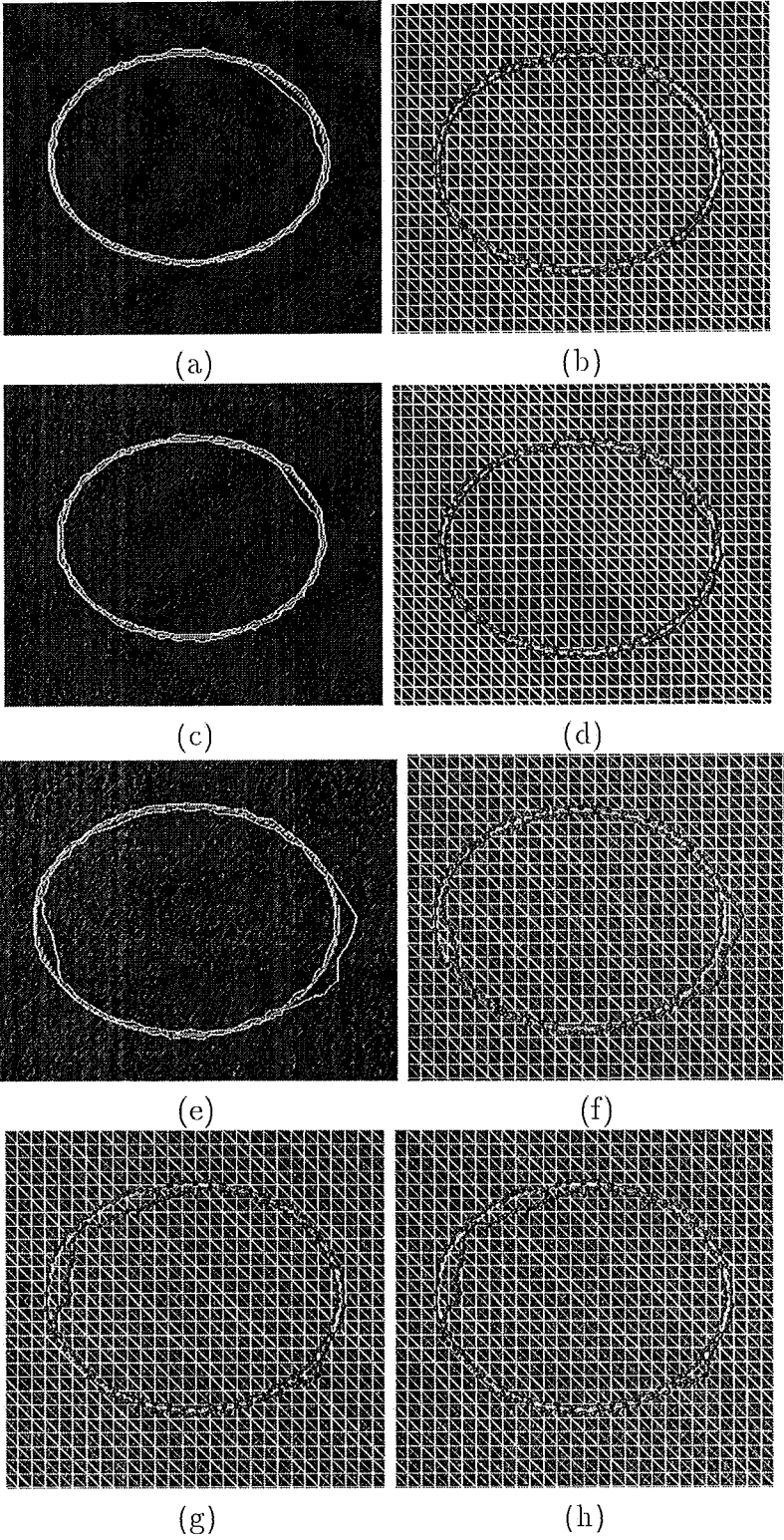


Figura 9.9: Dual-T-Snakes: Solução e solução projetada na malha 5×5 .

Imagem 9.9.a	loop 1	loop 2
Snake 0	0	40
Snake 1	7	90

(a)

Imagem 9.9.c	loop 1	loop 2
Snake 0	0	41
Snake 1	6	83

(b)

Imagem 9.9.e	loop 1	loop 2
Snake 0	4	34
Snake 1	0	112

(c)

Imagem 9.9.g	loop 1	loop 2
Snake 0	0	43
Snake 1	6	143

(d)

Tabela 9.4: (a)-(d) Performance para Dual-T-Snakes inicializado de acordo com Figura 9.8.

acima a snake interna chega na borda com $(6 + 143)$ interações (Tabela 9.4.d), neste caso temos $(17 + 151)$ e ainda estamos longe da borda (Figura 9.10). Isto acontece porque a força normal, que tem a função de *puxar* as snakes para a borda, muda de sentido a todo momento (devido ao ruído), criando instabilidade e oscilações no movimento.

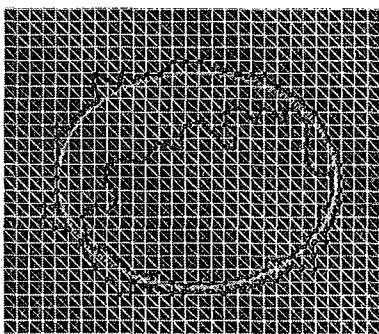


Figura 9.10: Instante $t = 168$ do Dual-T-Snakes para força normal dada por (5.54).

Pela Tabela 9.3 observamos apenas mudança do limiar para intensidade da imagem usado, o que se justifica pela mudança da média do ruído Gaussiano. Isto é um indicativo de *estabilidade* dos parâmetros com relação à mudança da intensidade de ruído da imagem, o que é desejável, como já ressaltado acima.

A Figura 9.11 mostra as soluções obtidas pelo Viterbi para os exemplos desta

seção.

Assim como na seção anterior, os parâmetros do Viterbi são dados por: $\alpha = 1200.0$, $\beta = 0.5$ and $\lambda = 140.0$. Novamente temos estabilidade dos parâmetros com relação ao ruído, o que é um bom resultado.

9.1.3 Background com Ruído e Artefatos

Neste caso, vamos analisar o comportamento do método quando aumentamos o ruído da imagem e introduzimos artefatos na mesma. A Figura 9.12.a mostra uma elipse imersa em um meio com ruído gaussiano com média 50.0 e variância 25.0. A imagem e a malha têm resoluções (256×256) e (3×3), respectivamente. A inicialização do Dual-T-Snakes está mostrada na Figura 9.12.b.

Os valores dos parâmetros são os seguintes: $k = 200.0$, $\lambda = 50.1$, $b = 50.0$, Ponto de Congelamento=5, $T = 50.0$.

Novamente, vemos a característica do Dual-T-Snakes de sair de mínimos locais. A snake externa ficou congelada na posição mostrada na Figura 9.13.c. Em seguida, a snake interna foi processada até estabilizar na posição indicada na Figura 9.13.e. Repete-se este processo até obter-se a solução final da Figura 9.13.f. O resultado final é bastante satisfatório.

Se mudamos a resolução da malha para (5×5) obtemos o resultado mostrado na Figura 9.14, partindo da mesma inicialização dada na Figura 9.12.

Aproveitaremos este exemplo para discutir um aspecto importante. A resolução da malha está diretamente relacionada ao número de snaxels e conseqüentemente com a complexidade do método. Assim, em imagens com alta resolução pode ser útil aplicar procedimentos multigrid para diminuir o custo computacional. Contudo, para tal método ser prático é necessário procedimentos sistemáticos para atualizar os valores dos parâmetros quando passamos de uma resolução para outra.

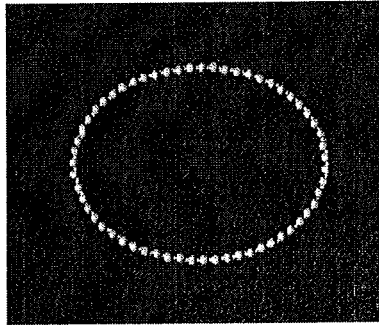
Na seção 9.1.1 nós reduzimos o parâmetro de suavização b à metade quando diminuimos a resolução. Neste caso, nós optamos por manter os mesmos valores dos parâmetros. O resultado obtido (Figura 9.14.f) é satisfatório.

A Figura 9.15 mostra as soluções finais obtidas pelo Viterbi. Os parâmetros são os mesmos da seção anterior.

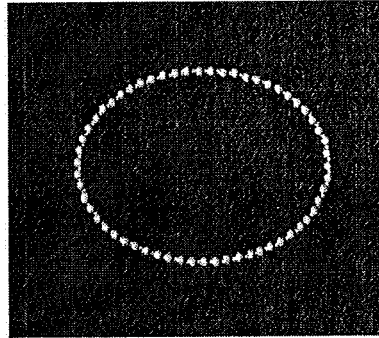
9.1.4 Mudanças Topológicas

Nos exemplos que seguem enfocamos as habilidades topológicas do Dual-T-Snakes. Iniciamos com uma imagem com três objetos e um fundo quase uniforme e acrescentamos ruído a esta imagem.

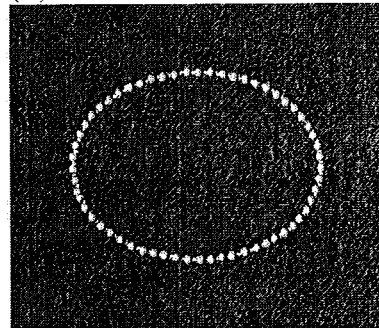
Assim, tomamos a imagem da Figura 9.16.a e usamos um detetor de bordas usual. O resultado obtido está mostrado na Figura 9.16.b e tem um background com pouco ruído.



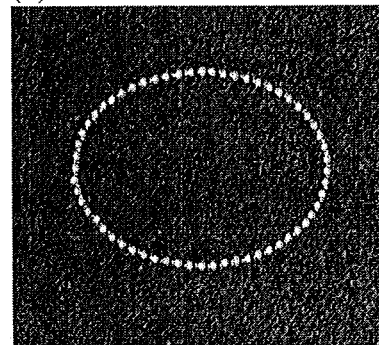
(a)



(b)

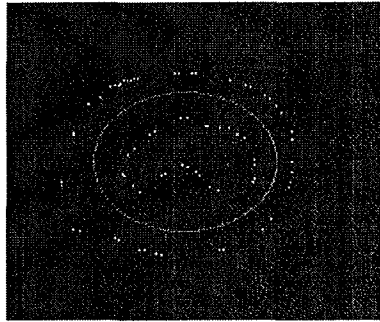


(c)

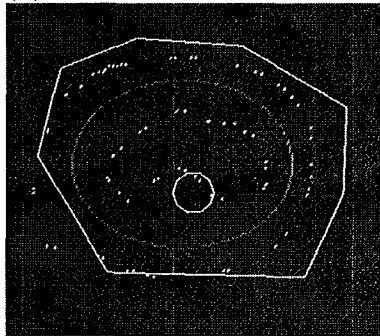


(d)

Figura 9.11: (a)-(d) Soluções do Viterbi para os exemplos das Figuras 9.9.



(a)



(b)

Figura 9.12: (a) Elipse em meio com Ruído e artefatos. (b) Inicialização do Dual-T-Snakes

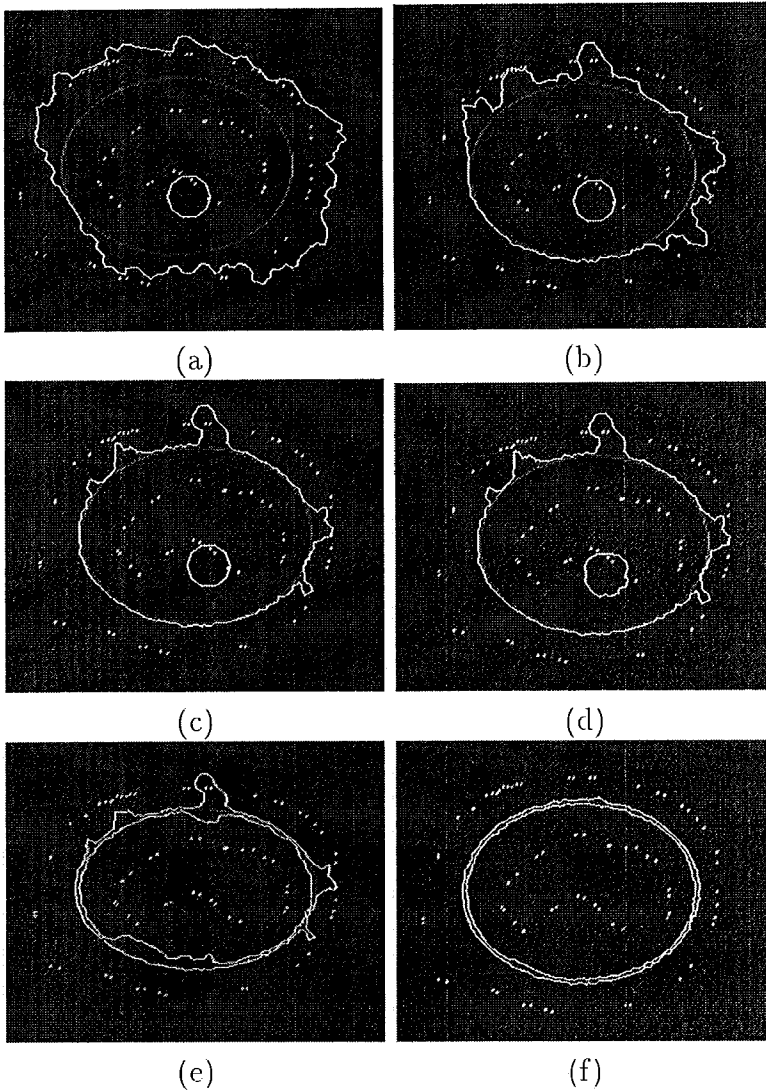


Figura 9.13: Evolução do Dual-T-Snakes para a Figura 9.12.b: (a)Interação 10. (b)Posição intermediária ($t = 30$) (c)Mínimo local da snake externa ($t = 41$). (d)Início do movimento da snake interna ($t = 42$). (e)Mínimo local da interna ($t = 152$). (f)Solução final.

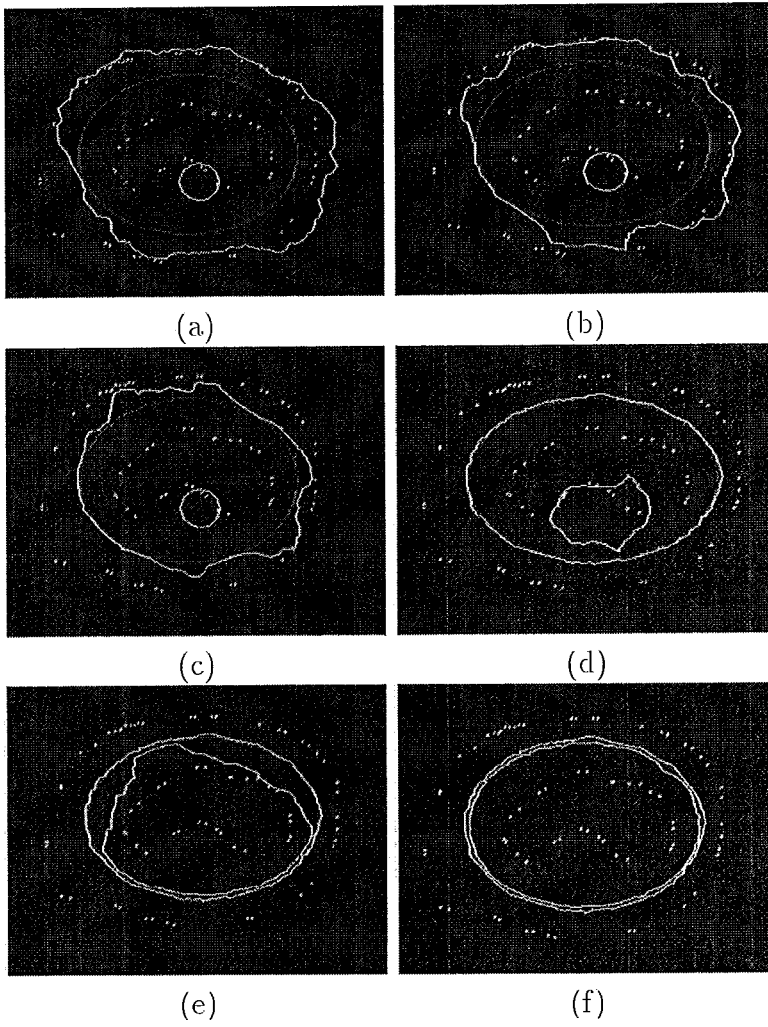


Figura 9.14: Dual-T-Snakes em malha 5×5 : (a)Interação 12. (b)Posição intermediária ($t = 21$) (c)Posição em $t = 31$. (d)Mínimo local da snake interna ($t = 90$). (e)Mínimo local da externa ($t = 191$). (f)Solução final.

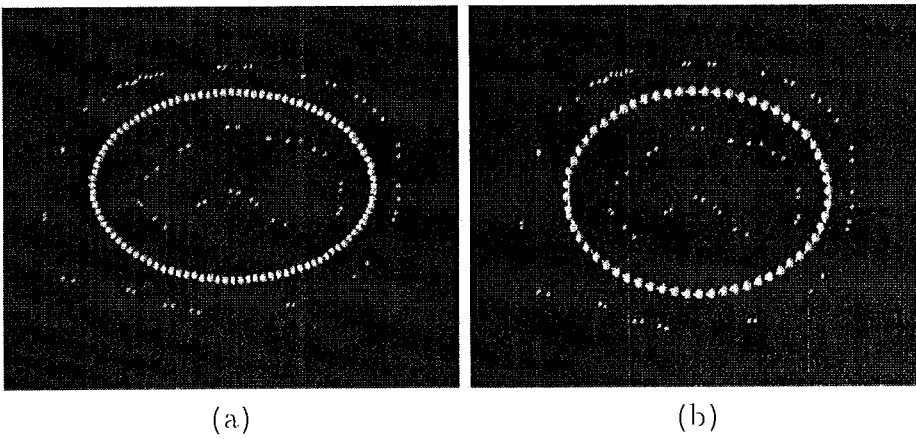
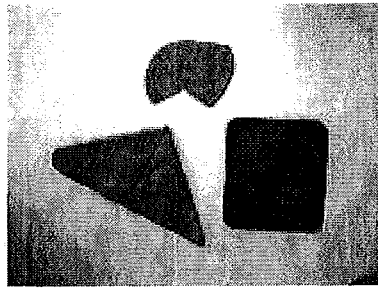
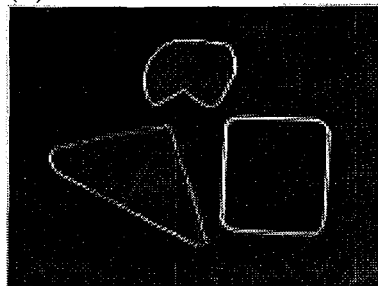


Figura 9.15: (a)-(b)Solução do Viterbi para as Figuras 9.13.f e 9.14.f, respectivamente.



(a)



(b)

Figura 9.16: (a) Figura contendo três objetos de interesse. (b) Detetor de bordas aplicado.

Em seguida, acrescentamos a esta imagem um ruído gaussiano com média 40.0 e variância 0.2. A Figura 9.17.a mostra a imagem obtida. As Figuras 9.17.b,c mostram a malha (3×3) e as T-Snakes usadas na inicialização do Dual-T-Snakes.

A Figura 9.18 mostra algumas situações intermediárias. A solução do Dual-T-Snakes está na Figura 9.19.

Novamente vemos alguns defeitos na solução final, os quais são decorrentes do critério de para usado, como já comentado na seção 9.1.2.

Os valores dos parâmetros usados são os seguintes: $k = 200.0$, $\lambda = 50.1$, $b = 50.0$, $T = 44.0$. O número de interações da snake externa, desde o início passando pelos splits e chegando à configuração final da Figura 9.19 foi 49. As snakes internas somaram 78 interações para chegar a posição final.

A Figura 9.20 mostra o mesmo exemplo, mas agora diminuindo a resolução da malha para 5×5 .

A Figura 9.21 mostra a imagem da Figura 9.16.b, mas agora acrescentado um ruído gaussiano com média 40.0 e variância 1.0. Os parâmetros possuem os mesmos valores listados acima. Observamos que o resultado para o objeto superior ficou ruim nas regiões com curvatura mais elevada. Para os outros objetos as bordas ficam bem limitadas pelas snakes, que é o resultado desejado nesta fase.

Este exemplo mostra uma dificuldade extra que surge ao tratarmos imagens

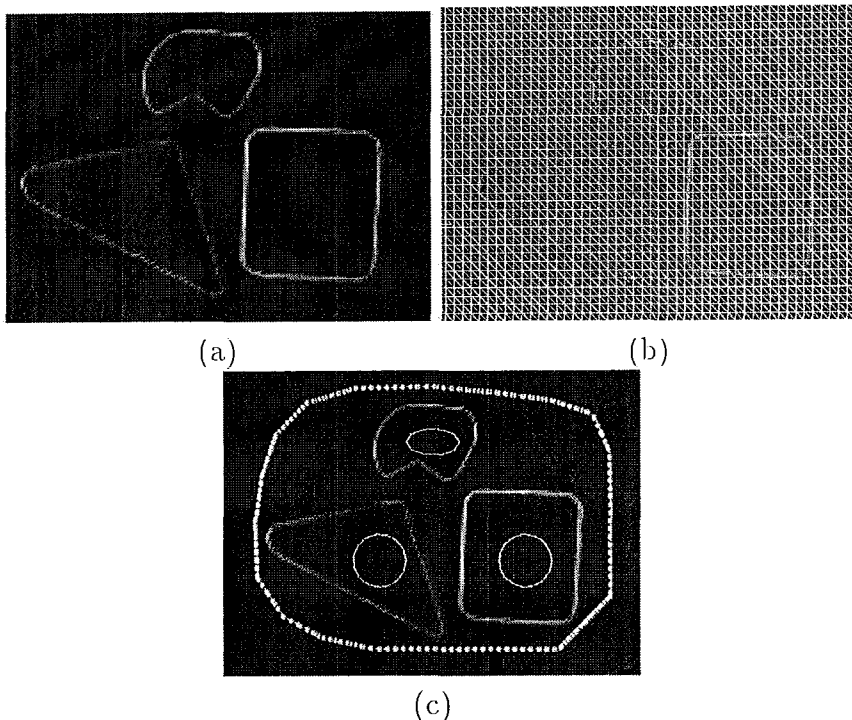


Figura 9.17: (a) Imagem a ser processada. (b) Malha usada (3×3). (c) Inicialização do Dual-T-Snakes: uma snake externa e uma snake interna a cada objeto.

com mais de um objeto de interesse: parâmetros ótimos para um objeto podem ser ineficientes para outro.

Isto decorre, por exemplo, de diferenças nas curvaturas dos objetos, ou padrões de intensidade variados nas bordas. Uma forma de corrigir este problema é aumentando a intensidade da força de imagem. Este procedimento faz com que a snake encontre mais resistência ao seu movimento nas proximidades da borda. Contudo, esta idéia traz a dificuldade de tornar a snake mais sensível a mínimos locais, o que pode ser tratado pelo Dual-T-Snakes.

Assim, fazendo $\lambda = 100.1$ obtemos um resultado satisfatório, o que pode ser verificado pela Figura 9.23.b.

Nas Figuras 9.22-9.23, mostramos duas seqüências de resultados obtidos para as imagens indicadas na Tabela 9.5, usando inicialmente malhas com resolução 3×3 e em seguida 5×5 .

Os valores dos parâmetros usados estão nas Tabelas 9.8-9.9. A performance do método pode ser avaliada pelas Tabelas 9.6 e 9.7 onde fornecemos o número de interações necessário para obter os resultados apresentados (nestas tabelas, o número de interações das snakes internas deve ser entendido como o a soma do número de interações de todas as snakes internas).

Neste momento, é interessante fazermos uma comparação entre estas seqüências

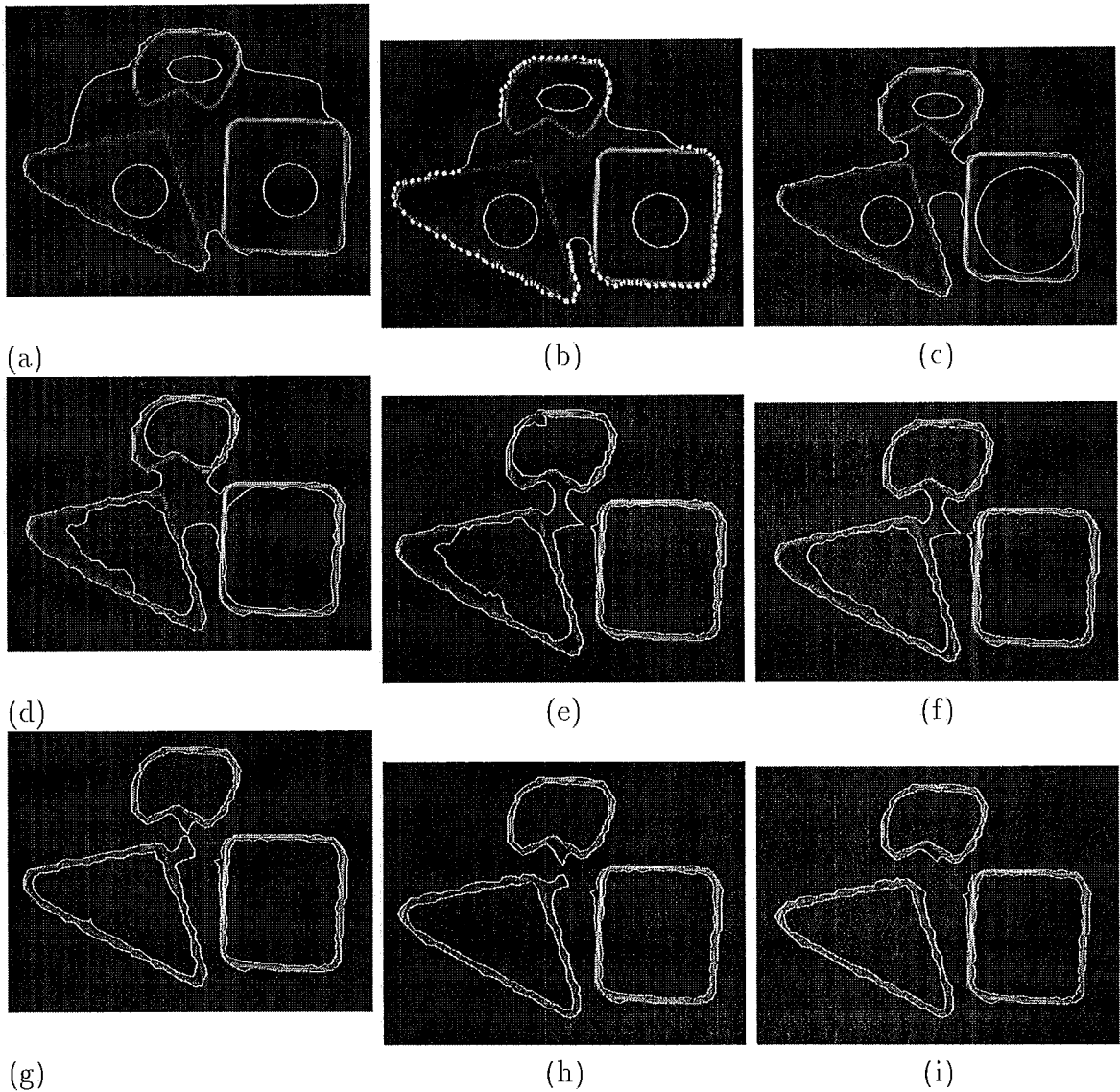


Figura 9.18: Evolução do Dual-T-Snakes para a Figura 9.17: (a) Interação 15. (b) Interação 25 (snake externa congelada). (c) Interação 35 (snake interna ao retângulo congelada). (d) Mínimo local da snake interior ao objeto triangular. (e) Snake externa sofre split. (f) Dual Snakes formados após split evoluem. (g) Novo split da snake externa. (h) Temos agora três dual snakes independentes. (i) Instantes finais do método.

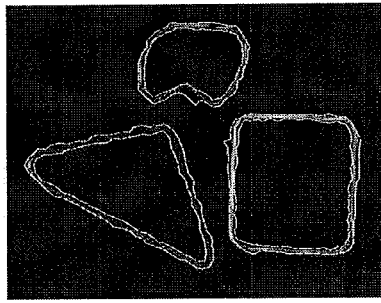


Figura 9.19: Solução do Dual-T-Snakes para a evolução da Figura 9.18.

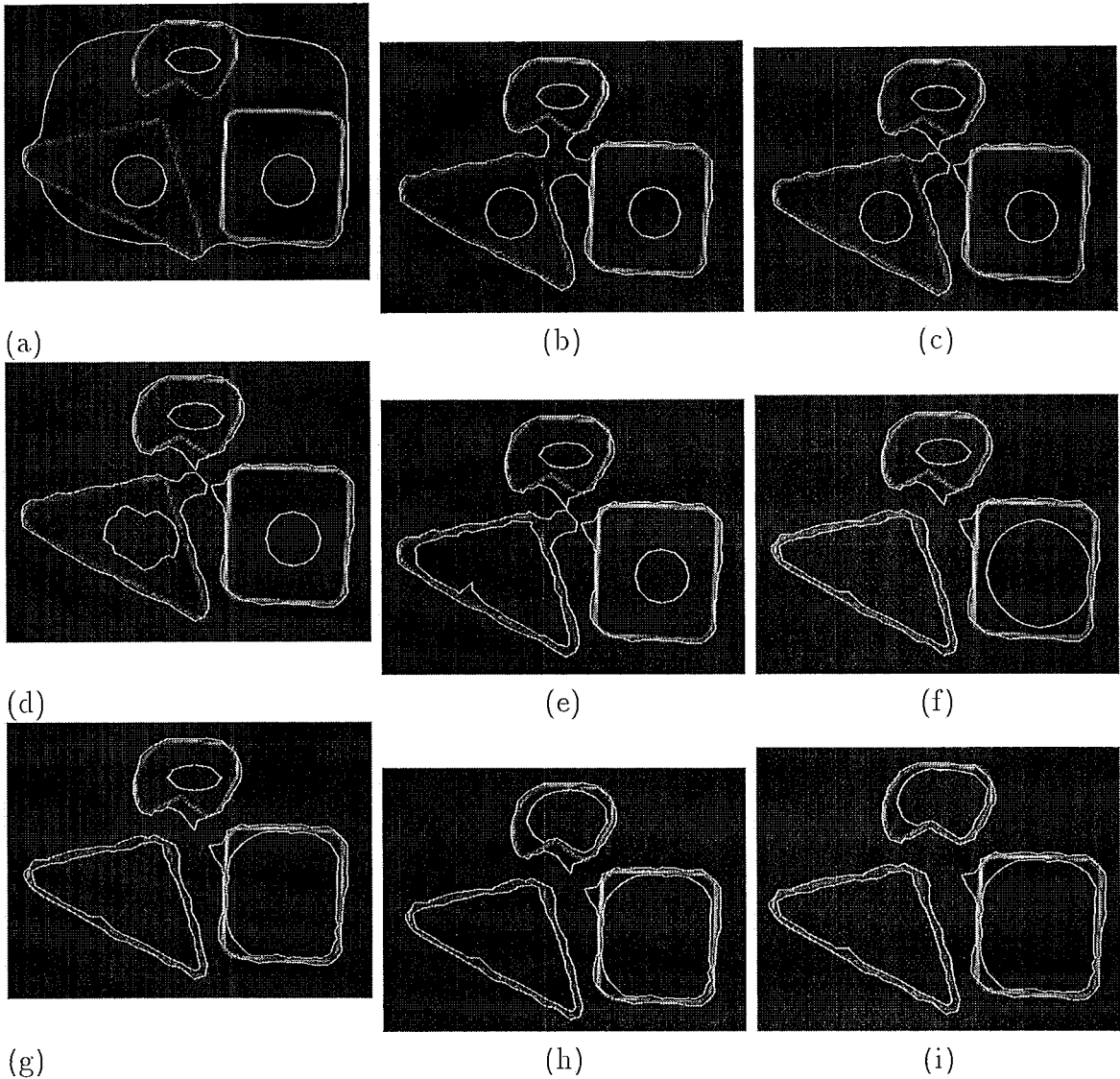


Figura 9.20: Evolução do Dual-T-Snakes para Figura 9.18.a usando malha (5×5) : (a) $t = 10$. (b) Pouco antes do split (interação 58). (c) Split na interação 60. (d) Novo split (interação 65). (e) Interação 98 (mínimo local da snake interna ao objeto triangular). (f) Interação 120. (g) Snake interior ao retângulo congelada. (h) Snake do objeto superior sendo processada. (i) Solução final.

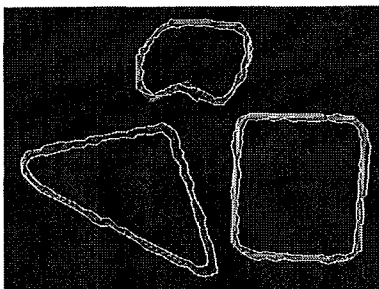


Figura 9.21: Dual-T-Snakes: Resultado insatisfatório para objeto superior.

Imagem	Média , Variância
9.22.a	$\mu = 40.0$, $\sigma = 1.0$
9.22.c	$\mu = 40.0$, $\sigma = 4.0$
9.22.e	$\mu = 40.0$, $\sigma = 20.0$
9.22.g	$\mu = 40.0$, $\sigma = 30.0$

Tabela 9.5: Parâmetros usados nos testes para o Dual-T-Snakes cujas soluções estão nas Figuras 9.22-9.23.

Imagem 9.22.a	Num. Interações
Snake Externa	67
Snakes Internas	104

(a)

Imagem 9.22.c	Num. Interações
Snake Externa	67
Snakes Internas	112

(b)

Imagem 9.22.e	Num. Interações
Snake Externa	69
Snakes Internas	102

(c)

Imagem 9.22.g	Num. Interações
Snake Externa	73
Snakes Internas	118

(d)

Tabela 9.6: (a)-(d)Performance para Dual-T-Snakes correspondente aos resultados mostrados na Figura 9.22.

de resultados.

Quanto aos valores dos parâmetros, pela Tabela 9.8 observamos um mesmo conjunto de valores para os casos com malha 3×3 . Ao passarmos para a malha 5×5 observamos apenas uma mudança no parâmetro de suavização b . Temos assim uma

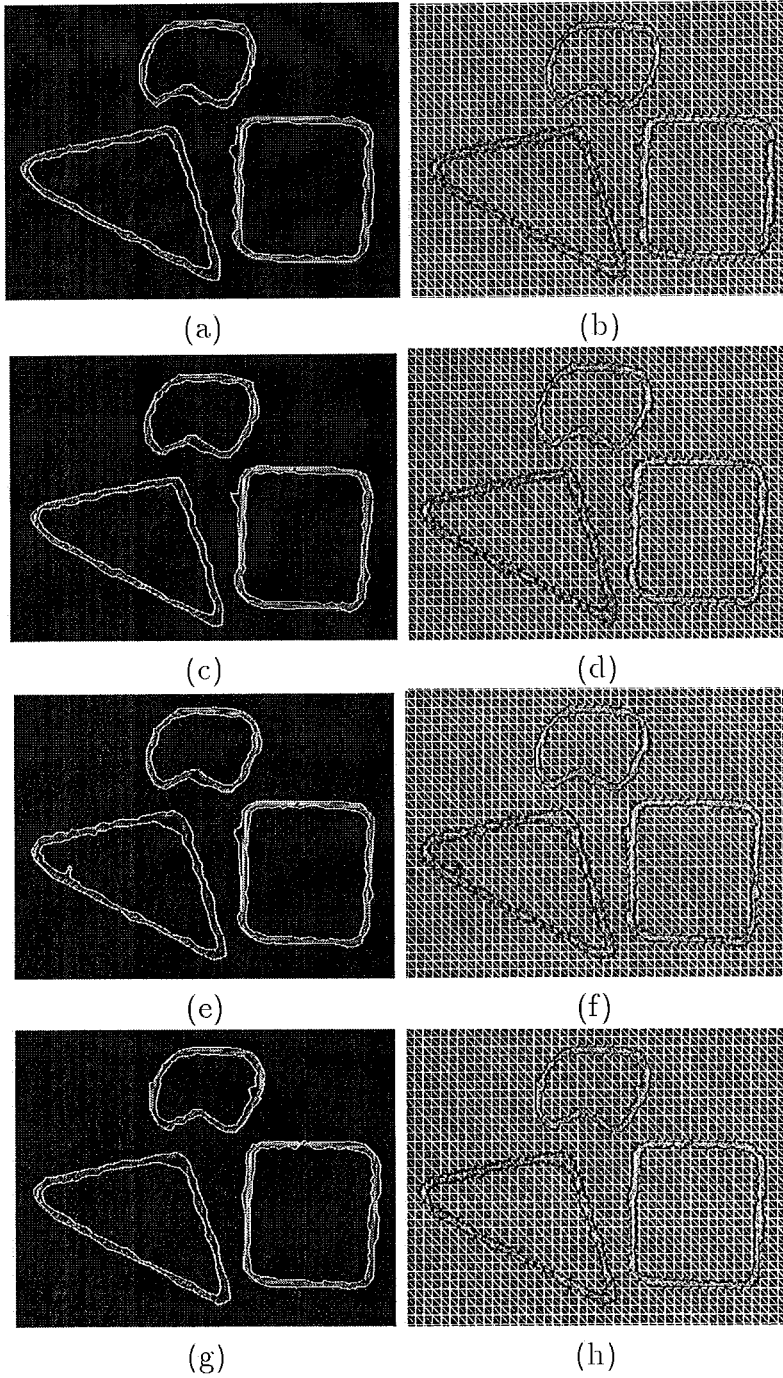


Figura 9.22: Dual-T-Snakes: Solução e sua projeção na malha 3×3 .

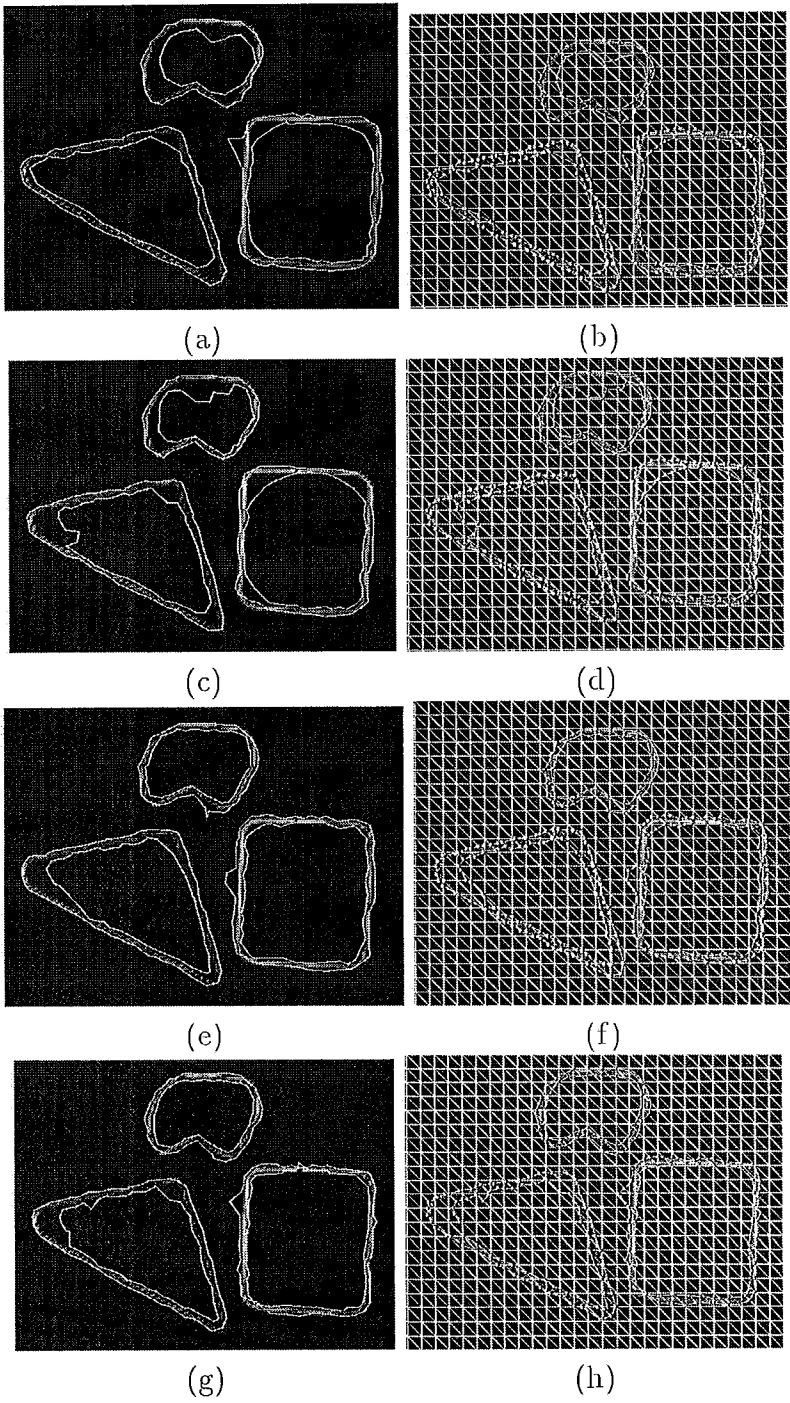


Figura 9.23: Dual-T-Snakes: Solução e sua projeção na malha 5×5 .

Imagem 9.23.a	Num. Interações
Snake Externa	71
Snakes Internas	79

(a)

Imagem 9.23.c	Num. Interações
Snake Externa	79
Snakes Internas	103

(b)

Imagem 9.23.e	Num. Interações
Snake Externa	83
Snakes Internas	84

(c)

Imagem 9.23.g	Num. Interações
Snake Externa	74
Snakes Internas	125

(d)

Tabela 9.7: (a)-(d) Performance para Dual-T-Snakes correspondente aos resultados mostrados na Figura 9.23.

Imagem	k	λ	b	T	P. Congelamento
Figura 9.22.a	200.0	100.1	50.0	44.0	5
Figura 9.22.c	200.0	100.1	50.0	44.0	5
Figura 9.22.e	200.0	100.1	50.0	44.0	5
Figura 9.22.g	200.0	100.1	50.0	44.0	5

Tabela 9.8: Parâmetros usados nos testes do Dual-T-Snakes correspondentes aos resultados mostrados na Figura 9.22. A malha usada é 3×3

Imagem	k	λ	b	T	P. Congelamento
Figura 9.23.a	200.0	100.1	10.0	44.0	10
Figura 9.23.c	200.0	100.1	10.0	44.0	10
Figura 9.23.e	200.0	100.1	20.0	44.0	10
Figura 9.23.g	200.0	100.1	10.0	44.0	10

Tabela 9.9: Parâmetros usados nos testes do Dual-T-Snakes correspondentes aos resultados mostrados na Figura 9.23. A malha usada é 5×5

boa estabilidade dos parâmetros nos dois conjuntos de teste.

Quanto ao ponto de congelamento, os valores escolhidos estão vinculados à dimensão das células da triangulação e o limiar T é escolhido próximo à média do ruído gaussiano usado.

A definição para a força normal é aquela da expressão 7.3. Quando uma T-Snake

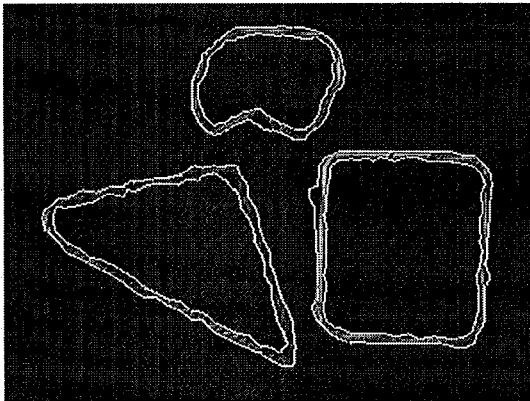
está congelada, a estratégia usada para retirá-la da posição de equilíbrio é variar o limiar T para a imagem.

De maneira geral, a qualidade do resultado é melhor quando usamos a resolução 3×3 para a triangulação. Isto decorre do critério de parada usado o qual interrompe o movimento das T-Snakes em posições mais distantes quando a malha é mais grossa.

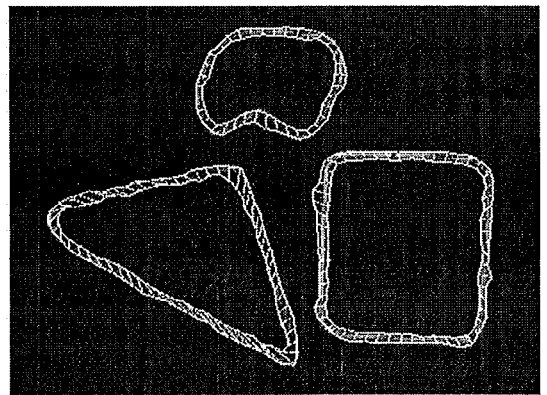
Um outro aspecto a ser destacado é com relação ao resultado da Figura 9.22.e. No objeto triangular desta figura vemos a snake interna com um detalhe com alta curvatura. A força de suavização nesta região é muito alta e assim defeitos deste tipo aparentemente não deveriam ocorrer.

O motivo deste comportamento vem da política adotada para decidir quando uma T-Snake está ou não congelada. Nestes casos, adotamos a postura de congelar uma T-Snake quando o número de snaxes com temperatura abaixo do ponto de congelamento for menor que 10% do número total de snaxels. Este procedimento tem a vantagem de evitar que uma T-Snake fique muito tempo no segundo loop em função de um pequeno número de snaxels ainda livres. Contudo, pode causar a interrupção do movimento sem que a força interna tenha exercido seu papel de suavização.

Uma maneira simples de corrigir este defeito seria, uma vez terminada a execução do Dual-T-Snakes usando o critério acima, deixar que o algoritmo evolua normalmente até que o ponto de congelamento seja atingido por *todos* os snaxels ou as snakes estejam próximas. O resultado da Figura 9.22.e é então suavizado, obtendo-se aquele da Figura 9.24.a. A Figura 9.24.b mostra o espaço de busca correspondente. Em geral, nos espaços de busca que usamos, nós discretizamos as snakes com um passo da ordem de 3 unidades e tomamos ao longo de cada segmento de reta um número de pontos da ordem do dobro da medida da aresta das células da triangulação (6, no caso da Figura 9.24.b).



(a)



(b)

Figura 9.24: (a) Solução do Dual-T-Snake suavizada. (b) Espaço de busca correspondente.

Outro aspecto são as intersecções entre as T-Snakes observadas nas Figuras 9.22.g e 9.23.g Isto provém da subjetividade das bordas decorrentes do limiar usado e principalmente da discrepância entre o parâmetro da força normal e das demais forças do modelo, o que em geral prejudica a suavidade do resultado final do Dual-T-Snakes. Contudo, para bordas do tipo degrau (Figura 2.1) teremos geralmente intersecções entre as snakes ou mesmo sobreposição (seção 9.2.4), uma vez que nestes casos a borda procurada não pertence a uma faixa da imagem com uma intensidade específica.

Do ponto de vista do espaço de busca, dois aspectos são fundamentais: não pode haver intersecção entre os segmentos que interligam as snakes e a distância entre dois pontos vizinhos em um mesmo segmento deve ser da ordem da distância entre dois pixels vizinhos da imagem. O primeiro aspecto é importante para que a solução obtida seja uma curva simples e o segundo é necessário para evitar custo computacional desnecessário.

Considerando estes fatos, a primeira possibilidade seria montar o espaço de busca com o resultado do Dual-T-Snakes tomando apenas o cuidado de evitar intersecção entre segmentos vizinhos. Outra possibilidade seria usar offsets para definir um espaço de busca mais eficiente. Nos casos, desta seção usamos a primeira opção. A seguir, veremos exemplos onde a segunda opção é mais eficiente.

A Figura 9.25 mostra as soluções para o Viterbi correspondentes à Figura 9.22. O espaço de busca usado tem o aspecto daquele da Figura 9.24 sendo definido com os parâmetros usados naquele caso.

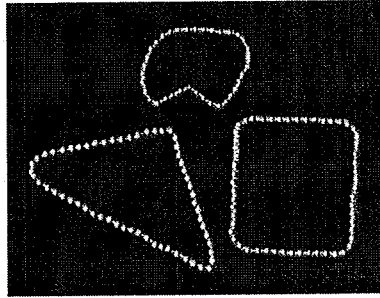
Os resultados são em geral satisfatórios, com exceção da parte inferior da região triangular. Esta região tem curvatura mais elevada e por isto exige um espaço de busca mais refinado para ser melhor coberta pelo método.

Para os resultados da Figura 9.23 a aplicação direta do Viterbi se torna mais complicada. Nestes casos, a falta de suavidade do resultado final e a distância entre as snakes torna o espaço de busca pouco eficiente. Assim, a melhor opção para este caso seria usar o resultado obtido para inicializar o Dual-T-Snakes em um malha mais fina (3×3 , por exemplo) e então obter um resultado eficiente para definir o espaço de busca. Na seção 9.2.4 a seguir veremos um exemplo de uma metodologia multigrig aplicada conjuntamente com o Dual-T-Snakes.

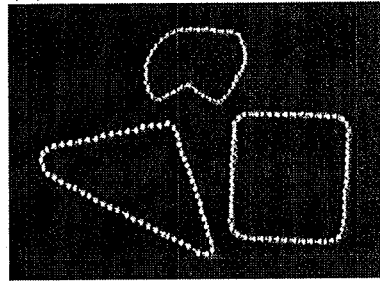
9.1.5 Conexões Fuzzy e Dual-T-Snakes

O exemplo a seguir é interessante para discutirmos a relação entre métodos de segmentação e snakes.

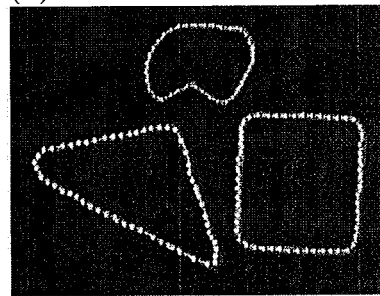
De maneira geral, snakes podem ser usadas conjuntamente com técnicas de tratamento de imagem, gerando métodos unificados de segmentação baseados na extração de contornos dos objetos (Giraldi *et al.*, 2000b; Zhu & Yuille, 1996).



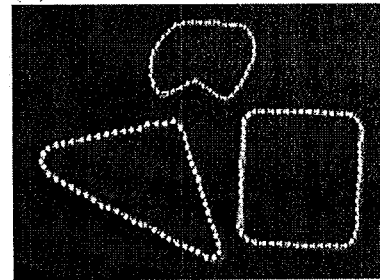
(a)



(b)



(c)



(d)

Figura 9.25: (a)-(d) Soluções do Viterbi para os exemplos das Figuras 9.22.

Por outro lado, o resultado da segmentação pode ser usado para controlar o movimento da snake em direção às bordas desejadas (Jones & Metaxas, June, 1998). É exatamente neste último sentido que usamos o método das conexões fuzzy desenvolvido no Apêndice E juntamente com o Dual-T-Snakes.

Vejam os exemplos. A imagem sintética da Figura 9.26 mostra dois objetos com interior ruidoso imersos em background com ruído exponencial de variância $\sigma = 1000.0$.

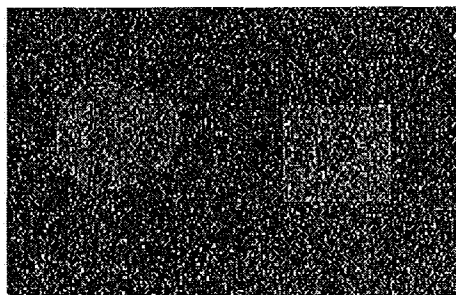


Figura 9.26: Objetos imersos em meio com ruído exponencial.

A aplicação direta do Dual-T-Snakes neste caso traz a dificuldade de definir o sentido da força normal devido a presença do ruído tanto no interior quanto no exterior dos objetos. Uma definição da força normal baseada na expressão 5.54 pode levar a instabilidades no movimento da snake diminuindo a performance e a qualidade do resultado.

Se usarmos a expressão 7.3 diminuimos as oscilações no movimento da snake, mas, por outro lado, a snake pode ficar presa nos diversos mínimos locais da energia da imagem, o que afeta novamente a performance do algoritmo.

Assim, fazer uma pré-segmentação da imagem e usar o resultado obtido para definir a força normal é uma possibilidade interessante.

O método das conexões fuzzy discutido no Apêndice E é uma opção para isto uma vez que este parte de pontos sementes estabelecidos pelo usuário. Se estes pontos pertencem ao interior dos objetos de interesse então podem ser usados para inicializar as snakes internas do Dual-T-Snakes. O resultado do método fuzzy, por sua vez, pode ser usado para definir o sinal da força normal, de acordo com a expressão ??.

A Figura 9.27 mostra o campo σ_0 obtido pelo método das conexões fuzzy multi-semente, segundo o algoritmo da seção E.3.

A Figura 9.28 mostra a solução obtida pelo Dual-T-Snakes. A força normal segue a definição dada pela expressão ?? e os valores dos parâmetros são os seguintes: $k = 200.0$, $\lambda = 50.1$, $b = 50.0$, Ponto de Congelamento=5, $T = 0.5$. A malha usada tem resolução 3×3 .

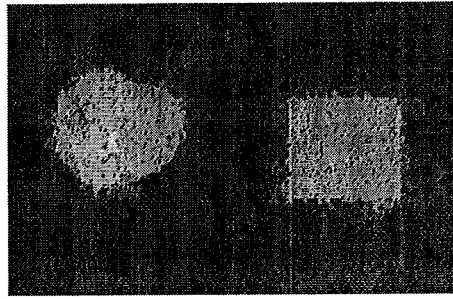


Figura 9.27: Campo σ_0 obtido pelo método das conexões fuzzy para a Figura 9.26.

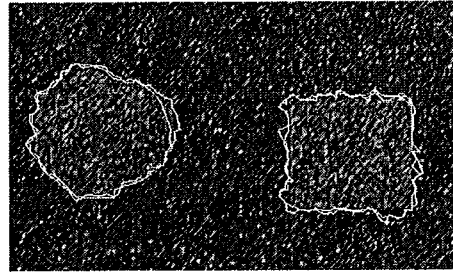
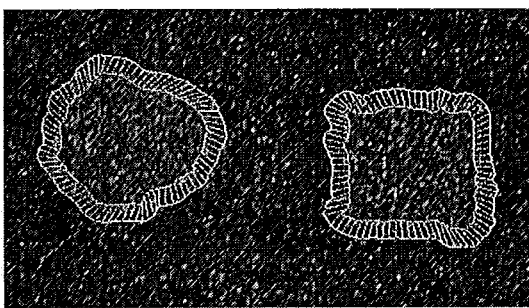
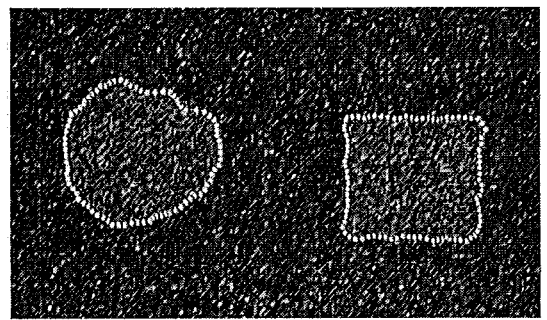


Figura 9.28: Solução do Dual-T-Snakes para normal definida segundo o campo σ_0 mostrado na Figure 9.27.

A proximidade entre as snakes neste caso torna pouco eficiente usar um espaço de busca definido entre as mesmas. Uma possibilidade mais segura é tomar dois offsets de uma das snakes e usar estas curvas para definir o espaço de busca. É exatamente o que fizemos neste caso e o resultado está mostrado na Figura 9.29.a. A solução do Viterbi está na Figura 9.29.b.



(a)



(b)

Figura 9.29: (a)Espaço de busca gerado a partir de off-sets. (b)Solução final.

O resultado obtido é aceitável, observando-se contudo defeitos, particularmente na borda da figura retangular. Estes defeitos têm como origem o alto ruído da imagem, o que dificulta a escolha de parâmetros ótimos para o Viterbi. Apesar disto, este exemplo mostra que para imagens muito ruidosas o método das conexões

fuzzy pode ser uma boa opção para a definição da força normal.

Isto decorre, em particular, pela definição das funções de afinidade dadas pelas expressões (E.15)-(E.17). As funções Gaussianas usadas garantem propriedades de suavização, o que torna o resultado menos sensível a variações bruscas dos gradientes, próprias de imagens com ruídos.

É importante ressaltar que a energia de campo usada no Viterbi é calculada sobre a imagem original e não sobre o resultado do método fuzzy. Os parâmetros usados nesta fase foram os seguintes: $\alpha = 1200.0$, $\beta = 0.5$ and $\lambda = 0.5$.

9.2 Imagens Médicas

Os exemplos desta seção serão constituídos por imagens de ressonância magnética, células, Raio-X e do Projeto do Homem Visível.

Nas imagens obtidas por ressonância magnética da cabeça de um paciente pudemos fazer uso da metodologia de inicialização da seção 8.3 para automatizar o posicionamento da snake externa do Dual-T-Snakes.

Em imagens contendo várias células, mostramos como podemos explorar este mesmo método de inicialização como ponto de partida para uma metodologia de segmentação baseada em extração de bordas dos objetos de interesse (Giraldi *et al.*, 2000b).

Nos demais exemplos de imagens contendo células mostramos a eficiência do Dual-T-Snakes em uma metodologia multigrid, bem como com a incorporação de técnicas para crescimento de regiões baseada na triangulação. Os resultados obtidos indicam que o método tem neste tipo de imagens um grande potencial de aplicação. Aliás, a eficiência de metodologias duais em imagens contendo várias células foi verificada em outras referências da literatura (P. Bamford, 1998).

A presença de regiões com alta curvatura e de artefatos no interior (ou exterior) dos objetos é outra característica destas imagens.

O último caso estudado é constituído por imagens de Raio-X do tórax onde veremos novamente as habilidades topológicas do Dual-T-Snakes.

9.2.1 Ressonância Magnética da Cabeça

Modelos anatomicamente realistas da cabeça de um paciente são necessários em várias aplicações médicas, tais como diagnósticos e modelagem de campos elétricos e magnéticos. No que diz respeito à modelagem de campos, sua aplicação está na área de tomografias funcionais para diagnósticos médicos não invasivos (Schimpf *et al.*, 1998).

O primeiro passo para a obtenção do modelo geométrico é a segmentação da imagem para a extração das bordas desejadas. Em uma imagem de ressonância

como a da Figura 9.30.a nós temos junto à camada óssea o couro cabeludo bem como outros tecidos internos.

Nos restringiremos à extração da camada óssea. Uma possibilidade bem conhecida para isto é o método de *marching cubes* (Lorenson & Cline, 1987). No entanto, este método pode falhar, particularmente devido à presença de ruído no background e bordas subjetivas. O resultado obtido deve passar por um pós-processamento para corrigir defeitos topológicos.

Contudo, uma observação extra permite fazer uso da metodologia do Capítulo 8 para a inicialização do Dual-T-Snakes, evitando possíveis defeitos topológicos comuns quando usamos métodos de busca em processamento de imagens.

Neste caso, o background apresenta a propriedade de escala local que pode ser caracterizada pela resolução dada por 5×5 e um limiar $T = 5.0$. Estes valores foram usados em um conjunto de teste contendo 79 imagens tendo fornecido resultados satisfatórios. A triangulação e Função Característica do Objeto (expressão (8.3)) para a fatia de número 60 estão representadas na Figura 9.30.b.

O resultado do método de inicialização é exatamente a curva externa da Figura 9.31.e. A curva interna é apenas um offset desta curva, tomado na direção do centróide da região limitada pela curva externa (estamos fazendo uso da convexidade da região interna à caixa craniana). O resultado do Dual-T-Snakes para esta fatia (de número 60) está na Figura 9.31.f.

As Figuras 9.31.a-d mostram os resultados da inicialização e do Dual-T-Snakes para as fatias de número 15 e 55. Os parâmetros do Dual-T-Snakes têm os seguintes valores: $k = 1.5.0$, $\lambda = 1.4$, $b = 2.0$.

Os bons resultados obtidos nestas figuras para o Dual-T-Snakes nem sempre se repetem durante este conjunto de dados. O limiar usado para caracterizar a camada óssea ($T = 70.0$) pode variar ao longo das fatias, o que dificulta a extração da caixa craniana por completo sem a intervenção de um especialista.

Com relação ao Viterbi, devemos também fazer considerações especiais. A Figura 9.30.a é um exemplo de uma fatia onde a camada óssea é demasiado larga. Antes de aplicar o método é preciso a opinião de um especialista para que se possa definir uma curva que caracterize eficientemente esta camada para algum tipo de aplicação.

Resultado do Viterbi para a fatia de número 15 está mostrado na Figura 9.32. Neste caso, a camada óssea não é muito grossa e assim sua extração é menos subjetiva.

9.2.2 Célula do Sangue

Na seção 9.1 testamos o Dual-T-Snakes em imagens onde as bordas desejadas correspondem a faixas no plano, imersas em um background com ruído. Nesta seção, veremos como podemos criar esta situação a partir de uma imagem real.

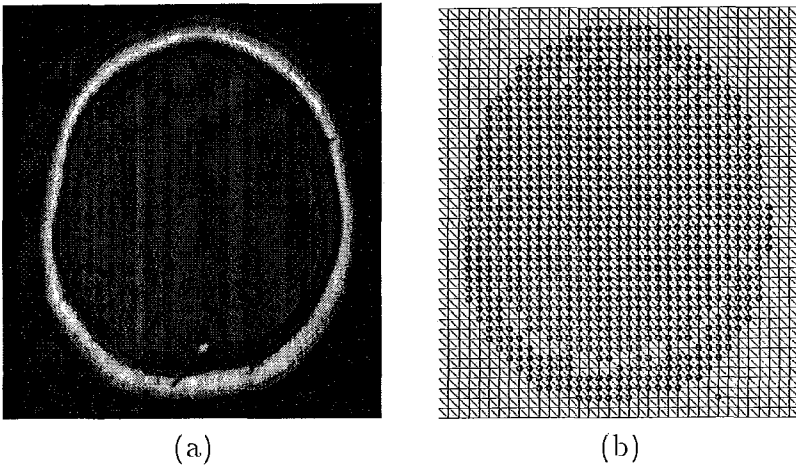


Figura 9.30: (a) Imagem original. (b) Triangulação com resolução 5×5 e Função Característica do Objeto.

A Figura 9.33 mostra a imagem de uma célula do sangue com resolução 312×270

Uma forma simples de criarmos a situação desejada é usando um filtro passa baixa e em seguida um passa-alta. Este procedimento fornece arestas espalhadas, ideais para a aplicação do Dual-T-Snakes. A Figura 9.34 mostra o resultado de uma filtragem deste tipo aplicada sobre a imagem da Figura 9.33

Como esperado, o resultado obtido não está livre de ruídos ou mesmo de artefatos no background. Contudo, a borda procurada está contida na faixa em destaque. No momento em que o Dual-T-Snakes extrair esta faixa, o Viterbi fornecerá o resultado final.

A inicialização e solução final do Dual-T-Snakes estão nas Figura 9.35.a,b, respectivamente. Os parâmetros usados nesta etapa foram os seguintes: $k = 15.0$, $\lambda = 14.0$, $b = 0.2$, $T = 16.0$.

A força normal é dada pela expressão 7.3 uma vez que temos certo nível de ruído no background.

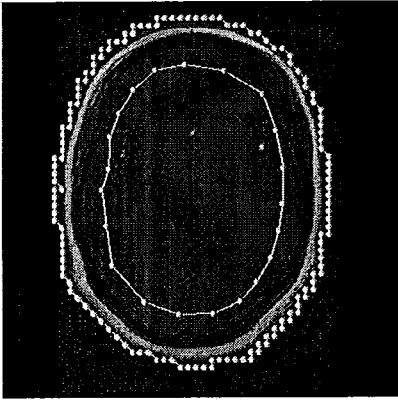
A Figura 9.36 mostra solução do Viterbi para os parâmetros com os seguintes valores: $\alpha = 1.0$, $\beta = 10.0$ and $\lambda = 50.0$. O resultado é bastante satisfatório.

9.2.3 Imagens com Várias Células

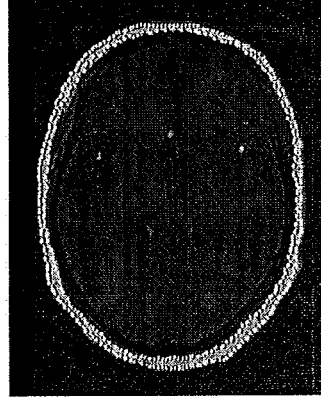
Imagens contendo várias células podem constituir uma aplicação interessante para o método de inicialização do Capítulo 8.

O ponto inicial é a definição da propriedade de escala local. Seja então a imagem da Figura 9.37 a qual mostra um conjunto de células da coluna cervical. Neste caso, uma simples inspeção mostra que a propriedade de escala local pode ser caracterizada por uma malha de resolução 5×5 e por um limiar dado por $T = 150.0$

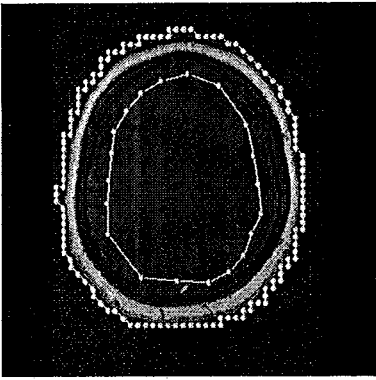
Outro aspecto, é a escala externa de separação entre células . Em algumas



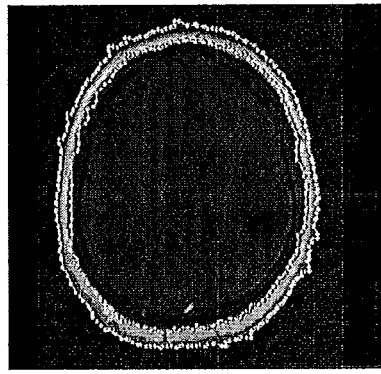
(a)



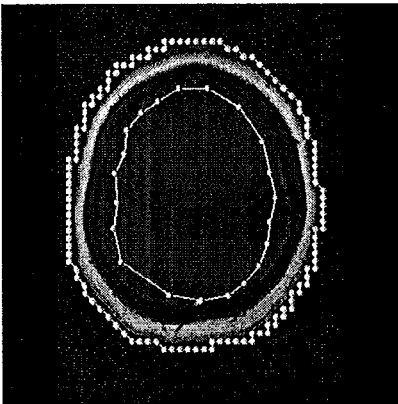
(b)



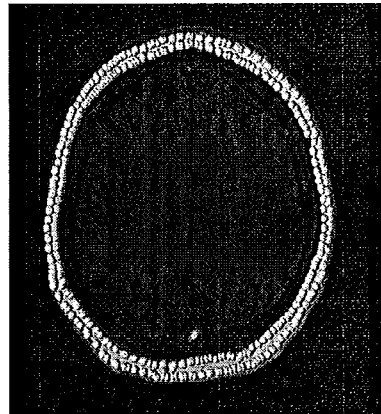
(c)



(d)



(e)



(f)

Figura 9.31: Inicialização e solução do Dual-T-Snakes para fatias 15, 55 e 60.

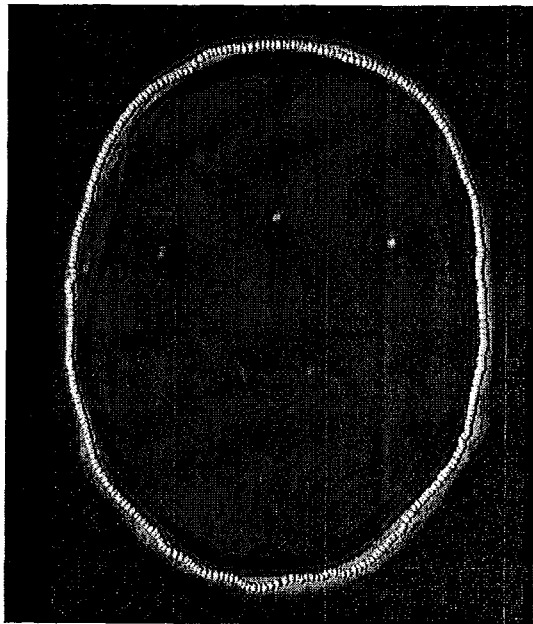


Figura 9.32: Solução obtida por programação dinâmica para a fatia de número 15.

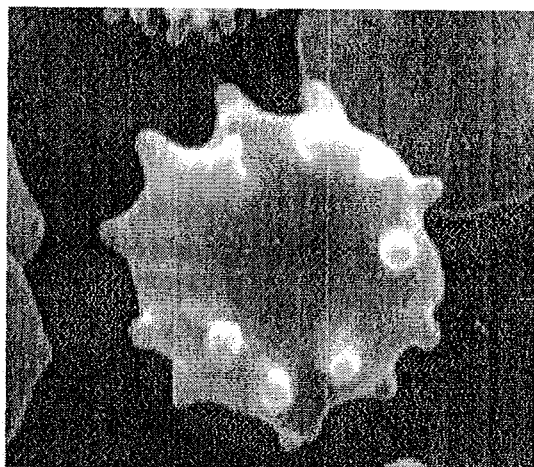


Figura 9.33: Células do sangue.

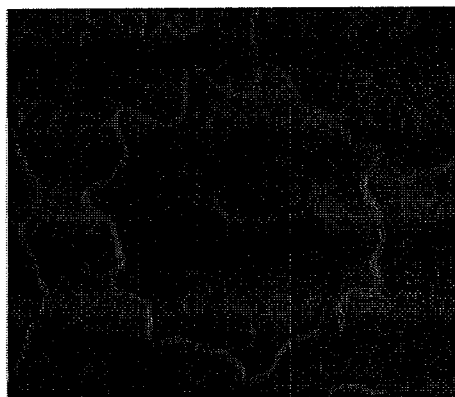
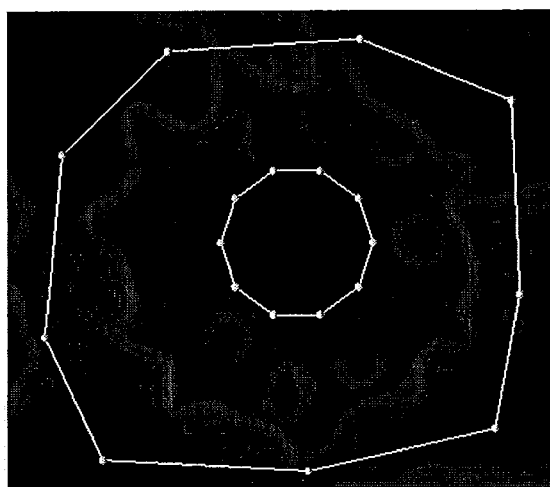
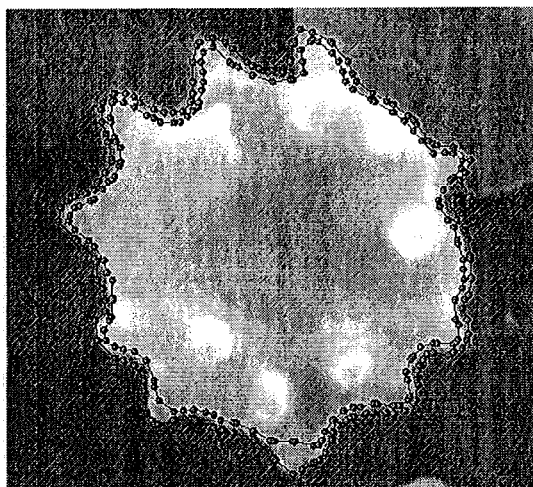


Figura 9.34: Imagem da Figura 9.33 filtrada.



(a)



(b)

Figura 9.35: (a) Inicialização do Dual-T-Snakes. (b) Solução final do Dual-T-Snakes.

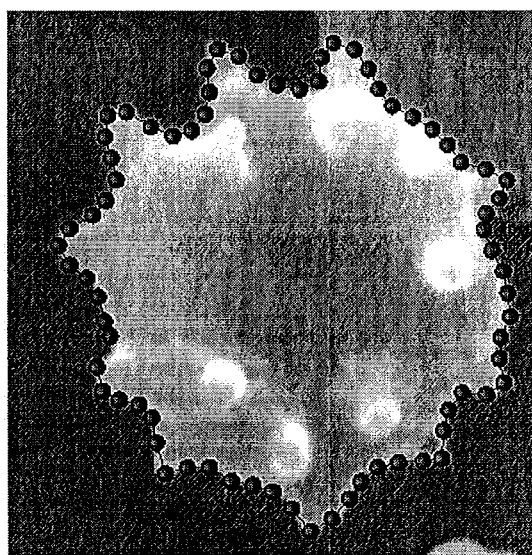


Figura 9.36: Solução final do Viterbi correspondente à Figura 9.33.

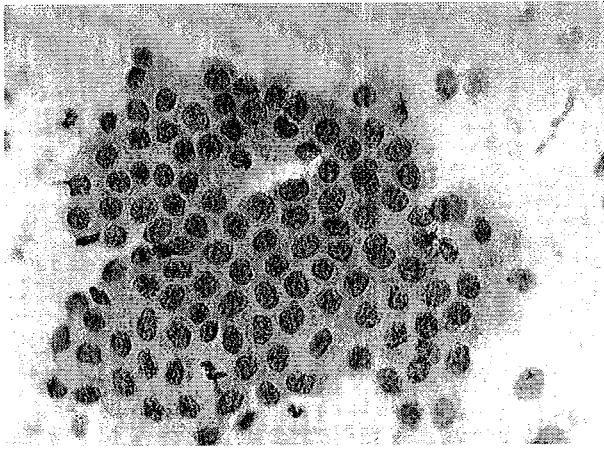


Figura 9.37: Células da coluna cervical.

regiões, esta escala é mais fina que a interna (*Problema de Duas Escalas*). Assim, como já ressaltado na seção 8.3, o resultado dos passos (1)-(4) pode conter regiões com mais de um objeto envolvidas por uma única curva, como pode ser verificado na Figura 9.38.

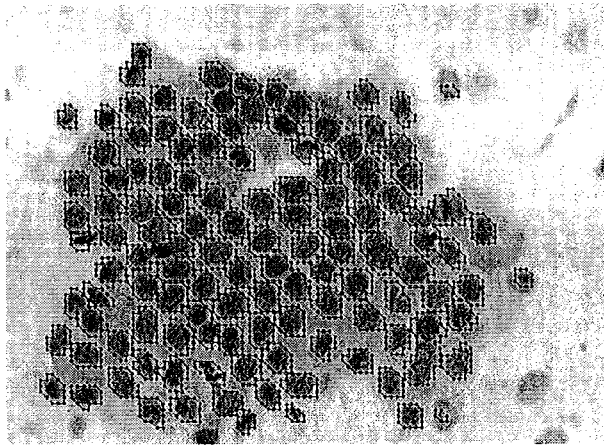


Figura 9.38: Aproximações lineares obtidas para malha 55 e limiar $T = 150.0$ aplicado sobre a imagem da Figura 9.37, sem usar limites para área.

Se usarmos um limite superior conveniente para a área, conseguiremos segmentar algumas destas regiões quando aumentamos a resolução usada. Assim, seguindo os passos (1)-(5) da seção 8.3, obtemos o resultado mostrado na Figura 9.39.

É importante ressaltar que o objetivo do passo (5) é principalmente suavizar o resultado obtido nos passos (1)-(4). Sendo assim, pode-se estabelecer um número máximo de interações e interromper o movimento da T-Snake quando este número é atingido. Um bom valor seria o próprio ponto de congelamento.

Observemos inicialmente, que o resultado obtido é uma segmentação *grosseira* da

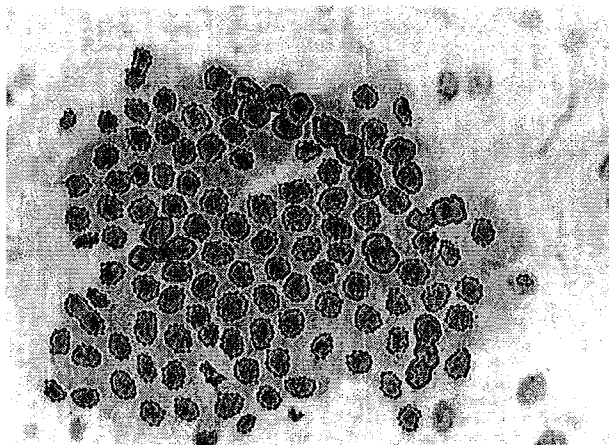


Figura 9.39: Resultado obtido para malha 1×1 e limiar $T = 150.0$, usando limites inferior e superior para área dados por 75.0 e 500.0, respectivamente.

imagem. Temos ainda regiões com mais de uma célula envolvidas por uma mesma curva, e existem células cujo padrão de intensidade não foi coberto pelo limiar usado. Nestas regiões, o problema não é a resolução usada, mas sim o limiar.

Se relaxarmos o limiar da intensidade de imagem para um valor $(T \pm \Delta T)$, poderíamos melhorar a segmentação nestas regiões (Giraldi *et al.*, 2000b). Contudo, as bordas obtidas poderiam avançar para o interior dos objetos, como mostrado na Figura (Figura 9.40). A solução automática para este problema é o ponto mais delicado do método.

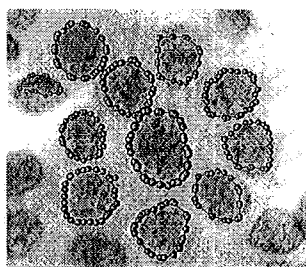


Figura 9.40: Resultado obtido para células da coluna usando malha 1×1 e limites inferior e superior para área dados por 75.0 e 500.0, respectivamente.

No presente momento, nos parece que a solução mais simples, e talvez mais eficiente, seria usar o resultado da Figura 9.39 como ponto de partida para um método semi-automático de segmentação. A questão em aberto é oferecer ferramentas para que o usuário possa alterar o resultado mostrado, dividindo regiões e melhorando bordas ainda mal definidas. Voltaremos a esta questão no Capítulo 10.

A Figura 9.41 mostra outro resultado. Neste caso temos uma imagem de células de porco. A Figura 9.41.c mostra as aproximações lineares obtidas e a Figura 9.41.d

mostra o resultado final para os seguintes parâmetros: $k = 1.5.0$, $\lambda = 1.0$, $b = 1.0$, $T = 105.0$, *Área Mínima*=100.0, *Área Máxima*=250.0. A resolução da malha é dada por 5×5 .

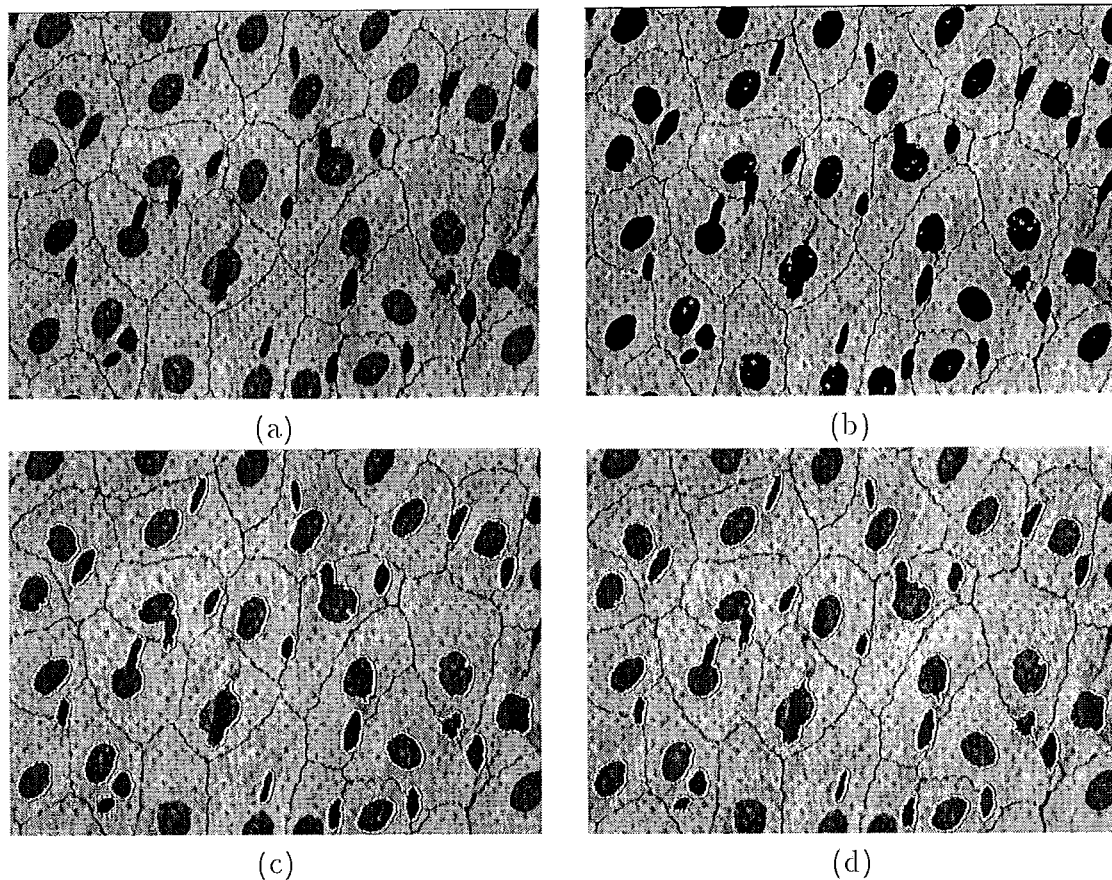


Figura 9.41: (a) Células de tecido animal (porco). (b) Aplicação do limiar $T = 105.0$ para a intensidade de imagem. (c) Variedade lineares por partes obtidas por (1)-(4). (d) Solução final.

Problemas de duas escalas podem ocorrer também quando temos apenas um objeto. A Figura 9.43 ilustra uma situação deste tipo.

Embora não seja uma figura de uma célula, o objeto amorfo nesta imagem tem características interessantes que podem ocorrer no contexto de imagens médicas também.

A escala externa neste caso se refere à separação entre partes do mesmo objeto. Nestas regiões a propriedade de escala local, que neste caso pode ser caracterizada por uma malha 9×9 e um limiar $T = 140.0$, é grosseira. Por isso, o resultado da aplicação do método de inicialização fornece uma aproximação muito distante daquela desejada.

Uma dificuldade deste caso, se comparado com os casos da seção 9.2.1 é que não podemos assumir convexidade do objeto de interesse. Se quisermos inicializar automaticamente o Dual-T-Snakes a partir da curva obtida, precisamos ser cuida-

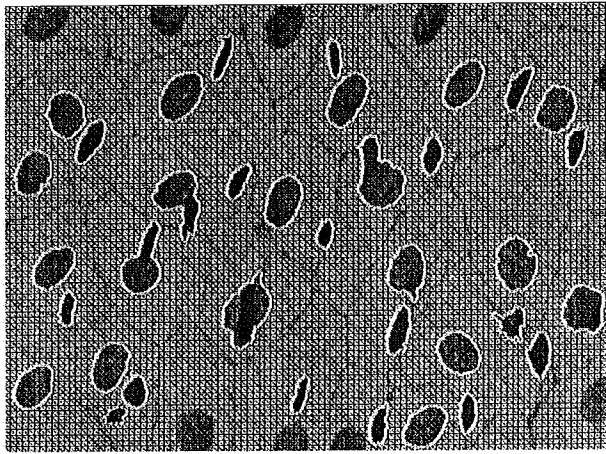


Figura 9.42: Resultado da Figura 9.41.d e malha 5×5 .

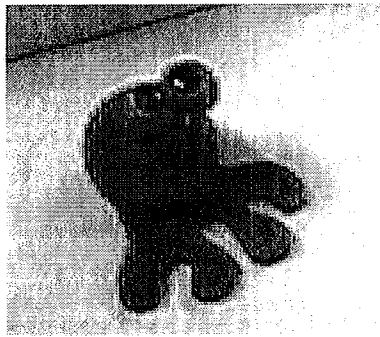


Figura 9.43: Imagem contendo objeto amorfo.

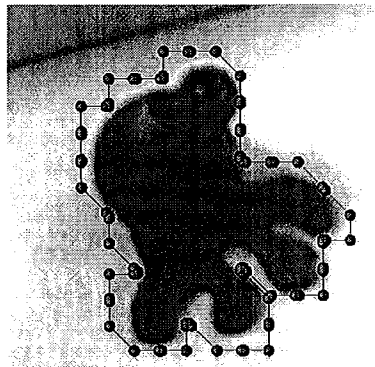


Figura 9.44: Inicialização obtida para a Figura 9.43 para malha 9×9 .

dosos para que a snake interna não corte as regiões com curvatura mais elevada do objeto. Isto implica em testes adicionais e pode ser complicado dependendo da complexidade da borda procurada.

Uma alternativa mais segura nestes casos seria usar uma malha com a resolução da própria imagem para o método de inicialização. O resultado obtido é uma aproximação bem mais detalhada da borda, como pode ser observado pela Figura 9.45.a.

Agora, partindo do algum conhecimento a priori do objeto, podemos fazer um offset do resultado obtido em direção ao interior do objeto, como mostra a Figura 9.45.a

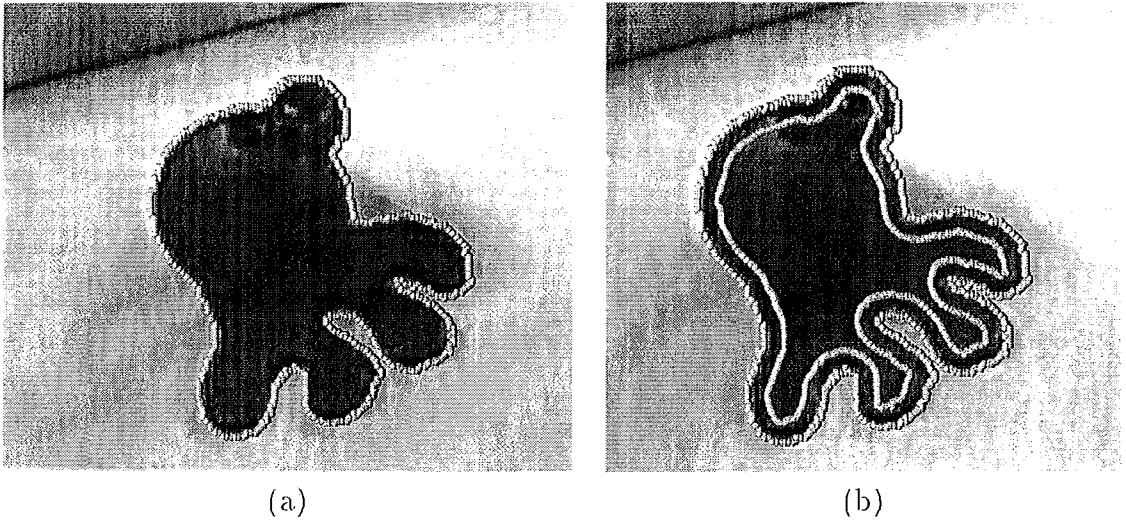


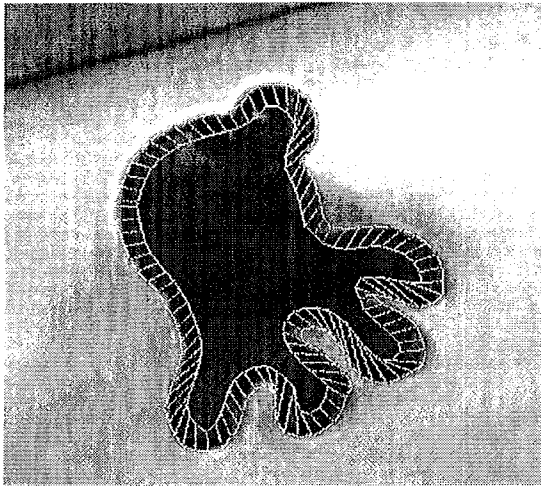
Figura 9.45: (a) Inicialização para malha com resolução da imagem. (b) Curva inicial e Offset.

Se quisermos mais segurança do resultado, poderemos usar dois offsets para limitar a borda desejada, o que será feito em um exemplo seguir. As Figura 9.46.a,b mostram o espaço de busca definido e resultado do Viterbi, respectivamente. Os parâmetros usados para o Viterbi são os seguintes: $\alpha = 1200.0$, $\beta = 10.0$ and $\lambda = 2.0$.

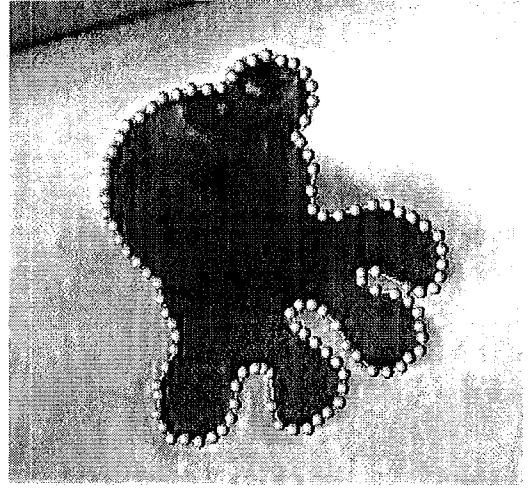
O offset é tomado na direção normal à curva resultante do método de inicialização. Possíveis auto-intersecções que venham ocorrer quando avançamos a curva da Figura 9.45.a são resolvidas naturalmente pelo método de reparametrização das T-Snakes (seção 5.3.2). A distância entre o offset e a curva inicial é o único parâmetro mais delicado e seu valor é totalmente dependente da aplicação em questão. Neste caso, usamos uma distância de 5 unidades entre as duas curvas da Figura 9.45.b.

9.2.4 Imagens de Micrografia Eletrônica

As imagens desta seção, foram obtidas via a técnica de micrografia eletrônica e retiradas do livro (Bloom & Fawcett, 1968), onde pode-se encontrar detalhes sobre



(a)



(b)

Figura 9.46: (a) Espaço de busca definido pela solução inicial e seu offset. (b) Resultado do Viterbi.

este método de aquisição.

No primeiro exemplo mostramos como o Dual-T-Snakes pode ser usado em uma metodologia multigríd.

Seja então a Figura 9.47 onde temos uma imagem com resolução 895×682 contendo uma estrutura interior de uma célula (*nucléolos*). Observemos inicialmente a presença de ruído e artefatos, bem como de regiões da fronteira com alta curvatura que exigem uma malha de resolução suficientemente alta para se obter a precisão desejada.

Contudo, quanto mais alta a resolução da malha maior o efeito do ruído e dos artefatos sobre o movimento das T-Snakes. Temos assim uma situação conflitante com um complicado *trade-off* entre eficiência e performance.

No contexto de modelos deformáveis, uma solução conhecida para este problema (Terzopoulos, 1986a) é usar uma malha com resolução mais grossa para obter uma primeira aproximação da borda. Em seguida, a malha é refinada e a solução melhorada até obter a precisão desejada.

Contudo, para o Dual-T-Snakes temos que levar em conta a necessidade de duas snakes em cada etapa. Em problemas como aqueles da seção 9.1 onde a solução do Dual-T-Snakes é uma faixa contendo a borda, bastaria inicializar o método na resolução mais elevada usando o resultado já obtido na resolução anterior.

Porém, no presente caso as bordas são do tipo degrau (Figura 2.1) e assim as snakes devem ficar aproximadamente sobrepostas. Logo, detalhes das regiões com curvatura mais alta não seriam extraídos ao passarmos para o próximo nível da malha. Usar um esquema de pirâmide usando filtros do tipo Sobel (Leymarie & Levine, 1993) poderia ser uma possibilidade para criar uma faixa contendo a borda

desejada, como foi o caso da seção 9.2.2. Contudo, isto tem seu custo também.

Considerando estes fatos tomamos a seguinte postura: vamos resolver o Dual-T-Snakes/Viterbi em uma malha mais grossa e usar o resultado para gerar dois offsets que inicializam o Dual-T-Snakes na resolução mais fina.

Vamos então resolver o Dual-T-Snakes para uma malha de resolução 15×15 , e usando os seguintes parâmetros: $k = 200.0$, $\lambda = 50.0$, $b = 10.0$, $T = 140.0$, *Ponto de Congelamento*=15.

Diferentemente dos casos anteriores, vamos permitir o passo máximo de cada snaxel em uma interação com sendo 2 unidades, uma vez que sabemos a priori que o resultado desta fase será uma aproximação grosseira da borda desejada.

A Figura 9.48.a mostra o resultado obtido.

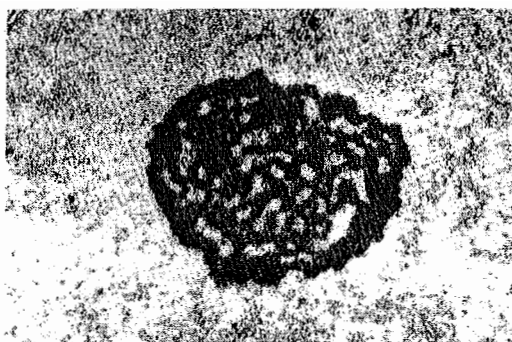
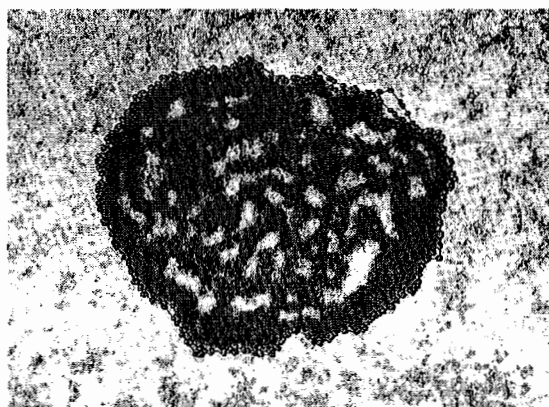
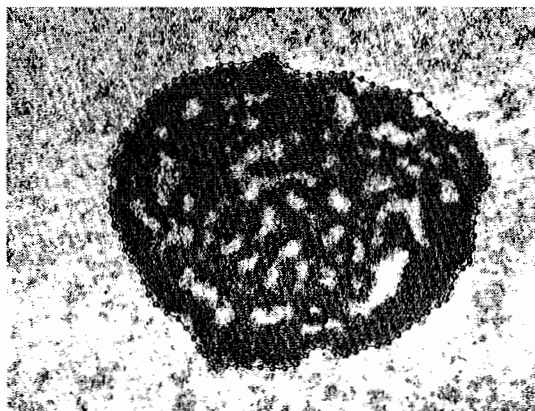


Figura 9.47: Estrutura interior (*nucleolos*) de uma célula.



(a)



(b)

Figura 9.48: Segmentação da imagem da Figura 9.47: (a) Solução do Dual-T-Snakes para malha 15×15 . (b) Solução correspondente do Viterbi.

Como já esperado, as snakes estão quase sobrepostas ao longo da borda, com exceção da região com curvatura mais elevada. A Figura 9.48.b mostra a soluçã

do Viterbi para o espaço de busca definido por este resultado com os seguintes parâmetros: $\alpha = 1200.0$, $\beta = 0.5$ and $\lambda = 0.0$.

De acordo com os comentários acima, usamos agora dois offsets desta solução, um em direção ao interior e outro em direção ao exterior do objeto.

Estas curvas são obtidas avançando a curva da Figura 9.48.b na sua direção normal. Como já comentado anteriormente, possíveis auto-intersecções que possam ocorrer neste procedimento são resolvidas naturalmente pela método de reparametrização da T-Snake (seção 5.3.2) não exigindo portando nenhum recurso extra.

A Figura 9.49.a mostra os off-sets usados para inicializar o Dual-T-Snakes para uma malha de resolução 5×5 . A distância entre cada off-set e a curva geradora é de 22 unidades. A Figura 9.49.b mostra a solução do Dual-T-Snakes nesta fase.

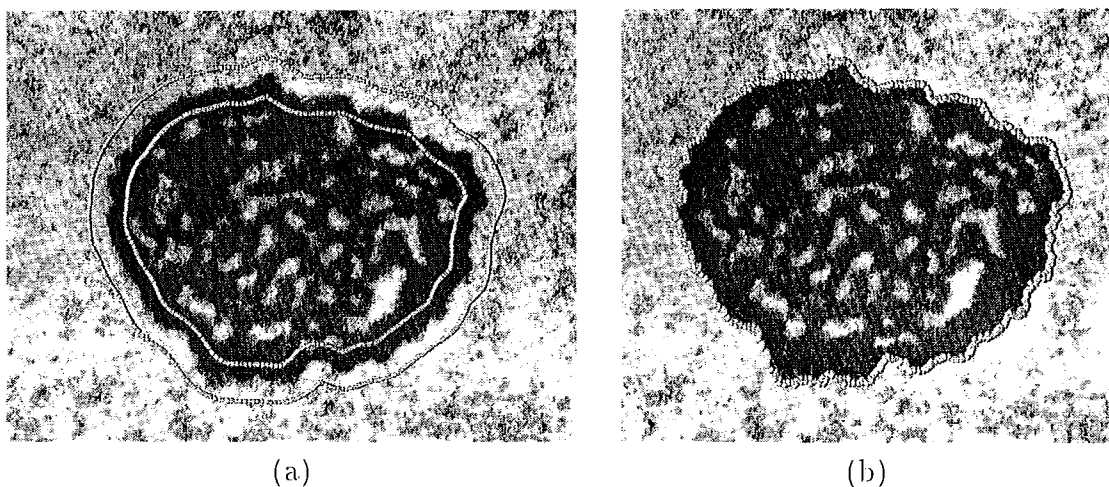


Figura 9.49: (a) Offsets para inicializar Dual-T-Snakes. (b) Solução do Dual-T-Snakes para malha 5×5 . (b)

A solução final obtida pelo Viterbi está na Figura 9.50 e os parâmetros são dados por: $\alpha = 1200.0$, $\beta = 0.5$ and $\lambda = 0.0$.

Observemos que para obter a solução parcial da Figura 9.48.b não há necessidade de aplicar duas buscas uma vez que partimos do pressuposto de que esta é apenas uma aproximação grosseira da borda. Quanto à distância entre os off-sets e a curva geradora, o valor da diagonal das células da triangulação é sempre um valor de referência.

9.2.5 Micrografia Eletrônica de Célula de Gato

A imagem a seguir mostra um exemplo envolvendo operações topológicas no contexto de imagens de células.

A Figura 9.51 mostra a imagem original a ser processada. Sua resolução é de 600×600 .

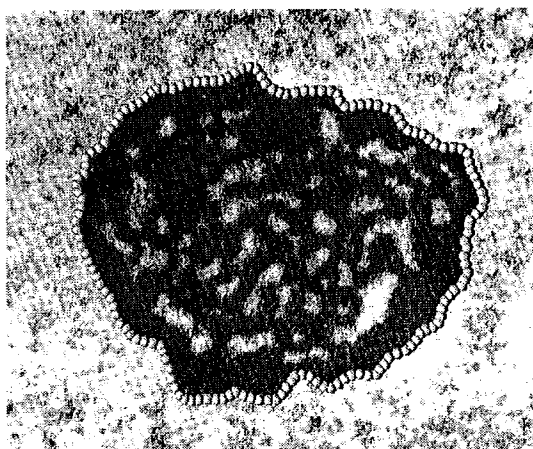


Figura 9.50: Solução final para imagem 9.47.

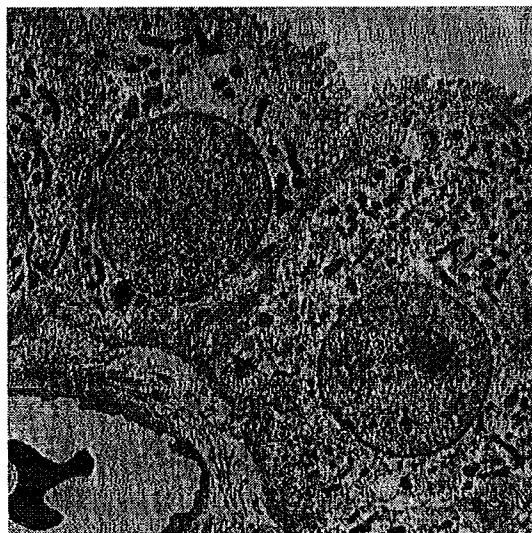


Figura 9.51: Imagem original.

A exemplo da Figura 9.47 esta imagem apresenta o interior e o exterior das células com padrões de intensidade de difícil caracterização. Neste caso em particular, vemos a presença de pequenas estruturas e artefatos os quais podem dificultar o movimento das snakes.

O primeiro ponto portanto é definir o tipo de pré-processamento da imagem. Uma observação que pode direcionar esta etapa é de que na borda das células temos uma estreita faixa com padrões de intensidade da ordem de $T = 80.0$. Contudo, existem pequenas estruturas com intensidades também dentro deste limiar. Assim, uma possibilidade interessante neste caso seria usar métodos de suavização que *respeitem* esta faixa, mas borrem o interior dos objetos.

Uma possibilidade simples para isto é a suavização direcional (Jain, 1989). Uma opção mais elaborada seria usar métodos de difusão não lineares do tipo daqueles discutidos na seção 2.2.

A Figura 9.52.a mostra a mesma imagem filtrada via suavização direcional e a Figura 9.52.b o resultado quando aplicamos o limiar dado acima:

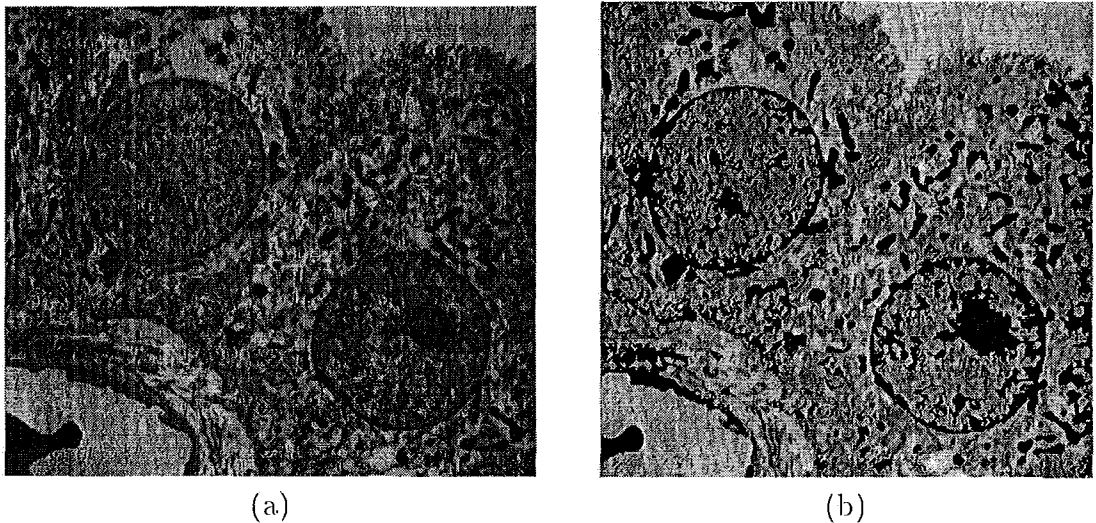


Figura 9.52: (a)Célula de Gato: Imagem filtrada por suavização direcional. (b) Resultado da aplicação do limiar $T = 80.0$.

A Figura 9.53.a mostra a inicialização do Dual-T-Snakes e a Figura 9.53.b mostra a solução parcial obtida com os parâmetros: $k = 200.0$, $\lambda = 50.0$, $b = 10.0$, $T = 80.0$, *Ponto de Congelamento*=28 e malha com resolução 20×20 .

O critério usado para retirar uma T-Snake do ponto de congelamento é a variação do limiar. A força normal é definida pela expressão 7.3.

Este exemplo mostra uma dificuldade do Dual-T-Snakes: a presença de artefatos pode fazer com que o critério de parada baseado na proximidade entre as snakes nunca seja atingido.

É exatamente o que ocorre com o dual snake da direita na Figura 9.53.b. Este tipo de comportamento obriga o estabelecimento de um número máximo de inte-

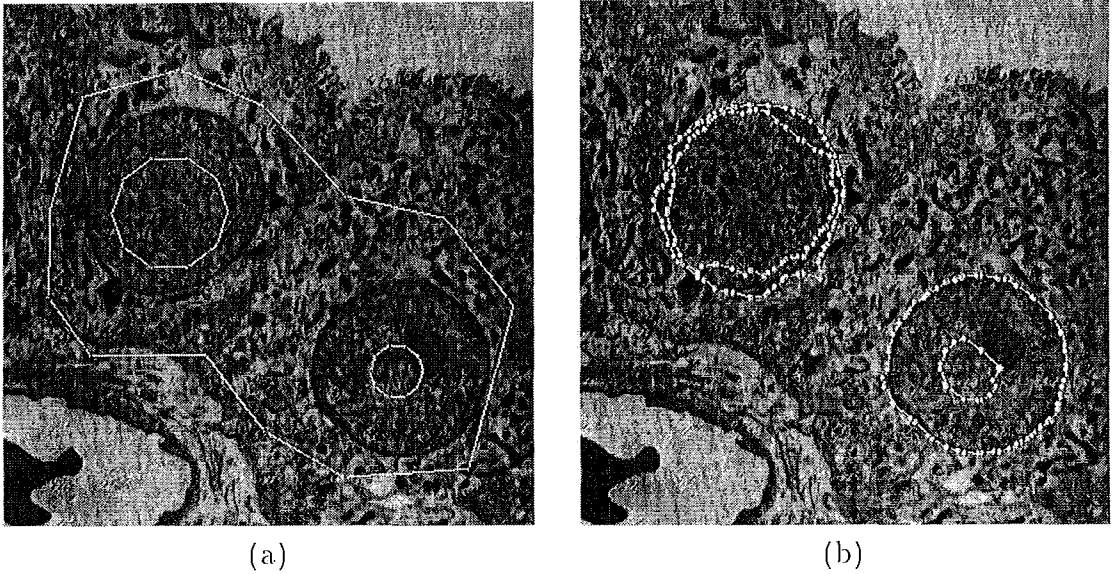


Figura 9.53: Célula de Gato: (a) Inicialização do Dual-T-Snakes. (b) Solução parcial do Dual-T-Snakes.

rações (um múltiplo do ponto de congelamento) a partir do qual assumimos que a instância atual do Dual-T-Snakes não será capaz de retirar as T-Snakes do ponto de congelamento.

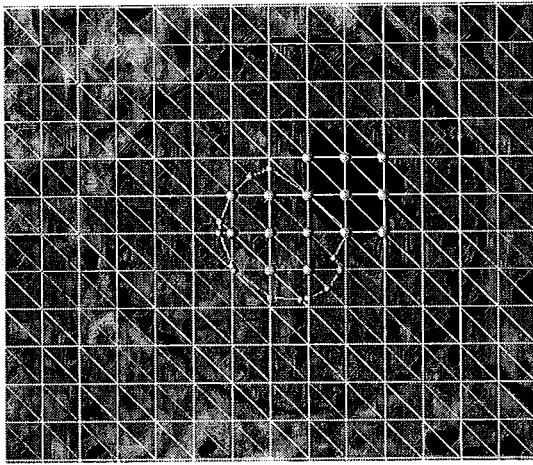
Nestes casos, poderíamos reduzir a dimensão da malha na tentativa de retirar a snake interna da posição atual. Contudo, para a figura acima, esta possibilidade não se mostrou eficiente.

O problema maior com relação à célula da direita é que o limiar usado deixa uma falha no interior da célula (Figura 9.52.b). A forma que nos pareceu mais eficiente para resolver esta dificuldade vem da seguinte observação: se os artefatos formam uma barreira para o movimento da T-Snake então poderíamos incorporar estes "objetos" ao interior da T-Snake usando um procedimento de busca simples nas vizinhanças da curva em questão.

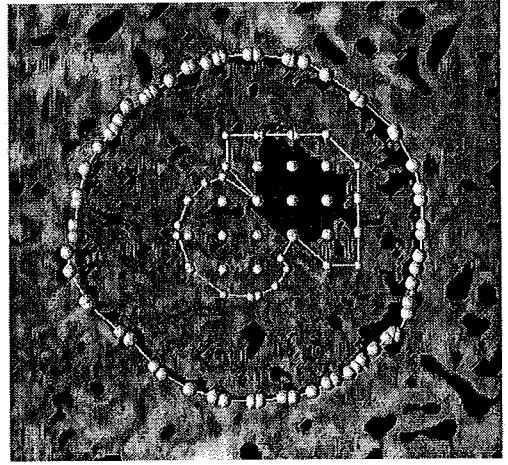
Para visualizar esta idéia, observe a Figura 9.54.a onde marcamos os nós da triangulação interiores à T-Snake bem como os nós vizinhos desta onde a intensidade está abaixo do limiar escolhido. A variedade linear por partes dada pelo algoritmo door-in-door-out pode ser usada para re-inicializar o Dual-T-Snakes para a célula indicada (Figura 9.54.b). Com isto, obtemos um procedimento mais eficiente e robusto para tratar estas situações.

A solução do Dual-T-Snakes está finalmente apresentada na Figura 9.55. O resultado para a célula da esquerda é idêntico ao apresentado na Figura 9.53.b uma vez que neste caso o critério de parada baseado na triangulação foi satisfeito. Para a célula da direita, observamos que a snake interna pode avançar, graças ao procedimento acima, chegando finalmente ao resultado desejado nesta fase.

A Figura 9.56 mostra a solução final obtida pelo Viterbi com os parâmetros dados



(a)



(b)

Figura 9.54: Célula de Gato: (a)Crescimento de região baseado na triangulação. (b) Re-inicializando o Dual-T-Snakes para a célula da esquerda.

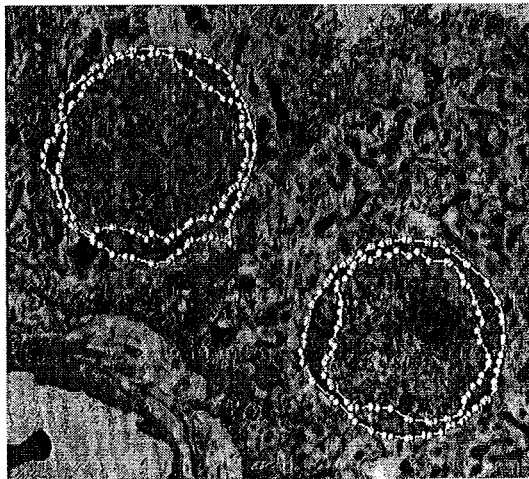


Figura 9.55: Solução final do Dual-T-Snakes juntamente com crescimento de região para células de Gato.

por: $\alpha = 1200.0$, $\beta = 10.0$ and $\lambda = 2.0$. O resultado obtido tem boa qualidade.

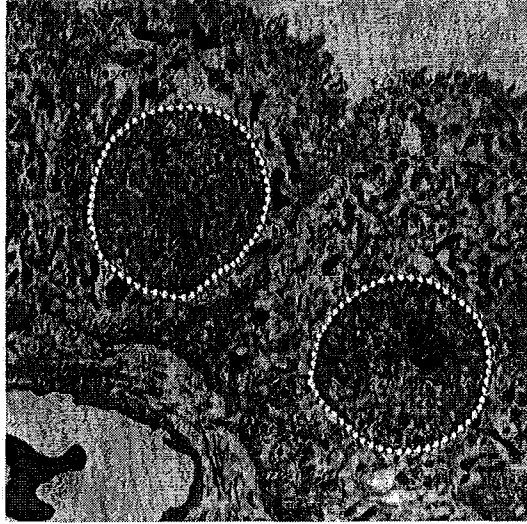


Figura 9.56: Solução final para células de Gato.

9.2.6 Homem Visível

Neste projeto um cadáver de um homem foi congelado e fatiado e estas fatias por sua vez foram digitalizadas em um equipamento especial. O resultado é um conjunto de dados colorido e muito detalhado do corpo humano.

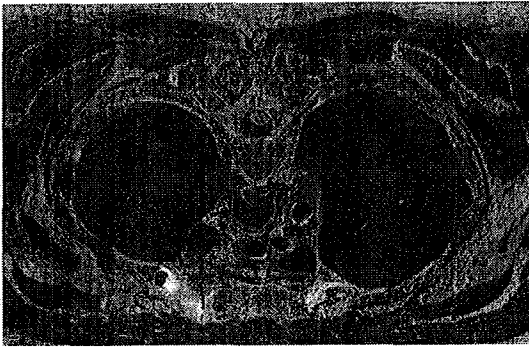
No Brasil, este projeto tem despertado interesse de pesquisadores em mecânica de fluidos interessados em aplicar métodos de simulação numérica na análise do fluxo sanguíneo e dos fenômenos relacionados. Esta é uma área da bioengenharia com muitos problemas em aberto e multidisciplinar por natureza, uma vez que envolve conhecimentos em anatomia, biologia, fluidos e processamento de imagens.

As imagens da Figura 9.57 constituem dois exemplos (aqui convertidos em tons de cinza) do conjunto de dados citado. Estas imagens foram cedidas pelo pesquisador Raúl A. Feijóo, do Laboratório de Computação Científica (LNCC), onde desenvolve-se atualmente uma linha de pesquisa em bioengenharia voltada para hemodinâmica.

Estas imagens mostram dois cortes próximos, na altura do peito. As estruturas em tons mais escuros correspondem a artérias, ao esôfago e aos pulmões (estruturas maiores). O bom contraste observado e a escala interna destas estruturas tornam a aplicação do método de inicialização do Capítulo uma boa opção para a etapa de segmentação.

Neste caso, a propriedade de escala local é bem caracterizada por uma malha 5×5 e o limiar para a imagem dado por $T = 35.0$.

A imagem da Figura 9.58.a mostra as aproximações lineares obtidas pelo algoritmo door-in-door-out e a Figura 9.58.b mostra a solução final do método do Capítulo



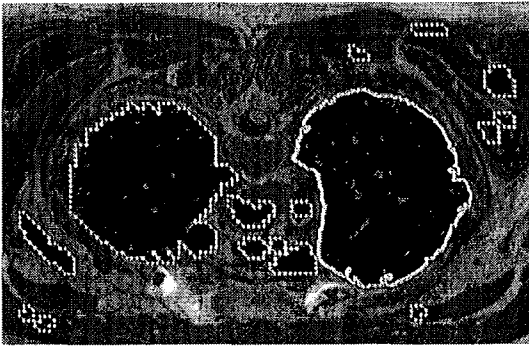
(a)



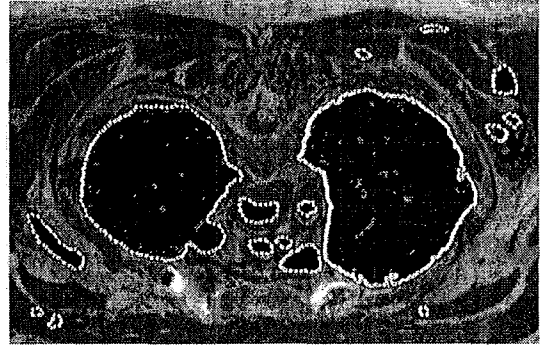
(b)

Figura 9.57: (a)-(b) Imagens do Projeto Homem Visível.

8.



(a)



(b)

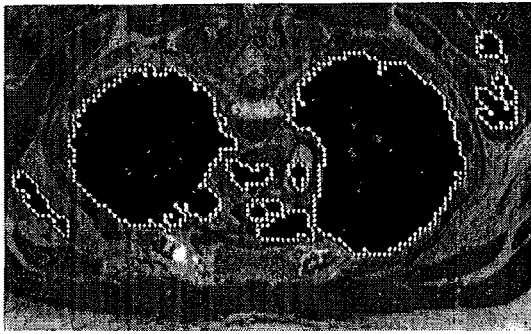
Figura 9.58: Homem Visível: (a) Variedades lineares obtidas. (b) Resultado obtido.

A Figura 9.59.a mostra as aproximações lineares para a segunda imagem. Observe que duas estruturas ficaram envolvidas por uma mesma snake. Isto veio do fato de não termos imposto um limite superior de área neste caso. A Figura 9.59.b mostra este caso resolvido, com as estruturas segmentadas corretamente.

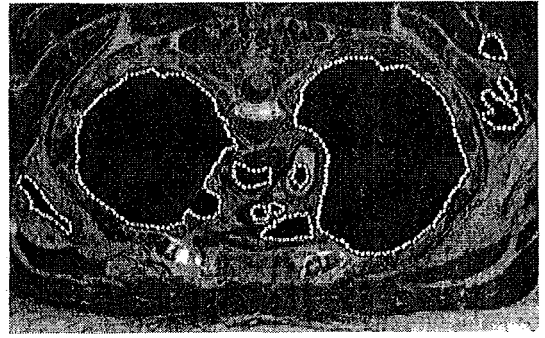
9.2.7 Raio-X do Tórax

As radiografias do tórax constituem um importante recurso para diagnóstico de distúrbios e lesões pulmonares. O advento da radiografia digital possibilitou a pesquisa de métodos automáticos para a análise destas imagens e auxílio ao diagnóstico. Um ponto fundamental para isto é a segmentação das mesmas (van Ginneken & ter Haar Romery, n.d.).

Neste caso também, a extração do contorno que delimita cada pulmão tem papel de destaque. Em (van Ginneken & ter Haar Romery, n.d.) encontramos uma técnica com esta finalidade baseada em heurísticas e detecção de pontos de borda orientados em direções definidas previamente.



(a)



(b)

Figura 9.59: Homem Visível: (a)Variedades lineares obtidas. (b) Inicialização final para Figura 9.57.b.

A Figura 9.60 mostra uma imagem deste tipo, onde vemos o resultado de uma exposição frontal de um paciente. As dificuldades de processamento deste tipo de imagem são várias.

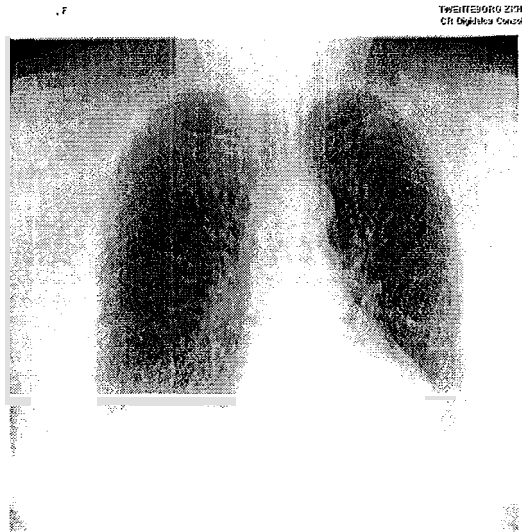


Figura 9.60: Radiografia do tórax.

Primeiramente, observemos que o contraste nas bordas dos pulmões é muito pouco definido, o que implica que a snake terá pouca resistência ao seu movimento quando próxima às bordas desejadas. Além disso, acrescenta-se o problema dos padrões de intensidade no interior dos pulmões ser de difícil caracterização.

O background é mais uniforme, o que poderia fornecer um limiar para definir a força normal. Contudo, o tênue contraste nas bordas torna esta possibilidade pouco eficiente.

Apesar destas dificuldades, este é um caso onde podemos explorar o potencial do Dual-T-Snakes uma vez que temos dois objetos de interesse.

Como nos exemplos anteriores, o primeiro ponto é o pré-processamento da imagem. Uma vez que temos pouco contraste nas bordas a suavização gaussiana não parece ser adequada num primeiro momento, embora pudesse ser interessante para suavizar o interior dos objetos.

Optamos assim por usar inicialmente um filtro da mediana, para uma janela de dimensão 5×5 , aplicado 5 vezes sobre a imagem. Este procedimento suaviza mais intensamente o interior dos objetos do que suas fronteiras, o que é um resultado desejado neste momento.

Em seguida, aplicamos um filtro gaussiano para diminuir o efeito de possíveis artefatos criados na filtragem anterior. O resultado está na Figura 9.61.a. A Figura 9.61.b mostra o resultado da aplicação de um limiar $T = 180.0$ sobre a imagem assim obtida. Este limiar será usado na definição da força normal dada pela expressão (5.54).

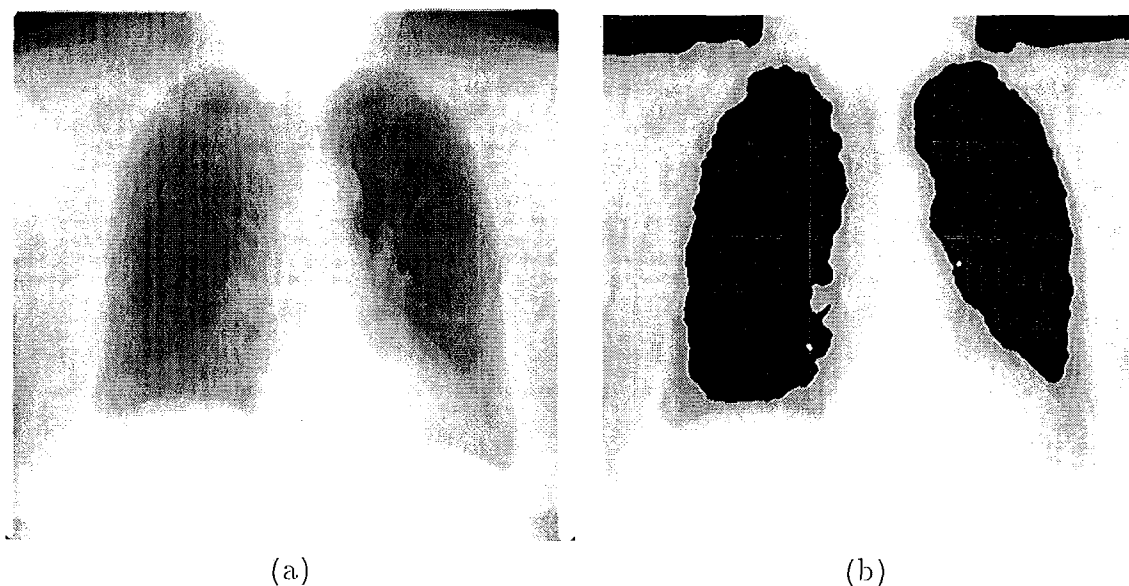
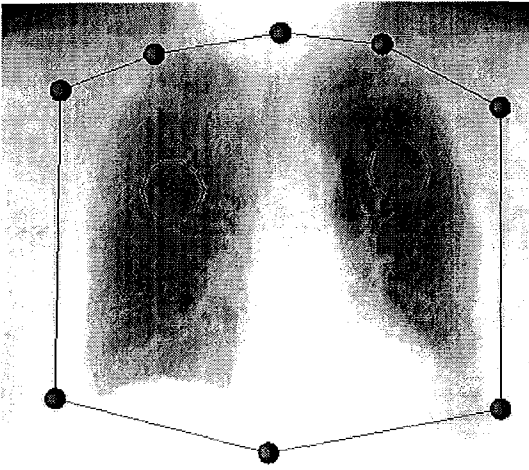


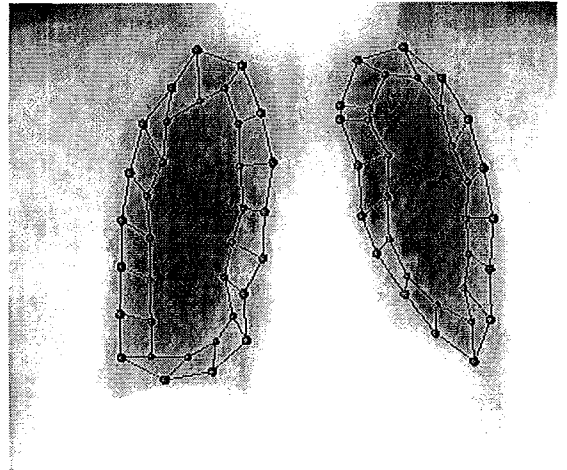
Figura 9.61: Radiografia do tórax: (a) Imagem filtrada. (b) Resultado da aplicação do limiar $T = 180.0$.

A Figura 9.62.a mostra a inicialização do Dual-T-Snakes e o espaço de busca definido a partir da solução obtida. Os parâmetros usados são os seguintes: $k = 15.0$, $\lambda = 14.0$, $b = 0.2$. A malha tem resolução 4×4 .

O resultado obtido está longe do desejado, principalmente nas regiões inferiores. O pré-processamento usado tem o efeito positivo de diminuir os contrastes entre os diversos artefatos no interior dos objetos. Isto é interessante pois do contrário a snake interna poderia ficar presa em uma região interna e o método *puxaria* a snake externa em direção a esta região perdendo totalmente a borda desejada, sobre a qual o contraste é pouco definido.



(a)



(b)

Figura 9.62: Radiografia do tórax: (a) Inicialização do Dual-T-Snakes. (b) Espaço de busca obtido.

Contudo, não resolvemos o problema das regiões onde temos pouca definição. A aplicação do Dual-T-Snakes de forma mais eficiente depende do aprimoramento do pré-processamento para este problema específico, o que não é objetivo deste trabalho.

Capítulo 10

Conclusões e Perspectivas

Neste trabalho nós enfocamos os modelos de contornos ativos 2D partindo do modelo original (Kass *et al.* , 1988).

Descrevemos as limitações básicas deste modelo e analisamos algumas soluções encontradas na literatura bem como os modelos alternativos correspondentes. Dentre estes modelos, o modelo Dual (Gunn & Nixon, 1997; Gunn, 1996a) e o método das T-Snakes (McInerney & Terzopoulos, 1995; McInerney, 1997) tiveram lugar central uma vez que uma contribuição desta tese é a combinação dos mesmos gerando o método das Dual-T-Snakes (Giraldi *et al.* , 2000d; Giraldi *et al.* , 2000c). Basicamente, o Dual-T-Snakes consiste em incorporar ao método das T-Snakes um *modelo Dual Generalizado*: um *contorno externo* envolvendo os objetos de interesse contrai e se divide enquanto que *contornos internos* aos objetos se expandem em direção às bordas procuradas.

Com isto, pudemos colocar as habilidades topológicas das T-Snakes juntamente com a capacidade de rejeitar mínimos locais do modelo Dual evolucionário original em uma única metodologia. Os resultados experimentais do Capítulo 9 colocam em evidências estas qualidades do Dual-T-Snakes. Estes resultados demonstram também a robustez do novo método com relação a mínimos locais.

Uma vez que o Dual-T-Snakes não faz uso de correspondência entre as snakes, ficamos livres da necessidade de um modelo de forma, como é o caso do método Dual original. Pudemos assim estender o método dual para formas e topologias mais gerais.

Do ponto de vista das T-Snakes, o Dual-T-Snakes trouxe efetivamente mais robustez quanto aos mínimos locais da energia de imagem pelo balanço da energia/afinidade e pela nova noção de proximidade (baseada na triangulação) entre as snakes internas e externas.

Em geral, o resultado do Dual-T-Snakes são duas curvas limitando a região onde está a borda procurada. Fazendo proveito desta redução do espaço de busca, nós propomos na seção 6.2 a utilização do modelo Dual Baseado em Programação Dinâmica, uma vez que as snakes internas e externas necessárias para a aplicação

do algoritmo de Viterbi (seção 5.2) são exatamente o resultado do Dual-T-Snakes.

Os exemplos com imagens sintéticas do Capítulo 9 evidenciam a principal classe de aplicação do Dual-T-Snakes: imagens com background e objetos de interesse com padrões de intensidade pouco homogêneos.

A complexidade computacional do Dual-T-Snakes é $O(N_T)$, onde N_T é o número de triângulos da divisão simplicial do espaço. Uma vez que usamos sempre snakes internas e externas, o custo computacional efetivo é sempre da ordem do dobro daquele correspondente a usar apenas uma T-snake envolvendo os objetos.

A inicialização do Dual-T-Snakes pode ser feita pelo usuário, ou pelo método do Capítulo 8, este último constituindo-se em uma contribuição desta tese no contexto de inicialização de modelos deformáveis (Giraldi *et al.*, 2000e; Giraldi *et al.*, 2000b; Giraldi *et al.*, 2000a).

Esta metodologia usa os elementos básicos das T-Snakes (*estatísticas da imagem, triangulação e funções características*), bem como um conhecimento prévio da *escala local* dos objetos, para inicializar um modelo deformável próximo da fronteira dos mesmos. A complexidade deste método também é da ordem do número de triângulos da divisão simplicial do espaço.

Na seção 9.2.1 tivemos exemplos de como este método de inicialização pode ser usado para definir a snake externa e em seguida a interna, desde que tenhamos informações adicionais da geometria dos objetos.

Os exemplos do Capítulo 9 mostram que a escolha de parâmetros não é crítica para o Dual-T-Snakes. A idéia é abrir mão de uma maior suavidade da solução, uma vez que o resultado que se deseja com o Dual-T-Snakes é apenas diminuir o espaço de busca para em seguida obter a solução desejada pelo método de Viterbi.

O exemplos do Capítulo 9 indicam que a política de variação do limiar é mais eficiente para retirar uma T-Snake de um mínimo local no caso de imagens com ruídos. Neste caso também, os resultados apresentados com imagens sintéticas evidenciam a maior eficiência da definição da normal dada pela equação 7.3 com relação àquela dada pela equação (5.54).

O aumento da resolução da malha se mostrou eficiente para extrair detalhes da borda em regiões com curvatura mais alta podendo também ser usada para retirar uma T-Snake de um ponto de congelamento.

Quando o resultado do Dual-T-Snakes apresentar intersecção entre as snakes ou mesmo sobreposição das mesmas (seção 9.2.4), o espaço de busca pode ser obtido através de offsets de uma das snakes ou mesmo da solução para a primeira busca do Viterbi. Os offsets podem ser tomados na direção normal à solução parcial, uma vez que as possíveis auto-intersecção que podem ocorrer são resolvidas naturalmente pelo método de reparametrização das T-Snakes.

A utilização de métodos multigrid com o Dual-T-Snakes pode prejudicar a sua-

vidade da solução desta fase, caso não atualizemos os parâmetros ao passarmos para uma malha mais fina. Contudo, uma vez que podemos abdicar de uma maior suavidade para a solução do Dual-T-Snakes a utilização de metodologias deste tipo pode ser eficiente uma vez que a escolha dos parâmetros fica simplificada. Os resultados das seções 9.2.4 e 9.1.4 apontam exatamente nesta direção.

O exemplo da seção 9.2.5 onde usamos um método de crescimento de região baseado na triangulação parece ser uma proposta eficiente para avançar uma T-Snake presa em artefatos da imagem (mínimo local) que não possam ser eliminados eficientemente pela variação do limiar usado na definição da força normal.

Como limitações do Dual-T-Snakes, temos primeiramente a necessidade de limitar o passo máximo da snake a uma unidade. Isto implica em perda de performance. A necessidade de interação com o usuário na sua inicialização é outra limitação do método.

Quanto ao método de inicialização do Capítulo 8, sua principal limitação está vinculada à necessidade de um conhecimento a priori da escala local dos objetos de interesse.

Com relação ao Viterbi, o maior problema encontrado é a necessidade de um espaço de busca mais refinado nas regiões com curvatura mais elevada (Figuras 9.25). Se tomarmos um espaço de busca mais fino ao longo de toda a região limitada pelas snakes poderemos comprometer a performance do algoritmo. Por outro lado, a identificação automática das regiões que necessitam de um espaço de busca mais refinado dependem de testes adicionais que também possuem seu custo.

10.1 Perspectivas Futuras

Os dois métodos que constituem a contribuição deste trabalho (*Dual-T-Snakes e Inicialização de Modelos Deformáveis*) são fundamentados na metodologia das T-Snakes, a qual tem extensão para 3D (T-Superfícies, Apêndice C) com eficiência já comprovada na literatura (McInerney & Terzopoulos, 1999; McInerney, 1997). Assim, toda a metodologia deste trabalho pode ser estendida para superfícies. Os fundamentos matemáticos para isto são desenvolvidos no Apêndice C.

Uma vez que estamos limitando o passo da snake a no máximo um pixel por interação, podemos testar a possibilidade de usar uma noção de afinidade baseada na energia de imagem sobre a snake; isto é, dada uma T-Snake congelada em um mínimo local, aplicaríamos força adicional em regiões onde o campo externo seja menos intenso. A detecção destas regiões da snake poderia ser feita usando-se a metodologia apresentada em (Durikovic *et al.*, 1995). Tal procedimento não seria eficiente para bordas subjetivas e implicaria em redefinir o papel da força normal no Dual-T-Snakes. Além disto, sua extensão para 3D não parece possível.

Uma possibilidade para reduzir o esforço computacional do método de segmentação baseado no Dual-T-Snakes e em Programação Dinâmica (seção 6.2) seria usar métodos multi-resolução baseados em Quadrees ou Pirâmides. Neste caso, primeiramente aplicaríamos o Dual-T-Snakes para a imagem com resolução mais baixa. Em seguida, usaríamos uma técnica de minimização em multi-escala (Heitz *et al.* , 1994; Mignotte & Meunier, February, 2000) para fazer o tracking das bordas desenhadas pelos diferentes níveis da estrutura.

Podemos abrir a possibilidade de mais de um ponto semente em um mesmo objeto tratar objetos com vários padrões de intensidade. Cada ponto semente seria o centro de uma T-Snake interior. Estas snakes expandiriam, sofrendo merges durante a sua evolução até que finalmente tenhamos um dual snakes com apenas duas snakes.

Esta possibilidade traz a perspectiva de explorar mais intensamente a simbiose natural entre o método das conexões fuzzy (Udupa & Samarasekera, 1996) e o Dual-T-Snakes discutida no Apêndice E. Os pontos sementes para o método fuzzy seriam usados para inicializar o Dual-T-Snakes e o campo gerado seria usado para definir o sentido da força normal das snakes, como feito na seção 9.1.5.

A interação com o usuário é um recurso que não exploramos no contexto do Dual-T-Snakes. Embora o grande desafio do método seja deixá-lo totalmente automático, a possibilidade do usuário estabelecer vínculos para a snake pode ser útil, particularmente para problemas onde o contraste na fronteira dos objetos de interesse é baixo, caso típico das imagens de Raios-X do tórax analisadas na seção 9.2.7.

A interação com o usuário vem naturalmente ligada ao desenvolvimento de um sistema de segmentação/extração de bordas com interfaces independentes do Data Explorer. Este sistema poderia incorporar não apenas o Dual-T-Snakes mas também o método de inicialização do Capítulo 8. Como já apontado na seção 8.3, este método pode ser ponto de partida para uma metodologia semi-automática de segmentação baseada em extração de bordas. Neste caso, a idéia básica é usar o resultado do método como uma pré-segmentação da imagem a qual seria melhorada pela ação do usuário.

Assim, regiões onde temos mais de um objeto envolvido por uma única snake, e que não foram separados na resolução mais fina, poderia ser *divididas* apropriadamente pela ação do especialista. Tal proposta pressupõe que tenhamos políticas eficientes de interação com o resultado inicial.

O *live wire* - paradigma de extração de bordas baseado em interação com o usuário (Falcão *et al.* , 1996)- poderia ser uma primeira opção neste sentido, juntamente com metodologias semi-automáticas para efetuar splits (cortes, no caso) de regiões.

O desenvolvimento de um sistema usando estes elementos é mais uma perspectiva futura para este trabalho. Em (Liang *et al.* , 1999b; Liang *et al.* , 1999a) encontramos

um sistema que unifica live wire e snakes em um método único. Contudo, este sistema não pressupõe mudanças topológicas.

A análise de convexidade do Apêndice D pode ser interessante no estudo de modelos para tracking baseados em sistemas de equações diferenciais de segunda ordem (Peterfreund, 1999).

Os resultados da teoria de grupos de Lie sumarizados no Apêndice B e a discussão da seção B.3 devem ser tomados como ponto de partida para uma revisão ou mesmo generalização de modelos globais de snakes baseados em conceitos de invariância, tais como o AI-Snakes (seção 5.1.3).

Apêndice A

Programação Dinâmica no Contínuo

Seguindo a descrição apresentada em (Dreyfus, 1965), consideremos o problema de encontrar a curva $w = w(z)$, satisfazendo $w(z_0) = w_0$ e $w(z_1) = w_1$, e que minimiza um funcional $J[w]$ do tipo:

$$J[w] = \int_{z_0}^{z_1} F(z, w, w') dz, \quad (\text{A.1})$$

onde:

$$w' = \frac{dw}{dz}. \quad (\text{A.2})$$

Consideremos como ponto inicial da curva um ponto (x, y) , onde $x \leq z_1$ em vez do ponto inicial (z_0, w_0) . O problema torna-se então encontrar a curva admissível $w = w(z)$, $x \leq z \leq z_1$, com $w(x) = y$ e $w(z_1) = w_1$, a qual minimiza o funcional:

$$J[w] = \int_x^{z_1} F(z, w, w') dz. \quad (\text{A.3})$$

Introduzimos agora uma função auxiliar, chamada *função de valor ótimo*, através da definição:

$$S(x, y) = \min_P \left[\int_x^{z_1} F(z, w, w') dz \right], \quad (\text{A.4})$$

onde P é o conjunto que contém as curvas admissíveis, saindo do ponto inicial (x, y) e satisfazendo a condição $w(z_1) = w_1$.

Supondo agora $x < z_1$, tomemos inicialmente um pequeno incremento Δz tal que $x + \Delta z \leq z_1$. Assim, a integral em (A.4) toma a forma:

$$S(x, y) = \min_P \left[\int_x^{x+\Delta z} F(z, w, w') dz + \int_{x+\Delta z}^{z_1} F(z, w, w') dz \right], \quad (\text{A.5})$$

Nós, então, consideramos dentre as curvas admissíveis apenas aquelas passando por (x, y) , assumindo valor arbitrário em $x + \Delta z$, mas que são ótimas no intervalo $[x + \Delta z, z_1]$; isto é. montamos agora o novo problema:

$$S_{new}(x, y) = \min_P \left[\int_x^{x+\Delta z} F(z, w, w') dz + S(x + \Delta z, w(x + \Delta z)) \right], \quad (\text{A.6})$$

Vamos supor válido o *Princípio de Otimalidade* (Dreyfus, 1965):

O caminho ótimo entre dois pontos é também ótimo entre quaisquer dois outros pontos sobre o mesmo .

Por este princípio podemos igualar (A.5) and (A.6):

$$S(x, y) = S_{new}(x, y), \quad (\text{A.7})$$

obtendo assim a expressão básica desta seção (Dreyfus, 1965):

$$S(x, y) = \min_P \left[\int_x^{x+\Delta z} F(z, w, w') dz + S(x + \Delta z, w(x + \Delta z)) \right], \quad (\text{A.8})$$

Façamos agora aproximações de primeira ordem com a finalidade de obter uma forma diferencial para a expressão (A.8). Para isto, vamos primeiramente supor que w' é contínua no intervalo $[x, x + \Delta z]$, e que F também é contínua nos seus argumentos. Com isto podemos assumir que a contribuição de $w = w(z)$ ao valor da integral em (A.8) sobre o interval $[x, x + \Delta z]$ pode ser aproximado por:

$$\int_x^{x+\Delta z} F(z, w, w') dz \approx F(x, y, y') \Delta z + o(\Delta z), \quad (\text{A.9})$$

onde:

$$y' = \frac{dw}{dz}(z = x), \quad (\text{A.10})$$

e $o(\Delta z)$ denota termos em Δz tais que:

$$\lim_{\Delta z \rightarrow 0} \left[\frac{o(\Delta z)}{\Delta z} \right] = 0, \quad (\text{A.11})$$

Analogamente para a função w podemos escrever:

$$w(z + \Delta z) = y + y' \Delta z + o(\Delta z), \quad (\text{A.12})$$

supondo continuidade para w e w' em $[x, x + \Delta z]$.

Com estas aproximações, podemos escrever a expressão (A.8) na forma:

$$S(x, y) = \min_{y'} [F(x, y, y') \Delta z + o(\Delta z) + S(x + \Delta z, y + y' \Delta z + o(\Delta z))], \quad (\text{A.13})$$

Esta expressão é ponto de partida para a formulação diferencial do problema contínuo em programação dinâmica.

Supondo a existência da segunda derivada de $S(x, y)$ e que a mesma seja limitada, nós podemos expandir o lado direito da equação (A.13) em torno de (x, y) e escrever:

$$S(x, y) = \min_{y'} [F(x, y, y') \Delta z + o(\Delta z) + S(x, y) + S_x(x, y) \Delta z + S_y(x, y) y' \Delta z + o(\Delta z)]. \quad (\text{A.14})$$

Subtraindo $S(x, y)$ de ambos os lados da equação (A.14) e tomando o limite para Δz tendendo a zero, obteremos a equação:

$$0 = \min_{y'} [F(x, y, y') + S_x(x, y) + S_y(x, y) y']. \quad (\text{A.15})$$

Esta é a equação diferencial parcial que deve ser satisfeita pela função $S(x, y)$. A utilidade prática desta equação é dificultada pelo problema de minimização envolvido. Uma forma de simplificar esta equação é observar que uma condição necessária para que a mesma seja satisfeita será:

$$0 = F(x, y, y'(x, y, S_y)) + S_x + S_y y'(x, y, S_y), \quad (\text{A.16})$$

onde aqui fazemos uso explícito do fato de y' ser depende de x, y e S_y .

Supondo agora que F seja continuamente diferenciável com respeito a y' , que y' é limitada e sem vínculos, então, a derivada parcial com respeito a y' da expressão entre colchetes em (A.15) será zero. O resultado assim obtido, juntamente com a expressão em (A.16), forma o seguinte sistema de equações diferenciais parciais:

$$F_{y'}(x, y, y') + S_y = 0. \quad (\text{A.17})$$

$$F(x, y, y') + S_x + S_y y' = 0, \quad (\text{A.18})$$

Resolvendo (A.17) e (A.18) para S_y e S_x , respectivamente, obteremos:

$$S_y = -F_{y'}(x, y, y'), \quad (\text{A.19})$$

$$S_x = -F(x, y, y') + F_{y'} y'. \quad (\text{A.20})$$

Supondo que $S(x, y)$ tenha derivadas de segunda ordem contínuas, podemos derivar (A.19) com relação a x , (A.20) com relação a y , igualando ao final as expressões obtidas. Após um simples re-agrupamento de termos, obtemos a equação diferencial a seguir:

$$F_{y'y'} y'_x + F_{y'y'} y'_y + (F_{xy'} + F_{yy'} - F_y) = 0, \quad (\text{A.21})$$

a qual é independente da função $S(x, y)$.

Esta equação é do tipo *quase-linear*; ou seja, pode ser colocada na forma:

$$P(x, y, y') y'_x + Q(x, y, y') y'_y + R(x, y, y') = 0, \quad (\text{A.22})$$

a qual é linear nas derivadas parciais da função solução $y'(x, y)$. Existe uma teoria especial, chamada *teoria de características*, para este tipo de equação diferencial (Dreyfus, 1965) o que torna este problema de particular interesse teórico.

É importante ressaltar que a equação diferencial em (A.21) fornece uma condição necessária, mas não suficiente para garantir a globalidade da solução.

Apêndice B

Grupos de Lie e Invariância

Neste capítulo discutimos alguns resultados e conceitos da teoria de Grupos de Lie que são relevantes para problemas em processamento de imagens e visão computacional envolvendo as noções de invariância e espaços de escalas

Estes conceitos são naturalmente tratados dentro da teoria dos Grupos de Lie. Esta teoria está baseada nas idéias de grupo de transformações e variedades diferenciáveis.

Do ponto de vista geométrico, o conceito de variedades diferenciáveis separa claramente o objeto e o sistema de coordenadas usado para sua descrição.

Ao estudarmos a geometria intrínseca de um objeto, as propriedades de interesse não estão vinculadas ao sistema de coordenadas usado para descrevê-lo. A base matemática para a descrição de um objeto, que seja invariante por transformações de coordenadas, está no *cálculo tensorial*.

Do ponto de vista físico, mais próximo portanto de visão computacional, este problema está diretamente ligado ao sistema de unidades fundamentais escolhidas para descrever um fenômeno e o comportamento das grandezas derivadas quando re-escalamos as fundamentais. A *Lei Universal de Invariância* estabelece que as leis físicas devem ser independentes da escolha das grandezas fundamentais (Florack *et al.*, 1992).

Esta lei é básica para a *Análise Dimensional*, cuja formulação mais rigorosa está no teorema dos π (seção B.2). Este resultado, por sua vez, é básico para o trabalho (Florack *et al.*, 1992) que formula o espaço de escalas partindo de considerações algébricas em lugar de analíticas, como foi o caso de (Witkin, 1983).

É importante observar que ao convoluirmos uma imagem com o Kernel Gaussiano, estamos de fato operando sobre o objeto (variedade), transformando-o.

A questão agora é determinar quantidades que sejam invariantes sob transformações do objeto propriamente dito. A base matemática da teoria de invariantes em questão está na teoria de grupos de Lie que é o tema central deste apêndice.

Iniciamos com conceitos necessários à definição de Variedades Diferenciáveis para em seguida apresentarmos os Grupos de Lie e os conceitos da teoria de invariantes.

Passamos em seguida ao Teorema dos π citado acima para então finalizarmos com o método para cálculo de invariantes e aplicações.

B.1 Grupos de Lie

B.1.1 Espaços Topológicos

Dar uma *topologia* sobre um conjunto E é dar um conjunto Θ de partes de E gozando das seguintes propriedades (Honig, 1976):

- e1) A reunião $O = \bigcup_{i \in I} O_i$ de uma família qualquer O_i de elementos de Θ ainda é um elemento de Θ ;
- e2) A intersecção $O = O_1 \cap O_2$ de dois conjuntos elementos de Θ ainda é um conjunto de Θ ;
- e3) $E \in \Theta$.

Nestas condições, o par (E, Θ) é chamado *espaço topológico*. Os elementos de E são denominados genericamente de *pontos* e os conjuntos elementos de Θ são denominados conjuntos *abertos*.

Lembremos que $\bigcup_{i \in \emptyset} O_i = \emptyset$, e portanto o conjunto vazio é aberto.

Dado um ponto $p \in E$, chamamos uma *vizinhança* de p a qualquer conjunto V_p que contém um conjunto aberto O com $p \in O$.

Um espaço topológico (E, Θ) é denominado *separado* ou de *Hausdorff* se quaisquer pontos distintos $p \neq q$ têm vizinhanças V_p e V_q sem ponto comum.

B.1.2 Variedades Diferenciáveis

Seja M um espaço de Hausdorff como definido acima. Uma estrutura diferenciável de dimensão n em M é uma família de aplicações biunívocas $\varphi_\alpha : U_\alpha \rightarrow \mathbb{R}^n$ onde U_α é um aberto de M e $\varphi_\alpha(U_\alpha)$ é um aberto de \mathbb{R}^n , tais que:

- 1) $\bigcup_{\alpha} U_\alpha = M$.
- 2) Para todo par α, β , a aplicação $\varphi_\beta \circ \varphi_\alpha^{-1} : \varphi_\alpha(U_\alpha \cap U_\beta) \rightarrow \varphi_\beta(U_\alpha \cap U_\beta)$ é diferenciável.
- 3) A família $\{(U_\alpha, \varphi_\alpha)\}$ é maximal relativamente às condições (1) e (2).

Uma *variedade diferenciável* de dimensão m é um espaço de Hausdorff com uma estrutura diferenciável de dimensão m (Olver, 1986).

Seja $p \in U_\alpha$ e $\varphi_\alpha(p) = (x_1(p), \dots, x_m(p))$. Então, o conjunto U_α é chamado uma *vizinhança coordenada* de p e $x_i(p)$ são chamados *coordenadas locais* de p . O par $(U_\alpha, \varphi_\alpha)$ é chamado uma parametrização (ou um *sistema de coordenadas*) de M em p .

Uma *estrutura analítica* de dimensão m é definida como acima bastando substituir em (2) *diferenciável* por *suave*. Neste caso, M é chamada uma variedade suave.

B.1.3 Definição de Grupo

Definição: Um *grupo* é um conjunto G não vazio juntamente com uma lei de composição interna em G denotada por (\cdot) (Olver, 1986):

$$(g, h) \mapsto g \cdot h, \text{ onde } g, h, g \cdot h \in G,$$

genericamente chamada *multiplicação*, com as seguintes propriedades:

(1) Associativa. Se $g, h, k \in G$, então:

$$g \cdot (h \cdot k) = (g \cdot h) \cdot k.$$

(2) Elemento Identidade. Existe um único elemento $e \in G$, chamado *elemento identidade*, tal que a propriedade

$$e \cdot g = g \cdot e = g,$$

para todo $g \in G$.

(3) Inverso. Para todo $g \in G$ existe um elemento denotado por g^{-1} , denominado seu *inverso*, com a propriedade:

$$g \cdot g^{-1} = e.$$

Neste caso, costuma-se dizer que o par (G, \cdot) é um grupo, ou simplesmente que G é um grupo para os casos onde a operação de grupo esteja claramente sub-entendida.

Exemplos:

(a) Seja $G = \mathbb{Z}$ o conjunto dos números inteiros com operação de adição. As propriedades acima são naturalmente satisfeitas, sendo o *inverso* de um inteiro x seu oposto $-x$. O mesmo se aplica para $G = \mathbb{R}$ também com a adição.

(b) Seja $G = GL(n, \mathbb{R})$ o conjunto das matrizes reais e inversíveis com a operação de multiplicação. Neste caso, a propriedade associativa é satisfeita, o elemento identidade é exatamente a matriz identidade I usual e o inverso de uma matriz A é dado pela matriz inversa usual A^{-1} .

B.1.4 Definição de Grupos de Lie

Um *grupo de Lie r -paramétrico* é um grupo G que é também uma variedade suave de dimensão r tal que ambas as operações de grupo (Olver, 1986):

$$m : G \times G \longrightarrow G, \quad m(g, h) = g \cdot h, \quad g, h \in G,$$

e a inversão

$$i : G \rightarrow G, \quad i(g) = g^{-1}, \quad g \in G,$$

são aplicações suaves entre variedades.

Exemplos:

(a) Seja $G = \mathfrak{R}^r$ com a estrutura diferenciável trivial dada pela aplicação identidade e seja a operação de adição usual em \mathfrak{R}^r identificando o inverso de um vetor x com seu oposto $-x$. Uma vez que, tanto a adição de vetores quanto a inversão são operações suaves, \mathfrak{R}^r é um exemplo de um grupo de Lie r -paramétrico.

(b) Consideremos agora o conjunto $G = SO(2)$ das rotações no plano:

$$SO(2) = \left\{ \begin{pmatrix} \cos(\theta) & -\text{sen}(\theta) \\ \text{sen}(\theta) & \cos(\theta) \end{pmatrix}; 0 \leq \theta < 2\pi \right\},$$

com a operação usual de multiplicação de matrizes. Observe que as propriedades de grupo são satisfeitas e portanto $SO(2)$ é um grupo. A estrutura diferenciável pode ser obtida pela observação de que podemos identificar $SO(2)$ com o círculo unitário (Olver, 1986):

$$S^1 = \{(\cos(\theta), \text{sen}(\theta)); 0 \leq \theta < 2\pi\}.$$

Assim, $SO(2)$ é um grupo de Lie 1-paramétrico.

B.1.5 Grupo de Transformações Locais

Seja M uma variedade suave. Um grupo local de transformações atuando em M é dado por um grupo de Lie G , um subconjunto aberto U , com

$$\{e\} \times M \subset U \subset G \times M,$$

o qual é o domínio de definição da *ação do grupo*, e uma aplicação suave $\Psi : U \rightarrow M$ com as seguintes propriedades:

(a) Se $(h, x) \in U$, $(g, \Psi(h, x)) \in U$, e também $(g \cdot h, x) \in U$, então:

$$\Psi(g, \Psi(h, x)) = \Psi(g \cdot h, x).$$

(b) Para todo $x \in M$.

$$\Psi(e, x) = x.$$

(c) Se $(g, x) \in U$, então $(g^{-1}, \Psi(g, x)) \in U$ e:

$$\Psi(g^{-1}, \Psi(g, x)) = x$$

Note que, para o caso de $M = U$ a propriedade (c) segue diretamente de (a) e (b).

B.1.6 Órbitas

Na definição abaixo, o termo *subvariedade* de uma variedade diferenciável M deve ser entendido como um subconjunto de M que também é uma variedade diferenciável (Olver, 1986).

Uma órbita de um grupo de transformações locais em M é o *menor* subconjunto (não vazio) invariante pela ação das transformações. Mais precisamente, um subconjunto $\tilde{O} \subset M$ é uma *órbita* desde que ele satisfaça as seguintes propriedades:

- (a) Se $x \in \tilde{O}$, $g \in G$ e $g \cdot x$ está definido, então $g \cdot x \in \tilde{O}$,
- (b) Se $\tilde{U} \subset \tilde{O}$, e \tilde{U} satisfaz a propriedade (a) então ou $\tilde{U} = \tilde{O}$ ou \tilde{U} é vazio.

Seja G um grupo local de transformações atuando sobre uma variedade M . Então dizemos que:

(a) O grupo G atua *semi-regularmente* (ação semi-regular) se todas as órbitas de G são da mesma dimensão das sub-variedades de M ,

(b) O grupo G atua *regularmente* se a ação é semi-regular e adicionalmente, para todo $x \in M$ existem vizinhanças arbitrariamente pequenas U de x com a propriedade de que cada órbita de G intercepta U em um subconjunto conexo por partes.

B.1.7 Campos Vetoriais em Variedades

Um conceito fundamental na teoria de grupos de Lie e grupo de transformações são as *transformações infinitesimais*. Para apresentar esta idéia, necessitamos primeiramente da noção de campo vetorial em uma variedade. Iniciemos com a definição de vetor tangente.

Seja C uma curva suave em uma variedade M , parametrizada por $\phi : I \subset \mathbb{R} \rightarrow M$. Em coordenadas locais $x = (x^1, \dots, x^m)$ a curva C é dada por m funções suaves $x \circ \phi(\varepsilon) = (\phi^1(\varepsilon), \dots, \phi^m(\varepsilon))$ da variável real ε . Em cada ponto $p = \phi(\varepsilon)$ de C a curva tem um *vetor tangente*, dado no sistema de coordenadas local pela derivada:

$$\frac{d(x \circ \phi)}{d\varepsilon} = \left(\frac{d\phi^1}{d\varepsilon}, \frac{d\phi^2}{d\varepsilon}, \dots, \frac{d\phi^m}{d\varepsilon} \right) \quad (\text{B.1})$$

É importante observar que o conceito de vetor tangente é independente do sistema local de coordenadas usado numa vizinhança de x (Olver, 1986; Carmo, 1979).

O conjunto de todos os vetores tangentes a todas as possíveis curvas em M que passam por p é chamado *espaço tangente a M em p* , sendo denotado por $TM|_p$.

Se M é uma variedade diferenciável suave de dimensão m então $TM|_p$ é um espaço vetorial de dimensão m com:

$$\left\{ \frac{\partial}{\partial x^1}, \dots, \frac{\partial}{\partial x^m} \right\}, \quad (\text{B.2})$$

fornecendo uma base para $TM|_p$, a qual está representada acima no sistema de coordenadas $x = (x^1, \dots, x^m)$.

Para distinguir entre vetores tangentes e a noção de campo de vetores dada a seguir, adotaremos a seguinte notação:

$$v|_{p=\phi(\varepsilon)} = \dot{\phi}^1(\varepsilon) \frac{\partial}{\partial x^1} + \dot{\phi}^2(\varepsilon) \frac{\partial}{\partial x^2} + \dots + \dot{\phi}^m(\varepsilon) \frac{\partial}{\partial x^m}, \quad (\text{B.3})$$

para o vetor tangente a C em $p = \phi(\varepsilon)$.

A coleção de todos os espaços tangentes correspondentes a todos os pontos $p \in M$ é chamado o *fibrado tangente* de M , sendo denotado por

$$TM = \bigcup_{p \in M} TM|_p. \quad (\text{B.4})$$

Um *campo vetorial* v em M fornece um vetor tangente $v|_p \in TM|_p$ a cada ponto $p \in M$, com $v|_p$ variando suavemente de um ponto para outro. Em coordenadas locais (x^1, \dots, x^m) , um campo vetorial é dado por:

$$v(x) = \xi^1(x) \frac{\partial}{\partial x^1} + \xi^2(x) \frac{\partial}{\partial x^2} + \dots + \xi^m(x) \frac{\partial}{\partial x^m},$$

onde $\xi^i(x)$ é uma função suave de x .

Uma *curva integral* de um campo vetorial v é uma curva suave e paramétrica $\phi(\varepsilon)$ cujo vetor tangente em qualquer ponto coincide com o valor de v no mesmo ponto; isto é:

$$\dot{\phi}(\varepsilon) = v|_{\phi(\varepsilon)},$$

para todo ε . Em coordenadas locais, $x \circ \phi(\varepsilon) = (\phi^1(\varepsilon), \dots, \phi^m(\varepsilon))$ deve ser a solução do sistema autônomo de equações diferenciais ordinárias

$$\frac{dx^i}{d\varepsilon} = \xi^i(x), \quad i = 1, \dots, m, \quad (\text{B.5})$$

onde $\xi^i(x)$ são as coordenadas de v em x .

Para $\xi^i(x)$ suaves, os teoremas padrões de existência e unicidade para sistemas de equações diferenciais ordinárias garantem que existe uma única solução para (B.5) dada a condição inicial (Sotomayor, 1979):

$$\phi(0) = x_0.$$

Isto implica a existência de uma única curva integral maximal $\phi : I \rightarrow M$ passando por um dado ponto $\phi(0) \in M$.

B.1.8 Fluxos e Geradores Infinitesimais

Se v é um campo vetorial, nós denotamos a curva integral maximal de (B.5), passando por x em M , por $\Psi(\varepsilon, x)$ e chamamos Ψ o *fluxo* gerado por v (Olver, 1986). Assim, para cada $x \in M$, e ε em algum intervalo I_x contendo 0, $\Psi(\varepsilon, x)$ será um ponto na curva integral passando através de x em M . O fluxo de um campo vetorial tem as seguintes propriedades (Olver, 1986):

(a) $\Psi(\delta, \Psi(\varepsilon, x)) = \Psi(\delta + \varepsilon, x)$, $x \in M$, para todos $\varepsilon, \delta \in \mathfrak{R}$ tais que ambos os lados da equação estejam definidos,

(b) $\Psi(0, x) = x$, para qualquer $x \in M$,

(c)

$$\frac{d}{d\varepsilon} \Psi(\varepsilon, x) = v|_{\Psi(\varepsilon, x)}$$

As duas primeiras propriedades significam que o fluxo gerado por um campo vetorial comporta-se como um grupo 1-parametrizado de transformações atuando em M . A terceira propriedade simplesmente diz que v é tangente à curva integral $\Psi(\varepsilon, x)$ que passa por um dado $x \in M$. O campo vetorial v é chamado o *gerador*

infinitesimal da ação do grupo assim definido, uma vez que pelo Teorema de Taylor, usando coordenadas locais, temos:

$$\Psi(\varepsilon, x) = x + \varepsilon\xi(x) + O(\varepsilon^2), \quad (\text{B.6})$$

onde:

$$\xi(x) = \frac{d\Psi(\varepsilon = 0, x)}{d\varepsilon}.$$

As órbitas de um grupo 1-paramétrico são as curvas integrais maximais do campo vetorial v .

Intensamente, se $\Phi(\varepsilon, x)$ é um grupo 1-paramétrico de transformações atuando em M , então seu gerador infinitesimal v é dado por:

$$v|_{x=} = \frac{d\Phi(\varepsilon = 0, x)}{d\varepsilon}. \quad (\text{B.7})$$

A existência e unicidade da solução de (B.5), para o campo vetorial v assim definido, garante que o fluxo gerado coincide com a ação (local) de $\Phi(\varepsilon, x)$ em M no domínio comum de definição. Assim, existe uma correspondência um-a-um entre grupos 1-paramétricos (locais) de transformações e seus geradores infinitesimais (Olver, 1986).

B.1.9 Teoria de Invariantes

A teoria que segue é útil no estudo de grupos de simetria para sistemas de equações algébricas, isto é, sistemas de equações da forma:

$$F_j(x) = 0, \quad j = 1, \dots, l \quad (\text{B.8})$$

onde $F_1(x), \dots, F_l(x)$ são funções suaves com valores reais definidas para todo x em uma variedade M . Uma solução de um sistema deste tipo é um ponto x tal que $F_j(x) = 0$, para $j = 1, \dots, l$. Um grupo de simetria do sistema (B.8) é um grupo local de transformações G atuando em M tal que G transforma soluções do sistema em outras soluções do mesmo.

Do ponto de vista dos problemas em visão computacional, nosso objetivo com os conceitos que seguem é primeiramente dar a base teórica para a apresentação do Teorema dos π , usado em (Florack *et al.*, 1992) para a demonstração da unicidade do Kernel Gaussiano sob as restrições de simetria discutidas no Capítulo 2.

Em seguida, estes conceitos aparecem de maneira mais prática, no contexto de determinação de invariantes, seguindo o trabalho de (Gool *et al.*, 1995).

Subconjuntos invariantes

Seja G um grupo local de transformações em uma variedade M . Um subconjunto $\mathfrak{S} \subset M$ é chamado G -invariante e G é chamado um grupo de simetria de \mathfrak{S} , se para qualquer $x \in \mathfrak{S}$ e $g \in G$ tais que $g \cdot x$ esteja definido tivermos $g \cdot x \in \mathfrak{S}$.

Funções Invariantes

Seja G um grupo de transformações locais atuando em uma variedade M . Uma função $F : M \rightarrow N$, onde N é outra variedade, é chamada uma função G -invariante se para todo $x \in M$ e todo $g \in G$ tal que $g \cdot x$ esteja definido tivermos:

$$F(g \cdot x) = F(x).$$

Invariância Local

Seja G um grupo local de transformações atuando em uma variedade M . Um subconjunto $\mathfrak{S} \subset M$ é chamado *localmente G -invariante* se para todo $x \in \mathfrak{S}$ existe uma vizinhança $\tilde{G}_x \subset G$ da identidade em G tal que $g \cdot x \in \mathfrak{S}$ para todo $g \in \tilde{G}_x$.

Uma função suave $F : U \rightarrow N$ onde U é um subconjunto aberto de M , é chamada *localmente G -invariante* se para todo $x \in U$ existe uma vizinhança $\tilde{G}_x \subset G_x$ do elemento identidade e em G tal que $F(g \cdot x) = F(x)$ para todo $g \in \tilde{G}_x$.

F é chamada *globalmente invariante* se $F(g \cdot x) = F(x)$ para todo $x \in U$, $g \in G$ tal que $g \cdot x \in U$.

Invariantes e Dependência Funcional

Sejam $\zeta^1(x), \dots, \zeta^k(x)$ funções suaves com valores reais, definidas em uma variedade M . Então:

(a) ζ^1, \dots, ζ^k são chamadas *funcionalmente dependentes* se para todo $x \in M$ existe uma vizinhança U de x e uma função suave com valores reais $F(z^1, \dots, z^k)$, não identicamente nula em qualquer subconjunto aberto de \mathfrak{R}^k , tal que:

$$F(\zeta^1(x), \dots, \zeta^k(x)) = 0, \tag{B.9}$$

para todo $x \in U$.

(b) ζ^1, \dots, ζ^k são chamadas *funcionalmente independentes* se elas não são funcionalmente dependentes quando restritas a qualquer subconjunto aberto $U \subset M$; em outras palavras, se $F(z^1, \dots, z^k)$ é tal que (B.9) vale para todo x em algum subconjunto aberto $U \subset M$, então $F(z^1, \dots, z^k) \equiv 0$ para todo z em algum subconjunto aberto de \mathfrak{R}^k (o qual está contido na imagem de U).

Teorema: Seja G atuando semi-regularmente (seção B.1.6) em uma variedade com dimensão m . Seja s a dimensão das órbitas correspondentes. Se $x_0 \in M$, então existem precisamente $m - s$ invariantes locais, funcionalmente independentes, $\zeta^1(x), \dots, \zeta^{m-s}(x)$ definidos em uma vizinhança de x_0 . Além disso, qualquer outro invariante definido próximo de x_0 é da forma

$$\zeta(x) = F(\zeta^1(x), \dots, \zeta^{m-s}(x)), \quad (\text{B.10})$$

para alguma função suave F . Se a ação de G é regular, então os invariantes podem ser tomados como sendo globalmente invariantes em uma vizinhança de x_0 .

Demonstração: Ver (Olver, 1986).

B.2 Análise Dimensional

No caso de grupos de transformações de escala, a Teoria acima é a base do Teorema dos π o qual é fundamental para os métodos em análise dimensional.

Em qualquer problema físico (e também em processamento de imagens) existem quantidades fundamentais, tais como comprimento, frequência, etc, os quais podem ser escaladas independentemente umas das outras. Sejam (z^1, z^2, \dots, z^m) estas quantidades. Assim o grupo de *transformações de escala* age sobre estas quantidades de acordo com:

$$(z^1, z^2, \dots, z^r) \longrightarrow (\lambda_1 z^1, \lambda_2 z^2, \dots, \lambda_r z^r). \quad (\text{B.11})$$

onde os fatores de escala $\lambda = (\lambda_1, \lambda_2, \dots, \lambda_r)$ são números reais positivos arbitrários. Assim o correspondente grupo é exatamente o produto cartesiano de r cópias do grupo multiplicativo \mathfrak{R}^+ dos números reais positivos (Olver, 1986).

Quantidades derivadas, tais como velocidade, luminância da imagem como função das coordenadas espaciais, etc; são afetadas pela transformação das quantidades fundamentais dada por (B.11) acima. Se chamarmos estas quantidades por (x^1, x^2, \dots, x^m) e, assumindo homogeneidade dimensional (Martins *et al.*, 1979), a ação do grupo de escalas nas quantidades derivadas toma a forma:

$$\lambda \cdot (x^1, x^2, \dots, x^m) = (\lambda_1^{\alpha_{11}} \lambda_2^{\alpha_{21}} \dots \lambda_r^{\alpha_{r1}} x^1, \dots, \lambda_1^{\alpha_{1m}} \lambda_2^{\alpha_{2m}} \dots \lambda_r^{\alpha_{rm}} x^m). \quad (\text{B.12})$$

onde os expoentes $\alpha_{i,j} \in \mathfrak{R}^+$, $i = 1, \dots, r$, $j = 1, \dots, m$, são prescritos pela dependência das quantidades x^j com relação às fundamentais z^i .

Se uma grandeza derivada permanece inalterada quando as fundamentais são re-escaladas segundo (B.11) então esta quantidade é dita *adimensional*.

É possível também termos funções $F(x^1, x^2, \dots, x^m) = 0$ invariantes pela transformação de escala (B.11), as quais são denominadas *unit-free*.

Teorema dos π : Sejam z^1, z^2, \dots, z^r as quantidades fundamentais as quais são escaladas independentemente de acordo com $z^i \rightarrow \lambda_i z^i$. Sejam x^1, x^2, \dots, x^m as quantidades derivadas as quais escalam de acordo com (B.12) para alguma matriz de constantes $A = (\alpha_{i,j})_{r \times m}$. Seja s o posto de A . Então existem $m - s$ quantidades adimensionais:

$$\pi^k = (x^1)^{\beta_{1k}} (x^2)^{\beta_{2k}} \dots (x^m)^{\beta_{mk}}, \quad \beta_{ik} \in \mathfrak{R}^+; \quad k = 1, \dots, m - s; \quad i = 1, \dots, m, \quad (\text{B.13})$$

com a propriedade que qualquer outra quantidade adimensional pode ser escrita como uma função dos $\pi^1, \pi^2, \dots, \pi^{m-s}$. Se $F(x^1, x^2, \dots, x^m) = 0$ é *unit-free*, então existe uma relação equivalente $\tilde{F}(\pi^1, \pi^2, \dots, \pi^{m-s}) = 0$ a qual pode ser expressa somente em termos das quantidades adimensionais em (B.13):

$$\tilde{F}(\pi^1, \pi^2, \dots, \pi^{m-s}) = 0.$$

Demonstração: Ver (Olver, 1986).

Calculando Invariantes

Vejamos como usar a teoria de Lie para calcular invariantes. O procedimento descrito abaixo poderia ser compactado através da derivada de Lie. No entanto, uma vez que a proposta desta seção é efetivamente o cálculo dos invariantes faremos uma apresentação menos formal seguindo aquela dada em (Gool *et al.*, 1995).

Suponha então que as grandezas de interesse estão agrupadas no vetor coluna $\vec{m} = (m_1, m_2, \dots, m_{n_m})^T$. Vamos assumir que existe um grupo de Lie G n_p - *paramétrico* cuja ação no espaço destas grandezas modela as mudanças de interesse.

Formalmente,

$$\vec{m}' = g \cdot \vec{m}, \quad (\text{B.14})$$

onde \vec{m}' denota simplesmente o vetor transformado (não confundir com derivada).

Um invariante f é então uma função $f(\vec{m})$ do vetor de medidas cujo valor não é afetado pela ação do grupo G :

$$f(\vec{m}') = f(g \cdot \vec{m}) = f(\vec{m}), \quad \text{para todo } g \in G. \quad (\text{B.15})$$

Uma vez que G é um grupo de Lie temos que g pode ser expressa como uma função (suave) $g = g(p_1, p_2, \dots, p_{n_p})$ dos parâmetros p_i do grupo.

Se variarmos suavemente o vetor de parâmetros teremos:

$$g(t) = g(p_1(t), p_2(t), \dots, p_{n_p}(t)). \quad (\text{B.16})$$

Substituindo (B.16) na expressão (B.15) é imediato que:

$$\frac{d}{dt} f(\vec{m}'(t)) = 0. \quad (\text{B.17})$$

pois do contrário f não seria um invariante.

Pela regra de Cadeia:

$$\sum_{i=1}^{n_m} \sum_{j=1}^{n_p} \frac{\partial f}{\partial m'_i} \frac{\partial m'_i}{\partial p_j} \frac{dp_j}{dt} = 0. \quad (\text{B.18})$$

Duas observações importantes devem ser feitas neste ponto. Uma vez que a equação diferencial (B.18) deve ser válida para qualquer função $g(t) = g(p_1(t), p_2(t), \dots, p_{n_p}(t))$, e portanto para todo $\frac{dp_j}{dt}$, é necessário que:

$$\sum_{i=1}^{n_m} \frac{\partial f}{\partial m'_i} \frac{\partial m'_i}{\partial p_j} = 0, \text{ para todo } j = 1, 2, \dots, n_p. \quad (\text{B.19})$$

A segunda observação vem de uma generalização do resultado B.6 segundo a qual os operadores diferenciais:

$$\sum_{i=1}^{n_m} \frac{\partial m'_i}{\partial p_j} \Big|_{p=\epsilon} \left(\frac{\partial}{\partial m_i} \right), \quad (\text{B.20})$$

são *geradores infinitesimais* do grupo de Lie G (Olver, 1986). Assim, basta resolver a equação (B.19) para $t = 0$, para obtermos a condição necessária e suficiente para que f seja um invariante. Assim, temos finalmente o seguinte sistema de equações diferenciais:

$$\sum_{i=1}^{n_m} \frac{\partial m'_i}{\partial p_j} \Big|_{p=\epsilon} \left(\frac{\partial f}{\partial m_i} \right) = 0, j = 1, 2, \dots, n_p. \quad (\text{B.21})$$

As soluções deste sistema fornecem os invariantes procurados. O teorema do final da seção B.1.9 fornece número de soluções funcionalmente independentes deste sistema, uma vez conhecida a dimensão das órbitas do grupo G .

B.3 Aplicações

Um exemplo simples pode esclarecer os conceitos acima. Tomemos então o caso do movimento rígido no plano. O conjunto de transformações em questão, denotado por $M(2)$, é caracterizado por três parâmetros: o ângulo de rotação θ e as coordenadas do vetor de translação dadas por (t_1, t_2) .

Sejam duas transformações $A, B \in M(2)$. Usando coordenadas homogêneas, estas transformações podem ser escritas na forma:

$$A = \begin{pmatrix} \cos \theta & -\text{sen} \theta & t_1 \\ \text{sen} \theta & \cos \theta & t_2 \\ 0 & 0 & 1 \end{pmatrix}, \quad B = \begin{pmatrix} \cos \phi & -\text{sen} \phi & t_3 \\ \text{sen} \phi & \cos \phi & t_4 \\ 0 & 0 & 1 \end{pmatrix}.$$

O conjunto $M(2)$ munido da operação de composição de transformações definida pelo produto (usual) $C = A \cdot B$ é um Grupo de Lie de dimensão 3. Sua ação sobre o \mathfrak{R}^2 (seção B.1.5) é definida por uma aplicação:

$$\Psi : M(2) \times \mathfrak{R}^2 \rightarrow \mathfrak{R}^2;$$

dada pelas seguinte expressão:

$$\begin{aligned} x'_1 &= x_1 \cos(\theta) - y_1 \text{sen}(\theta) + t_1 \\ y'_1 &= x_1 \text{sen}(\theta) + y_1 \cos(\theta) + t_2 \end{aligned} \tag{B.22}$$

Observemos no entanto, que ao considerarmos movimentos rígidos estamos pensando em transformações que preservam a distância entre dois pontos $(x_1, y_1), (x_2, y_2) \in \mathfrak{R}^2$. Assim, um vetor de grandezas mais completo seria $\vec{m} = (x_1, y_1, x_2, y_2)^T$, o qual será transformado no vetor $\vec{m}' = (x'_1, y'_1, x'_2, y'_2)^T$ dado por:

$$\begin{aligned} x'_1 &= x_1 \cos(\theta) - y_1 \text{sen}(\theta) + t_1 \\ y'_1 &= x_1 \text{sen}(\theta) + y_1 \cos(\theta) + t_2 \\ x'_2 &= x_2 \cos(\theta) - y_2 \text{sen}(\theta) + t_1 \\ y'_2 &= x_2 \text{sen}(\theta) + y_2 \cos(\theta) + t_2 \end{aligned} \tag{B.23}$$

Formalmente, estamos procurando invariantes para o grupo $M(2)$ atuando sobre $\mathfrak{R}^2 \times \mathfrak{R}^2$; ou seja, funções reais $f(x_1, y_1, x_2, y_2)$ tais que:

$$f(x'_1, y'_1, x'_2, y'_2) = f(x_1, y_1, x_2, y_2) \tag{B.24}$$

O sistema de equações B.21 toma a seguinte forma:

$$\begin{aligned} -y_1 \frac{\partial f}{\partial x_1} + x_1 \frac{\partial f}{\partial y_1} - y_2 \frac{\partial f}{\partial x_2} + x_2 \frac{\partial f}{\partial y_2} &= 0, \\ \frac{\partial f}{\partial x_1} + \frac{\partial f}{\partial x_2} &= 0, \\ \frac{\partial f}{\partial y_1} + \frac{\partial f}{\partial y_2} &= 0. \end{aligned} \tag{B.25}$$

a qual (como já esperado) possui solução analítica dada por:

$$f(x_1, y_1, x_2, y_2) = F((x_1 - x_2)^2 + (y_1 - y_2)^2), \tag{B.26}$$

onde $F : \mathbb{R}^2 \rightarrow \mathbb{R}$ é uma função diferenciável qualquer.

Se tivéssemos usado como vetor de grandezas apenas as coordenadas (x_1, y_1) obteríamos apenas a função $f = (x_1)^2 + (y_1)^2$ como único invariante funcionalmente independente. A quantidade de coordenadas do vetor de grandezas é uma escolha que depende primeiramente da aplicação desejada.

Com este exemplo simples podemos apresentar o esquema da Figura B.1, o qual identifica os principais passos para a aplicação prática da teoria de Lie no cálculo de invariantes:

Apesar de matematicamente rigorosa, a aplicação da Teoria de Lie segundo os passos indicados no esquema não se faz, em geral, de maneira tão direta como no exemplo apresentado.

Primeiramente, a solução do sistema de equações diferenciais correspondente pode exigir o conhecimento de resultados específicos sobre Teoria de Lie em equações diferenciais (Olver, 1986).

Além disso, os invariantes encontrados podem envolver derivadas de alta ordem, o que pode trazer problemas de estabilidade numérica quando aplicados.

Soma-se ainda a dificuldade inicial de definir claramente o Grupo de Lie G a ser usado, uma vez que o conjunto de transformações de interesse pode não ser um grupo com relação à composição de transformações considerada.

Em (Gool *et al.*, 1995) estes problemas são analisados e alguns exemplos mais elaborados são estudados no sentido de apresentar algumas diretrizes para resolvê-los e o potencial da teoria acima na prática.

Um destes exemplos pode ser um caminho interessante para usarmos Grupos de Lie no contexto das snakes, ou mais especificamente, das AI-Snakes da seção 5.1.3. Nesta formulação considera-se invariância por transformações afins mas não se cogita a possibilidade de reparametrizar a snake durante sua evolução. Tal procedimento poderia ser necessário dependendo da proximidade da curva inicial e da solução procurada.

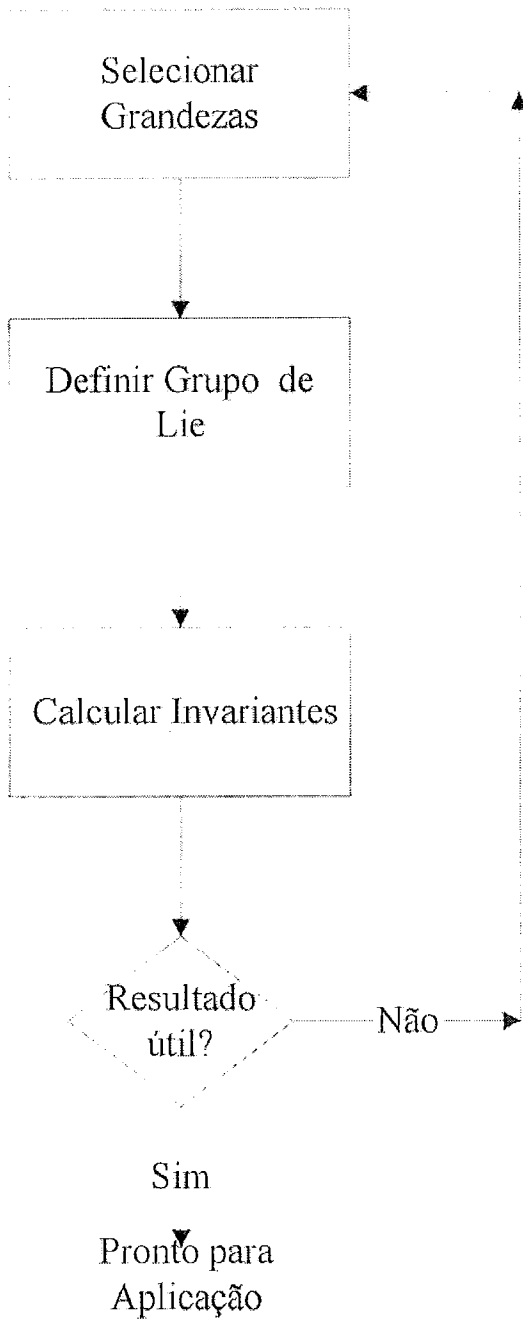


Figura B.1: Esquema com principais etapas para aplicação da Teoria de Lie em visão computacional.

Para facilitar a apresentação, vamos retomar a formulação contínua de modelo original da seção 3.1.

Seja então uma curva parametrizada $c(s)$, tal que:

$$c : [0, 1] \rightarrow \mathbb{R}^2$$

Observemos que o operador de energia da snake dado pela expressão (3.6) depende não apenas das coordenadas espaciais x, y dos pontos da curva mas também de derivadas destas com relação ao parâmetro s . No caso do modelo original temos derivadas até segunda ordem. Em modelos vinculados usados para tracking podemos ter derivadas de ordem mais elevada (Black & Yuille, 1993). Façamos então um desenvolvimento geral.

Ao reparametrizarmos a curva segundo uma função $s = \phi(\tau)$ as derivadas da coordenada x (e analogamente para a coordenada y) são transformadas segundo a expressão:

$$\begin{pmatrix} \frac{dx}{d\tau} \\ \frac{d^2x}{d\tau^2} \\ \frac{d^3x}{d\tau^3} \\ \vdots \\ \frac{d^m x}{d\tau^m} \end{pmatrix} = \begin{pmatrix} \frac{d\phi}{d\tau} & 0 & 0 & \dots & 0 \\ \frac{d^2\phi}{d\tau^2} & \left(\frac{d\phi}{d\tau}\right)^2 & 0 & \dots & 0 \\ \frac{d^3\phi}{d\tau^3} & 3\frac{d\phi}{d\tau}\frac{d^2\phi}{d\tau^2} & \left(\frac{d\phi}{d\tau}\right)^3 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{d^m\phi}{d\tau^m} & \dots & \dots & \dots & \left(\frac{d\phi}{d\tau}\right)^m \end{pmatrix} \begin{pmatrix} \frac{dx}{dt} \\ \frac{d^2x}{dt^2} \\ \frac{d^3x}{dt^3} \\ \vdots \\ \frac{d^m x}{dt^m} \end{pmatrix} \quad (\text{B.27})$$

onde m é a derivada de ordem mais alta.

Supondo que ϕ seja monótona, pode ser mostrado que vetores contendo subseqüentes derivadas de ϕ (iniciando com a derivada de ordem zero) transformam-se linearmente sob reparametrização, segundo a matriz de transformação abaixo:

$$\begin{pmatrix} 1 & 0 & 0 & 0 & \dots & 0 \\ 0 & a & 0 & 0 & \dots & 0 \\ 0 & b & a^2 & 0 & \dots & 0 \\ 0 & c & 3ab & a^3 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \dots & \dots & \dots & \dots & a^m \end{pmatrix} \quad (\text{B.28})$$

onde $a = \frac{d\phi}{d\tau}$, $b = \frac{d^2\phi}{d\tau^2}$, $c = \frac{d^3\phi}{d\tau^3}$, etc.

O grupo assim definido é um Grupo de Lie com dimensão $(m + 1)$.

Voltando ao caso das AI-Snakes, teríamos dois conjuntos de transformações a considerar: o conjunto das transformações afins e o grupo de reparametrização definido pela matriz (B.28) acima. Contudo, pela linearidade das transformações afins, segue imediatamente que estes grupos comutam entre si.

Do ponto de vista da teoria de invariantes, isto implica que o conjunto de invariantes procurado é a intersecção dos conjuntos de invariantes para cada grupo separadamente.

Se considerarmos transformações afins com determinante $a_{11}a_{22} - a_{12}a_{21} = 1$, encontramos em (Gool *et al.* , 1995) o seguinte invariante comum:

$$\frac{12 |x^{(1)}x^{(2)}| |x^{(2)}x^{(3)}| - 5 |x^{(1)}x^{(3)}| + 3 |x^{(1)}x^{(2)}| |x^{(1)}x^{(4)}|}{9 |x^{(1)}x^{(2)}|^{8/3}} \quad (\text{B.29})$$

o qual é denominado *curvatura afim* sendo conhecida em geometria diferencial (as barras verticais indicam determinante da matriz formada pelos vetores coluna). Invariantes deste tipo são denominados *diferenciais* uma vez que envolvem apenas derivadas das funções coordenadas em um ponto.

Voltando às snakes, o primeiro ponto a ser levantado é quanto à possibilidade de desenvolver um modelo de contorno ativo, análogo às AI-Snakes, mas agora contínuo em lugar de discreto e que leve em consideração a possibilidade de reparametrizar a snake durante sua evolução. Para tal, teremos que considerar os efeitos da mudança de parâmetros juntamente com a invariância por transformações afins, que é precisamente o ponto central do desenvolvimento acima.

Como já apontado na seção 4.4, não há referências de aplicações da Teoria de Lie em modelos de contorno ativo. Somente futuras pesquisas poderão mostrar se o desenvolvimento acima é de fato útil na prática.

Em visão computacional, contudo, os Grupos de Lie vêm ganhando atenção, particularmente com relação a métodos multi-escala e invariância.

Como já destacado acima, o Teorema dos π , juntamente com certas restrições de simetria conduz à unicidade do Kernel Gaussiano para espaços de escala lineares (Florack *et al.* , 1992).

Este fato é o ponto de partida para um modelo geométrico multi-escala em visão (Florack *et al.* , 1992; Hendee & Wells, 1997). A idéia central destes trabalhos é usar uma formulação independente de coordenadas e que permita uma interpretação geométrica do espaço de escalas correspondente ao Kernel Gaussiano. Desta forma pode-se passar naturalmente para espaços de escala não lineares via conceitos bem conhecidos em geometria Riemanniana (Florack *et al.* , 1995).

Finalizando, os trabalhos usando Grupos de Lie e seu ferramental matemático (grupos, variedades e transformações) vêm mostrando o potencial destes conceitos em visão computacional e indicando que esta é uma área que pode trazer bons

resultados tanto a nível de generalizações (Florack *et al.* , 1995), como de solução de problemas que possam ser formulados envolvendo conceitos de invariância (Gool *et al.* , 1995).

Apêndice C

Variedades Lineares por Partes e T-Superfícies

Neste capítulo fornecemos os elementos teóricos necessários para estender o método das T-Snakes da seção 5.3.2 para superfícies gerando o método das T-Superfícies (McInerney & Terzopoulos, 1997; McInerney, 1997). Com isto, ficamos em condições de propor a extensão do Dual-T-Snakes para 3D.

C.1 Definições Básicas

Seja E um espaço Euclidiano de dimensão finita.

Seja $\sigma \subset E$. Então:

- (1) $co(\sigma)$ denota o fecho convexo de σ ;
- (2) $aff(\sigma)$ denota o fecho afim σ ;
- (3) $tng(\sigma) = aff(\sigma) - \sigma$ denota o espaço tangente de σ o qual é o espaço linear obtido pela translação de $aff(\sigma)$ para a origem;
- (4) $int(\sigma)$ denota o interior relativo de σ com respeito ao espaço $aff(\sigma)$;
- (5) $\partial(\sigma)$ denota a fronteira relativa de σ com respeito ao espaço $aff(\sigma)$.

Células: Se $u \in E$, $u \neq 0$ e $\gamma \in \mathfrak{R}$, então o conjunto $\{x \in E \mid u^*x \geq \gamma\}$ é chamado um *semi-espaço* e sua fronteira $\{x \in E \mid u^*x = \gamma\}$ é chamado um hiperplano. Uma intersecção finita de semi-espaços é chamado um poliedro ou célula. Assim, células são conjuntos convexos fechados, podendo ser limitados ou não.

Trivialmente qualquer espaço afim de dimensão finita é uma célula.

A dimensão de uma célula é identificada com a dimensão de seu espaço tangente, isto é, $\dim(\sigma) = \dim(tng(\sigma))$, e chamaremos uma célula *m-dimensional* simplesmente por *m-célula*.

Uma *face* τ de uma célula σ é um subconjunto convexo $\tau \subset \sigma$ tal que para todo $x, y \in \sigma$ e $\lambda \in \mathfrak{R}$ vale a seguinte propriedade:

$$x, y \in \sigma, 0 < \lambda < 1, (1 - \lambda)x + \lambda y \in \tau \implies x, y \in \tau.$$

Uma face de dimensão k é chamada uma k -face. Trivialmente, uma célula σ é uma face de si própria. Todas as outras faces de σ são chamadas *faces próprias*.

Usando o teorema de separação para conjuntos convexos, pode ser mostrado que um subconjunto $\tau \subset \sigma, \tau \neq \sigma$ é uma face de σ se e somente se existe um semi-espaço ξ tal que $\sigma \subset \xi$ e:

$$\tau = \sigma \cap \partial\xi.$$

Na linguagem usual, $\partial\xi$ é chamado um hiperplano que suporta σ em τ .

Desta apresentação, segue imediatamente que faces são células e que qualquer célula tem um número finito de faces. Além disso, qualquer intersecção de um número finito de faces é também uma face. Uma face própria de máxima dimensão, isto é, $\dim(\tau) = \dim(\sigma) - 1$ é chamada uma *facet* de σ . 0-faces correspondem a vértice de σ e 1-faces são também chamadas arestas de σ . Simpleses são casos particulares de células simples e assim as definições dadas aqui são compatíveis com aquelas da seção 4.3.

Variedade Linear Por Partes: Seja \widehat{M} uma família de $(N + 1)$ - células. Para $0 \leq k \leq N + 1$, usaremos a seguinte notação:

$$\widehat{M}^k = \left\{ \tau \mid \tau \text{ é uma } k\text{-face de alguma célula } \sigma \in \widehat{M} \right\}.$$

Além disso, indicamos:

$$\left| \widehat{M} \right| = \bigcup_{\sigma \in \widehat{M}} \sigma.$$

Nós chamamos \widehat{M} uma *Variedade Linear Por Partes* de dimensão $(N + 1)$ se e somente se as seguintes condições são satisfeitas:

(1) a intersecção $\sigma_1 \cap \sigma_2$ de duas células $\sigma_1, \sigma_2 \in \widehat{M}$ é vazia ou uma face comum de ambas as células,

(2) uma facet $\tau \in \widehat{M}^N$ é comum a no máximo duas células de \widehat{M} ;

(3) a família \widehat{M} é localmente finita, isto é, qualquer subconjunto relativamente compacto de $\left| \widehat{M} \right|$ intercepta somente um número finito de células de \widehat{M} .

C.2 Aproximando Variedades Diferenciáveis Definidas Implicitamente

Seja $H : \mathfrak{R}^{N+K} \rightarrow \mathfrak{R}^N$ uma aplicação suave tal que $0 \in \text{range}(H)$. Pelo Teorema da Função Implícita sabemos que o conjunto:

$$M = \{x \in \mathfrak{R}^{N+K} \mid H(x) = 0, x \text{ é um ponto regular de } H\},$$

é uma variedade suave de dimensão K .

Nesta seção nós descrevemos um algoritmo para gerar uma variedade linear por partes de dimensão K que aproxima M .

Vamos supor uma triangulação Γ em \mathfrak{R}^{N+K} . Se usarmos uma triangulação do tipo Freudenthal temos a vantagem de que dado um simplex qualquer um de seus vizinhos pode ser obtido por regras de pivotamento envolvendo apenas vetores de inteiros (Allgower & Georg, 1990).

Assim como na seção 4.3 nós denotamos por H_Γ a aproximação linear por partes de H com respeito a Γ . As definições para pontos regulares e valores regulares dadas em 4.3 permanecem válidas também neste contexto.

Neste caso, se zero é um valor regular de H_Γ , o conjunto $H_\Gamma^{-1}(0)$ possui a estrutura de uma variedade linear por partes. Mais precisamente:

Seja zero um valor regular de H_Γ . Se $\sigma \in \Gamma$ possui uma intersecção não nula com $H_\Gamma^{-1}(0)$, então $M_\sigma = \sigma \cap H_\Gamma^{-1}(0)$ é um politopo K -dimensional, e a família

$$M_\Gamma = \{M_\sigma \mid \sigma \in \Gamma, \sigma \cap H_\Gamma^{-1}(0) \neq \emptyset\},$$

é uma Variedade Linear por Partes de dimensão K .

Demonstração: Ver (Allgower & Georg, 1990).

O algoritmo a seguir descreve os passos fundamentais de um algoritmo de continuação para obter a variedade M_Γ que aproxima M , embora a suposição de suavidade para H pode ser relaxada sem trazer qualquer problema para a aplicação do algoritmo.

Nós fazemos então apenas a suposição de que zero é um valor regular da aproximação linear por partes H_Γ de H . Analogamente á seção 4.3 nós dizemos que um simplex $\sigma \in \Gamma$ é um *simplex de borda* se ele contém uma N -face completamente rotulada com respeito a H_Γ .

Algoritmo de Continuação Para Variedades de Dimensão K qualquer.

1. Inicialização

Encontrar um simplex de borda $\sigma \in \Gamma$;

Montar a lista atual de simplices de borda: $\Sigma = \{\sigma\}$;

Montar $V(\sigma)$ =conjunto dos vértices de σ ;

2. Repetir enquanto $V(\sigma) \neq \emptyset$ para algum $\sigma \in \Sigma$,

Tomar $v \in V(\sigma)$;

Obter por pivotamento de v em \tilde{v} um simplex $\tilde{\sigma}$ vizinho de σ ;

Se $\tilde{\sigma}$ não é simplex de borda descartar $\tilde{\sigma}$, retirar v do conjunto $V(\sigma)$. Retornar a passo (2).

Se $\tilde{\sigma} \in \Sigma$ então retirar v de $V(\sigma)$ e \tilde{v} de $V(\tilde{\sigma})$. Retornar ao passo (2).

Atualizar Σ fazendo $\Sigma \doteq \Sigma \cup \{\tilde{\sigma}\}$;

Se tivermos $K = 2$ e uma triangulação do tipo CF (Figura C.1), então os possíveis politopos são aqueles da Figura C.2, a menos de translações e rotações:

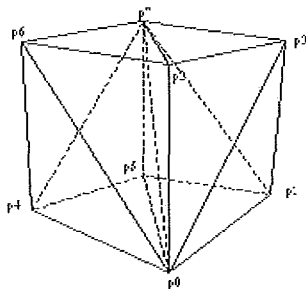


Figura C.1: Subdividindo um cubo (célula) para obter triangulação tipo CF.

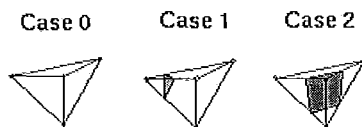


Figura C.2: Politopos para triangulação tipo CF.

Neste caso, podemos substituir o algoritmo acima pelo algoritmo dos *Tetra-Cubes* para fazer melhor uso das simplificações decorrentes do número reduzido de casos a considerar (basicamente três).

C.3 T-Superfícies

Uma das vantagens do método das T-Snakes descrito na seção 5.3.2 é a possibilidade de estendê-lo para 3D, mantendo a mesma metodologia do caso 2D. A base teórica do novo modelo - denominado T-Superfícies (McInerney & Terzopoulos, 1997; McInerney, 1997) - foi desenvolvida acima.

Como no caso 2D, os elementos básicos das T-Superfícies são: triangulação do espaço, função característica e modelo discreto para a snake.

Do ponto de vista topológico, uma T-Superfície é uma Variedade Linear por Partes de dimensão 2. Do ponto de vista dinâmico, uma T-Superfície é uma malha triangular elástica e fechada. Os pontos da malha são denominados *nós* (ou *snaxels*) e os triângulos denominados *elementos triangulares*.

A cada nó v_i estão associadas forças elásticas, força normal e forças de imagem, gerando uma equação de evolução idêntica àquela da seção 5.3.2.

Uma vez evoluída a superfície é preciso projetar a mesma sobre a triangulação e atualizar a função característica. O processo de reparametrização segue os mesmos passos do caso 2D, salvo as devidas modificações necessária para atualizar a função característica. Esta função tem exatamente o mesmo papel do caso 2D. Ela distingue o interior do exterior das superfícies (snakes 3D) permitindo tratar mudanças topológicas da snake 3D pela simples aplicação do algoritmo de continuação da seção anterior para determinar todas as componentes conexas em questão.

A atualização eficiente da função característica é novamente o ponto central. O algoritmo para isto é análogo àquele descrito na seção 5.3.2, observando-se agora que temos que tratar o prisma definido por cada elemento triangular $\{p_1, p_2, p_3\}$ na posição anterior e seu correspondente $\{p_{1n}, p_{2n}, p_{3n}\}$ na nova posição.

Durante sua evolução, cada elemento triangular pode passar sobre zero, apenas um, ou vários nós da triangulação do espaço. O poliedro formado conectando os pontos $\{p_1, p_2, p_3\}$ com seus correspondentes $\{p_{1n}, p_{2n}, p_{3n}\}$ é dividido em oito poliedros (Figura C.4). Para determinar se um nó da triangulação foi *queimado* durante a evolução da superfície, devemos determinar se este nó é interior a algum destes poliedros. O algoritmo usado é uma extensão daquele descrito na seção 5.3.2 para as T-Snakes.

Primeiramente, particionamos o domínio da imagem em oito subespaços através de três planos. Estes planos são definidos unindo-se os dois nós de cada aresta do elemento triangular na posição corrente com o nó da malha (Figura C.3).

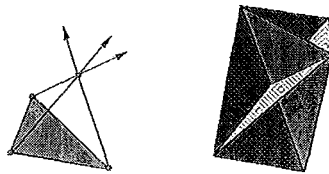


Figura C.3: Formação dos espaços usados para algoritmo de classificação em 3D.

A seguir, classificamos os três nós do elemento triangular na nova posição $\{p_{1n}, p_{2n}, p_{3n}\}$ em um dos oito subespaços (Figura C.4). O resultado desta classificação é usado para determinar rapidamente se um nó da triangulação foi queimado.

O algoritmo das T-Superfícies é exatamente aquele das T-Snakes, usando-se agora o algoritmo de continuação da seção anterior para determinar as componentes conexas da superfície.

A triangulação usada em (McInerney & Terzopoulos, 1997; McInerney, 1997) é do tipo Coxeter-Freudenthal pela sua simplicidade (Figura C.1).

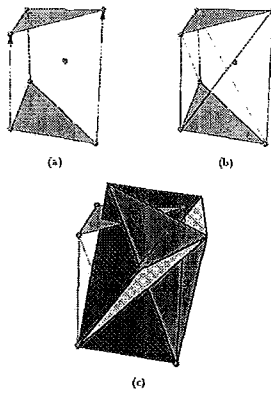


Figura C.4: (a)Elemento triangular se moveu para nova posição queimando nó da triangulação. (b)Poliedro formado pelo elemento triangular na posições nova e antiga. (c) Partição do espaço.

Apêndice D

Análise de Convexidade para Snakes

Neste Apêndice analisamos a conexidade do modelo de snakes procurando com isto estabelecer relações entre o comportamento dos modelos das seções 3.1 e 3.2 e os parâmetros correspondentes.

Dentre as propostas encontradas na literatura para tratar os problemas da não conexidade da energia do modelo original, encontramos a utilização de métodos multi-escala, reparametrização, forças adicionais, programação dinâmica, métodos duais, interação com o usuário e ajuste de parâmetros.

Nos métodos multi-escala, o modelo é resolvido para uma resolução mais baixa (escala grossa) da imagem e sua solução nesta resolução é usada como condição inicial para a escala seguinte (Leymarie & Levine, 1993). À medida que o método avança para escalas mais finas espera-se ficar mais próximo do mínimo (global) desejado. Esta metodologia tem o custo computacional extra de se montar a representação multi-escala da imagem e, caso a seqüência de escalas utilizada não seja adequada, a convergência para o contorno procurado pode não ocorrer.

O princípio básico dos métodos multi-escala consiste no fato de que, ao passarmos de uma resolução mais grossa para a seguinte, acrescentamos detalhes na imagem (pequenos artefatos e ruídos) os quais *perturbam* a snake. Esta perturbação se reflete em um desequilíbrio de forças sobre a curva podendo fazer com que a mesma atinja uma posição mais próxima da borda desejada.

O efeito de perturbar a snake pode ser obtido também através de reparametrização da curva. Na prática, isto equivale a criar mais snaxels, diminuindo assim as forças internas.

O modelo de Balloon da seção 3.4 pode ser visto também como uma tentativa para vencer os efeitos da não conexidade do modelo original. Neste caso, a força normal adicional tem o papel de "conduzir" a snake para o mínimo global diminuindo assim a sensibilidade com relação à inicialização decorrente da falta de conexidade.

Assim como (Cohen, 1991), Gang Xu et al. (Xu *et al.*, 1994) sugerem também

a adição de forças internas extras mas de uma forma mais sistemática. Neste caso, a curva é aproximada localmente pelo círculo definido pelo centro e raio de curvatura e a partir desta aproximação é obtida uma relação envolvendo os parâmetros das forças internas. Esta relação fornece uma estimativa para a força normal interna a qual é usada para redefinir os parâmetros do modelo durante a evolução da snake. O método desenvolvido por Gang Xu et al. (Xu *et al.* , 1994) tem ainda a vantagem de ser menos sensível à escolha de parâmetros. Contudo, o método torna a curva menos sensível a contornos mal definidos (*subjetivos*), tem o custo adicional da determinação da aproximação local da curva e sua extensão para o caso 3D não é óbvia.

Os métodos envolvendo programação dinâmica e contornos duais foram discutidos em detalhe nos Capítulos 4 e 5.

A interação com o usuário é um ponto delicado no contexto de snakes. Embora o principal desafio nestes métodos seja prescindir totalmente da intervenção humana, isto pode ser difícil de conseguir na prática, não apenas pela não conexidade da energia mas também pela subjetividade inerente à definição de borda de objetos numa imagens.

Dentre as metodologias usadas para interação com o usuário temos a utilização de pontos atratores ou repulsores e a criação de modelos híbridos usando o método de snakes e métodos de extração de bordas baseados em paradigmas interativos, tais como o *live-wire* (Liang *et al.* , 1999b; Falcão *et al.* , 1996).

Outro método para lidar com este problema consiste primeiramente em procurar condições que garantam a conexidade da energia (3.6) na região de interesse. Tendo a garantia de conexidade, um método de otimização pode ser aplicado com segurança de convergência.

O ponto de partida deste Apêndice é procurar condições que garantam a conexidade da energia (3.6) na região de interesse. Em (Davatzikos & Prince, 1999) encontramos semelhante análise, baseada em um modelo discreto de snakes.

A apresentação a seguir está baseada em nosso relatório técnico (Giraldi & Oliveira, 1999). O principal objetivo deste trabalho é estender a análise de conexidade da energia apresentada em (Davatzikos & Prince, 1999) de forma a incorporar o espaço de velocidades da snake. A principal consequência desta análise é encontrar condições necessárias para garantir a convergência para o contorno desejado de uma versão para dimensão finita do modelo da seção 3.2.

Neste sentido, iniciamos revisando o trabalho de (Davatzikos & Prince, 1999). A seguir apresentamos o modelo de snake de interesse, o qual se constitui em um sistema dinâmico cuja análise qualitativa é apresentada na seção D.2.2. Finalmente, apresentamos a análise quantitativa do mesmo a qual interliga os parâmetros dinâmicos (massa linear μ e amortecimento γ) com a energia potencial do mesmo.

D.1 Análise de Convexidade do Modelo Original

A análise de conexidade do modelo (3.6) poderia em princípio ser feita via segunda variação (Dubrovin *et al.*, 1984) do funcional de energia. Infelizmente, os resultados de semelhante análise são de pouca utilidade prática o que dirigiu a pesquisa na direção de versões discretas do modelo (3.6).

Com esta simplificação, o funcional (3.6) se torna uma função não-linear. O problema de otimização correspondente pode ser resolvido então por um método do tipo *steepest descent* (Chapra & Canale, 1988), desde que se possa garantir conexidade da energia discretizada. Daí a importância do estudo de conexidade do modelo.

Consideramos então a curva (snake) como uma coleção de pontos

$$\left\{ q_i = (x_i, y_i)^T, i = 0, 1, \dots, N \right\}, \quad (\text{D.1})$$

onde $q_i = c(i/N)$ e c é a curva.

Na análise desta seção estaremos supondo contornos abertos e os parâmetros ω_1 e ω_2 do modelo (3.6) constantes. Desta forma, o vetor de pontos livres da curva (discreta) será:

$$d = (q_1, q_2, \dots, q_{N-2}, q_{N-1}). \quad (\text{D.2})$$

Assim, nós podemos obter uma aproximação discreta expressão 3.6 pela aplicação de diferenças finitas do tipo usado na seção 3.2.3, obtendo-se

$$\bar{E}(d) = \bar{E}_{int}(d) + \bar{E}_{ext}(d), \quad (\text{D.3})$$

onde $\bar{E}_{int}(d)$ e $\bar{E}_{ext}(d)$ são as versões discretas das energias interna (funcional (3.4)) e externa (funcional (3.5)) do modelo original. As expressões exatas para $\bar{E}_{int}(d)$ e $\bar{E}_{ext}(d)$ pode ser encontradas em (Davatzikos & Prince, 1999).

A condição necessária para d minimizar $\bar{E}(d)$ é dada por:

$$\nabla \bar{E}(d) = \omega_1 N (2A_1 d - b_1) + \omega_2 N^3 (2A_2 d - b_2) + \nabla_d P = 0, \quad (\text{D.4})$$

onde b_1, b_2 são vetores de dimensão $2(N-1)$ relacionados com as condições de fronteira, $A_1 = \text{diag}(B_1)$ e $A_2 = \text{diag}(B_2)$ com B_1 sendo uma matriz simétrica do tipo Toeplitz, 3-diagonal, cuja primeira linha é $(2, -1, \dots, 0)$ e B_2 é uma matriz simétrica e 3-diagonal (para detalhes ver (Davatzikos & Prince, 1999)).

Se $\bar{E}(d)$ é estritamente convexa, então (D.4) é também condição suficiente, e d é o único ponto de mínimo para $\bar{E}(d)$.

No que segue, o domínio de definição do potencial P é denotado por R . Assim, o domínio D onde d toma valores é dado por:

$$D = \{ (q_1, q_2, \dots, q_{N-2}, q_{N-1}) \in R^{2N-2}; q_i \in \mathfrak{R}^2, i = 1, \dots, N-1 \}, \quad (\text{D.5})$$

D.1.1 Análise de Convexidade

O objetivo em (Davatzikos & Prince, 1999) é estabelecer condições para as quais $\bar{E}(d)$ é estritamente convexa. O fato de $\bar{E}(d)$ ser uma função escalar definida em D dado acima implica que $\bar{E}(d)$ será estritamente convexa em uma região R se a matriz Hessiana de $\bar{E}(d)$ (aqui denotada por $D^2 \bar{E}(d)$) for definida positiva em R ; isto é, se o menor autovalor de $D^2 \bar{E}(d)$ for maior que zero:

$$\lambda_{\min} (D^2 \bar{E}(d)) > 0 \iff \lambda_{\min} (D^2 \bar{E}_{int}(d) + D^2 \bar{E}_{ext}(d)) > 0, \quad (\text{D.6})$$

onde $D^2 \bar{E}_{int}(d)$ e $D^2 \bar{E}_{ext}(d)$ são as Hessianas de $\bar{E}_{int}(d)$ e $\bar{E}_{ext}(d)$, respectivamente.

Usando o fato de que o menor autovalor da soma de duas matrizes simétricas é maior ou igual à soma dos menores autovalores das duas matrizes, nós encontramos a seguinte condição suficiente para garantir a conexão de $\bar{E}_{ext}(d)$:

$$\lambda_{\min} (D^2 \bar{E}_{int}(d)) + \lambda_{\min} (D^2 \bar{E}_{ext}(d)) > 0. \quad (\text{D.7})$$

Podemos mostrar que os autovalores de $D^2 \bar{E}_{int}(d)$ coincidem com aqueles da matriz $2\omega_1 N A_1 + 2\omega_2 N^3 A_2$ (Davatzikos & Prince, 1999). Além disso, uma vez que A_1 é diagonal de blocos, seus autovalores coincidem com aqueles de B_1 . De maneira similar, os autovalores de A_2 coincidem com aqueles de B_2 . Portanto, concluímos que:

$$\lambda_{\min} (D^2 \bar{E}_{int}(d)) \geq 2\omega_1 N \lambda_{\min} (B_1) + 2\omega_2 N^3 \lambda_{\min} (B_2) > 0. \quad (\text{D.8})$$

Uma vez que B_1 é Toeplitz 3-diagonal, seus autovalores podem ser encontrados via uma fórmula recursiva (Davatzikos & Prince, 1999), obtendo-se o seguinte autovalor mínimo:

$$\lambda_{\min} (B_1) = 2 \left(1 - \cos \left(\frac{\pi}{N} \right) \right). \quad (\text{D.9})$$

Os autovalores de B_2 satisfazem uma relação de recursão a qual não possui uma solução explícita, mas apresenta um limite inferior dado por (Davatzikos & Prince, 1999):

$$\lambda_{\min}(B_2) \geq \lambda_{\min}(B_1)^2. \quad (\text{D.10})$$

Substituindo as expressões (D.9)-(D.10) em (D.8) encontramos o seguinte resultado:

$$\lambda_{\min}(D^2 \bar{E}_{int}(d)) \geq 4\omega_1 N \left(1 - \cos\left(\frac{\pi}{N}\right)\right) + 8\omega_2 N^3 \left(1 - \cos\left(\frac{\pi}{N}\right)\right)^2. \quad (\text{D.11})$$

Analisaremos a seguir a energia externa.

D.1.2 Hessiana da Energia Externa

Primeiramente, nós devemos notar que a matriz $D^2 \bar{E}_{ext}(d)$ é diagonal de blocos com os elementos da diagonal dados por:

$$D_i = \frac{1}{N} \begin{bmatrix} \frac{\partial^2 P(q_i)}{\partial x_i^2} & \frac{\partial^2 P(q_i)}{\partial x_i \partial y_i} \\ \frac{\partial^2 P(q_i)}{\partial x_i \partial y_i} & \frac{\partial^2 P(q_i)}{\partial y_i^2} \end{bmatrix}, \quad i = 1, \dots, N-1, \quad (\text{D.12})$$

onde $q_i = (x_i, y_i)^T$.

Assim, os autovalores de $D^2 \bar{E}_{ext}(d)$ podem ser determinados encontrando os autovalores das matrizes 2×2 acima.

Uma solução direta produz:

$$\lambda_{i1} = \frac{P_{xx}(q_i) + P_{yy}(q_i)}{2N} + \frac{\sqrt{(P_{xx}(q_i) - P_{yy}(q_i))^2 + 4P_{xy}^2(q_i)}}{2N}, \quad (\text{D.13})$$

$$\lambda_{i2} = \frac{P_{xx}(q_i) + P_{yy}(q_i)}{2N} - \frac{\sqrt{(P_{xx}(q_i) - P_{yy}(q_i))^2 + 4P_{xy}^2(q_i)}}{2N}, \quad (\text{D.14})$$

para $i = 1, \dots, N-1$. O autovalor mínimo é:

$$\lambda_{\min}(D^2 \bar{E}_{ext}(d)) = \min\{\lambda_{i1}, \lambda_{i2}; i = 1, \dots, N-1\}. \quad (\text{D.15})$$

Nós podemos dar um passo além para eliminar a dependência destas expressões com relação à localização específica dos pontos $q_i = (x_i, y_i)^T$. Para isto, vamos definir:

$$h_1 = \frac{P_{xx}(x, y) + P_{yy}(x, y)}{2N} + \frac{\sqrt{(P_{xx}(x, y) - P_{yy}(x, y))^2 + 4P_{xy}^2(x, y)}}{2N}, \quad (\text{D.16})$$

$$h_2 = \frac{P_{xx}(x, y) + P_{yy}(x, y)}{2N} - \frac{\sqrt{(P_{xx}(x, y) - P_{yy}(x, y))^2 + 4P_{xy}^2(x, y)}}{2N}, \quad (\text{D.17})$$

Vejamos como é possível simplificar estas expressões. Primeiramente, nós consideramos a curva isopotencial passando por um ponto $(x, y)^T$ e assumimos a princípio que sua curvatura é não nula. Nós podemos então definir um sistema de coordenadas polares (local) cuja origem coincide com o centro de curvatura da curva (ver (Xu *et al.*, 1994) para detalhes). Nós agora consideramos o potencial como uma função das coordenadas locais r e ϕ (Figura D.1), e procuramos expressões para P_{xx} , P_{yy} , e P_{xy} nas novas coordenadas.

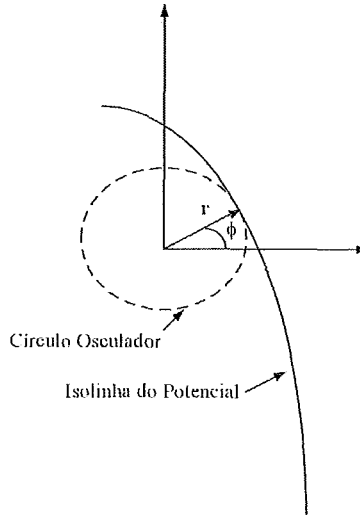


Figura D.1: Sistema de coordenadas locais no centro de curvatura da isolinha de potencial passando por um ponto q .

Primeiramente, pode ser mostrado que $P_\phi = P_{\phi\phi} = 0$ (Davatzikos & Prince, 1999). Além disso, notamos que os potenciais usados para extração de bordas possuem curvas impotenciais aproximadamente paralelas próximo às bordas, especialmente quando estas são suaves.

Por esta observação, é possível assumir $P_{r\phi} \approx 0$, e finalmente, aproximar (D.16)-(D.17) por:

$$h_1 \approx \frac{P_{rr}(x, y) + \frac{P_y(x, y)}{r(x, y)}}{4} + \left| P_{rr}(x, y) - \frac{P_y(x, y)}{r(x, y)} \right|, \quad (\text{D.18})$$

$$h_1 \approx \frac{P_{rr}(x, y) + \frac{P_y(x, y)}{r(x, y)}}{4} - \left| P_{rr}(x, y) - \frac{P_y(x, y)}{r(x, y)} \right|. \quad (\text{D.19})$$

Pode ser mostrado que (Davatzikos & Prince, 1999):

$$\min_R \{h_1(x, y), h_2(x, y)\} = \min_R \{e_1(x, y), e_2(x, y)\}, \quad (\text{D.20})$$

onde:

$$e_1(x, y) = \frac{P_{rr}(x, y)}{2}, \quad e_2(x, y) = \frac{P_r(x, y)}{2r(x, y)} \quad (\text{D.21})$$

Usando esta expressão e o fato de que:

$$\lim_{N \rightarrow \infty} 2N^2 \left(1 - \cos\left(\frac{\pi}{N}\right)\right) = \pi^2. \quad (\text{D.22})$$

nós podemos eliminar a dependência de N em (D.11) e finalmente escrever a condição de conexidade procurada na seguinte forma:

$$A(R'') + \omega_1 \pi^2 + \omega_2 \pi^4 > 0, \quad (\text{D.23})$$

onde $A(R'') = \min_{R''} \{e_1(x, y), e_2(x, y)\}$ e R'' é um subconjunto de R contendo a região de interesse.

Com esta análise é possível garantir que um método de otimização convergirá para a solução de (D.4) em R'' . No entanto, nada garante que esta solução seja a borda desejada. A expressão acima fornece limites inferiores para os parâmetros do modelo. Estaremos mais seguros da eficiência do método se limitarmos superiormente o intervalo de variação de ω_1 e ω_2 .

Em (Davatzikos & Prince, 1995) encontramos um trabalho nesta direção no contexto de processamento de imagens tomográficas do córtex cerebral. Neste trabalho parte-se de um modelo de snakes sem rigidez ($\omega_2 = 0$) e uma força externa específica. Usando-se uma representação em série de Fourier para a snake, pode-se interpretar as equações de Euler-Lagrange correspondentes como um filtro no espaço de frequências aplicado sobre os dados. A frequência de corte deste filtro (*cutoff frequency*) está diretamente relacionada ao parâmetro ω_1 e fornece o limite superior desejado (para maiores detalhes ver (Davatzikos & Prince, 1995)).

Este procedimento foi aplicado originalmente no contexto de modelos deformáveis em visão computacional (Witten, 1993) tendo resultado em um algoritmo elegante para o tracking no espaço de escalas linear (suavização Gaussiana). Contudo, estes métodos estão fora do escopo deste trabalho.

D.2 Snakes em Dimensão Finita

No modelo de contorno ativo original usado acima a otimização da energia pode ser feita via um método do tipo steepest descent, por exemplo.

Como mencionado na seção 3.2 um outro método para otimizar a energia da snake consiste em imergir a curva em um meio viscoso e resolver as equações dinâmicas correspondentes [Black-Yuille (1992)]. Esta metodologia é mais robusta em relação aos mínimos locais, mas é ainda sensível à não conexidade da energia. Nesta seção procuramos quantificar esta sensibilidade o que resulta em uma análise baseada em conceitos de sistemas dinâmicos e com um apelo geométrico característico do estudo das equações diferenciais ordinárias. Por outro lado, a análise que segue pode ser vista como uma generalização daquela apresentada acima, envolvendo agora também o espaço de velocidades.

Seja então um espaço de curvas admissíveis de dimensão finita, isto é, estaremos supondo que as bordas dos objetos de interesse possam ser representadas na forma:

$$c(s) = \sum_{i=0}^N q_i f_i(s) = B^T Q, \quad (\text{D.24})$$

onde $B = \{f_0, f_1, \dots, f_N\}$ é um conjunto de funções linearmente independentes (base do espaço) e $Q = (q_1, q_2, \dots, q_{N-2}, q_{N-1})^T$ é um vetor coluna de pontos do plano, geralmente denominados pontos de controle da curva.

Os exemplos mais comuns de modelos deformáveis deste tipo são as D-NURBS (Qin & Terzopoulos, 1996), Splines Ativas (Black & Yuille, 1993) e B-Snakes (Brigger *et al.*, n.d.), os quais são baseados em funções polinomiais por partes pertencentes ao espaço das funções quadrado integráveis (?).

A configuração instantânea do sistema é dada por um ponto em um espaço de configurações de dimensão $(2N + 2)$, cujas coordenadas (*generalizadas*) são dadas pelos $N + 1$ pontos do vetor Q acima. Uma vez que a curva se deforma, os pontos de controle se movimentam e assim temos um vetor de velocidades \dot{Q} associado com Q . O estado completo do sistema (snake) é definido pelo par $\left(Q, \dot{Q} \right)$.

Uma vantagem de usar curvas da forma (D.24) é que as equações de movimento correspondentes (equações de Euler-Lagrange) formam um sistema de equações diferenciais ordinárias. Estas equações são as análogas aquelas obtidas na seção 3.2, mas agora para dimensão finita.

Para desenvolver estas equações, procederemos analogamente ao que foi feito na seção 3.2. Assim, definimos:

Energia Cinética

$$T(\dot{c}) = \frac{1}{2} \int \mu \left\| \frac{\partial c}{\partial t} \right\|^2 ds = \frac{\mu}{2} \dot{Q}^T M_0 \dot{Q}. \quad (\text{D.25})$$

onde $M_0 = \int B B^T ds$ e μ é a *massa linear*.

Dissipação de Energia

$$\frac{\partial U_{at}}{\partial t}(\dot{c}) = -\frac{\gamma}{2} \dot{Q}^T M_0 \dot{Q}. \quad (\text{D.26})$$

onde γ é o *coeficiente de amortecimento*.

Energia Potencial Elástica

$$E_{int} = \omega_1 Q^T M_1 Q + \omega_2 Q^T M_2 Q. \quad (\text{D.27})$$

onde ω_1 e ω_2 são os parâmetros de elasticidade e rigidez definidos na seção 3.1 e:

$$M_1 = \int \frac{dB}{ds} \cdot \frac{dB^T}{ds} ds, \quad M_2 = \int \frac{d^2B}{ds^2} \cdot \frac{d^2B^T}{ds^2} ds. \quad (\text{D.28})$$

Energia Potencial Externa

$$E_{ext} = \int P(c(s, t)) ds. \quad (\text{D.29})$$

Energia Potencial do Modelo

$$E_p = E_{int} + E_{ext}. \quad (\text{D.30})$$

A Lagrangeana correspondente é dada por:

$$L(c, \dot{c}, c', c'') = T - E_p - U_{at}, \quad (\text{D.31})$$

As equações de Euler-Lagrange, derivadas via Princípio de Hamilton (Goldstein, 1981), para o sistema descrito pelas equações (D.25)-(D.30) compõe o modelo de snake que usaremos nesta seção. Estas equações possuem a seguinte forma (Black & Yuille, 1993):

$$\frac{d}{dt} \left(\frac{\partial T}{\partial \dot{Q}} \right) - \frac{\partial T}{\partial Q} - \frac{\partial}{\partial \dot{Q}} \left(\frac{\partial U_{at}}{\partial t} \right) + \frac{\partial E_p}{\partial Q} = 0 \quad (\text{D.32})$$

Substituindo (D.25)-(D.30) nesta expressão nós encontramos o seguinte conjunto de equações:

$$\mu \ddot{Q} + \gamma \dot{Q} + M_0^{-1} (\omega_1 M_1 + \omega_2 M_2) Q = M_0^{-1} F, \quad (\text{D.33})$$

onde:

$$F = -\frac{1}{2} \int \frac{\partial P(c(s, t))}{\partial Q} ds. \quad (\text{D.34})$$

A análise de parâmetros apresentado na seção 3.2.1 se aplica igualmente para este modelo. Contudo, a restrição à dimensão finita permite agora reduzir o modelo de snakes a um sistema dinâmico, não mais envolvendo derivadas parciais no espaço (como no Capítulo 3, o que simplifica sua análise).

Vejamoinicialmente, o efeito da massa e coeficiente de amortecimento no comportamento do método numérico.

D.2.1 Análise Numérica

Uma vez que o sistema (D.33) é constituído por equações diferenciais ordinária, podemos discretizá-lo usando apenas diferenças finitas no tempo. O esquema numérico adotado é definido pelas expressões a seguir:

$$\dot{Q}(t - \tau) \approx \frac{1}{2\tau} [Q(t) - Q(t - 2\tau)], \quad (\text{D.35})$$

$$\ddot{Q}(t - \tau) \approx \frac{1}{\tau^2} [Q(t) - 2Q(t - \tau) + Q(t - 2\tau)]. \quad (\text{D.36})$$

Procedendo identicamente ao que foi feito na seção 3.2.3 encontramos o seguinte sistema linear:

$$AQ^{t+\tau} = B^{t,t-\tau}. \quad (\text{D.37})$$

onde:

$$A = \left[\left(\frac{\mu}{\tau^2} + \frac{\gamma}{2\tau} \right) M_0 + K \right], \quad (\text{D.38})$$

$$B^{t,t-\tau} = 2F(\bar{t} - \tau) + \left(\frac{2\mu}{\tau^2} \right) M_0 Q^t - \left(\frac{\mu}{\tau^2} - \frac{1}{2\tau} \right) M_0 Q^{t-\tau}, \quad (\text{D.39})$$

O número de condição da matriz A na norma 2 é dado por (?):

$$\kappa_2(A) = 1 + \frac{\lambda_{\max}(M_0) - \lambda_{\min}(M_0)}{\lambda_{\min}(M_0)} + \frac{\lambda_{\max}(M_0)}{\beta}, \quad (\text{D.40})$$

onde:

$$\beta = \left(\frac{\mu}{\tau^2} + \frac{\gamma}{2\tau} \right). \quad (\text{D.41})$$

Portanto, obtemos conclusões equivalentes àsquelas da seção 3.2.3, ou seja, que melhoramos a estabilidade numérica do sistema linear (D.37) quando aumentamos os valores de μ e λ ou diminuimos o passo no tempo τ .

Contudo, novamente temos um trade-off entre estabilidade, performance e eficiência do modelo. As próximas seções têm por objetivo final *quantificar* este trade-off.

D.2.2 Análise Qualitativa

A análise do esquema numérico apresentada acima foi um caminho para obter relações entre os parâmetros e o comportamento do modelo. Outra possibilidade neste sentido é analisar o modelo contínuo. Iniciemos por uma análise simples do sistema (D.33).

Do ponto de vista mecânico, temos um sistema físico cuja energia mecânica (potencial mais cinética) decresce em uma taxa dada pela expressão (D.26). Assim, uma vez que não há ganho de energia mecânica, o sistema atingirá o estado estacionário em algum ponto do espaço de configuração. Das equações (D.33) nós podemos mostrar (abaixo) que este ponto é um extremo da energia potencial.

Assim, independentemente das condições iniciais e parâmetros, esta análise simples mostra que um extremo da energia potencial é sempre atingido. Neste ponto, duas questões surgem naturalmente:

(1) Uma vez que a energia potencial não é convexa em geral, como encontrar condições iniciais e parâmetros apropriados para atingir o mínimo local (ou global) desejado?

(2) E quanto à taxa com a qual a solução do sistema converge para o mínimo desejado?

A primeira questão é discutida em (Giraldi & Oliveira, 1999) utilizando-se formalismo Hamiltoniano (Goldstein, 1981) e está fora do escopo desta apresentação.

Nesta seção e na próxima apresentaremos uma resposta para a segunda questão a qual liga a análise de conexidade da energia potencial e os parâmetros dinâmicos (μ e γ) do modelo. O resultado a ser obtido mostra a relação estreita entre a dinâmica

do sistema (D.33) e os autovalores da Hessiana da energia potencial dada por (D.30), sendo portanto uma extensão da análise da seção D.1 incorporando agora o espaço de velocidades.

Primeiramente, vamos colocar o sistema de equações diferenciais ordinárias de segunda ordem da expressão (D.30) na forma:

$$\ddot{Q} = f(Q, \dot{Q}), \quad (\text{D.42})$$

Vamos reduzir este sistema para um sistema de primeira ordem pela transformação usual (Sotomayor, 1979):

$$Q_1 = Q, \quad Q_2 = \dot{Q}. \quad (\text{D.43})$$

Assim, a equação (D.33) torna-se:

$$\begin{aligned} \dot{Q}_1 &= Q_2, \\ \dot{Q}_2 &= f(Q_1, Q_2) \end{aligned} \quad (\text{D.44})$$

Chamando:

$$x = \begin{pmatrix} Q_1 \\ Q_2 \end{pmatrix}, \quad X_1 = Q_2 \text{ e } X_2 = f(Q_1, Q_2), \quad (\text{D.45})$$

o sistema (D.44) torna-se:

$$\dot{x} = X(x), \quad (\text{D.46})$$

onde $X(x) = (X_1, X_2)^T$. O sistema (D.46) é autônomo e de primeira ordem. Se a energia E_P é de classe C^2 então o campo X será de classe C^1 e o problema de Cauchy:

$$\dot{x} = X(x), \quad x(0) = x_0, \quad (\text{D.47})$$

tem solução única (Sotomayor, 1979).

Da teoria de sistemas dinâmicos nós sabemos que os pontos que resolvem a equação:

$$X(x) = 0, \quad (\text{D.48})$$

chamados *pontos singulares (ou críticos)*, são fundamentais para a caracterização qualitativa da solução de (D.46). Um resultado fundamental no contexto das snakes emerge da análise destes pontos. Primeiramente, devemos notar que estes são exatamente os extremos da energia E_p .

Para mostrar isto, vamos denotar por \bar{x} uma solução de (D.48). Assim, pelas definições de Q_1 e Q_2 dadas nas expressões (D.43) teremos:

$$\dot{Q} = 0, \tag{D.49}$$

para todo t , logo:

$$Q = Const. \tag{D.50}$$

Se nós tomarmos o modelo de snake (D.33) nós veremos que tal ponto, o qual é uma solução do sistema (D.44), é também uma solução da equação:

$$\frac{1}{2} (\omega_1 M_1 + \omega_2 M_2) Q = F, \tag{D.51}$$

e assim, é também um extremo da energia E_p . Inversamente, se um ponto \bar{x} é uma borda ($Q_2 \equiv 0$ portanto) e satisfaz a equação (D.51) então é um ponto singular de X .

Portanto, as informações básicas da topologia do espaço de fase para (D.46) estão relacionadas às soluções de (D.51) as quais são de fato os pontos que procuramos. Para que o modelo (D.33) seja eficiente, qual deve ser a relação entre esta topologia e a borda que procuramos?

A única possibilidade é que a borda seja um atrator, isto é, um nó ou foco estável (Figuras D.2.a). As outras possibilidades para pontos singulares estão também representados na Figura D.2 e fica claro as dificuldades inerentes, isto é, ou encontramos subespaços instáveis nos quais a solução *foge* do ponto desejado (Figuras D.2.b-e) ou então teremos componentes cíclicas para a solução (foco instável da Figura D.3) o que vai dificultar, ou mesmo impedir, a convergência.

No caso da Figura D.3 existe ainda um problema relacionado com a *estabilidade estrutural* do sistema correspondente. De fato, o sistema neste caso é *estruturalmente instável* o que implica em problemas numéricos devido bifurcações que podem ocorrer.

Concluindo, para que a borda procurada seja um atrator, nós temos que escolher os parâmetros (μ , γ , ω_1 e ω_2) de tal forma que os autovalores de DX tenham parte real estritamente negativa (Sotomayor, 1979) na solução \bar{x} da equação (D.51). Este é o resultado central desta seção, o qual pode ser colocado da seguinte forma:

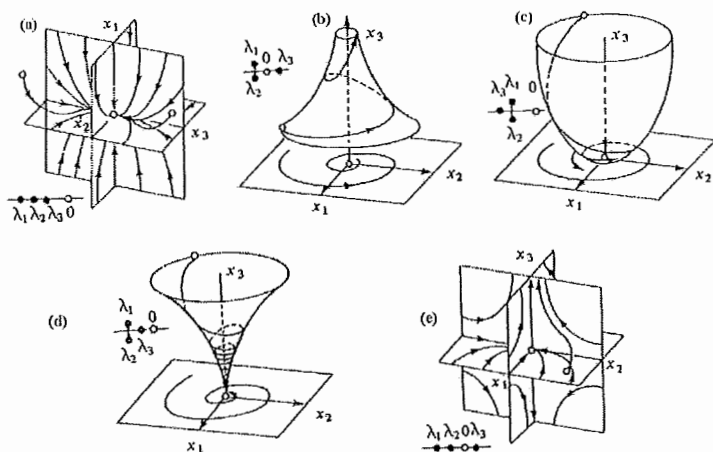


Figura D.2: Sistemas dinâmicos hiperbólico 3D.

Os parâmetros do modelo devem ser escolhidos de tal forma que a borda procurada seja um atrator ou foco estável na topologia do espaço de fase de (D.46).

Na próxima seção, nós demonstraremos que uma condição suficiente para isto é que a energia E_p tenha um mínimo (local ou global) na borda procurada.

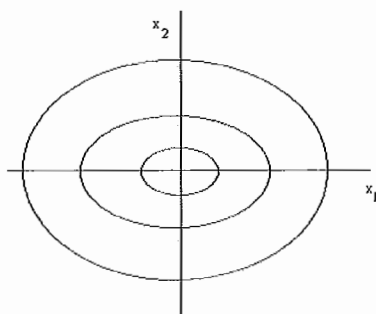


Figura D.3: Foco instável.

D.2.3 Convexidade e Atratores

Provaremos nesta seção a seguinte afirmação:

Um ponto singular x de X é um atrator ou foco estável se a energia dada por (D.30) é estritamente convexa em uma vizinhança de x , isto é, se x é um mínimo local de E_p .

Para demonstrar este fato, nós devemos analisar os autovalores da matriz DX em um ponto singular x supondo que a energia potencial é estritamente convexa em uma vizinhança V_x de x . Se estes autovalores têm parte real não-nula então nós

podemos usar o Teorema de Hartman para linearizar o campo X em V_x (Sotomayor, 1979).

Portanto, da teoria de sistemas dinâmicos lineares, se todos os autovalores de DX têm parte real estritamente negativa nós teremos um nó ou foco estável e a afirmação acima fica demonstrada.

De (D.44)-(D.45) nós vemos que a matriz DX tem a forma:

$$DX = \begin{pmatrix} 0 & I_{N \times N} \\ -\frac{1}{\mu} M_0^{-1} D^2(E_p) & -\frac{\gamma}{\mu} I_{N \times N} \end{pmatrix}, \quad (D.52)$$

onde $I_{N \times N}$ é a matriz identidade de ordem N , $D^2(E_p)$ é a matriz Hessiana da energia potencial E_p e M_0^{-1} é a inversa de M_0 definida em (D.25).

A equação de autovalores/autovetores para $DX(x)$ é da forma:

$$DX \cdot v = \sigma v, \quad (D.53)$$

onde (\cdot) significa multiplicação de matriz por vetor.

Usando (D.52) acima nós podemos escrever esta expressão na forma:

$$\begin{aligned} v_2 &= \sigma v_1, & (D.54) \\ -\frac{1}{\mu} M_0^{-1} D^2(E_p) \cdot v_1 - \frac{\gamma}{\mu} I \cdot v_2 - \sigma v_2 &= 0. \end{aligned}$$

Isolando v_2 na primeira destas equações e substituindo o resultado na segunda obtemos a seguinte expressão:

$$-\frac{1}{\mu} M_0^{-1} D^2(E_p) \cdot v_1 - \sigma \frac{\gamma}{\mu} I \cdot v_1 - \sigma^2 v_1 = 0. \quad (D.55)$$

a qual depende somente de v_1 :

Relembremos que nós estamos apenas interessados no sinal dos autovalores σ_i , $i = 1, \dots, 2N$, supondo que os autovalores α_i , $i = 1, \dots, 2N$ da Hessiana $D^2(E_p)$ são estritamente positivos. Primeiramente, notemos que a equação (D.55) pode ser colocada na forma:

$$D^2(E_p) \cdot v_1 = \beta M_0 \cdot v_1, \quad (D.56)$$

onde:

$$\beta = -(\sigma\gamma + \mu\sigma^2). \quad (D.57)$$

Assim, nós temos um *problema de autovalores/autovetores generalizado*, bem comum no estudo de modos de vibração para estruturas mecânicas (Bath & Wilson, 1976). Pode ser mostrado que se M_0 e $D^2(E_p)$ são definidas positivas e simétricas, os autovetores procurados são $M_0, D^2(E_p)$ – *ortogonais*. isto é:

$$\Phi_i^T M_0 \Phi_j = \delta_{ij}, \quad \Phi_i^T D^2(E_p) \Phi_j = \delta_{ij} \beta_i. \quad (\text{D.58})$$

Usando estas relações, é direto mostrar que a equação (D.56) é equivalente a:

$$\beta_i + \sigma\gamma + \mu\sigma^2 = 0, \quad i = 1, \dots, N, \quad (\text{D.59})$$

cujas soluções são dadas por:

$$\sigma_i^\pm = \frac{-\gamma \pm \sqrt{\Delta_i}}{2\mu}, \quad i = 1, \dots, N, \quad (\text{D.60})$$

onde:

$$\Delta_i = \gamma^2 - 4\mu\beta_i. \quad (\text{D.61})$$

Para x ser um atrator é necessário que:

$$4\mu\beta_i > 0, \quad i = 1, \dots, N. \quad (\text{D.62})$$

Uma vez que M_0 representa uma norma (portanto é definida positiva e simétrica), uma condição suficiente para a condição (D.62) ser verificada é que $D^2(E_p)$ seja definida positiva, ou seja, se a borda procurada seja um mínimo local da energia E_p . Portanto, a afirmação do início desta seção está demonstrada.

D.3 Discussão

Algumas considerações podem ser feitas a partir da discussão acima. Primeiramente, lembremos que o chamado *Amortecimento Crítico* ocorre quando:

$$\gamma^2 - 4\mu\beta_i = 0, \quad (\text{D.63})$$

sendo caracterizado por uma convergência rápida para a solução, na direção Φ_i (*modo* correspondente). Como primeira consequência do resultado (D.62) nós vemos

a impossibilidade de conseguir amortecimento crítico em mais de um modo neste modelo.

Uma forma de contornar esta dificuldade é substituir a matriz γM_0 pela série de Caughey dada por (Bath & Wilson, 1976):

$$C = M_0 \sum_{i=0}^{i=p-1} a_i [M_0^{-1} K]^i \tag{D.64}$$

uma vez que agora teríamos varias constantes de amortecimento. A expressão (D.64) tem a vantagem de manter a mesma transformação para diagonalizar a matriz de amortecimento (Bath & Wilson, 1976), mantendo portanto o desenvolvimento acima.

Analisando agora a expressão (D.60) observamos que esta interliga os dois elementos básicos do modelo dinâmico de snakes dado pela equação (D.64): a energia potencial (através do sinal dos autovalores β_i) e a dinâmica da snake (através dos parâmetros μ e γ). Neste sentido, (D.60) generaliza (Davatzikos & Prince, 1999), agora incorporando o espaço de velocidades.

Além disso, este resultado pode ser visto como um complemento para a discussão da seção D.2.1. Nesta última, nós observamos que quanto maiores μ e γ melhor era a estabilidade numérica, mas havia um trade-off com relação à performance/eficiência do modelo. A expressão (D.60) quantifica (ao menos teoricamente) este trade-off, mostrando também que a perda de performance para valores altos de μ e γ se deve ao fato de que estes podem fazer Δ_i mais negativo e assim o termo oscilatório correspondente mais alto.

O principal ponto agora seria como precisar a vizinhança V_x . A energia E_p não poderia fornecer a resposta a esta questão porque esta não envolve o espaço de velocidades. Tal observação aponta na direção da necessidade de uma quantidade (física) cuja análise de conexidade seja determinante da dinâmica do modelo e que incorpore o espaço de velocidades. Tal quantidade é exatamente a energia mecânica do sistema cujas propriedades de interesse surgem naturalmente na *Formulação Hamiltoniana* do modelo de snakes.

Esta formulação é desenvolvida no contexto de snakes em (Giraldi & Oliveira, 1999) mas está fora dos propósitos desta dissertação.

Apêndice E

Método da Conexões Fuzzy

As imagens capturadas por instrumentos de aquisição (câmeras, tomógrafos, etc) possuem indefinições que dependem de fatores tais como limitações de resolução, tanto espacial quanto temporal, parâmetros físicos, e outras propriedades dos dispositivos. Isto implica que a definição dos objetos de interesse a partir de padrões de intensidade vai sofrer imprecisões gerando subjetividade.

Em decorrência disto, argumenta-se em (Udupa & Samarasekera, 1996) que a noção de *objeto* em uma imagem digital deve ser definida formalmente via conceitos *fuzzy*.

Dentre os métodos de segmentação baseados em conceitos desta teoria, o método das conexões fuzzy (Udupa & Samarasekera, 1996) é particularmente interessante para este trabalho, uma vez que este envolve a escolha inicial de pontos semente, os quais podem ser usados para inicializar o Dual-T-Snakes.

Outro aspecto interessante deste método é que seu resultado pode ser incorporado facilmente na definição da força normal do Dual-T-Snakes, constituindo-se assim em uma metodologia alternativa para nosso trabalho.

Este método já foi usado conjuntamente com modelos deformáveis (Jones & Metaxas, June, 1998). Vejamos seus elementos básicos.

E.1 Subconjuntos Fuzzy

Seja X um conjunto de referência (não vazio). Um *subconjunto fuzzy* A de X é um conjunto de pares ordenados dado por:

$$A = \{(x, \mu_A(x)) \mid x \in X\}, \quad (\text{E.1})$$

onde:

$$\mu_A : X \longrightarrow [0, 1], \quad (\text{E.2})$$

é a função de inclusão (*membership function*) of A em X . Nós dizemos que A é não vazia se existir $x \in X$ tal que $\mu_A(x) \neq 0$. O subconjunto fuzzy vazio de X , denotado por Φ , satisfaz $\mu_\Phi(x) = 0$, para todo $x \in X$.

A *união e intersecção* entre dois subconjuntos fuzzy A e B de X é definida como segue:

$$A \cup B = \{(x, \mu_{A \cup B}(x)) \mid x \in X\}, \quad (\text{E.3})$$

onde para todo $x \in X$, $\mu_{A \cup B}(x) = \max\{\mu_A(x), \mu_B(x)\}$.

$$A \cap B = \{(x, \mu_{A \cap B}(x)) \mid x \in X\}, \quad (\text{E.4})$$

onde para todo $x \in X$, $\mu_{A \cap B}(x) = \min\{\mu_A(x), \mu_B(x)\}$.

Uma relação fuzzy ρ em X é um subconjunto fuzzy de $X \times X$ dado por:

$$\rho = \{(x, y), \mu_\rho(x, y)\} \mid (x, y) \in X \times X\}$$

onde

$$\mu_\rho : X \times X \longrightarrow [0, 1]$$

Nós representaremos uma imagem digital *n-dimensional* por um par $\Omega = (C, f)$, onde C é um array n-dimensional de pares ordenados de inteiros e f é uma função que associa a cada pixel da cena um array *m-dimensional* de inteiros que correspondem à intensidade de imagem.

E.2 Método da Conexões Fuzzy

Este método foi proposto inicialmente em (Udupa & Samarasekera, 1996). Posteriormente foi entendido para admitir a possibilidade de segmentar um cena com vários objetos de interesse, gerando o método das conexões fuzzy multi-sementes, que será descrito nesta seção.

Este método parte de um conjunto de pontos sementes sobre a imagem e tem a vantagem de incorporar toda a informação da imagem em um único campo, denominado σ , podendo ser usado tanto para imagens em tons de cinza quanto para imagens multivariadas, tais como imagens coloridas e com textura (Jones & Metaxas, June, 1998).

A definição a seguir é fundamental.

Dado um inteiro positivo M e o domínio D da imagem I , uma M -semisegmentação de I é uma função que para cada $c \in D$ associa um vetor $\sigma^c = (\sigma_0^c, \sigma_1^c, \dots, \sigma_M^c)$, tal que $\sigma_0^c \in [0, 1]$; para todo valor de m nós temos $\sigma_m^c = 0$ ou σ_0^c , e $\sigma_0^c = \max\{\sigma_i^c; i = 1, \dots, M\}$.

Para deixar esta definição mais clara para o contexto que nos interessa, vamos supor dois pontos sementes ($M = 2$) na imagem. Se estes pontos pertencem a diferentes objetos, digamos O_1 e O_2 o campo σ deve ser tal que $\sigma_1^c > 0$ e $\sigma_2^c = 0$ se $c \in O_1$ (analogamente para $c \in O_2$).

Assim, de acordo com a definição acima, nós definimos $\sigma_0^c = \max\{\sigma_i^c; i = 1, 2\}$. Se c pertence ao background a situação ideal seria $\sigma_1^c = \sigma_2^c = 0$. Na prática, contudo, precisamos usar um limiar para o campo σ_0 a fim de distinguir os objetos do background.

O método de conexões fuzzy multi-semente (*multiseed fuzzy connectedness*) é um método para calcular o campo σ . Iniciemos com o caso onde temos apenas um objeto de interesse.

Uma *afinidade fuzzy* em D é uma função $\psi : D^2 \rightarrow [0, 1]$ que estima os pontos (pixels) da imagem mais prováveis de pertencer a objeto O de interesse.

Dado um par de pontos (c, d) chamamos o valor $\psi(c, d)$ de *ψ -intensidade do par (c, d)* .

Chamamos uma *cadeia* em $U \subseteq D$ de $c^{(0)}$ até $c^{(K)}$ como sendo uma seqüência $\langle c^{(0)}, \dots, c^{(K)} \rangle$ de pontos (pixels) em U . A *ψ -intensidade* da cadeia é dado por:

$$\psi - \text{intensidade}_{cadeia} = \min_{1 \leq k \leq K} \psi(c^{(k-1)}, c^{(k)}) \quad (\text{E.5})$$

Se $K = 0$ então a *ψ -intensidade* da cadeia é definida como sendo 1.

Para segmentação de múltiplos objetos, nós devemos ter uma afinidade fuzzy para cada objeto, o que leva a um campo vetorial $\Psi = (\psi_1, \dots, \psi_M)$ em D^2 , onde cada componente ψ_i é uma função de afinidade fuzzy associada ao um objeto $O_i; i = 1, \dots, M$ da cena.

Sejam $U \subseteq D, W \subseteq D$ e $c \in D$. Seja $\Lambda_{U,W,c}$ o conjunto das cadeias em U partindo de um pixel em W até c . Nós denotamos por $\mu_{i,U,W}(c)$ ao valor:

$$\mu_{i,U,W}(c) = \max_{\langle d^{(0)}, \dots, d^{(K)} \rangle \in \Lambda_{U,W,c}} \psi - \text{intensidade}_{cadeia} \quad (\text{E.7})$$

Finalmente, o valor de $\sigma_i^c, i = 1, \dots, M$ é tal que

$$\sigma_i^c = \mu_{i,D,V_i}(c), \quad (\text{E.8})$$

onde V_i contém os pontos sementes para o objeto O_i . Um método de programação dinâmica é usado para obter $\mu_{i,D,V_i}(c)$ (Udupa & Samarasekera, 1996), o qual se baseia no seguinte teorema:

Sejam $V_m \subset D, 1 \leq m \leq M$, os conjuntos de pontos sementes, supostos não vazios. Então, existe uma única M -semisegmentação σ de D com a propriedade de que para cada $c \in D$, se para $1 \leq n \leq M$

$$\begin{aligned} s_n^c &= 1, \text{ se } c \in V_n, \\ s_n^c &= \max_{d \in V_n} (\min(\mu_{n,D,V_n}(d), \psi_n(d, c))), \text{ caso contrário,} \end{aligned} \quad (\text{E.9})$$

então para $1 \leq m \leq M$:

$$\begin{aligned} \sigma_m^c &= s_m^c, \text{ se } s_m^c \geq s_n^c \text{ para todo } 1 \leq n \leq M, \\ \sigma_m^c &= 0, \text{ caso contrário.} \end{aligned} \quad (\text{E.10})$$

Na próxima seção apresentamos o algoritmo que implementa o método e procuramos esclarecer os conceitos acima.

E.3 Algoritmo

Na inicialização do algoritmo a ser apresentado, fazemos

$$\begin{aligned} \sigma_m^c &\leftarrow 1, \text{ se } c \in V_m, \text{ para algum } m, \\ \sigma_m^c &\leftarrow 0, \text{ caso contrário.} \end{aligned} \quad (\text{E.11})$$

Definimos uma fila de prioridades dada por $H = V_1 \cup V_2 \cup \dots \cup V_m$. Usamos também uma variável l para guardar durante toda a execução o maior ψ_i - *intensidade_cadeia* para pontos na fila de prioridades H .

O coração do algoritmo é a expressão E.9.

A seguir apresentamos o pseudo-código correspondente.

```
for c ∈ D
  do for m ← 0 to M
```

Referências Bibliográficas

- Adams, A. & Bischof, L. 1994. Seeded Region Growing. *IEEE Trans. on Pattern Analysis and Machine Intel.*, **16**(6).
- Allgower, E. L. & Georg, K. 1990. *Numerical Continuation Methods: An Introduction*. Springer-Verlag Berlin Heidelberg.
- Alvarez, L.; Lions, Pierre-Louis & Morel, Jean-Michel. 1992. Image Selective Smoothing and Edge Detection by Nonlinear Diffusion II. *SIAM J. Num. Analysis*, **29**(3), 845–866.
- Amini, A. A.; Weymouth, T. E. & Jain, R. C. 1990. Using Dynamic Programming for Solving Variational Problems in Vision. *IEEE Trans. on Pattern Analysis and Machine Intel.*, **12**(9), 855–867.
- Bamford, P. & Lovell, B. A Water Immersion Algorithm for Cytological Image Segmentation.
- Bamford, P. & Lovell, B. 1995 (June). Robust Cell nucleous segmentation using a viterbi search based active contour. *Pages 840–845 of: Proc. Of the Fifth Int. Conf. On Computer Vision (ICCV'95), Cambridge, MA, USA.*
- Bath, K. & Wilson, E. L. 1976. *Numerical Methods in Finite Elements Analysis*. Prentice-Hall, Inc.
- Berger, M. J. 1986. Data Structures for Adaptive Grid Generation. *SIAM J. Sci. Stat. Comput.*, **7**, 904–916.
- Berger, M. J. & Olinger, J. 1984. Adaptive Mesh Refinement for Hyperbolic Partial Differential Equations. *J. Comput. Physics*, **54**, 484–512.
- Black, A. & Yuille, A. (eds). 1993. *Active Vision*. MIT Press.
- Bloom, W. & Fawcett, D. W. 1968. *A Textbook of Histology*. 8nd edn. W. B. Saunders Company.


```

do  $\sigma_m^c \leftarrow 0$ 
 $H \leftarrow \emptyset$ 
for  $m \leftarrow 1$  to  $M$ 
do for  $c \in V_m$ 
do if  $\sigma_0^c = 0$  then  $H \leftarrow H \cup \{c\}$ 
 $\sigma_0^c \leftarrow \sigma_m^c \leftarrow 1$ 
 $l \leftarrow 1$ 
while  $l > 0$ 
for  $m \leftarrow 1$  to  $M$ 
do remover um ponto  $d$  de  $V_m$ 
 $C \leftarrow \{c \in D; \sigma_m^c < \min(l, \psi_m(d, c))\}$ 
while  $C \neq \emptyset$ 
do remover um ponto  $c$  de  $C$ 
 $t \leftarrow \min(l, \psi_m(d, c))$ 
if  $l = t$  and  $\sigma_m^c < l$  then  $V_m \leftarrow V_m \cup \{c\}$ 
if  $\sigma_0^c < t$  then
if  $\sigma_0^c = 0$  then  $H \leftarrow H \cup \{c\}$ 
for  $n \leftarrow 1$  to  $M$ 
do  $\sigma_n^c \leftarrow 0$ 
if  $\sigma_0^c \leq t$  then  $\sigma_0^c \leftarrow \sigma_m^c \leftarrow t$ 
while  $Chave\_Maxima(H) = l$ 
Remover\_Chave\_Maxima( $H$ )
 $l \leftarrow Chave\_Maxima(H)$ 
for  $m \leftarrow 1$  to  $M$ 
 $V_m \leftarrow \{c \in H; \sigma_m^c = l\}$ 

```

E.4 Dual-T-Snakes e Conexões Fuzzy

O método descrito acima fornece um pré-processamento da imagem. Seu resultado para imagens com ruídos pode ser útil para definir o sentido da força normal do Dual-T-Snakes como segue.

Sejam $F_{ij}(v_i)$ a força normal no snaxel i da snake $j = 0, 1, \dots, M$, onde M é o número de objetos de interesse. De acordo com nossas definições do Capítulo 6, estaremos supondo $j = 0$ para a snake externa e $j \neq 0$ para as demais.

Então, uma vez calculado o campo σ segundo o algoritmo acima, podemos definir a força normal da seguinte forma:

$$F_{ij}(v_i) = Val_j(v_i) kn_i, \quad (\text{E.12})$$

onde:

$$Val_j(v_i) = 1 \text{ se } \sigma_j^{v_i} = \sigma_0^{v_i} > T_\sigma \quad (\text{E.13})$$

$$Val(v_i) = 0 \text{ caso contrário,} \quad (\text{E.14})$$

onde T_σ é um limiar para σ_0^c .

Do ponto de vista do Dual-T-Snakes, este procedimento é interessante pois os pontos sementes devem ser escolhidos pelo usuário em regiões com padrões de intensidade significativos, e assim são interessantes também para a inicialização das snakes internas.

E.5 Função de Afinidade

Uma possibilidade para a função de afinidade usada para imagens em tons de cinza é a seguinte:

$$\psi(c, d) = \mu_\alpha(c, d) [\omega_1 g_1(I(c), I(d)) + \omega_2 g_2(F(c), F(d))], \quad (\text{E.15})$$

$$g_1 = \exp \left\{ -1/2 [(1/2)(I(c) + I(d)) - m_{1i}]^2 / \sigma_{1i}^2 \right\}, \quad (\text{E.16})$$

$$g_2 = \exp \left\{ -1/2 [|I(c) - I(d)| - m_{2i}]^2 / \sigma_{2i}^2 \right\}, \quad (\text{E.17})$$

onde I é a intensidade de imagem (em níveis de cinza neste caso), (m_{1i}, σ_{1i}) , (m_{2i}, σ_{2i}) são a intensidade média e desvio padrão da imagem e de seu gradiente em uma vizinhança em torno dos pontos sementes.

A função $\mu_\alpha(c, d)$ é tal que:

$$\mu_\alpha(c, d) = 1, \text{ se } |c - d| = 1, \quad (\text{E.18})$$

$$\mu_\alpha(c, d) = 0, \text{ caso contrário.}$$

- Brigger, P.; Engel, R. & Unser, M. *B-spline Snakes and JAVA interface: An Intuitive Tool for General Contour Outlining*. Tech. rept. Biomedical Imaging Group, Swiss Fed. Inst. of Tech.
- Bulpitt, A. J. & Efford, N. D. 1996. An efficient 3D deformable model with self-optimising mesh. *Image and Vision Computing*, **14**, 573–580.
- Burt, P. J.; Hong, Tsai-Hong & Rosenfeld, A. 1981. Segmentation and Estimation of Image Region Properties Through Cooperative Hierarchical Computation. *IEEE Trans. on Systems, Man, and Cybernetics*, **SMC-11**(12), 802–809.
- Cameron, G. 1995. Modular Visualization Environments: Past, Present, and Future (Special Focus). *IEEE Computer Graphics*, May.
- Canny, J. F. 1983. *Finding Edges and Lines in Images*. Tech. rept. MIT, <http://www.mit.edu/>.
- Carmo, M. P. 1979. *Geometria Riemanniana*. Livros Técnicos e Científicos Editora S.A.
- Catte, F.; Lions, Pierre-Louis; Morel, Jean-Michel & Coll, T. 1992. Image Selective Smoothing and Edge Detection by Nonlinear Diffusion. *SIAM J. Num. Analysis*, **29**(1), 182–193.
- Chapra, C. & Canale, R. P. 1988. *Numerical Methods for Engineers*. McGraw-Hill International Editions.
- Cohen, L. D. 1991. On Active Contour Models and Balloons. *CVGIP:Image Understanding*, **53**(2), 211–218.
- Cohen, L.D. & Cohen, I. 1993. Finite-element methods for ACTIVE contour models and balloons for 2-D and 3-D images. *IEEE Trans. on Pattern Analysis and Machine Intel.*, **15**(11).
- Corporation. IBM. 1993. *IBM Visualization Data Explorer: Programmer's Reference*. 4nd edn. IBM.
- Curwen, R. & Blake, A. 1993. *Active Vision*. MIT Press. Chap. 3, "Dynamic Contours: Real-Time Active Splines", pages 39–57.
- Davatzikos, C. A. & Prince, J. L. 1995. An Active Contour Model for Mapping the Cortex. *IEEE Transactions on Medical Imaging*, **14**(1), 65–80.
- Davatzikos, C.A. & Prince, J.L. 1999. Convexity Analysis of Active Contour Algorithms. *Image and Vision Computing*, **17**(1), 27–36.

- Dolgoplova, T. F. & Ivanov, V. K. 1966. On numerical differentiation. *USSR Comp. Math. and Math. Phys.*, **6**, 570–576.
- Dreyfus, S. 1965. *Dynamic Programming and the Calculus of Variations*. New York: Academic.
- Dubrovin, B. A.; Fomenko, A. T. & Novikov, S. P. 1984. *Modern Geometry - Methods and Applications (The Geometry of Surfaces, Transformation Groups, and Fields)*. Vol. 1. Springer-Verlag.
- Durikovic, R.; Kaneda, K. & Yamashita, H. 1995. Dynamic contour: a texture approach and contour operations. *The Visual Computer*, **11**, 277–289.
- Falcão, Alexandre X.; Udupa, J.K.; Samarazekera, S. & Hirsch, B. E. 1996. User-Steered image boundary segmentation. *Proceedings of SPIE on Medical Imaging, Newport Beach, CA*.
- Florack, L. M. J.; Romery, B. M. Ter Haar; Koenderink, J. & Viergever, M. A. 1992. Scale and the differential structure of images. *Image and Vision Computing*, **10**(6), 376–388.
- Florack, L.M.; Salden, A.H.; Romery, B.M. Ter Haar; Koenderink, J.J & Viergever, M.A. 1995. Nonlinear scale-space. *Image and Vision Computing*, **13**(4), 279–294.
- Geiger, D.; Gupta, A.; Costa, L. A. & Vlontzos, J. 1995. Dynamic Programming for Detecting, TRacking, and Matching Deformable Contours. *IEEE Trans. Pattern Anal. Mach. Intell.*, **17**(3), 294–302.
- Giraldi, G.; Strauss, E.; Apolinario, A. & Oliveira, A. F. 2000a. An Initialization Method for Deformable Models. *In: Computer Graphics and Imaging (CGIM'2000)*.
- Giraldi, G. A. & Oliveira, A. F. 1999. *Convexity Analysis of Snake Models Based on Hamiltonian Formulation*. Tech. rept. Universidade Federal do Rio de Janeiro, Dep. Eng. Sistemas e Computação, <http://www.cos.ufrj.br/relatorios/reltec99/>.
- Giraldi, G. A. & Oliveira, A. F. 2000. Dual-Snake Model in the Framework of Simplicial Domain Decomposition. *In: International Symposium on Computer Graphics, Image Processing and Vision (SIBGRAPI'199 (Technical Poster)*.
- Giraldi, G. A.; Strauss, E. & Oliveira, A. F. 2000b. A Boundary Extraction Approach Based on Multi-resolution Methods and the T-Snakes Framework. *In:*

International Symposium on Computer Graphics, Image Processing and Vision (SIBGRAPI'2000 (to appear).

- Giraldi, G. A.; Strauss, E. & Oliveira, A. F. 2000c. A Boundary Extraction Method Based on Dual-T-Snakes and Dynamic Programming. *In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '2000).*
- Giraldi, G. A.; Gonçalves, L. M. & Oliveira, A. F. 2000d. Dual Topologically Adaptable Snakes. *Pages 103–106 of: Proceedings of the Fifth Joint Conference on Information Sciences (JCIS'2000, Vol. 2) - Third International Conference on Computer Vision, Pattern Recognition, and Image Processing.*
- Giraldi, G. A.; Strauss, E. & Oliveira, A. F. 2000e. An Initialization Method for Active Contour Models. *In: Proceedings of the 2000 International Conference on Imaging Science, Systems, and Technology (CISST'2000).*
- Goldstein, H. 1981. *Classical Mechanics*. 2nd edn. Addison-Wesley.
- Golub, G. H. & Loan, C. F. Van. 1985. *Matrix Computation*. The Johns Hopkins University Press.
- Gool, L. V.; Moons, T.; Powrls, E. & Oosterlinck, A. 1995. Vision and Lie's approach to invariance. *Image and Vision Computing*, 13(4), 259–277.
- Gunn, S. R. 1996a. *Dual Active Contour Models for Image Feature Extraction*. Ph.D. thesis, Faculty of Engineering and Applied Science, Dep. of Eletronics and Comp. Science.
- Gunn, S. R. & Nixon, M. S. 1997. A Robust Snake Implementation; A Dual Active Contour. *IEEE Trans. Pattern Anal. Mach. Intell.*, 19(1), 63–68.
- Gunn, Steve R. 1996b (May). *Dual Active Contour Models for Image Feature Extraction*. Ph.D. thesis, Faculty of Engineering and Applied Science, Department of Eletronics and Computer Science.
- Heitz, F.; Perez, P. & Bouthemy, P. 1994. Multiscale Minimization of Global Energy Functions in Some Visual Recovery Problems. *CVGIP: Image Understanding*, 59(1), 125–134.
- Hendee, W. R. & Wells, P. N. T. 1997. *The Perception of Visual Information*. 2nd edn. Springer.
- Herman, G. T. & Carvalho, B. M. Multiseede Segmentation Using Fuzzy Connectedness. *Short Paper, Medical Image Processing Group, Dep. Radiology, Un. Pennsylvania.*

- Hirsch, C. 1988. *Numerical Computation of Internal and External Flows*. Vol. 1. Livros Técnicos e Científicos Editora S.A., RJ-Brasil.
- Honig, C. S. 1976. *Aplicações da Topologia à Análise*. Editora Edgar Blucher, Ltda, SP-Brasil.
- Hummel, A. 1986. Representations Based on zero-crossings in scale-space. *Pages 204–209 of: Proc. IEEE Computer Vision and Pattern Recognition Conf.*
- Ip, H H. S. & Shen, D. 1998. An affine-invariant active contour model (AI-snake) for model-based segmentation. *Image and Vision Computing*, 16(2), 135–146.
- Jain, A. K.; Duin, R. & Mao, J. 2000. Statistical Pattern Recognition: A Review. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(1), 4–37.
- Jain, Anil K. 1989. *Fundamentals of Digital Image Processing*. Prentice-Hall, Inc.
- Jolion, J. M. & Montanvert, A. 1992. The Adaptive Pyramid: A framework for 2D Image Analysis. *CVGIP: Image Understanding*, 55(3), 339–348.
- Jones, Timothy N. & Metaxas, Dimitris N. June, 1998. Image Segmentation Based on the Integration of Pixel Affinity and Deformable Models. *In: Proceedings of CVPR'98*.
- Junior, R. I. & Iorio, V. M. 1988. *Equações Diferenciais Parciais: uma introdução*. Livros Técnicos e Científicos Editora S.A., RJ-Brasil.
- Kass, M.; Witkin, A. & Terzopoulos, D. 1988. Snakes: Active Contour Models. *International Journal of Computer Vision*, 1(4), 321–331.
- Koenderink, J. 1984. The structure of Images. *Biol. Cybern.*, 50, 363–370.
- Lai, K. F. & Chin, R. T. 1995. Deformable Contours: Modeling and Extraction. *IEEE Trans. Pattern Anal. Mach. Intell.*, 17(11), 1084–1090.
- Leymarie, F. & Levine, M. D. 1993. Tracking Deformable Objects in the Plane Using an Active Contour Model. *IEEE Trans. Pattern Anal. Mach. Intell.*, 15(6), 617–634.
- Liang, J.; McInerney, T. & Terzopoulos, D. 1999a. Iterative Medical Image Segmentation with United Snakes. *In: MICCAI'99*.
- Liang, J.; McInerney, T. & Terzopoulos, D. 1999b. United Snakes. *In: ICCV'99*.
- Lima, E. L. 1985. *Curso de Análise*. Vol. 2. Livros Técnicos e Científicos Editora S.A.

- Lorensen, W. E. & Cline, H. E. 1987. Marching Cubes: A High Resolution 3D Surface Construction Algorithm. *Computer Graphics*, **21**(4), 163–169.
- MA, Song De & Li, Bingcheng. 1998. Derivative computation by multiscale filters. *Image and Vision Computing*, **16**, 43–53.
- Malladi, R.; Sethian, J. A. & Vemuri, B. C. 1995. Shape Modeling with Front Propagation: A Level Set Approach. *IEEE Trans. Pattern Anal. Mach. Intell.*, **17**(2), 158–175.
- Marr, D. C. & Hildreth, E. C. 1980. Theory of Edge Detection. *Proc. Roy. Soc. London B*, **207**, 187–217.
- Martins, N.; Pauli, R. U. & Mauad, F. C. 1979. *Física para a Universidade: Vol.1, Análise Dimensional*. Editora Pedagógica e Universitária LTDA, E.P.U, São Paulo.
- McInerney, T. & Terzopoulos, D. 1995 (June). Topologically Adaptable Snakes. *Pages 840–845 of: Proc. Of the Fifth Int. Conf. On Computer Vision (ICCV'95), Cambridge, MA, USA*.
- McInerney, T. & Terzopoulos, D. 1997 (March). Medical Image Segmentation Using Topological Adaptable Surfaces. *In: Proc. CVRMed'97*.
- McInerney, T. & Terzopoulos, D. 1999. Topology Adaptive Deformable Surfaces for Medical Image Volume Segmentation. *IEEE Trans. on Medical Imaging*, **18**(10), 840–850.
- McInerney, T. J. 1997. *Topologically Adaptable Deformable Models for Medical Image Analysis*. Ph.D. thesis, Department of Computer Science, University of Toronto.
- Mignotte, M. & Meunier, J. February, 2000. An Unsupervised Multiscale Approach for the Dynamic Contour-Based Boundary Detection Issue in Ultrasound Imagery. *Pages 366–369 of: Proceedings of the Fifth Joint Conference on Information Sciences (JCIS'2000, Vol. 2) - Third International Conference on Computer Vision, Pattern Recognition, and Image Processing*.
- Mumford, D. & Shah, J. 1989. Optimal Approximations by Piecewise Smooth Functions and Associated Variational Problems. *Comm. Pure Appl. Math.* **42**, 577–684.
- Nayfeh, A. H. & Mook, D. T. 1979. *Nonlinear Oscillations*. John Wiley Sons, a Wiley-Interscience Publication.

- Olver, P. J. 1986. *Applications of Lie Groups to Differential Equations*. Springer-Verlag, Berlin, (1986).
- P. Bamford, B. Lovell. 1998. *Unsupervised Cell Nucleus Segmentation with Active Contours*. Tech. rept. Dep. Eletrical and Comp. Eng., Univ. of Queensland, Australia, <http://www.cssip.elec.uq.edu.au/>.
- Perona, P. & Malik, J. 1990. Scale-Space and Edge Detection Using Anisotropic Diffusion. *IEEE Trans. on Patter Analisis and Mach. Intell.*, **12**(7), 629–639.
- Peterfreund, N. 1999. Robust Tracking of Position and Velocity With Kalman Snakes. *IEEE Trans. on Pattern Analysis and Machine Intel.*, **21**(6).
- Poggio, Tomaso; Torre, Vincent & Koch, Christof. 1985. Computational Vision and regularization theory. *Nature*, **317**(26), 314–319.
- Qin, Hong & Terzopoulos, D. 1996. D-NURBS: A Physics-Based Framework for Geometric Design. *IEEE Trans. on Pattern Analysis and Machine Intel.*, **2**(1), 85–96.
- R. Fisker, J. M. Carstensen. 1998 (August 16-20). On parameter estimation in deformable models. *Pages 762–766 of: 14th International Conference on Pattern Recognition*.
- Rogers, David F. 1985. *Procedural Elements for Computer Graphics*. MacGraw-Hill Book Company.
- Samtaney, R.; Silver, D.; Zabusky, N. & Cao, J. 1994. Visualizing Features and Tracking Their Evolution. *IEEE Computer*, July, 20–27.
- Sapiro, G. 1995 (August). *Color Snakes*. Tech. rept. Hewlett-Packard Labs, guille@hpl.hp.com.
- Sapiro, G. 1997. Color Snakes. *Computer Vision and Image Understanding*, **68**(2), 247–253.
- Schimpf, P.; Hauelsen, J.; Ramon, C. & Nowak, H. 1998. Realistic computer modeling of electric and magnetic fields of human head and torso. *Parallel Computing*, **24**, 1433–1460.
- Sethian, J. A. 1996. *Level Set Methods: Evolving Interfaces in Geometry, Fluid Mechanics, Computer Vision and Materials Sciences*. Cambridge University Press, 1996.

- Sethian, S. Osher J. A. 1988. Fronts Propagation with Curvature-Dependent Speed: Algorithms Based on Hamilton-Jacobi Formulations. *Journal of Computational Physics*, **79**, 12–49.
- Snyder, M. A. 1991. On the Mathematical Foundations of Smoothness Constraints for the Determination of Optical Flow and for Surface Reconstruction. *IEEE Trans. on Pattern Analysis and Mach. Intell.*, **13**(11), 1105–1114.
- Sotomayor, Jorge. 1979. *Lições de Equações Diferenciais Ordinárias*. Gráfica Editora Hamburg, Ltda.
- Terzopoulos, D. 1986a. Image Analysis Using Multigrid Relaxation Methods. *IEEE Trans. Pattern Anal. Mach. Intell.*, **8**(2), 129–139.
- Terzopoulos, D. 1986b. Regularization of Inverse Visual Problems Involving Discontinuities. *IEEE Trans. on Pattern Analysis and Machine Intel.*, **8**(4).
- Udupa, J. & Samarasekera, S. 1996. Fuzzy connectedness and object definition: Theory, algorithms and applications in image segmentation. *Graphical Models, Image Processing*, **58**(3).
- van Ginneken, B. & ter Haar Romery, B. M. *Automatic segmentation of lungs fields in chest radiographs*. Tech. rept. <http://www.isi.uu.nl>.
- Williams, C. K. I.; Revo, M. & Hinton, G. E. 1997. Instantiating Deformable Models with a Neural Net. *Computer Vision and Image Understanding*, **68**(1), 120–126.
- Witkin, A. 1983. Scale-Space Filtering. *Pages 1019–1021 of: Int. Joint Conf. Artificial Intelligence, Karlsruhe, Germany*.
- Witten, G. 1993. Scale Space Tracking and Deformable Sheet Models for Computational Vision. *IEEE Trans. Pattern Anal. Mach. Intell.*, **15**(7), 697–706.
- W.J. Niessen, B.M. ter Haar Romery & Viergever, M.A. 1998. Geodesic Deformable Models for Medical Image Analysis. *IEEE Trans. on Medical Imaging*, **17**(4).
- Xu, Gang; Segawa, E. & Tsuji, S. 1994. Robust Active Contours With Insensitive Parameters. *Pattern Recognition*, **27**(7), 879–884.
- Zhu, S. C. & Yuille, A. 1996. Region Competition: Unifying Snakes, Region Growing, and Bayes/MDL for Multibang Image Segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, **18**(9), 884–900.