

UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA ELÉTRICA
DEPARTAMENTO DE ENGENHARIA DA COMPUTAÇÃO E AUTOMAÇÃO INDUSTRIAL

Reconstrução Tridimensional por Ajuste de Superfícies Paramétricas

por Roberto Marcondes Cesar Junior

orientador Prof. Dr. Roberto de Alencar Lotufo

Dissertação submetida à Faculdade de Engenharia Elétrica da Universidade Estadual de Campinas, para preenchimento dos pré-requisitos parciais para obtenção do Título de Mestre em Faculdade de Engenharia Elétrica.

Este exemplar é a coleção final da tese
defendida por Roberto Marcondes Cesar Junior
em 26/11/93
pelo Conselho de Engenharia Elétrica da Comissão

janeiro 1994

Resumo

Este trabalho apresenta os principais aspectos sobre uma abordagem de reconstrução tridimensional (3D) de objetos por ajuste de superfícies paramétricas. A reconstrução 3D se insere como um tópico da visualização volumétrica.

O processo de reconstrução 3D pode ser dividido em 3 etapas gerais : segmentação de contornos; casamento de segmentos entre contornos; e interpolação ou aproximação da superfície paramétrica. Cada uma dessas etapas é apresentada e discutida. São apresentados os detalhes do desenvolvimento e implementação de um método de reconstrução 3D por superfícies de Coons. A etapa de segmentação e representação de contornos é feita com auxílio de curvas B-Spline. Para a descrição dos segmentos, foi criado um descritor que é calculado a partir do polígono de controle de cada segmento. O casamento de segmentos é feito por um método baseado no algoritmo de busca em grafos conhecido como A^* . O ajuste de superfícies é feito por "patches" de Coons.

Esta dissertação contribui em 3 aspectos principais: melhoria do método de segmentação de contornos proposto por Medioni, criação de um descritor para segmentos de B-Spline e desenvolvimento de um método de casamento de segmentos que se baseia no algoritmo A^* de busca em grafos. São apresentados os resultados utilizando imagens sintéticas e experimentais, obtidas por digitalização por "scanner" e por um processo de tomografia Computadorizada de raio-x.

Abstract

An approach to three-dimensional (3D) reconstruction of objects by parametric surfaces, a topic from volume visualization, is described.

The parametric surface 3D reconstruction can be divided in three steps : contour segmentation; segment matching; and parametric surface formation. These steps are presented and discussed. A method for 3D reconstruction by Coons surfaces is developed and presented. B-Splines are used for contour segmentation and representation. The guiding-polygon is used for contour description. Segment matching is achieved by an algorithm based on the A^* graph search method. Surface formation is performed by Coons's blending.

This thesis contributes in 3 main aspects: improvement of the contours segmentation method proposed by Medioni, definition of a B-Spline segment descriptor, and developement of a segment matching method, based on the A^* graph search method. Some results from both synthetic and experimental images are presented.

Agradecimentos

"(...) with a little help from my friends (...)"

No lugar de agradecimentos, pois nada que eu dissesse daria conta de quanto devo, fica aqui uma pequena e humilde lembrança, com todo amor, a tantas pessoas cujos caminhos acabaram cruzando o meu, me ensinando e criando um sorriso enorme em meu rosto...

(...inclusive aquelas que não sabem porque...)

...meus pais, meus irmãos, minha vó Nair, meu saudoso vô João, meus tios, meus compadres Renato e Edilene, meu afilhado e todos os outros que, desde pequeno, me ensinam e alegram...

...Dri, linda...

...meu orientador, Lotufo...

...pessoas fantásticas que tive tanta sorte de conhecer em Rio Preto, companheiros de intermináveis conversas sobre vida, ciência, música, arte, livros, novelas ou qualquer outra coisa, onde o que menos importa é o assunto, Sebaka, Marquinhos, Silvya, Paty, Lisa, Vá, Oscar, Jaca, Renatão, Benga, Lia e o resto do povo da Lua, Mary Blown, Sel e todas que já foram cabiludas um dia, pessoal da moradia, Mazza, Bê e outros agitadores, Buça, Véio, e o resto do povo do samba, grande canalha do Bino, Japa-Sam, Elaine, Haza, Vadinho, Giba, Marcão, Rô, Pablito, Dedéia, Bel, Geisa, Lí, Mayuca, Alê e qualquer um que já tenha ajudado no CIP, Kibe, Luis Paulo, Conterrânea, Nenê, Tibas, Rodrigo, Gêngis, Pastel, Zé Lôco, Paulinho do sax...

...meus amigos que ficaram em Pinda, em especial o Marelo, Bicudo e Robertinho...

...as pessoas que vim conhecer ou reencontrar aqui em Campinas, Lili, Sussu, Fridão, Clecinho, Jari (Sumo Sacerdote de Lós), Éder, Ratão, Mari, Rininha, Bel, Helinho,

os músicos e o resto do povo da vila São João, Claudião e a vó, todos da capoeira, Cintia, Tião, Mirão, Sonia, Fabinho, Birigui e todos que já tenham ido numa festa da Bel, Dani, Zoninha, Mandruvis (o velho lobo do bar), Geraldinho, Fernandinho...

...Lê, Silvinha, Marcião e Zenon, GRANDE pessoal aqui de casa...

...e, claro, o pessoal aqui da Elétrica, Alexandre (grande cumpanheiro), Gonzaga (seu Luis, respeita Januário), Marião, Armando, Myriam, Jorginho, Olguinha, Renato, Castanho, Taninha, Berutti, Ilka, Francesco, Ulisses, Gonzalo, Déia, Ricardo, Lú, Valéria, Ju, Ari, Gugu, Alencar, Felipão, Samarone, Marcinho, Perla...

...meus professores de Rio Preto, em especial o Chaim, Farid, Inês, Maurílio, Eliana, Tadini, Cordeiro, Chahine, Marcelo, Socorro, Júlio, Eurípides, Miro, Sueli, Adalberto...

...alguns daqui que me ajudaram muito nesta dissertação, Márcio, Pedro, Ting (ajuda sem tamanho), Hermano, Clésio...

...FAPESP, que bancou minha iniciação científica desde 89, e agora o mestrado...

...e todos os outros que, pela loucura que é terminar uma dissertação, acabaram ficando de fora, pelo meu esquecimento...

...para meu pai, que para o azar da minha mãe, adora acordar cedo, e para o prof. Odelar,
um cara que sabe das coisas, e que tanto me ensinou sobre ciência...

"(...) - Pois não, Sr. Settembrini. Essa licença lhe dei uma vez por tôdas. É muito amável da sua parte. E o termo "filho enfermiço da Vida" é bonito. Quanta coisa não inventam os escritores ! Não sei se devo orgulhar-me dêsse título ; mas êle soa bem, indiscutivelmente ! Pois é, eu me dedico um pouquinho a êsses "filhos da Morte". Acho que é a isso que o senhor se refere. Às vezes, quando tenho tempo, e sem que o regime sofra com isso, ocupo-me com os casos graves e sérios; compreende ? com aquêles que não estão aqui para divertir-se e para entregar-se à licenciosidade, mas que estão morrendo.

- Está escrito : "Deixai que os mortos enterrem os seus mortos !" - replicou o italiano.

Hans Castorp ergueu os braços e expressou com a sua fisionomia que existia muita coisa que estava escrita, isso e também aquilo, de maneira que era difícil discernir o melhor e inspirar-se nêle. (...)"

Diálogo entre o jovem Hans Castorp e o pedagogo Ludovico Settembrini no fantástico "A Montanha Mágica", de Thomas Mann, 1924.

Conteúdo

RESUMO	i
ABSTRACT	ii
AGRADECIMENTOS	iii
CONTEÚDO	ii
LISTA DE FIGURAS	vi
1 Introdução	1
1.1 Objetivos	1
1.2 Organização do trabalho	4
2 Revisão sobre Reconstrução 3D	5
2.1 Abordagens para Reconstrução 3D	7
2.1.1 Modelo Abstrato da Aquisição dos Dados	7
2.1.2 Reconstrução 3D por Ajuste de Superfícies	8
2.1.3 Reconstrução 3D por Processamento de Voxels Binários	13
2.1.4 "Rendering" de Volume Semitransparente	16
2.2 Reconstrução 3D por Ajuste de Superfícies Paramétricas	17
2.3 Pré-processamento	22

2.3.1	Segmentação	23
2.3.2	Correspondência entre contornos e Subdivisões	25
3	Segmentação e Representação de Contornos	29
3.1	Introdução	29
3.2	Representação de Curvas no Domínio Espacial : Revisão Bibliográfica . . .	33
3.3	Segmentação e Representação de Contornos por B-Splines	36
3.3.1	O Algoritmo Modificado	39
3.3.2	Estimativa da Curvatura	40
3.3.3	Estimativa da Curvatura em Curvas Paramétricas.	41
3.3.4	Estimativa da Curvatura usando B-Splines.	42
3.3.5	Ajuste dos Pontos de Segmentação e de Controle	43
3.4	Descrição de Contornos através do Polígono de Controle	46
4	Casamento de Segmentos	55
4.1	O problema de correspondência	55
4.2	O Algoritmo A^*	58
4.2.1	Busca em um Espaço de Estados	58
4.2.2	"Best-First Search" e o A^*	59
4.2.3	As Listas OPEN e CLOSED	61
4.2.4	A Função Heurística	62
4.2.5	O Algoritmo A^*	62
4.3	O Processo de Casamento de Segmentos	67
4.3.1	Exemplo de Casamento entre Dois Contornos	67
4.3.2	O Grafo de Busca	71
4.3.3	A Função Custo $c(m, n)$	71
4.3.4	As Funções h' , explode(n) e goal(n)	72

5	Ajuste e Visualização de Superfícies Paramétricas	77
5.1	Introdução	77
5.2	Superfícies Paramétricas	78
5.2.1	Curvas Paramétricas Cúbicas e Representação Matricial	79
5.2.2	Curvas e Superfícies de Hermite	81
5.2.3	Superfícies de Coons	85
5.2.4	Superfícies de Bézier	89
5.2.5	Superfícies B-Spline	89
5.2.6	B-Splines Racionais	89
5.3	Geração de Superfícies	90
5.3.1	Avaliação Iterativa das Equações	91
5.3.2	Técnicas de Subdivisão	92
5.4	Visualização	92
6	Implementação e Resultados Obtidos	94
6.1	Introdução	94
6.2	Estruturas de Dados	95
6.3	Etapas da Reconstrução 3D e Ambientes Utilizados	98
6.4	Resultados Obtidos	100
6.4.1	Pré-processamento	100
6.4.2	Representação e Segmentação	100
6.4.3	Casamento de Contornos	101
6.4.4	Ajuste de Superfícies	102
6.5	Análise dos Resultados e Principais Problemas Encontrados	102
7	Conclusão	114
7.1	Sobre o que foi feito: Comentários finais	114

7.2	Sobre o que se pode fazer: Sugestões para trabalhos futuros	115
7.2.1	Pré-processamento: Correspondência entre Contornos	115
7.2.2	Segmentação e Representação de Contornos	116
7.2.3	Descrição e Casamento de Contornos	117
7.2.4	Ajuste de Superfícies	119

BIBLIOGRAFIA

120

Lista de Figuras

2.1	Plano cortando objeto de interesse.	8
2.2	Curva obtida em uma secção.	8
2.3	Dois planos seccionais.	8
2.4	Planos seccionais paralelos.	9
2.5	Obtenção da matriz 3D de dados.	9
2.6	Formação da superfície por triangularização.	10
2.7	Triangularização entre dois contornos.	10
2.8	União das superfícies entre cada par de contornos	10
2.9	Triângulo formado pelos vértices q_i, q_k, p_j	11
2.10	Matriz de dados em forma de voxels.	14
2.11	Esfera centrada na origem.. . . .	19
2.12	Curva resultante da intersecção entre a esfera e o plano.	19
2.13	Segmentos de curva sobre a superfície.	20
2.14	Formação da superfície "lofted".	20
2.15	Contornos consecutivos P e Q.	20
2.16	Formação da superfície entre os contornos P e Q.	21
2.17	Curva segmentada.	21
2.18	Contornos P e Q segmentados.	22
2.19	Esquema do processo de ajuste de superfícies por segmentação.	22

2.20	Imagem original, segmentada e contornos extraídos.	25
2.21	Caso mais simples de correspondência entre contornos.	26
2.22	Problema de ramificações : (a) superfície com subdivisão ; (b) correspondência entre contornos extraídos.	26
2.23	Situações complexas no problema de correspondência.	27
2.24	Método de correspondências : (a) contornos ; (b) grafo criado ; (c) árvore geradora do grafo.	28
2.25	Reconstrução 3D de um toro : (a) toro seccionado ; (b) contornos extraídos.	28
3.1	Representação do objeto "quadrado" por seu contorno (a) e sua região (b).	30
3.2	Descrição do contorno por quatro vetores e um ponto.	31
3.3	Aproximação poligonal : (a) de um contorno fechado; (b) de uma onda.	34
3.4	Vizinhança do ponto p_i no contorno.	40
3.5	Avaliação da curvatura através da vizinhança.	40
3.6	Ajuste da função S para o cálculo analítico da curvatura.	41
3.7	Pontos com curvatura elevada na mesma vizinhança.	43
3.8	Exemplo de ponto de segmentação que deve ser eliminado pelo cálculo do ângulo.	45
3.9	Exemplos das situações em que os ajustes desenvolvidos se aplicam.	46
3.10	Contorno segmentado.	46
3.11	Segmento de B-Spline e vetores do polígono de controle.	47
3.12	Vetores v_1 e v_2 na origem.	48
3.13	Ponto médio do segmento.	49
3.14	Gráfico $T' \times C'$	52
3.15	Gráfico $T \times C$ normalizado.	52
3.16	Situações que o atributo h procura diferenciar.	54
4.1	Caso geral do problema de casamento entre dois contornos.	57

4.2	Exemplo do funcionamento do algoritmo "best-first search".	60
4.3	Espaço de estados e "best-first search".	61
4.4	Avaliação de h' : (a) $h' \leq h$ e (b) $h' > h$	65
4.5	Estruturas usadas pelo A^*	67
4.6	Situação inicial para o casamento dos contornos B e C	68
4.7	Casamento dos pontos b_0 e c_0	69
4.8	Primeira explosão no espaço de estados.	70
4.9	Casamento dos segmentos b_0b_1 e c_0c_1	70
4.10	Segunda explosão no espaço de estados.	70
4.11	Situação de junção no casamento (entre os segmentos c_1c_2 e c_2c_3).	71
4.12	Situação de sucesso no espaço de estados.	72
4.13	Os contornos B e C casados.	72
4.14	Processo de casamento visto em uma grade.	73
5.1	Bézier e polígono de controle	92
6.1	Representação do espaço de estados por uma árvore binária.	97
6.2	(a) representação de um nó do espaço de estados; (b) e (c) exemplo de uma configuração do espaço de estados e sua estrutura correspondente.	104
6.3	Tabela de resultados com letras digitalizadas (legenda : ap. pol. 1 - aproximação poligonal pelos pontos de segmentação; ap. pol. 2 - aproximação poligonal por pontos de segmentação e de controle; ncont - número de pontos do contorno; nseg - número de pontos de segmentação; nrep - número total de pontos (segmentação + controle).	105
6.4	Tabela de resultados com imagem de tomografia.	106
6.5	Contorno representado por B-Spline pelo método original.	107
6.6	Contorno representado por B-Splines com ajuste global.	107
6.7	Representação por B-Splines com pontos de segmentação e de controle ($\square$).	108
6.8	Tabela de resultados do casamento de segmentos.	109

6.9	Representação do casamento de segmentos entre os contornos.	109
6.10	Casamento de segmentos.	110
6.11	Contornos segmentados obtidos a partir de imagem de tomografia.	111
6.12	Representação do casamento de segmentos dos contornos da cabeça.	111
6.13	Reconstrução 3D por ajuste de superfícies de Coons dos contornos da fig. 6.10.112	
6.14	Reconstrução 3D por ajuste de superfícies de Coons dos contornos da fig. 6.8.112	
6.15	Estrutura "wireframe" de uma reconstrução 3D.	113
6.16	Estrutura "wireframe" de uma reconstrução 3D de um joelho.	113
7.1	Distâncias dos pontos de controle à reta suporte do segmento.	116
7.2	Situação em que o teste de distância pode alterar a estrutura do segmento.	116
7.3	Casamento com junção em ambos contornos.	118

Capítulo 1

Introdução

1.1 Objetivos

A visualização volumétrica é uma área da computação de imagens que trata de métodos para a reconstrução de modelos de objetos tridimensionais [12]. A reconstrução tridimensional (3D) de imagens é o processo pelo qual obtém-se o modelo tridimensional do objeto a partir de suas informações bidimensionais [35].

As técnicas para reconstrução 3D de imagens podem ser classificadas em 3 categorias principais: processamento de voxels binários, "rendering" de volume semitransparente e ajuste de superfícies [34].

No processamento de voxels binários, a matriz formada pelo empilhamento das secções é tratada como uma matriz de pequenos cubos, que são a unidade básica de informação tridimensional. O termo que denomina esses cubos é **voxel** (do inglês "volume element").

A abordagem de "rendering" de volume semitransparente é uma generalização da anterior, onde uma função C_λ de cor, com λ = azul, verde ou vermelho, e uma função α de opacidade são definidas para cada voxel. As imagens são formadas pela composição dos voxels segundo os raios de projeção sobre um plano de visualização. A cor e transparência de cada pixel são calculadas com base nas funções C_λ e α .

Na terceira categoria, uma superfície é ajustada segundo os pontos do contorno

de cada secção. Para cada secção (plano seccional), determina-se a curva que corresponde à intersecção desse plano com o objeto, e une-se cada par de curvas, relativas a secções consecutivas, com uma primitiva geométrica.

Os métodos de reconstrução 3D por ajuste de superfícies podem ser genericamente divididos em 3 etapas principais:

- (1) Segmentação dos contornos;
- (2) Casamento de segmentos entre os contornos;
- (3) Ajuste de "patches";

A segmentação resulta numa representação mais compacta do contorno, preservando suas características de forma. A segmentação de contornos pode ser feita de diversas maneiras, mas a forma mais utilizada se baseia nos métodos de aproximação poligonal.

O casamento de segmentos é realizado para o estabelecimento de uma relação de correspondência entre os contornos. Dois aspectos principais estão ligados ao casamento de contornos: critério de correspondência e o algoritmo de casamento.

O ajuste de "patches" define a superfície final. As superfícies mais utilizadas para a reconstrução 3D são aquelas formadas por triangularização, mas outros tipos de superfície podem ser utilizados.

Esta dissertação apresenta um estudo sobre a implementação de um método de reconstrução tridimensional (3D) por ajuste de superfícies paramétricas. O método desenvolvido nesta dissertação trata do caso mais simples, onde existe apenas 1 objeto e um contorno por secção.

O método de segmentação desenvolvido realiza a representação de contornos por segmentos de B-Splines. Ele foi criado a partir de uma modificação do método proposto por Medioni em [27]. A modificação proposta nesta dissertação melhora o método original sob dois aspectos: descrição mais compacta dos dados e melhor caracterização estrutural dos segmentos.

O critério de correspondência para o casamento de contornos se baseia num descritor de segmentos proposto. Esse descritor possui 4 atributos : o vetor diretor, que

expressa informação de sentido do segmento; o ponto médio, que localiza o segmento no plano; o tamanho relativo do segmento; e um vetor caracteriza estruturalmente o segmento.

O casamento de segmentos, segunda etapa do método proposto de reconstrução 3D, foi desenvolvido a partir da generalização do método de casamento proposto por Kehtarnavaz em [61]. O algoritmo de casamento implementado se baseia em uma técnica de busca em grafos conhecida como A^* . A generalização do método é conseguida pela incorporação de informação heurística ao A^* . Essa informação, que é definida pelo número de segmentos e o custo médio das arestas do grafo, não é utilizada por Kehtarnavaz [61].

A etapa de ajuste e visualização de "patches" de Coons é feita por 3 ambientes : o ambiente de modelagem de curvas e superfícies "Model", o programa para "rendering" de superfícies SCANLINE, e um programa de visualização de estruturas " wireframe". Esses programas, que foram desenvolvidos no DCA - FEE - UNICAMP, foram modificados para que pudessem ser intergrados ao método de reconstrução 3D.

A visualização volumétrica é uma linha de pesquisa do Grupo de Computação de Imagens do Departamento de Computação e Automação da Faculdade de Engenharia Elétrica - FEE - UNICAMP. Nesse sentido, esta dissertação se insere como sendo um tópico dos estudos e programas desenvolvidos pelos professores e alunos de pós-graduação e iniciação científica. São desenvolvidas pesquisas nas abordagens de processamento de voxels binários e "rendering" de volume semitransparente, além do ajuste de outros tipos de superfícies, como as obtidas por triangularização.

Esta dissertação contribui em 3 aspectos principais: melhoria do método de segmentação de contornos proposto por Medioni em [27], criação de um descritor para segmentos de B-Splines e desenvolvimento de um método de casamento de segmentos que se baseia no algoritmo A^* de busca em grafos. Foi feita uma extensa revisão bibliográfica sobre cada uma das etapas do processo de reconstrução 3D por ajuste de superfícies paramétricas.

A técnica desenvolvida foi testada em imagens reais e simuladas, e comprovam sua eficiência e limitações.

1.2 Organização do trabalho

No próximo capítulo, são apresentados uma introdução às 3 principais abordagens de reconstrução 3D (sec. 2.1), os primeiros conceitos de reconstrução 3D por ajuste de superfícies paramétricas (sec. 2.2), e considerações sobre o pré-processamento para essas aplicações (sec. 2.3).

O capítulo 3 trata da etapa de representação, segmentação e descrição dos contornos por B-splines.

No capítulo 4, o problema de casamento de segmentos entre contornos consecutivos é apresentado. O método de casamento desenvolvido, baseado no algoritmo A^* , é definido e discutido.

Uma revisão às superfícies paramétricas é apresentada no capítulo 5.

No capítulo 6, são discutidos os aspectos principais da implementação de cada etapa do método de reconstrução 3D desenvolvido. São apresentados ainda resultados obtidos em testes dos programas criados e adaptados.

Finalmente, o capítulo 7 apresenta os comentários finais acerca do que foi feito, e conclui esta dissertação com uma discussão sobre possíveis trabalhos futuros.

Capítulo 2

Revisão sobre Reconstrução 3D

A visualização volumétrica é uma área da computação de imagens que trata de métodos para a reconstrução de modelos objetos tridimensionais [12]. A reconstrução tridimensional (3D) de imagens é o processo pelo qual obtém-se um modelo tridimensional do objeto a partir de informações bidimensionais deste [35]. Três componentes principais são identificadas nas atividades que fazem uso do processo de visualização volumétrica : visualização, análise e manipulação [47].

A visualização trata de métodos relativos à exibição do modelo em um dispositivo apropriado (por exemplo, a tela do computador). A reconstrução 3D é de importância fundamental para o processo de visualização.

Na análise, a meta principal é se fazer medições no modelo do objeto de forma absoluta, comparativa e combinativa.

A manipulação tem por finalidade alterar o modelo do objeto de forma interativa, tanto para auxílio na visualização e análise quanto em simulações como, por exemplo, planejamento de cirurgias em aplicações médicas.

Aplicações de visualização volumétrica são encontradas em estudos de espécimes biológicas [8], análise anatômica e morfológica de órgãos em ciências biomédicas [55], interpretação sísmica 3D [4], CAD/CAM [19], e outros.

Em microscopia, embora muitas vezes o objeto de estudo seja tridimensional, os microscópios monoculares não permitem que essa estrutura seja observada. Se esse objeto

for suficientemente transparente, o microscópio pode ser focalizado em diversos níveis, fornecendo secções do objeto. Se, por outro lado, o objeto for opaco, então deverá ser cortado, e cada lâmina possuirá uma secção deste. É desejável a reconstituição da estrutura 3D original, para que possa ser manipulada e analisada.

Em aplicações de CAD, existe uma técnica para a criação de superfícies denominada "lofting", onde o usuário especifica uma seqüência de curvas, arbitrariamente posicionadas no espaço modelo, e o sistema gera a superfície que interpola essas curvas. Essa técnica, particularmente, pode gerar superfícies complexas a partir de um conjunto de curvas seccionais paralelas.

A distribuição da temperatura de uma região aérea pode ser empregada para se inferir a temperatura em regiões não amostradas. Um conjunto de leituras, feitas em diferentes rotas por um avião de reconhecimento, é usado para a criação de curvas espaciais, onde as coordenadas x e y correspondem à posição do avião, e a coordenada z à temperatura lida. Uma superfície que interpola essas curvas fornece a informação inferida desejada.

A microscopia eletrônica e o método de análise por difração de raios-x fornecem dados 3D para a visualização de macro-moléculas, como o DNA ou o RNA.

A partir de imagens seccionais, obtidas por tomografia computadorizada de raio-x, um modelo 3D pode ser criado e usado no planejamento de cirurgias, fornecendo informações mais completas sobre, por exemplo, uma fratura.

Equipamentos para a aquisição de dados tiveram grande desenvolvimento nos últimos anos, incluindo a tomografia computadorizada, a ressonância magnética, medicina nuclear [20], microscopia eletrônica [30, 31], e difração de raio-x [45].

As técnicas para reconstrução 3D de imagens são classificadas em 3 categorias principais : ajuste ou interpolação de superfícies, processamento de voxels binários e "rendering" de volume semitransparente [34].

Para a interpolação de superfícies, dados os pontos de contorno representados em secções, deve-se interpolar uma superfície por esses pontos. Para cada secção (plano seccional), determina-se a curva que corresponde à intersecção desse plano com o objeto, e une-se cada par de curvas, relativas a secções consecutivas, com uma primitiva geométrica. Quando duas secções consecutivas estiverem muito distantes uma da outra, a interpolação

pode gerar uma imagem pouco natural. Neste caso, pode ser desejável a criação de curvas intermediárias.

No processamento de voxels binários, a matriz formada pelo empilhamento das secções é tratada como uma matriz de pequenos cubos, que são a unidade básica de informação tridimensional. O termo que denomina esses cubos é **voxel** (do inglês "volume element").

A abordagem de "rendering" de volume semitransparente é uma generalização da anterior, onde uma função C_λ de cor, com λ = azul, verde ou vermelho, e uma função α de opacidade são definidas para cada voxel. As imagens são formadas por uma composição, projetando-se raios através dos voxels sobre um plano de visualização. A cor e transparência de cada pixel são calculadas com base nas funções C_λ e α .

2.1 Abordagens para Reconstrução 3D

2.1.1 Modelo Abstrato da Aquisição dos Dados

Muitas aplicações de processamento de imagens e computação gráfica (síntese) envolvem o "display" de uma superfície reconstruída a partir de uma seqüência de contornos planares. Esses contornos são obtidos de dados adquiridos por um dispositivo a partir do objeto tridimensional original, no caso de aplicações para processamento de imagens e visualização. Em aplicações de CAGD ("Computer Aided Geometric Design", Projeto Geométrico Auxiliado por Computador), o usuário especifica os contornos em várias secções de uma superfície que deseja desenvolver.

Pode-se encarar o processo de formação dos dados como a intersecção entre um plano seccional e o objeto de interesse (fig. 2.1). As informações contidas numa secção de um objeto são resultantes dessa intersecção, que pode ser pensada como uma imagem em níveis de cinza. Os contornos da superfície a ser reconstruída são obtidos através de um processo de segmentação e extração de contornos em cada secção [81] (fig. 2.2).

Esses conceitos se assemelham um pouco à idéia de curvas de nível de uma superfície.

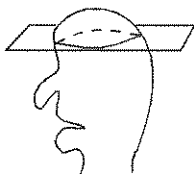


Figura 2.1: Plano cortando objeto de interesse.

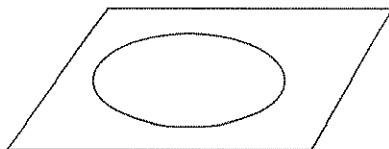


Figura 2.2: Curva obtida em uma secção.

Agora, se o processo descrito acima for repetido em posições diferentes do objeto, obtém-se um conjunto de secções deste (fig 2.3). Neste trabalho, será tratado o caso em que as secções são obtidas por cortes paralelos (fig. 2.4).

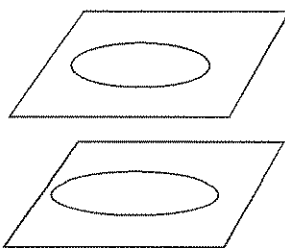


Figura 2.3: Dois planos seccionais.

Ordenando e empilhando-se as secções na orientação do eixo z , forma-se uma matriz tridimensional de dados (fig. 2.5). Cada elemento dessa matriz tem associado um valor obtido pelo dispositivo de aquisição de dados.

2.1.2 Reconstrução 3D por Ajuste de Superfícies

O problema de interpolação de superfícies a dados tridimensionais é colocado de formas diferentes na literatura consultada. As várias caracterizações do problema dão origem a métodos diferentes de interpolação. Porém, uma característica presente na maioria dos métodos dessa abordagem é o tratamento explícito de contornos (curvas) em cada secção

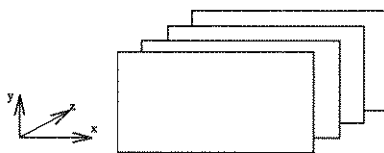


Figura 2.4: Planos seccionais paralelos.

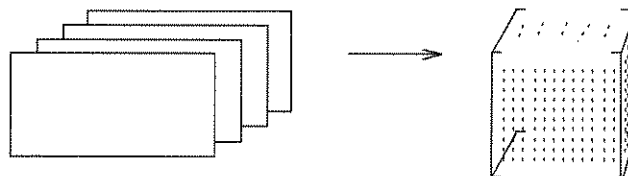


Figura 2.5: Obtenção da matriz 3D de dados.

do objeto.

Essa característica aproxima muito os conceitos dessa abordagem a idéia de curvas de nível de uma superfície. Isso é devido à não discretização do espaço, quando os dados são tratados. Essa observação só não é válida em [79], onde mesmo essa abordagem é tratada como se os dados fossem voxels num espaço discretizado, ao contrário dos outros trabalhos, onde os dados são pontos dispersos no espaço.

A intersecção entre esses planos paralelos e o objeto de interesse fornece, depois da segmentação, os contornos, que são curvas das superfícies exteriores e interiores do objeto. Esses contornos podem ser fechados, abertos ou pontos. O objetivo do algoritmo de reconstrução de superfícies é a formação de **retalhos de superfícies** ("surface patches") entre os contornos presentes em secções adjacentes para, dessa forma, conseguir uma aproximação do objeto original. Um exemplo desse processo é a formação dessa superfície por triangularização (fig. 2.6). Esse tipo de algoritmo de reconstrução foi um dos primeiros a surgir [35]. Os triângulos que formarão a superfície são formados por segmentos de reta ligando pontos de contornos adjacentes (fig. 2.7).

O problema de reconstrução 3D é transformado no problema de construção de uma superfície, composta por triângulos, entre cada par de contornos que pertencem a secções consecutivas do objeto. Assim, a estrutura final da superfície reconstruída é resultante da união das superfícies geradas entre cada par de contornos consecutivos (fig. 2.8).

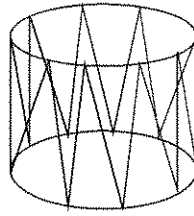


Figura 2.6: Formação da superfície por triangulação.

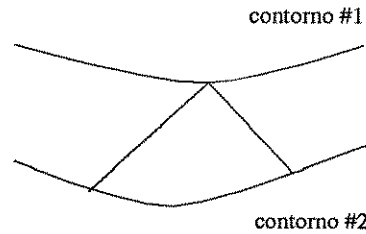


Figura 2.7: Triangulação entre dois contornos.

Segundo [35], sejam os contornos consecutivos $P = p_0, p_1, \dots, p_m$ e $Q = q_0, q_1, \dots, q_n$, onde p_i e q_j são pontos componentes dos contornos P e Q . Se os contornos são fechados, tem-se $p_0 = p_m$ e $q_0 = q_n$. A superfície que será contruída entre os contornos P e Q , como já explicado anteriormente, deverá ser composta por triângulos. Assim, um plano triangular entre os contornos P e Q deverá ter como vértices uma das formas p_i, p_k, q_j ou q_i, q_k, p_j . A figura 2.9 ilustra um dos casos.

Dessa forma, se os pontos q_i, q_k, p_j forem considerados vértices de um grafo, e $q_i p_j, p_j q_k, q_k q_i$ arestas, tem-se que a superfície que interpola os contornos P e Q é formada pelos triângulos compreendidos pelas arestas de um grafo toroidal. A construção desse grafo pode ser baseada em funções custo, como área da superfície construída, volume do sólido

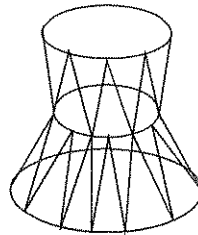


Figura 2.8: União das superfícies entre cada par de contornos

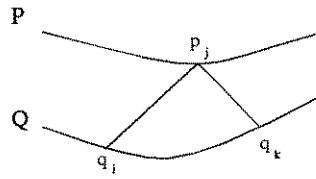


Figura 2.9: Triângulo formado pelos vértices q_i, q_k, p_j

compreendido por essa superfície, comprimento das arestas e outras.

Em [58] é apresentada uma caracterização mais geral de um contorno numa secção. Tem-se que um contorno pode ser :

- (1) uma curva fechada;
- (2) uma curva aberta;
- (3) um ponto;

O método de triangularização apresentado em [58] permite a manipulação de muitos contornos por plano, mapeamento parcial de contornos e oferece, ainda, um mecanismo para intervenção humana, durante o processo de interpolação. Essa intervenção é necessária, por causa da deficiência do método em tratar casos de subdivisões ("branching") da superfície. Esse problema será melhor explicado na secção 2.3.2.

Também baseado em triangularização é o método introduzido em [14]. O critério utilizado é o de minimização da área da superfície. Esse trabalho apresenta soluções para o problema de correspondência entre os contornos e ramificações da superfície. Depois da triangularização, uma superfície paramétrica, onde os vértices dos triângulos são usados como pontos de controle, é ajustada para se conseguir um melhor resultado visual.

Uma outra forma de se construir a superfície entre os contornos (também formada por triângulos) é apresentada em [38]. Ao invés da construção explícita de sua superfície, um sólido, formado por poliedros com faces triangulares, é construído. A superfície é formada pela intersecção do sólido por planos tangentes aos contornos P e Q. Esse sólido é contruído utilizando-se conceitos de geometria computacional ligados à triangularização de Delaunay [25]. É interessante notar que métodos de geometria computacional começam a aparecer e serem estudados cada vez mais em problemas de reconstrução 3D [63].

Tem-se ainda em [61, 60] a utilização de informação estrutural para a reconstrução de superfícies. Estes são os primeiros trabalhos em que técnicas de reconhecimento sintático de padrões são utilizadas [48]. A inovação destes métodos consiste na aproximação poligonal [61] ou por segmentos de curva [60] dos contornos, e na decomposição destes em cadeias de primitivas. Essas cadeias irão servir como informação sobre a estrutura global da superfície a ser interpolada, enquanto que, nos outros métodos, a superfície é criada utilizando-se apenas informações locais dos contornos. As vantagens da utilização de informações estruturais em problemas ligados ao reconhecimento de padrões e processamento de imagens são discutidas em [71]. No primeiro trabalho, a superfície é criada por triangularização, numa abordagem similar a [35], enquanto que no segundo, tem-se a formação de superfícies paramétricas, e as curvas não precisam ser obtidas em secções, podendo ter 3 graus de liberdade no espaço.

A caracterização do problema em [57] é feita segundo o conceito de voxels, abordagem também seguida em [79]. Nesse trabalho, é apresentado um sistema de reconstrução 3D baseado na interpolação de superfícies paramétricas, sendo que as discussões são direcionadas a reconstrução da articulação do joelho. Os métodos que usam o conceito de voxels na abordagem de superfícies geralmente se baseiam na construção de isosuperfícies ("isosurfaces") [3] formadas pelos pontos que possuem uma mesma propriedade.

Existem ainda métodos que procuram ajustar superfícies paramétricas aos contornos. Em [61] [52], os contornos são segmentados, e esses segmentos definem os "patches" da superfície aproximadora. Em [61], são utilizadas superfícies de Coons, enquanto que em [52], são usadas superfícies de Hermite.

O método proposto por [87] pode ser traduzido no algoritmo 2.1.

Este método permite a interpolação de superfícies com subdivisões ("branching"), além de fornecer um resultado relativamente mais "suave" do que os outros, em relação às superfícies geradas. Os contornos intermediários são gerados por um método de interpolação dinâmica elástica ("dynamic elastic interpolation") [18], enquanto que a interpolação dos contornos é feita por uma superfície quadrática.

Com base na análise dos métodos levantados na literatura, pode-se apontar os pontos relevantes quanto ao desenvolvimento de algoritmos para interpolação de superfícies. Os primeiros métodos que surgiram para essa interpolação são baseados na técnica de

algoritmo 2.1 (Algoritmo de [87]) :

- 1 - *Algoritmo de interpolação de superfície.*
- 2 - *É dado um par de contornos consecutivos C_i e C_{i+1} .*
- 3 - *Gerar um conjunto de contornos intermediários.*
- 4 - *Interpolar os contornos.*
- 5 - *Fim.*

triangularização. Isso equivale a dizer que a superfície final forma um objeto poliédrico. Este fato pode acarretar uma característica indesejável da superfície gerada : a falta de "suavidade", dando origem a uma imagem com aspecto grosseiro ("coarse") [87].

Além disso, o problema de subdivisões ("branching") da superfície só veio a ser resolvido de forma sistemática em [38], quando a interpolação é feita por triangularização. Mas, mesmo assim, como foi observado anteriormente, com uma formulação diferente do problema, não mais utilizando a abordagem proposta por [35]. A diversificação dos métodos de interpolação de superfícies começa a percorrer caminhos que levam a análise da estrutura global dos contornos, como proposto em [61].

Finalmente, tem-se em [60, 87, 57], métodos que utilizam a interpolação por outros tipos de superfícies. O aspecto dessas superfícies mostra-se mais "suave" que as obtidas por triangularização, porém com um refinamento matemático mais elaborado.

2.1.3 Reconstrução 3D por Processamento de Voxels Binários

Para esta abordagem a matriz de dados obtida é tratada como um conjunto de voxels, como mostrado na figura 2.10.

Considerar-se-á a presença de apenas um objeto no cubo de voxels, o que não invalida a discussão subsequente.

Agora, dada a matriz de entrada, o primeiro passo para se realizar a visualização do objeto é identificar os elementos que compõem esse objeto. Uma propriedade importante desses elementos é que eles formam um conjunto conexo [12]. Esta fase do processo de

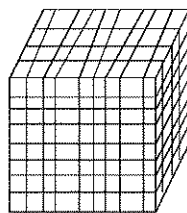


Figura 2.10: Matriz de dados em forma de voxels.

reconstrução 3D é denominada segmentação.

Uma vez identificados os elementos que compõem o objeto, ser-lhes-á atribuído o valor 1 de luminosidade, e 0 aos restantes. Isso resultará numa representação binária da cena, ou seja, uma imagem binária [41].

O próximo passo é identificar a face externa do objeto, dando origem a uma superfície a ser visualizada. Não é difícil perceber que essa superfície será formada por faces dos voxels mais externos do objeto.

Assim, se essa superfície for identificada, pode-se projetá-la num plano de visualização, utilizando-se técnicas de computação gráfica para dar a ilusão de terceira dimensão na visualização do objeto.

Um último fato importante a se perceber é a explícita utilização de técnicas de processamento de imagens e visão computacional nessa abordagem. Técnicas de segmentação, conceitos sobre imagens binárias e detecção de contornos podem ser utilizadas para o entendimento dessa abordagem. Porém, não se deve esquecer que os métodos são aplicados a imagens bidimensionais, e devem ser reavaliados e redefinidos para dados tridimensionais.

Segmentação

No caso da tomografia computadorizada, os dados da imagem são obtidos pela medição da atenuação do raio-x em cada voxel. Essa medição é expressa em relação a água, segundo a "escala de Hounsfield" (G. Hounsfield, premio Nobel 1979). Tem-se que, por exemplo, o osso possui uma densidade ou atenuação de raio-x maior que os tecidos ou o ar. Portanto, o osso possui um número de Hounsfield maior que o do ar. Assim, a segmentação dos ossos por limiar pode ser feita descartando-se os voxels que possuem um número de

Hounsfield menor que um determinado valor [10].

Existem métodos que fazem a detecção de bordas depois de se conseguir a imagem binária. Porém, em imagens com grande quantidade de ruído e inconsistências, o cálculo de um valor para se realizar a segmentação torna-se difícil e muito suscetível a erros. Desse fato surge a importância do desenvolvimento de métodos para a detecção de bordas em imagens com vários níveis de cinza. A detecção de bordas pode ser feita pela utilização de operadores de gradientes que fornecem a variação de intensidade de luz em cada ponto.

Nos trabalhos levantados na literatura, tem-se [10, 12] e [50], utilizando segmentação por limiar, e o desenvolvimento de métodos de detecção de bordas em [39, 4], sendo que os dados utilizados por este último são provenientes de dados sísmicos 3D.

É muito importante notar que a segmentação do objeto é um passo a ser cumprido também pela abordagem de interpolação de superfícies, onde os contornos referentes ao objeto em cada uma de suas secções devem ser identificados. Tem-se ainda que, em [39], primeiro identifica-se os contornos do objeto em cada plano seccional, e a borda do objeto é obtida "empilhando-se" esses contornos.

Formação e visualização da superfície

O resultado da segmentação de um objeto do resto da imagem é um conjunto conexo de voxels. Para se fazer a visualização desse objeto, deve-se identificar a superfície formada pela borda deste. Isso pode ser conseguido calculando-se as faces dos voxels mais externos do objeto que estão entre um voxel com valor de intensidade 1 e outro com valor de intensidade 0 (em imagens binárias!). Esse é um passo que consome muito tempo, nessa abordagem.

Em [12] é apresentado um algoritmo que visita 2/3 das faces componentes dessa superfície uma vez e 1/3 duas vezes. Uma vez identificada uma face, o algoritmo calcula as faces que estão conectadas a esta e que fazem parte da superfície. Em [39], como já foi citado, esse passo é conseguido "empilhando-se" contornos identificados nos planos seccionais. Em [4] o algoritmo de "surface tracking" apresentado baseia-se em operadores de detecção de bordas.

Finalmente, a visualização da superfície identificada é feita utilizando-se métodos

de computação gráfica conhecidos como "rendering" de superfícies.

As desvantagens existentes nessa abordagem também estão presentes na interpolação de superfícies. Dentre elas, a principal é a classificação binária de dados, resultando em perda de informação, como dados entre superfícies e as dificuldades em se tratar objetos muito pequenos ou mal definidos [79]. Por outro lado, uma grande vantagem dessas duas abordagens, em relação a última, é a facilidade em se trabalhar com a superfície resultante, tanto na compactação dos dados para o armazenamento, quanto na utilização de métodos de computação gráfica para se fazer a visualização.

2.1.4 "Rendering" de Volume Semitransparente

"Rendering" de volumes é uma técnica para visualização de dados tridimensionais, computando-se projeções num plano de imagens, através de um conjunto de voxels. Esta técnica é uma variante da abordagem apresentada na seção anterior. A principal diferença que esta abordagem apresenta, em relação as outras duas, é a não utilização de classificação binária dos dados. Esta abordagem não realiza a reconstrução de um modelo do objeto, como as duas anteriores, mas faz a visualização diretamente a partir do volume de dados.

Em [20] tem-se a descrição geral dos passos que compõem os principais métodos dessa abordagem. Dada a matriz de voxels, tem-se as transformações C_λ e α associadas a cada elemento dessa matriz.

O primeiro passo a ser realizado por um sistema de visualização volumétrica por "rendering" de volumes é a classificação [20], que atribuirá a cada voxel as porcentagens de cada material presentes nele. Por exemplo, utilizando-se a densidade de Hounsfield, pode-se identificar principalmente 4 tipos de material componentes do corpo humano : ar, gorduras, tecidos e ossos.

A técnica de classificação estima a probabilidade de um material estar homogeneamente presente num voxel. Essa probabilidade representa a quantidade de material nesse elemento. Portanto, se um voxel possui probabilidade de 0.95 de ser formado só por osso, então é atribuído a ele um percentual de 95% de osso.

Essa técnica é conhecida como classificação estatística ou probabilística. A transformação $\alpha(x)$ de opacidade, $x = (i,j,k)$, terá seus valores definidos de acordo com o resultado da classificação [53]. Paralelamente, utilizando-se um modelo de iluminação, são calculados os valores assumidos pela transformação C_λ em cada voxel.

O próximo passo é a formação da imagem. Faz-se uma composição das matrizes formadas pelos valores de C_λ e α e um mapeamento dessa composição a um plano de imagem, ou o plano de visualização (e.g. a tela do computador). Raios correspondendo a cada pixel atravessam as matrizes de C_λ e α , e uma soma da contribuição de cada voxel na formação da imagem é feita ao longo desses raios. As equações que definem essa soma são encontradas em [53, 20, 54].

A principal característica da abordagem de "ray-casting" é o tratamento direto e explícito dos dados tridimensionais para a visualização do objeto [45]. Dessa característica decorre uma grande vantagem dessa abordagem em relação as outras duas, que dispensa a necessidade da classificação binária dos dados [34].

Decorre também a maior desvantagem dos métodos dessa abordagem, que é o excessivo esforço computacional que o processamento demanda. A razão desse fato é o tratamento de todos os voxels, com o custo computacional proporcional ao tamanho do volume [79].

Existe outra diferença também interessante entre essa abordagem e as outras. Enquanto que, nas duas primeiras, formava-se uma superfície para posterior visualização, nessa abordagem a etapa de visualização está embutida no próprio método de reconstrução.

Acontece que a visualização de superfícies possui muitos métodos conhecidos e testados com relativo sucesso (inclusive testados em imagens médicas [86]), enquanto que, para dados tridimensionais, isso não é verdade [79].

2.2 Reconstrução 3D por Ajuste de Superfícies Paramétricas

As superfícies paramétricas encontram aplicações das mais variadas em síntese e processamento de imagens [82]. Essa referência identifica duas classes de aplicações que diferenciam essas superfícies. A primeira diz respeito a um uso **sintético** das superfícies,

encontrado nas aplicações ligadas à concepção de um objeto. Um projetista interativamente cria ou modifica um modelo desejado, examinando e melhorando até que seja aceitável. A segunda está relacionada a um uso **analítico** dessas superfícies, onde elas estarão representando informações que são medidas de certa forma; assim como uma curva pode ser ajustada a um conjunto de pontos (por exemplo, temperatura medida em diversas altitudes), uma superfície pode ser ajustada a "propriedades medidas" de um objeto real. Exemplo : contornos gerados a partir de uma imagem obtida por tomografia.

Para a continuação dessa discussão, são apresentadas definições e exemplos para a introdução de conceitos importantes.

definição 2.1 (superfícies paramétricas) *Uma superfície paramétrica é uma superfície do tipo*

$$P(u, v) = (x(u, v), y(u, v), z(u, v))$$

onde u e v são os parâmetros.

exemplo 2.1 *As equações paramétricas de uma esfera de raio unitário centrada na origem são dadas por (fig. 2.11)*

$$x(u, v) = \cos u * \cos v$$

$$y(u, v) = \cos u * \sin v$$

$$z(u, v) = \sin u$$

$$u \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right]; \quad v \in [0, 2\pi].$$

Fixando-se o valor de um dos parâmetros e variando o outro obtém-se uma curva sobre a superfície (fig. 2.12).

exemplo 2.2 *Na equação paramétrica do exemplo anterior, fixar um valor de u equivale a calcular a curva resultante da intersecção da esfera e de um plano paralelo ao plano XoY . A equação da curva resultante pode ser obtida analiticamente. Tome-se $u = 0$. Logo*

$$x(0, v) = \cos v$$

$$y(0, v) = \sin v$$

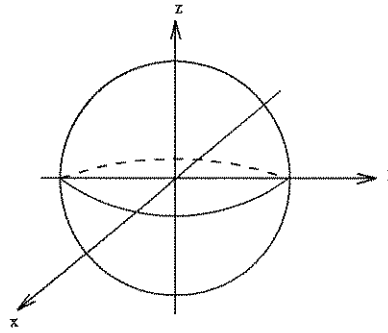


Figura 2.11: Esfera centrada na origem..

$$z(0,v) = 0 ; \quad v \in [0, 2\pi]$$

que será a equação paramétrica de uma circunferência no plano $z = 0$.

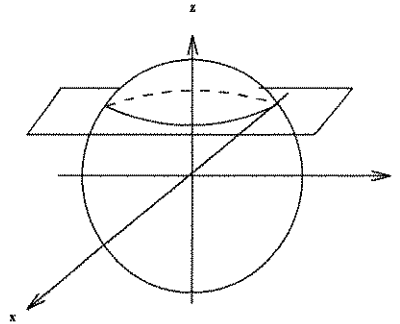


Figura 2.12: Curva resultante da intersecção entre a esfera e o plano.

É claro que o parâmetro v pode ser tomado em apenas um sub-conjunto do seu domínio. Nesse caso, a curva resultante seria um segmento de circunferência.

Tome-se então $P(u, v)$ uma superfície paramétrica e sejam u_0 e u_1 dois pontos pertencentes ao domínio de u . Tem-se que $P(u_0, v)$ e $P(u_1, v)$ definem duas curvas sobre a superfície $P(u, v)$ (fig. 2.13).

definição 2.2 (superfícies "lofted") *Sejam $P(u_0, v)$ e $P(u_1, v)$ curvas como as apresentadas na fig. 2.13. Então*

$$S(u, v) = (1 - u)P(u_0, v) + uP(u_1, v)$$

é definida como uma superfície "lofted".

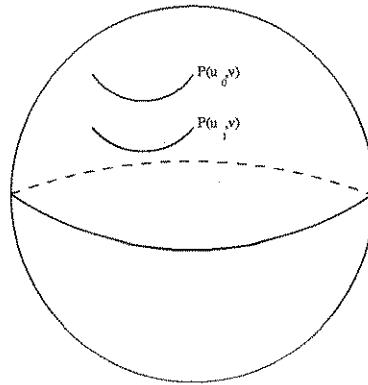


Figura 2.13: Segmentos de curva sobre a superfície.

Esse tipo de superfície "lofted" é produzida como uma interpolação linear entre duas curvas (fig. 2.14), sendo gerada a partir de segmentos de reta que unem essas curvas. A superfície $S(u, v)$ é uma aproximação de $P(u, v)$ entre os segmentos de curva $P(u_0, v)$ e $P(u_1, v)$.

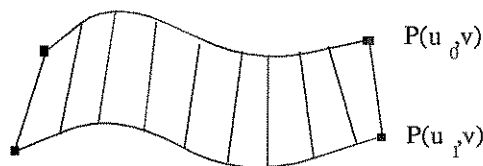


Figura 2.14: Formação da superfície "lofted".

De volta agora ao problema de reconstrução 3D, sejam 2 contornos consecutivos P e Q (fig. 2.15).

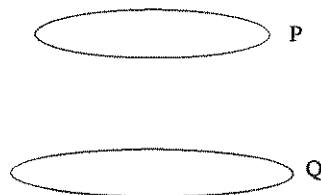


Figura 2.15: Contornos consecutivos P e Q .

Partindo do raciocínio anterior, uma forma de se ajustar uma superfície paramétrica a esses dois contornos, por exemplo uma superfície "lofted", pode ser descrita em dois passos :

(1) ajuste ou interpolação dos contornos P e Q por curvas paramétricas, $P(v)$ e $Q(v)$. Fazendo-se uma analogia à definição 1.2, tem-se $P(v) \equiv P(u_0, v)$ e $Q(v) \equiv P(u_1, v)$.

(2) cálculo da superfície $S(u, v)$ de acordo com a fórmula da definição 1.2 (fig. 2.16).

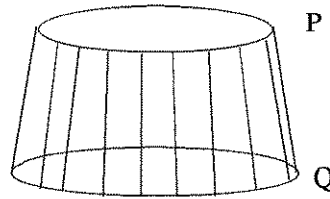


Figura 2.16: Formação da superfície entre os contornos P e Q.

Existem, no entanto, dificuldades escondidas nesse procedimento que podem impedir o sucesso desse processo. O passo (1) diz respeito a uma aproximação matemática, por meio de uma curva paramétrica, de uma curva planar qualquer. Mas, a menos que a curva a ser aproximada seja extremamente simples, o ajuste de uma curva paramétrica a toda curva pode ser inviável; uma abordagem preferível, e mais usada, é a da **aproximação por partes** [83]. Assim, a aproximação dos contornos P e Q seria feita segmentando-os e aproximando-os por segmentos de curva (fig. 2.17).

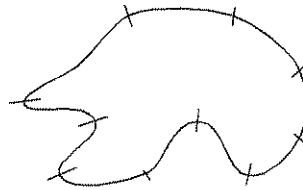


Figura 2.17: Curva segmentada.

Sejam agora $P(u) = p_0(u), p_1(u), \dots, p_m(u)$ e $Q(u) = q_0(u), q_1(u), \dots, q_n(u)$ as aproximações por partes dos contornos P e Q (fig. 2.18). Dessa forma, também a superfície será ajustada por partes. Esses "retalhos" de superfície, definidos por pares de segmentos de contornos consecutivos, são chamados "**patches**" [15]. Aqui aparece um segundo problema importante : como identificar quais os pares de segmentos que devem definir um "patch" ? É preciso fazer-se um "casamento" de segmentos ("**segment matching**") para que os pares sejam identificados. Também deve-se notar que o número de segmentos pode diferir

de contorno para contorno. Uma forma de resolver esse problema é a divisão ("split") ou junção ("merge") dos segmentos.

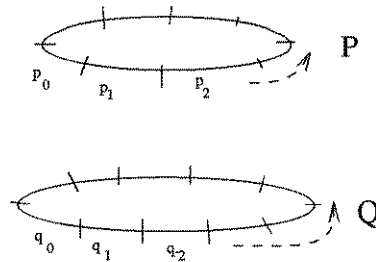


Figura 2.18: Contornos P e Q segmentados.

Finalmente, uma vez que os segmentos foram casados, o ajuste da superfície pode ser feito de muitas formas, dependendo do tipo da superfície que é ajustada. O problema de ramificações de superfícies não é tratado nessa abordagem inicial.

Um esquema dos passos envolvidos é mostrado na figura 2.19.

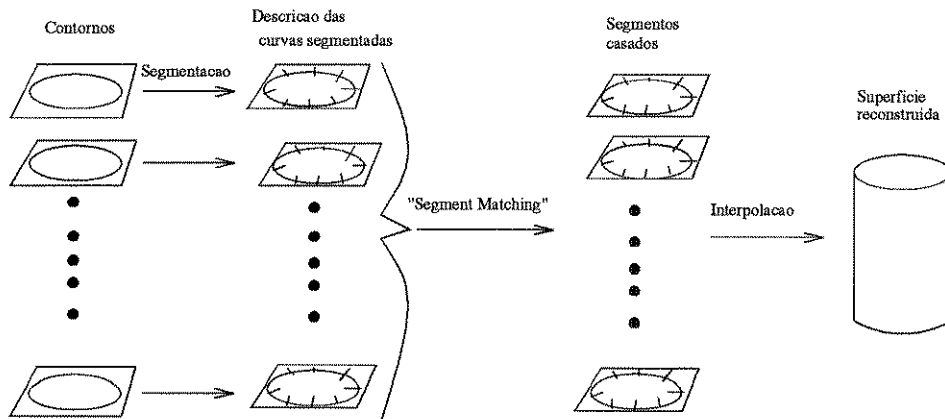


Figura 2.19: Esquema do processo de ajuste de superfícies por segmentação.

2.3 Pré-processamento

Uma vez que os dados 3D, representados nas secções consecutivas foram obtidos, as primeiras tarefas do processo de reconstrução 3D por ajuste de superfícies são :

- (1) Pré-processamento e segmentação dos contornos.

(2) Identificação de correspondência entre os contornos.

(3) Identificação e tratamento de subdivisões ("branching") entre contornos.

O pré-processamento é um tópico bastante geral relacionado a processamento de imagens em si; as tarefas envolvidas nessa etapa estão diretamente relacionadas ao tipo de imagem em questão, e às necessidades de cada aplicação. A filtragem de ruídos, realce de imagens ou de bordas são tarefas envolvidas nessa etapa. Os outros tópicos, por serem mais diretamente ligados ao tema desta dissertação, são abordados mais detalhadamente nas secções subseqüentes.

2.3.1 Segmentação

A segmentação procura, basicamente, dividir a imagem em regiões que possuam determinadas características comuns. Trata-se de uma partição da imagem. Uma outra forma de se entender a segmentação é encarando-a como um processo que separa os objetos do fundo de uma imagem.

Para problemas que tratam de dados tridimensionais, existem duas abordagens para a segmentação : em cada secção ou na própria matriz 3D.

Dentro dessas duas abordagens, por sua vez, pode-se dizer que existem 3 formas principais de se fazer a segmentação : análise de características, detecção de bordas e crescimento de regiões [49]. A segmentação por análise de características, que pode ser pensada como um problema de abordagem estatística para o reconhecimento de padrões, procura classificar cada elemento da imagem, seja ele um pixel ou um voxel, enquadrando-o em uma determinada classe pré-definida, baseado em uma ou mais características desse elemento. Um caso particular extremamente importante desse tipo de segmentação é a por **limiar** ("thresholding"). Os métodos de detecção de bordas procuram identificar descontinuidades de intensidade entre elementos vizinhos que compõem a imagem. Diretamente ligados à detecção de bordas estão os operadores de gradientes, que procuram realçar as bordas ao mesmo tempo que filtram ruídos na imagem. Por último, os métodos de crescimento de regiões começam com uma ou mais sementes que pertencem a uma determinada região, e seguem agregando outros elementos, segundo um critério de semelhança, até formar cada objeto. Por outro lado, a despeito da riqueza de estudos sobre segmentação, não existe

um método "perfeito" e geral de segmentação. O ideal de um processo de segmentação é que ele forneça regiões que possuam significado sobre a imagem e, muitas vezes, ocorre que as regiões que possuem esse significado não possuem as mesmas características visuais. A partir dessa constatação, existem métodos que procuram utilizar conhecimentos específicos de cada aplicação, definindo **modelos** dos objetos que são esperados em uma determinada imagem. Esses modelos podem tanto corresponder a trechos de bordas [49] ou ao objeto esperado [89], quanto a descrições deste.

Assim, [49] introduz uma abordagem sintática para a segmentação de imagens 2D por detecção de bordas. Um conjunto de primitivas, formado por retas e cantos ideais, é usado para a definição de uma gramática. A partir desta, um autômato é criado e utilizado para se calcular o melhor casamento entre o conjunto de primitivas e possíveis trechos de borda que ocorram na imagem. Por último, um procedimento de ligação de bordas é utilizado para completar o processo. Em [39] é apresentado um método para segmentação de imagens 3D, utilizando operadores de gradiente para a detecção de bordas e um algoritmo de busca em grafos para "boundary following". O operador de gradientes faz parte da função custo associada ao grafo.

Em [78] a segmentação é feita por análise de regiões utilizando o princípio "split-and-merge" [81]. O método se baseia na análise de características das regiões em testes de hipóteses estatísticos. Esse trabalho [78] traz uma extensa revisão de métodos de análise de homogeneidade de regiões, principalmente por histograma.

A segmentação e análise de imagens 3D possui outras aplicações que não a reconstrução 3D; em [77] são apresentados exemplos de segmentação 3D na abordagem espaço-temporal para análise de cenas em visão computacional, em que uma seqüência de imagens 2D, que variam no tempo, são tomadas como uma imagem 3D, onde o tempo é a terceira dimensão. O método de segmentação introduzido nesse trabalho também é voltado para a abordagem de modelos. O método se divide em duas partes que interagem : **partição** do volume em sub-volumes, e **identificação**, que justifica se a partição feita é válida ou não, tentando casar os sub-volumes com um conjunto de modelos 3D. O método proposto, além de obter a segmentação, procura fornecer uma descrição, em termos dos modelos 3D, de cada sub-volume segmentado. O modelo 3D usado, assim como as descrições dos objetos segmentados, são definidos segundo uma imagem 3D onde cada voxel é tomado como uma

variável aleatória. A descrição é feita usando técnicas de regressão linear.

A figura 2.20 mostra um exemplo de uma imagem original e outra segmentada, tornando-se binária.

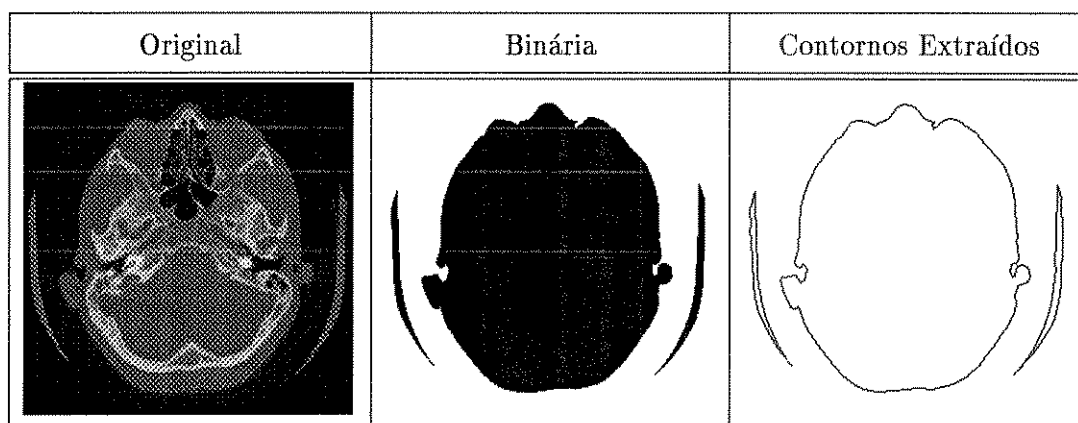


Figura 2.20: Imagem original, segmentada e contornos extraídos.

Uma vez que a imagem 3D, ou cada imagem seccional, foi segmentada, deve-se extrair os contornos de cada objeto de cada secção (fig. 2.20). No caso de uma segmentação geral, contornos diferentes podem corresponder a objetos diferentes na cena 3D original. Assim, em uma imagem obtida por tomografia de uma cabeça, existe um contorno que corresponde à pele, outro que corresponde ao osso, e assim por diante. O próximo passo é o estabelecimento da correspondência entre os contornos de cada secção. Além dos contornos poderem pertencer a objetos diferentes, contornos de um mesmo objeto podem se relacionar de maneiras complexas. Essas questões são discutidas na próxima secção.

2.3.2 Correspondência entre contornos e Subdivisões

A forma mais simples da disposição dos contornos ocorre quando só existe um objeto e os contornos se comportam de forma semelhante aos de um cilindro seccionado. Assim, em cada secção existe um só contorno simples e fechado. Nesse caso, a correspondência é direta, cada contorno de cada secção corresponde aos contornos das secções anterior e posterior (fig. 2.21).

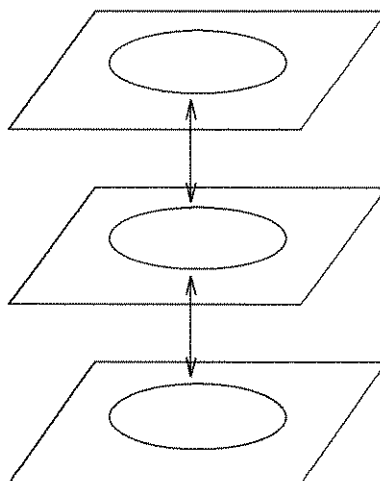


Figura 2.21: Caso mais simples de correspondência entre contornos.

É claro que existe uma infinidade de configurações mais complexas. Um dos casos mais clássicos de um problema mais complicado é o da ocorrência de **subdivisões** ou **ramificações** ("branching") em um objeto (fig. 2.22 (a)). Em um objeto desse tipo existe o problema de se corresponder um contorno a dois ou mais contornos em uma secção consecutiva (fig. 2.22 (b)). Existem muitas outras complicações que podem ocorrer, como objetos em forma de "bengala", objetos dentro de objetos, objetos que se cortam, e outras (fig. 2.23 (a), (b) e (c)).

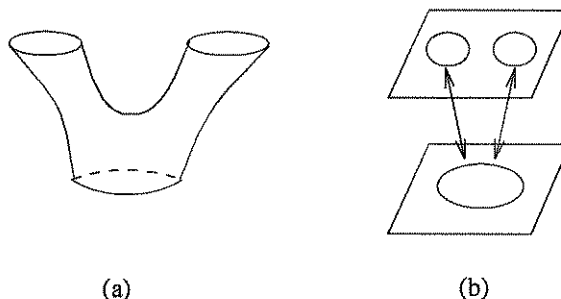


Figura 2.22: Problema de ramificações : (a) superfície com subdivisão ; (b) correspondência entre contornos extraídos.

Existem diversas abordagens para se resolver o problema de correspondência, que utilizam conceitos de teoria de classificação de superfícies de Morse [28], heurísticas e grafos.

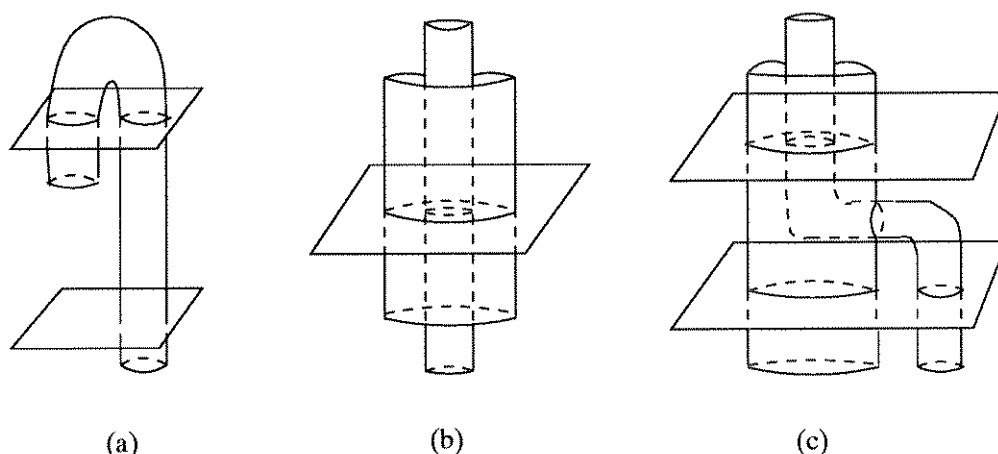


Figura 2.23: Situações complexas no problema de correspondência.

Em [6], é apresentado um método em que primeiramente é calculada a elipse que melhor aproxima cada contorno de cada secção. Em seguida, o método procura identificar cilindros, que são formados por elipses que possuam o centro sobre uma mesma reta. Uma vez que todos os cilindros foram calculados, o método procura identificar ligações entre esses cilindros, que podem significar possíveis casos de ramificações.

É apresentado em [14] um método de estabelecimento de correspondências baseado em grafos valorados. Nesse método também é calculada uma elipse aproximadora para cada contorno. De cada elipse são extraídos 4 atributos para caracterização do contorno : as coordenadas do centro da elipse, o diâmetro maior e o menor. Em seguida, é criado um grafo onde cada vértice representa um contorno (ou uma elipse), e as arestas representam ligações de cada contorno com todos os contornos das secções anterior e posterior (fig. 2.24 (a) e (b)).

A cada aresta é associado um custo, que é calculado pela distância euclidiana entre os pontos 4-dimensionais, cujas coordenadas são os atributos de cada elipse. Para se determinar a correspondência, o algoritmo calcula a árvore geradora do grafo que tiver o menor custo (fig. 2.24 (c)).

Esses dois trabalhos não resolvem a situação onde ocorrem ramificações e junções da superfície. Um caso que ilustra essa situação é o de um toro (fig. 2.25). Em [91] é apresentado um método mais geral, que faz a classificação das estruturas possíveis da

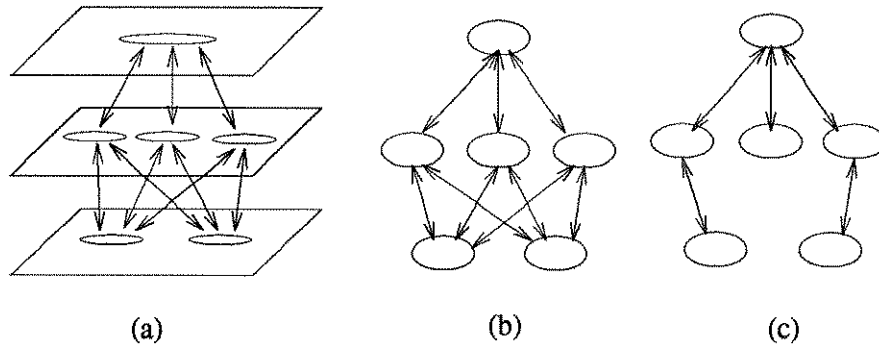


Figura 2.24: Método de correspondências : (a) contornos ; (b) grafo criado ; (c) árvore geradora do grafo.

superfície original a partir dos contornos, usando uma extensão da teoria de classificação de Morse [28]. Essa metodologia permite inclusive o tratamento de objetos mais complexos, como os da figura 2.23.

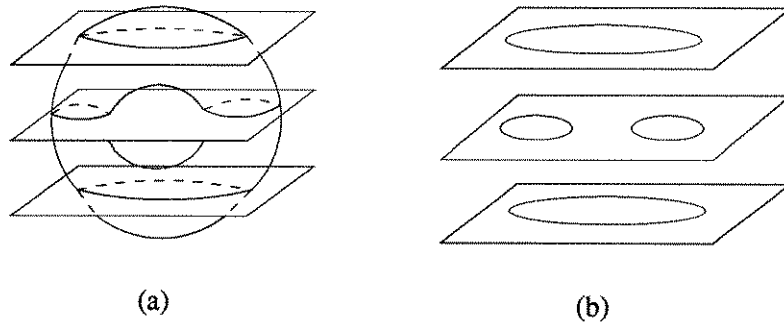


Figura 2.25: Reconstrução 3D de um toro : (a) toro seccionado ; (b) contornos extraídos.

É claro que a estrutura do objeto influencia diretamente o ajuste da superfície aos contornos. Existem estudos que tratam desses casos e, em particular, existem muitas propostas de soluções para o caso de ramificações. Existem soluções automáticas, outras que necessitam a intervenção do usuário e, geralmente, dependem diretamente da aplicação. Para o caso do ajuste de superfícies paramétricas, praticamente não existem trabalhos abordando situações mais complexas.

Capítulo 3

Segmentação e Representação de Contornos

3.1 Introdução

Existe uma expressão em inglês, *"to lick into shape"*, que significa *colocar em condições de bom funcionamento*. Com certeza, a maioria das pessoas que a empregam não imaginam o efeito que tal expressão causa em um pesquisador da área de visão computacional (VC) e reconhecimento de padrões (RP). Afinal, em uma imagem, uma das informações mais importantes a ser determinada é a **forma** ("shape") dos objetos dessa imagem [17]; e o "bom funcionamento" de um processo de reconhecimento de padrões, seja qual for a abordagem utilizada, depende diretamente de uma boa descrição/representação da forma de cada objeto. Segundo o mesmo [17] :

"One of the major challenges to computer vision is to represent shapes, or the important aspects of shapes, so that they may be learned, matched against, recollected and used(...)"

Existem 2 abordagens básicas relacionadas à representação das formas : contornos e regiões. A figura 3.1 ilustra a diferença entre essas duas abordagens. Exemplos de métodos de representação de formas por regiões são "octrees", "quadtrees", esqueletos, decomposição geométrica e morfológica [37]. Este capítulo discute os aspectos de **representação** ou **descrição de contornos**. Para ilustrar conceitos e métodos ligados aos

contornos, tome-se o contorno extraído de um quadrado, como na fig. 3.1 (a).

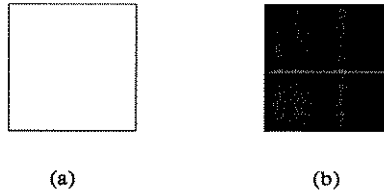


Figura 3.1: Representação do objeto "quadrado" por seu contorno (a) e sua região (b).

Uma primeira representação desse contorno é dada pelo conjunto ordenado:

$$P = \{p_0, p_1, \dots, p_n\} \quad (3.1)$$

$$p_i = (x_i, y_i), i = 0 \dots n$$

de pontos do contorno. Se o contorno for fechado, tem-se $p_0 = p_n$. Os métodos de descrição de contornos são classificados de diversas maneiras.

Primeiramente, um contorno pode ser analisado através de uma *transformação escalar* ou no próprio *domínio espacial* [81]. Um exemplo de uma transformação escalar sobre o contorno P da eq. 3.1 é :

$$Per(P) = \alpha$$

que associa ao contorno P o seu perímetro α . Muitos outros exemplos existem na literatura, como área definida pelo contorno, excentricidade, relação área/perímetro e outros. As transformações escalares são particularmente importantes na *abordagem estatística para o reconhecimento de padrões* [67].

Por outro lado, as técnicas que descrevem o contorno no domínio espacial procuram identificar *características estruturais* nesses. Suponha que cada lado do quadrado da fig. 3.1 (a) tem tamanho l . E sejam os vetores unitários (norma igual a 1) $\vec{a} = (1, 0)$, $\vec{b} = (0, 1)$, $\vec{c} = (-1, 0)$, $\vec{d} = (0, -1)$, como na fig. 3.2 (a). Então, o contorno 3.1 (a) pode ser descrito pela quádrupla $(l\vec{b}, l\vec{a}, l\vec{d}, l\vec{c})$ (fig. 3.2 (b)). Como o quadrado possui uma posição no plano, é necessária a identificação de um ponto para a completa descrição do contorno 3.1 (a). Na fig. 3.2 (c), esse ponto é dado por (x_0, y_0) . Assim, essa descrição será dada pela 5-upla $((x_0, y_0), l\vec{b}, l\vec{a}, l\vec{d}, l\vec{c})$.

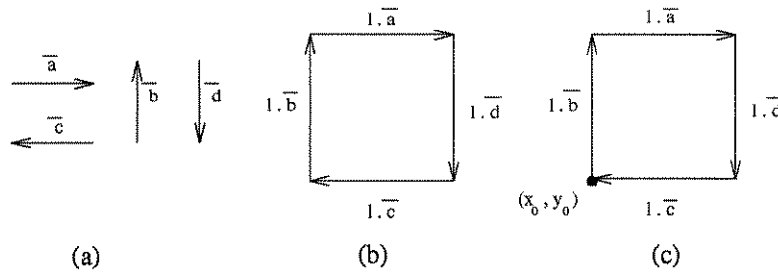


Figura 3.2: Descrição do contorno por quatro vetores e um ponto.

Esse tipo de descrição de contornos é importante na *abordagem estrutural para o reconhecimento de padrões* [48].

A representação de contornos, em quase todos seus métodos, possui características comuns, apesar de sua diversidade. O conhecimento dessas características facilita o entendimento da maioria dos métodos, além de funcionar como um referencial para a criação de parâmetros de avaliação e comparação dos métodos :

(i) **Extração de informações sobre a forma do contorno** : como foi citado no início deste capítulo, a representação do contorno deve facilitar o entendimento de características sobre ele; embora algumas vezes a representação possa ser o fim de um processamento de imagens, na maioria das aplicações não o é, servindo como uma etapa intermediária de um processo maior, como, por exemplo, no casamento de segmentos (ver cap. 4).

(ii) **Compressão de dados** : praticamente todas as formas de representação de contornos procuram efetuar uma redução no volume de dados, podendo atingir razões que vão de 2:1 a 25:1, dependendo do método e da imagem. Essa característica está relacionada a uma outra classificação dos métodos de descrição dos dados, que é a capacidade ou não do método reconstruir a imagem (contorno) original. Em [81], os métodos são classificados nos que *preservam informação* ou nos que *não preservam informação*. Porém, é fato que a maioria dos métodos permite uma *reconstrução parcial* do contorno original. Esse fato está diretamente relacionado com a próxima característica.

(iii) **Redução ou filtragem de ruídos** : imagens obtidas por um processo de digitalização, em contraposição àquelas geradas sinteticamente, costumam ser corrompidas com uma série de imperfeições. O método de representação, ao realizar a compressão de

dados, deve justamente filtrar os ruídos presentes no contorno.

(iv) **Dependência de cantos ou pontos de curvatura máxima** : embora esta característica esteja presente num número menor de métodos de representação, principalmente nos métodos que trabalham no domínio espacial, a identificação de pontos do contorno que possuem curvatura máxima, e o próprio cálculo da curvatura em geral, cumprem um papel central na maioria dos métodos de análise de contornos. Particularmente ligado à curvatura, um importante conceito é o de *canto* ("corner"), que é definido como um ponto de curvatura infinita no caso de curvas contínuas. Para o caso de métodos de aproximação por partes, que são abordados na secção 3.2, pode ser mostrado que o erro cometido pela aproximação está diretamente ligado à curvatura de cada ponto da curva a ser aproximada.

Antes de terminar esta secção, é preciso chamar atenção sobre a terminologia adotada. Os termos **representação** e **descrição** são amplamente usados na literatura, por vezes indistintamente. Como não existe uma característica marcante que possa diferenciá-los, optou-se também por usá-los com o mesmo significado. Também optou-se por traduzir "corner" por **canto**, no sentido de "canto da mesa". Nas situações em que esse termo for usado de forma ambígua, a devida atenção será chamada. Finalmente, o termo **contorno** será usado para designar curvas abertas e fechadas, obtidas a partir de um objeto da imagem. Um caso particularmente importante de curvas abertas são imagens de *ondas* ("waveforms") ou outras curvas unidimensionais da forma $y = f(x)$. Exemplos dessas imagens são o eletrocardiograma e gráfico de voz [80]. Nessas aplicações, pode-se analisar a onda através de seu espectro ou como sendo uma imagem. No reconhecimento de voz, uma combinação dessas duas abordagens por vezes é necessária.

Na próxima secção será apresentada uma revisão bibliográfica sobre representação de contornos.

3.2 Representação de Curvas no Domínio Espacial : Revisão Bibliográfica

A primeira forma de representação de curvas que possui um nível de abstração maior que a própria imagem binária é a do conjunto dos próprios pontos dessa curva, como apresentada na equação 3.1, ou seja, uma simples lista ordenada contendo seus pontos. Existem informações já podem ser extraídas dessa representação. Por exemplo, a curva possui $n + 1$ pontos, e se $p_n = p_0$, então a curva é fechada. Caso contrário, ela é aberta. Essas informações porém, não são completas para curvas com topologias mais complexas, como seria o caso de uma curva representando o número 6. Trabalhos como [36] sugerem que essas curvas sejam decompostas em componentes simples, onde cada um seria uma curva aberta ou fechada. O caso de curvas com *ramificações* ("branching") também pode ser tratado pela decomposição. Um exemplo é a curva em forma da letra "y".

Uma das formas mais usadas para a representação de curvas é o **código da cadeia**, proposto por Freeman [32]. Essa forma de representação possui características que foram amplamente estudadas na literatura. Diversas informações sobre a curva codificada são extraídas a partir dessa representação, como área, perímetro e centro de gravidade [41]. Em [33] é introduzida uma forma de encontrar pontos que possuam um grande valor, em módulo, de curvatura. Esses pontos definem uma *aproximação poligonal* do contorno representado. Já em [62], o contorno é dividido em segmentos de reta e arcos de circunferência, obtendo uma representação de nível mais elevado a partir do código da cadeia.

A abordagem estrutural para o reconhecimento de padrões possui uma característica peculiar para a descrição de contornos : o conceito de **primitivas**. As primitivas correspondem aos símbolos terminais de uma gramática formal [48], formando o seu conjunto um alfabeto segundo o qual os contornos são descritos. Esses são encarados como "palavras" ou "frases" compostas pelas primitivas. As técnicas de descrição de contornos que fazem uso desses conceitos são conhecidas como técnicas sintáticas. Em [76], os terminais são definidos a partir de uma aproximação linear de um contorno em forma de onda. Esses terminais descrevem, em função da inclinação de cada reta da aproximação poligonal, a forma como a curva se comporta. Essa descrição é usada para a detecção de picos na onda.

Em [64], que também faz uso de técnicas sintáticas, são definidos 4 tipos de primitivas : arcos que possam ser aproximados por curvas quadráticas; protrusões ou intrusões pontiagudas; segmentos lineares; e pequenos segmentos amorfos. Exemplos de aplicações dessa representação são apresentados em números manuscritos, glóbulos brancos no sangue e outros.

Em [60] são usados 3 tipos de terminais, correspondendo aos 3 primeiros apresentados em [64]. Nesse trabalho [60], a descrição é aplicada no casamento entre dois contornos.

A **aproximação poligonal de contornos** é uma das formas de representação de contornos mais estudadas [71]. A literatura relacionada a esse tópico é extremamente vasta e diversa. Assim, ao invés de uma completa revisão sobre esse tipo de representação, são referenciados métodos que possuem conceitos utilizados posteriormente.

Algebricamente, uma aproximação poligonal que aproxima o contorno P da eq. 3.1 é dada por

$$P' = \{p'_0, p'_1, \dots, p'_{n'}\}, \quad n' \leq n \quad (3.2)$$

O conjunto P' pode ou não ser um subconjunto de P . Esta é a primeira diferença entre os métodos de aproximação poligonal. Geometricamente, dois exemplos de aproximação poligonal são apresentados na figura 3.3.

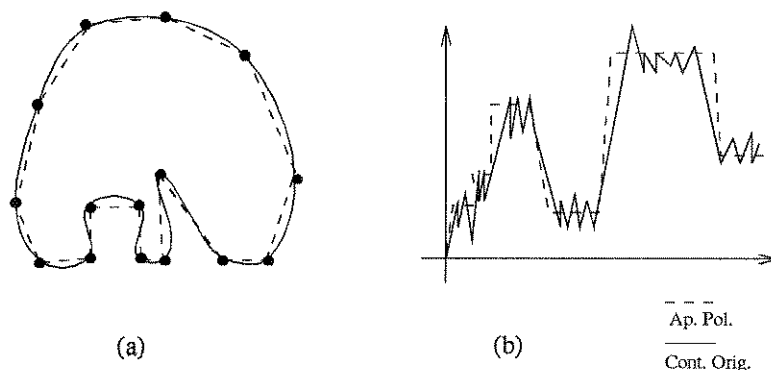


Figura 3.3: Aproximação poligonal : (a) de um contorno fechado; (b) de uma onda.

Outra diferença importante entre os métodos de aproximação poligonal está relacionada ao critério de aproximação. Dependendo da forma como esse critério é definido, pode-se dividir os métodos em duas classes : **globais** e **locais** [23]. De maneira geral, os

métodos globais procuram identificar uma aproximação poligonal possível para o contorno em questão utilizando erro global, enquanto que os métodos locais procuram, localmente, por pontos de alta curvatura em módulo, utilizando critério de erro local. Esses pontos são tomados como vértices da aproximação poligonal. Exemplos de métodos globais estão em [85, 83]; por outro lado [75, 74] apresentam métodos locais.

Uma característica presente na maioria dos métodos de aproximação poligonal é a necessidade da definição de limiares para a segmentação da curva a ser aproximada. Isso faz com que o método seja dependente de escala na maioria dos casos. Existem duas maneiras para contornar essa dependência:

(a) **normalização do critério de aproximação** : onde, por exemplo, o erro máximo de aproximação é normalizado em relação ao tamanho do contorno [29].

(b) **descrição hierárquica do contorno** : onde várias aproximações com diferentes valores para o erro máximo são obtidas e armazenadas. A estrutura que descreve essas aproximações deve permitir sua análise em vários níveis. Exemplos dessas estruturas são as "strip-trees" [17] ou os *super-segmentos* [24]. Um dos primeiros trabalhos a chamar a atenção para essa descrição hierárquica é [75].

Como foi citado anteriormente, um tipo particular de contorno que possui aplicações interessantes é aquele em forma de ondas. A figura 3.3(b) ilustra como a aproximação poligonal pode ser usada na descrição desses contornos. Em [80] é apresentada uma revisão de métodos que fazem a aproximação poligonal por partes de curvas desse tipo, e é também introduzido um método de segmentação e aproximação linear uniforme por partes, baseado em otimização discreta. São apresentadas aplicações em eletrocardiogramas e outros tipos de ondas.

Finalmente, existem métodos de representação de contornos por **curvas paramétricas**. Uma curva paramétrica pode ser definida de maneira geral como :

$$P(t) = \sum_{i=1}^n V_i B_i(t)$$

onde t é o parâmetro, $B_i(t)$ é a i -ésima função base, e V_i o i -ésimo ponto de controle da curva.

Em [1], as funções base escolhidas são gaussianas. Os parâmetros dessas funções

são usados como atributos para a decomposição e reconhecimento dos contornos. Segundo esse trabalho, a decomposição é única, o que permite que seus parâmetros sejam utilizados no processo de reconhecimento de formas.

A curva paramétrica usada em [59, 27] é a B-Spline. Maiores detalhes sobre esse tipo de representação são apresentados nas secções que seguem.

3.3 Segmentação e Representação de Contornos por B-Splines

Embora as curvas B-Spline tenham sido largamente utilizadas em muitas aplicações de computação gráfica, o mesmo não aconteceu em VC. Sua potencial utilização foi aventada em diferentes trabalhos e livros na década de 70 (por exemplo [73]), porém os problemas mais diretamente ligados à VC só começaram a ser encarados de maneira sistemática em trabalhos mais recentes [59, 27].

Uma **curva B-Spline** é definida pela equação :

$$P(u) = \sum_{i=0}^n p_i N_{i,k}(u) \quad (3.3)$$

onde $N_{i,k}(u)$ é a função base da B-Spline e $p_i, i = 0..n$, o conjunto de **pontos de controle**, formando o **polígono de controle** da curva [16]. O grau dos polinômios das funções base é determinado pelo parâmetro k . A função base pode ser definida como [7] :

definição 3.1 (B-Spline) *Uma função base de uma curva B-Spline, ou simplesmente uma função B-Spline, é uma função polinomial por partes, definida por :*

$$\begin{aligned} N_{i,1}(u) &= 1 && , \text{ se } t_i \leq u < t_{i+1} \\ &= 0 && , \text{ caso contrário} \end{aligned}$$

e

$$N_{i,k}(u) = \frac{u - t_i}{t_{i+k-1} - t_i} N_{i,k-1}(u) + \frac{t_{i+k} - u}{t_{i+k} - t_{i+1}} N_{i+1,k-1}(u)$$

onde k controla o grau do polinômio em u . Os valores t_i são conhecidos como **nós**, e formam uma partição do domínio da B-Spline, onde $t_i \leq t_{i+1}, \forall i$.

Agora, dado um conjunto de pontos que representam um contorno, como aquele da equação 3.1, como melhor representá-lo por uma curva ou um conjunto de curvas B-Spline? Uma primeira solução que poderia ser arriscada é a de se usar o próprio conjunto 3.1 como sendo o conjunto de pontos de controle da equação 3.3. Existem algumas considerações que devem ser feitas sobre essa solução. Em primeiro lugar, uma curva B-Spline "geralmente" não passa pelos pontos de controle. Esse termo "geralmente" está diretamente relacionado com a escolha dos nós t_i e com o conceito de pontos de controle múltiplos. Em relação aos nós t_i , existem duas formas de calculá-los que são usadas com maior frequência : para B-Spline uniforme periódica e não-periódica [88]. As B-Splines não periódicas são usadas em curvas abertas e, nesse caso, a curva passa apenas pelos pontos de controle extremos, ou seja, p_0 e p_n . Os nós são definidos, nesse caso, pela regra :

$$\begin{aligned} t_i &= 0 & , i < k \\ &= i - k + 1 & , k \leq i \leq n \\ &= n - k + 2 & , i > n \end{aligned}$$

As B-Splines periódicas são usadas para modelar curvas fechadas. Uma forma de gerar os nós é tomando-se $t_i = i$. Nesse caso, a curva B-Spline não passa por nenhum dos pontos de controle.

Pontos de controle múltiplos ocorrem quando se define, para um dado $a > 0$, $0 \leq i \leq i + 1 \leq \dots \leq i + a \leq n$, tais que :

$$p_i = p_{i+1} = \dots = p_{i+a}$$

Assim, a menos que sejam usadas outras regras para a definição dos nós, ou que todos os pontos do contorno sejam pré-definidos como múltiplos, a curva B-Spline irá apenas aproximar o contorno 3.1.

Uma desvantagem dessa solução é que ela não satisfaz as restrições (ii) e (iii) apresentadas na secção 3.1. Uma vez que todos os pontos do contorno original seriam usados, não haveria nenhuma compressão de dados e, muito menos, uma filtragem de ruídos.

Finalmente, como foi explicado no capítulo 2, o método de representação deve segmentar o contorno, para que os segmentos sejam usados na fase de casamento entre os contornos.

A compressão e a filtragem dos dados pode ser feita através da identificação de pontos importantes do contorno. Embora o conceito de importante seja relativamente subjetivo, diversos trabalhos nas áreas de visão computacional e reconhecimento de padrões apontam os pontos de grande curvatura como importantes em aplicações diferentes. Em certo sentido, é desejável que a representação do contorno preserve a informação desses pontos. Seja

$$\begin{aligned} P' &= \{p'_0, p'_1, \dots, p'_{n'}\} \\ P' &\subset P, n' \leq n \end{aligned}$$

curvatura local máxima, do conjunto P da equação 3.1. Trabalhos como [73, 17] utilizam um método para o cálculo dos pontos de controle de uma B-Spline cúbica ($k = 4$) que interpole esses pontos. Se o número de pontos a serem interpolados for $m = (n' + 1)$, então o método deverá resolver um sistema linear de $m + 1$ equações e incógnitas. Os coeficientes desse sistema linear forma uma matriz $(m + 1) \times (m + 1)$ da forma

$$\frac{1}{6} \begin{bmatrix} 4 & 1 & \cdots & 1 \\ 1 & 4 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \cdots & \cdots & 4 \end{bmatrix}$$

e a obtenção dessa matriz pode ser encontrada em qualquer um desses trabalhos.

Novamente, deve-se observar que mesmo esses métodos não definem uma segmentação do contorno original, visto que o contorno é aproximado por uma única B-Spline.

Esse problema é resolvido em [59, 27]. A idéia básica dos métodos apresentados nesses trabalhos é a utilização de dois tipos de pontos importantes do contorno. Em outras palavras, trata-se da definição de duas partições do contorno original. A primeira segmenta o contorno nos pontos de curvatura local máxima. Em seguida, em cada segmento, definidos pelos pontos de segmentação extremos, um conjunto de pontos importantes para aquele segmento são identificados. O conjunto formado pelos pontos extremos de cada segmento, mais os pontos intermediários identificados, é usado para o cálculo do polígono de controle de um segmento de B-Spline que representa esse segmento. Esse conjunto pode ser usado diretamente como polígono de controle, ou em um método para o cálculo da B-Spline que interpole esses pontos. O método apresentado em [27] está resumido no algoritmo 3.1

algoritmo 3.1 (Segmentação e Representação) :

- 1– *Algoritmo de representação por B-Splines;*
- 2 – *Para cada ponto do contorno, calcular a curvatura C_v ;*
- 3 – *Se $C_v > T_s$, então o ponto é classificado como de segmentação;*
- 4 – *Se $T_c < C_v < T_s$, então o ponto é classificado como de controle;*
- 5 – *Todos os demais pontos são descartados;*
- 6– *Fim.*

algoritmo 3.2 (Segmentação e Representação) :

- 1– *Algoritmo modificado de representação por B-Splines;*
- 2 – *Para cada ponto do contorno, calcular a curvatura C_v ;*
- 3 – *Se $C_v > T_s$, então o ponto é classificado como de segmentação;*
- 4 – *Se $T_c < C_v < T_s$, então o ponto é classificado como de controle;*
- 5 – *Todos os demais pontos são descartados;*
- 6 – *Ajustar os pontos de segmentação e de controle;*
- 7– *Fim.*

3.3.1 O Algoritmo Modificado

O método de segmentação e representação por B-Splines implementado é uma modificação do algoritmo 3.1. A modificação consiste na criação de um pós-processamento para eliminação de pontos indesejáveis. O algoritmo 3.2 mostra o método desenvolvido.

No algoritmo 3.2, assim como no restante deste capítulo, a curvatura calculada em um determinado ponto será denotada por C_v . Os valores T_s e T_c são limiares para a identificação de pontos de segmentação e de controle, respectivamente.

Como será discutido posteriormente, o método original implementado por Medioni em [27] acabou por fornecendo resultados indesejáveis para as aplicações de reconstrução 3D, principalmente em relação a um número excessivamente grande de segmentos na representação final. O passo 6, que é realizado fazendo-se uma análise global da estrutura final da representação, teve de ser criado e acrescentado ao algoritmo. O cálculo da cur-

vatura, também com auxílio de B-Splines, assim como o ajuste final da representação, são apresentados e discutidos nas próximas secções.

3.3.2 Estimativa da Curvatura

A estimativa da curvatura em um dado ponto de um contorno é um problema bastante estudado, possuindo diversas soluções, embora nenhuma seja ótima. O maior problema é que o contorno é uma curva discretizada, impedindo o cálculo analítico direto.

Segundo [75], existem duas formas principais de realizar esse cálculo. Sejam os pontos da figura 3.4, representando um trecho de um contorno. Deve-se avaliar a curvatura no ponto p_i . Seja ainda uma vizinhança $[p_{i-k}, \dots, p_{i+k}]$ do ponto p_i , $k > 0$. Uma forma é utilizar, de alguma maneira, os vetores $p_i \vec{p}_{i+j}$, $-k < j \leq k$, para o cálculo aproximado do ângulo formado em p_i . Por exemplo, pode-se ajustar pelo método dos mínimos quadrados uma reta aos pontos $\{p_{i-k}, \dots, p_i\}$ e $\{p_i, \dots, p_{i+k}\}$, e calcular o ângulo formado por essas retas (fig. 3.5). Dessa maneira, quanto mais próximo de 0° for esse ângulo, maior é a curvatura. Por outro lado, quanto mais próximo de 180° , menor a curvatura e, portanto, ele será descartado.

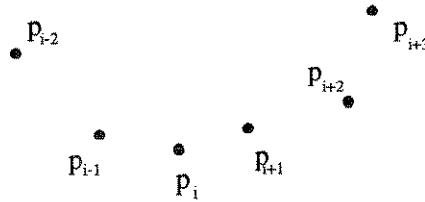


Figura 3.4: Vizinhança do ponto p_i no contorno.

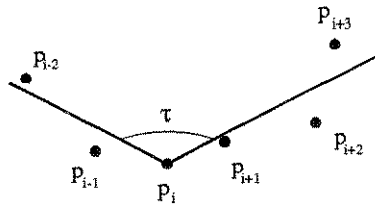


Figura 3.5: Avaliação da curvatura através da vizinhança.

A segunda é interpolar ou aproximar esses pontos por uma função S (fig. 3.6).

A curvatura pode ser calculada analiticamente a partir da expressão de S .

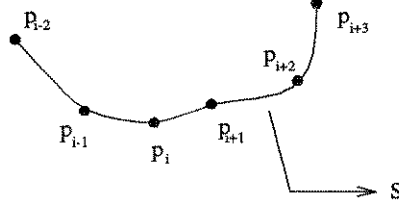


Figura 3.6: Ajuste da função S para o cálculo analítico da curvatura.

Essa é a forma implementada neste trabalho. Dada a vizinhança $[p_{i-2}, \dots, p_{i+2}]$, uma B-Spline é ajustada a esses pontos, e a curvatura calculada.

3.3.3 Estimativa da Curvatura em Curvas Paramétricas.

Seja um ponto $p = (x, y)$ no plano, pertencente a uma curva paramétrica S ,

$$\begin{aligned} x &= f(t) \\ y &= g(t) \end{aligned}$$

onde t é o parâmetro de S .

A derivada de S em um ponto $A(t = t_1)$ é dada por :

$$\left[\frac{dx}{dy} \right]_{t=t_1} = \left[\frac{\frac{dg}{dt}}{\frac{df}{dt}} \right]_{t=t_1}$$

A curvatura $C_v(t_1)$ é dada por :

$$\begin{aligned} C_v(t_1) &= \frac{\frac{d^2 y}{dx^2}}{\left(1 + \left(\frac{dy}{dx} \right)^2 \right)^{\frac{3}{2}}} = \\ &= \frac{\left(\left(\frac{df}{dt} \right) \left(\frac{d^2 g}{dt^2} \right) - \left(\frac{dg}{dt} \right) \left(\frac{d^2 f}{dt^2} \right) \right)}{\left(\left(\frac{df}{dt} \right)^2 + \left(\frac{dg}{dt} \right)^2 \right)^{\frac{3}{2}}} \end{aligned} \quad (3.4)$$

Suponha que a curva a ser ajustada é uma polinomial cúbica em t , com $t \in [0, 1]$ entre dois pontos A e B de S . As expressões de $f(t)$ e $g(t)$ podem ser representadas como :

$$\begin{aligned} x &= f(t) = a_1 t^3 + b_1 t^2 + c_1 t + d_1 \\ y &= g(t) = a_2 t^3 + b_2 t^2 + c_2 t + d_2 \end{aligned} \quad (3.5)$$

Substituindo a equação 3.5 em 3.4, e aplicando a equação resultante ao ponto $A(t = 0)$, tem-se que a curvatura nesse ponto é dada por :

$$C_v(0) = 2 \frac{c_1 b_2 - c_2 b_1}{(c_1^2 + c_2^2)^{\frac{3}{2}}} \quad (3.6)$$

3.3.4 Estimativa da Curvatura usando B-Splines.

Uma B-Spline cúbica paramétrica com nós igualmente espaçados é dada por

$$\begin{aligned} x(t) &= TMG_x \\ y(t) &= TMG_y \end{aligned}$$

onde $T = [t^3 \ t^2 \ t \ 1]$,

$$M = \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix}$$

e

$$\begin{aligned} G_x &= [x_{i-1} \ x_i \ x_{i+1} \ x_{i+2}]^t \\ G_y &= [y_{i-1} \ y_i \ y_{i+1} \ y_{i+2}]^t \end{aligned}$$

Fazendo os cálculos de $x(t)$ e $y(t)$, pode-se obter os coeficientes

de 3.5 :

$$\begin{aligned} b_1 &= \frac{1}{2}(x_{i-1} - 2x_i + x_{i+1}) \\ b_2 &= \frac{1}{2}(y_{i-1} - 2y_i + y_{i+1}) \\ c_1 &= \frac{1}{2}(x_{i+1} - x_{i-1}) \\ c_2 &= \frac{1}{2}(y_{i+1} - y_{i-1}) \end{aligned} \quad (3.7)$$

Segundo [27], uma suavização da curva pode ser conseguida calculando-se o deslocamento entre o ponto (x_i, y_i) e a B-Spline ajustada. O deslocamento, em uma B-Spline cúbica, é definido como :

$$\begin{aligned} \delta_x &= d_1 - x_i \\ \delta_y &= d_2 - y_i \end{aligned}$$

A curvatura é calculada após uma reavaliação das expressões 3.7 aos pontos deslocados, ou seja, aos pontos $p_i + \delta$, $\delta = (\delta_x, \delta_y)$. Feito esses cálculos, pode-se obter :

$$b'_1 = \frac{1}{12}((x_{i-2} + x_{i+2}) + 2(x_{i-1} + x_{i+1}) - 6x_i)$$

$$b'_2 = \frac{1}{12}((y_{i-2} + y_{i+2}) + 2(y_{i-1} + y_{i+1}) - 6y_i)$$

$$c'_1 = \frac{1}{12}((x_{i+2} - x_{i-2}) + 4(x_{i+1} - x_{i-1}))$$

$$c'_2 = \frac{1}{12}((y_{i+2} - y_{i-2}) + 4(y_{i+1} - y_{i-1}))$$

e novamente aplicar a equação 3.6 para o cálculo de C_v .

3.3.5 Ajuste dos Pontos de Segmentação e de Controle

Segundo o método proposto em [27] e resumido no algoritmo 3.1, os critérios para a classificação dos pontos são :

(1) Se $C_v > T_s$ e C_v for um máximo local, então o ponto é classificado como de segmentação.

(2) Se $T_c < C_v < T_s$, então o ponto é tomado como de controle.

O método original acabou fornecendo resultados que apresentavam características pouco desejáveis para sua utilização na reconstrução 3D.

A primeira delas é que, em muitos casos, verificou-se que não existe apenas 1 ponto com curvatura máxima local; por vezes, em uma dada vizinhança pré-definida, dois ou mais pontos possuem a mesma curvatura (fig. 3.7).



Figura 3.7: Pontos com curvatura elevada na mesma vizinhança.

A solução adotada para esses casos foi tomar apenas um desses pontos como de segmentação, transformando os restantes em pontos de controle do segmento a que pertencem.

algoritmo 3.3 (Ajuste por Vizinhança) :

- 1— *Algoritmo de ajuste por vizinhança da representação;*
- 2 — *Para cada ponto de segmentação P_i :*
 Se, dada uma vizinhança de P_i , existir outro ponto de
 segmentação e a curvatura desse ponto for menor que a de
 P_i , então transforme esse ponto em ponto de controle;
- 3 — *Para cada ponto de controle P_i :*
 Se, dada uma vizinhança de P_i , existir outro ponto de
 controle e a curvatura desse ponto for menor que a de
 P_i , então descarte esse ponto da representação;
- 4— *Fim.*

cessem. Da mesma maneira, o ajuste é feito aos pontos de controle. Dessa vez, em uma dada vizinhança, apenas 1 ponto é tomado como de controle, e os demais, descartados da representação. Pode-se usar a mesma vizinhança para os ajustes de pontos de segmentação e de controle. O algoritmo 3.3 resume os passos do ajuste por vizinhança.

Existem duas maneiras diferentes de implementar o ajuste acima: ajuste seqüencial ou em paralelo. No ajuste seqüencial, se um determinado ponto for testado e eliminado (e, portanto, existe um outro ponto do mesmo tipo em sua vizinhança), então ele não aparece na vizinhança de outro ponto. No ajuste em paralelo, todos os pontos são testados e aqueles que devem ser eliminados são marcados para eliminação posterior. Por exemplo, supondo que o ponto p_i de controle tem curvatura menor que o ponto p_{i+k} que está em sua vizinhança, então p_i deve ser eliminado da representação. Na implementação seqüencial, quando p_{i+k} for testado, o ponto p_i já não aparecerá na representação, uma vez que já foi eliminado. Na implementação em paralelo, p_i é marcado para eliminação, mas só desaparece da representação *após* o processamento de todos os pontos da representação. A principal diferença entre essas duas implementações ocorre quando pontos na mesma vizinhança possuem mesma curvatura. Enquanto que a implementação seqüencial pode eliminar um ponto melhor para a representação, a implementação em paralelo pode eliminar todos os pontos de uma determinada vizinhança, não escolhendo nenhum. A implementação desenvolvida nesta dissertação segue a linha seqüencial.

A segunda dificuldade diz respeito, principalmente, aos pontos de segmentação. Alguns contornos testados pelo método apresentaram, por vezes, uma segmentação onde os segmentos não representavam a informação estrutural desejada. A figura 3.8 (a) ilustra melhor esse problema. Ela mostra um dos lados da letra "A" que foi segmentado, sendo composto por dois segmentos. Embora a curvatura "global" no ponto de segmentação intermediário não seja grande, a curvatura calculada pelo método acabou classificando-o como de segmentação. Isso ocorre toda vez que o contorno apresenta uma configuração como a da figura 3.8 (b), onde o método fornece uma curvatura altíssima.

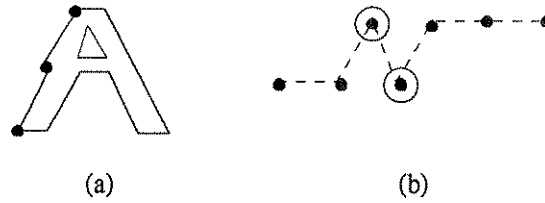


Figura 3.8: Exemplo de ponto de segmentação que deve ser eliminado pelo cálculo do ângulo.

A solução adotada foi a de calcular o ângulo formado entre cada ponto de segmentação e seus vizinhos (de segmentação) adjacentes. Supondo que (x_i, y_i) sejam as coordenadas do i -ésimo ponto de segmentação, pode-se calcular os vetores $p_i\vec{p}_{i-1}$ e $p_i\vec{p}_{i+1}$ como sendo :

$$p_i\vec{p}_{i-1} = (x_{i-1} - x_i, y_{i-1} - y_i)$$

$$p_i\vec{p}_{i+1} = (x_{i+1} - x_i, y_{i+1} - y_i)$$

e o cosseno do ângulo entre esses vetores é dado por :

$$\alpha = \frac{p_i\vec{p}_{i-1} \bullet p_i\vec{p}_{i+1}}{\|p_i\vec{p}_{i-1}\| \|p_i\vec{p}_{i+1}\|}$$

onde o símbolo $\bullet$ denota o produto escalar.

Como $-1 \leq \alpha \leq 1$, quanto mais próximo estiver α de -1, tanto mais próximo estará o ângulo de 180° e, portanto, esses pontos são descartados. Esse teste só é realizado nos pontos de segmentação.

A figura 3.9 mostra dois exemplos que ilustram os ajustes implementados.

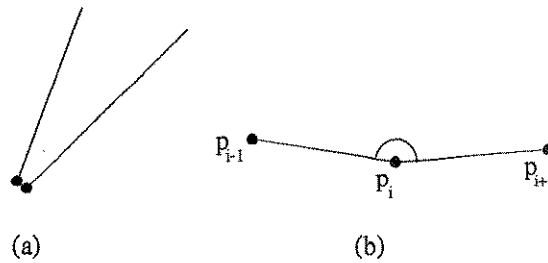


Figura 3.9: Exemplos das situações em que os ajustes desenvolvidos se aplicam.

3.4 Descrição de Contornos através do Polígono de Controle

Uma vez que o contorno foi segmentado e representado por segmentos de B-Spline (fig. 3.10), tem-se associado a cada segmento um polígono de controle composto pelos pontos de segmentação e os pontos de controle intermediários (fig. 3.11).

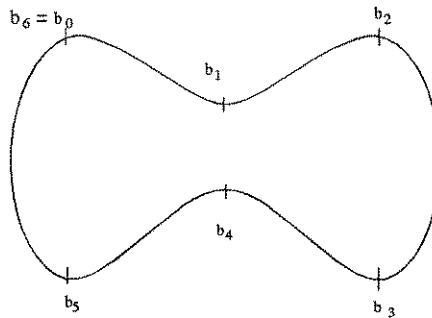


Figura 3.10: Contorno segmentado.

Um **segmento representado por uma B-Spline** é um segmento de curva que pode ser obtido a partir do polígono de controle P , onde

$$P = \{p_0, p_1, \dots, p_n\}$$

e $p_0 = b_i$ e $p_n = b_{i+1}$ (fig. 3.10 e 3.11). O problema de gerar o segmento a partir do polígono de controle já foi muito estudado, e algoritmos para resolvê-lo são encontrados em [7].

É interessante que se possa caracterizar o segmento de curva a partir do seu polígono de controle. Dessa forma, o conjunto de dados a ser processado para a descrição desse segmento será muito menor. Essa caracterização é necessária para a definição de um

descritor de segmentos, que será usado por uma medida de similaridade [66] entre os segmentos, para o posterior casamento desses.

Um conjunto de vetores $V = \{\vec{v}_0, \vec{v}_1, \dots, \vec{v}_{n-1}\}$ pode ser calculado a partir do conjunto P , onde

$$\vec{v}_i = p_i \vec{p}_{i+1} = p_{i+1} - p_i$$

A figura 3.11 ilustra a construção desses vetores.

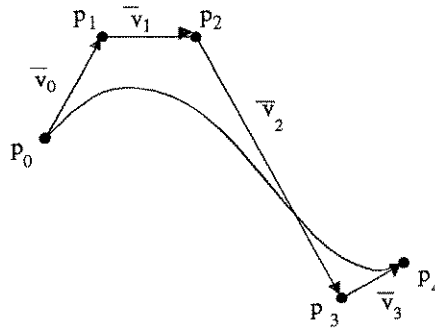


Figura 3.11: Segmento de B-Spline e vetores do polígono de controle.

Pode-se também definir o conjunto

$$C' = \{c'_0, c'_1, \dots, c'_n\}$$

onde cada c'_i é o valor da curvatura, calculado pelo algoritmo de segmentação e representação da secção 3.2, em cada ponto p_i . Pelo próprio funcionamento do algoritmo de segmentação, tem-se que

$$\|\vec{v}_i\| > 0, \forall i$$

Seja o conjunto $T' = \{t'_0, t'_1, \dots, t'_n\}$, onde

$$t'_0 = 0,$$

$$t'_i = \sum_{k=0}^{i-1} \|\vec{v}_k\|, \quad i = 1..n$$

Os conjuntos V , C' e T' são usados pelos descritores do segmento.

Existem diversas informações que podem ser extraídas da representação para a caracterização do segmento. São apresentadas algumas que foram encontradas na literatura. Os índices usados ilustram a aplicação de cada descritor no segmento da figura 3.11, para facilitar o seu entendimento. A generalização de cada um é direta.

- O vetor $p_0\vec{p}_4$: pode ser facilmente calculado pela expressão :

$$p_0\vec{p}_4 = p_4 - p_0 = (x_4 - x_0, y_4 - y_0)$$

É usado em [46], e dá uma idéia de orientação do segmento. Uma maneira de visualizar a diferenciação que esta característica apresenta entre os segmentos é tomar os vetores na origem. Suponha dois segmentos que possuam os vetores diretores $\vec{v}_1$ e $\vec{v}_2$, como na figura 3.12.

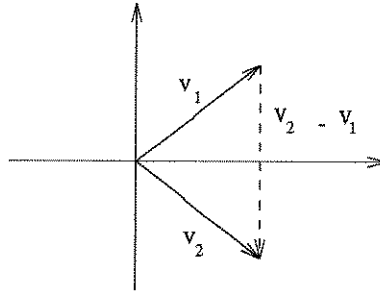


Figura 3.12: Vetores v_1 e v_2 na origem.

Existem duas maneiras de avaliar a distância entre os descritores $\vec{v}_1$ e $\vec{v}_2$, a saber :

- (i) norma de $\vec{v}_2 - \vec{v}_1$
- (ii) ângulo entre $\vec{v}_1$ e $\vec{v}_2$

É importante notar que a norma da diferença depende da norma dos vetores, enquanto que o ângulo não. Uma forma de anular essa dependência é através da normalização de $\vec{v}_1$ e $\vec{v}_2$. Por outro lado, essa dependência pode ser interessante para a diferenciação de segmentos de mesma orientação e tamanhos diferentes. □

- Ponto médio do segmento $\overline{p_0p_4}$: também pode ser facilmente calculado

pela equação :

$$(x_m, y_m) = \left(\frac{x_0 + x_4}{2}, \frac{y_0 + y_4}{2} \right)$$

Enquanto que o vetor $p_0\vec{p}_4$ dá uma idéia de orientação, o ponto médio relaciona a posição dos segmentos (fig. 3.13), diferenciando segmentos de mesma orientação mas posições diferentes.

Um parâmetro que deve ser usado para a avaliação desses descritores é o seu comportamento em relação à operação de junção ("merge") de 2 segmentos consecutivos. Por exemplo, no caso dos vetores $p_0\vec{p}_4$, basta que esses vetores sejam somados, no caso de uma junção de seus segmentos. No caso do ponto médio, este deve ser recalculado .□

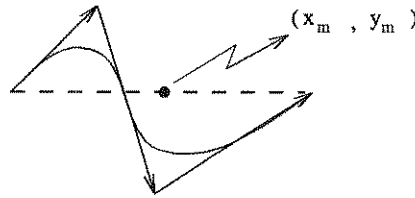


Figura 3.13: Ponto médio do segmento.

• **Somatória das normas dos vetores $\vec{v}_i$** : uma das grandezas apresentadas em [46, 24] para caracterizar um segmento é o tamanho deste. Ao invés de calcular o tamanho da curva, pode-se facilmente calcular o tamanho do polígono de controle, através da fórmula :

$$L = \sum_i \|\vec{v}_i\|$$

sendo que

$$\|\vec{v}_i\| = \sqrt{x_i^2 + y_i^2}$$

No caso da junção de dois segmentos j e $j + 1$, basta que se some L_j e L_{j+1} . Como o tamanho da curva, e por consequência de um segmento, é uma grandeza diretamente ligada à escala da representação, seu uso como atributo para caracterização do segmento pode comprometer certas aplicações. Trabalhos como [29], sugerem que essa grandeza seja normalizada pelo tamanho total da curva. Essa foi a solução adotada em nossa implementação .□

• **Somatória dos ângulos entre os vetores $\vec{v}_i$** : Também usada em [46, 24]. Dados dois vetores $\vec{u}$ e $\vec{v}$, o ângulo entre eles é dado por :

$$\text{ângulo}(\vec{u}, \vec{v}) = \arccos\left(\frac{\vec{u} \bullet \vec{v}}{\|\vec{u}\| \|\vec{v}\|}\right)$$

onde $\vec{u} \bullet \vec{v}$ denota o produto escalar entre os vetores.

Assim, se a_i for o ângulo entre $\vec{v}_i$ e $\vec{v}_{i+1}$, tem-se :

$$A = \sum_i a_i$$

A sendo a somatória .□

• **Medida de simetria em relação à mudança angular :** Apresentada em [46], é definida como :

$$Sim = \int_0^L \left(\int_0^S f(l) dl - \frac{A}{2} \right) ds$$

onde $\int_0^S f(l) dl$ mede a mudança angular até S ; A é a mudança angular total, e L é o tamanho do segmento. Também é apresentada em [46] uma forma de se calcular Sim com dados discretos :

$$Sim = G - \frac{1}{2}AL$$

e

$$G = \sum_i L_i \left(\sum_{l=1}^{i-1} a_l \right)$$

ou seja, G é calculado com base no comprimento de cada vetor v_i vezes a mudança angular até aquele ponto .□

• **Curvatura no ponto de controle :** ao invés de usar o ângulo entre cada par de vetores, uma outra medida que reflete a mudança de direção é a própria curvatura num determinado ponto, calculada pelo algoritmo de segmentação e representação. Assim, no lugar de calcular a mudança angular total, pode-se facilmente calcular a mudança total de curvatura :

$$C_t = \sum_i c_i$$

onde c_i denota a curvatura no vértice origem de v_i . Também a medida de simetria, apresentada no item anterior, pode ser calculada em função das curvaturas c_i , ao invés dos ângulos a_i . Neste caso, tem-se :

$$S = G - \frac{1}{2}C_tL$$

e

$$G = \sum_i L_i \left(\sum_{l=1}^{i-1} c_l \right). \square$$

• **Ângulo entre dois segmentos** : Apresentada em [46], é uma medida estrutural que relaciona os diferentes segmentos adjacentes entre si. $\square$

• **Curvatura nos pontos de segmentação** : Também é uma medida estrutural que relaciona 2 segmentos adjacentes. Como no caso das curvaturas nos pontos de controle, é calculada pelo algoritmo de segmentação. $\square$

• **Cardinalidade** : apresentada em [24], é dada pelo número de pontos de controle de cada segmento. $\square$

• **Excentricidade** : também introduzida em [24], pode ser calculada com base no retângulo básico [68], como sendo a razão entre o maior e o menor lado. Segundo [24], a excentricidade permite a diferenciação de segmentos com mesmo ângulos, mas de formatos diferentes. $\square$

• **Histograma e momentos** : Baseado nos conjuntos C' e T' , pode-se montar o conjunto de pares ordenados definido pelos pares $(t'_i, c'_i), i = 0..n$. Esse conjunto pode ser utilizado para uma caracterização estrutural do segmento (fig. 3.14). Em nossa implementação, esse conjunto é normalizado com transformações de translação e escala, para que obedeça às restrições de uma distribuição discreta de probabilidades, sendo assim tomado como um histograma normalizado. O novo conjunto será denotado por (t_i, c_i) (fig. 3.15). $\square$

Após o estudo de todas essas medidas, foi proposto o seguinte descritor para as aplicações de reconstrução 3D.

definição 3.2 (Descritor Bs) *Dado um polígono de controle $P = \{p_0, p_1, \dots, p_n\}$, define-se o descritor $Bs(P)$ como:*

$$Bs(P) = (\vec{d}, p_m, l, \vec{h})$$

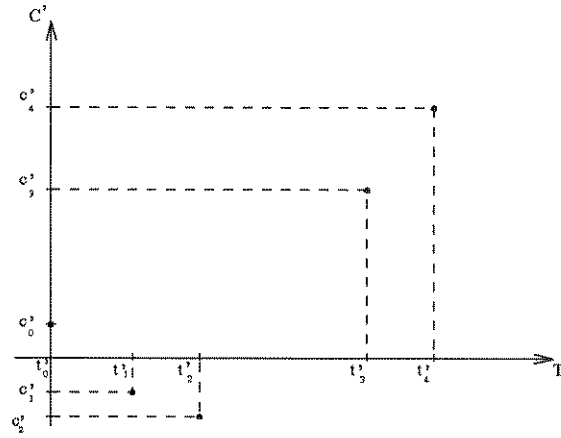


Figura 3.14: Gráfico $T' \times C'$.

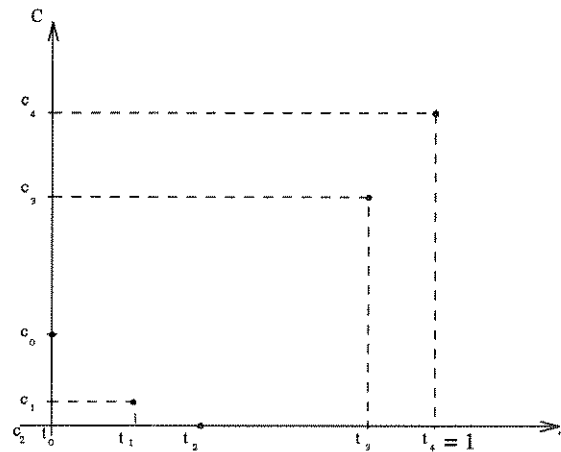


Figura 3.15: Gráfico $T \times C$ normalizado.

onde

$\vec{d}$: vetor normalizado $p_0\vec{p}_n$

p_m : ponto médio do segmento $\overline{p_0p_n}$

l : tamanho normalizado do segmento

$\vec{h}$: vetor obtido do histograma normalizado (t_i, c_i)

Cabem aqui alguns comentários sobre o descritor Bs . O atributo $\vec{d}$, que neste trabalho será denominado **vetor diretor do segmento**, possui 2 componentes (x_d, y_d) , e procura expressar uma informação sobre a direção do segmento. Deve ser normalizado com

tamanho 1. O ponto médio pm , também com duas componentes (x_{pm}, y_{pm}) , é particularmente importante em aplicações que dependam da posição da curva, i.e. aplicações que não são independentes de translação. O tamanho absoluto de um segmento pode ser calculado como sendo a somatória das normas dos vetores que compõem o polígono de controle :

$$l_a = \sum_{i=0}^{n-1} \|\vec{v}_i\|$$

Se L for o tamanho total do contorno, dado pela soma dos tamanhos absolutos de todos os segmentos, então o tamanho normalizado usado no descritor é dado por :

$$l = \frac{l_a}{L}$$

Como é assumido que a segmentação do contorno procura dividi-lo em partes estruturalmente significativas, o atributo tamanho relativo procura identificar segmentos que possuam, proporcionalmente, a mesma porção de contorno.

Finalmente, o vetor $\vec{h}$ pode ser montado atribuindo-se a cada um de seus elementos um valor obtido a partir do histograma (t_i, c_i) . Valores típicos são os momentos de probabilidade. Em nossas aplicações utilizamos apenas a média do histograma, fazendo com que $\vec{h}$ seja unidimensional. A média pode ser facilmente calculada pela fórmula :

$$m = \sum_{i=0}^n t_i c_i$$

Os valores obtidos a partir do histograma normalizado procuram descrever estruturalmente o segmento, diferenciando segmentos que possuam estruturas diferentes. Seja a figura 3.16. Suponha que a curvatura calculada em cada ponto de controle seja a mesma para ambos os segmentos. Nesse caso, todos demais os atributos podem coincidir, e o histograma seria a forma de diferenciar os dois segmentos.

Uma vez definido o descritor de segmentos, resta agora a criação de um mecanismo que permita medir a semelhança entre dois segmentos. Em outras palavras, falta a definição de uma medida de distância entre dois descritores. Uma função comumente utilizada é conhecida como **método dos mínimos quadrados** [23].

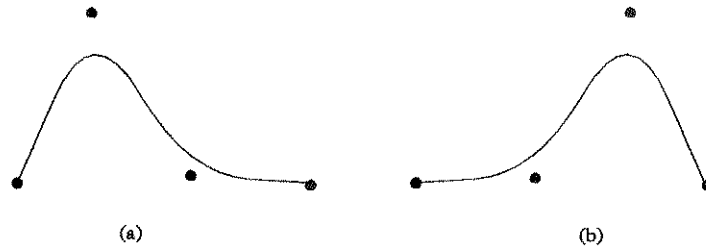


Figura 3.16: Situações que o atributo h procura diferenciar.

definição 3.3 (Método dos Mínimos Quadrados) *Dados dois descritores Bs , $A = \{a_0, a_1, \dots, a_5\}$ e $B = \{b_0, b_1, \dots, b_5\}$, onde as componentes de $\vec{d}$ e p_m foram representadas diretamente, a função de classificação $D(A, B)$ é definida como :*

$$D^2(A, B) = \sum_{i=0}^5 |a_i - b_i|^2 m_i f_i$$

onde m_i é um fator normalizador e f_i um fator de peso.

Os fatores de normalização m_i devem ser definidos de forma que todos os atributos tenham a mesma ordem de grandeza. Os fatores de peso f_i podem variar de aplicação para aplicação, dependendo da importância de cada atributo. A definição numérica desses fatores faz parte de um processo chamado aprendizagem [23]. Os pesos utilizados em nossas aplicações são apresentados no capítulo 6.

Capítulo 4

Casamento de Segmentos

4.1 O problema de correspondência

O casamento de contornos, segunda etapa no processo proposto de reconstrução 3D por superfícies, encontra formulações similares em outros problemas de visão computacional. Uma forma mais geral de interpretá-lo é encarando-o como um problema de estabelecimento de correspondência entre pontos de contornos diferentes. Ou seja, dados dois contornos P e Q , é desejado que se estabeleça uma correspondência entre eles, identificando, para os pontos de P , quais são seus "similares", ou "correspondentes" em Q .

Existe um problema mais genérico que é o de achar a correspondência entre pontos de pares de imagens, problema esse encontrado em análise de imagens [18]. Por exemplo, em estereoscopia binocular (visão estereoscópica), a correspondência entre 2 imagens, captadas por 2 dispositivos, é o primeiro passo para percepção de profundidade e a detecção de movimento [9]. Porém, a maioria dos métodos que tratam desse problema são caros computacionalmente. Por esse motivo, é desejável a redução do número de pontos a ser processados.

No caso da estereoscopia binocular, uma sugestão preferida é a de usar contornos, e o gradiente de intensidade nesses pontos, como elementos para o estabelecimento da correspondência. Na análise de movimento, a primeira tarefa consiste em medir-se o deslocamento dos elementos da imagem 2D, também denominado fluxo ótico. Uma das abordagens propostas se fundamenta na correspondência entre contornos, solução parecida

com a do problema em estereoscopia [9].

Segundo [18], além da estereoscopia e detecção de movimento, a correspondência entre contornos pode ser usada ainda em reconhecimento de escrita, verificação de assinaturas e geração de "frames" para desenho animado. Esse trabalho apresenta um método de casamento de segmentos, utilizando programação dinâmica, denominado "elastic matching". Dados dois contornos, representados por aproximação poligonal, dentre os quais um é considerado contorno inicial, e o outro contorno meta, uma sequência de contornos intermediários é criada. Essa sequência procura representar uma transformação do contorno inicial ao meta. O conceito básico por trás desse método está na identificação de um "campo de força" que, agindo sobre o contorno original, procura modificá-lo para que fique similar ao meta. A correspondência é obtida seguindo-se o caminho de cada ponto pelos contornos gerados.

Os trabalhos [87] e [51] utilizam abordagem similar a [18] para solucionar a correspondência. Em [87], o método de [18] é estendido para o caso de sub-divisões dos contornos intermediários permitindo, por exemplo, que exista um contorno inicial e dois ou mais meta. O método de [51] também utiliza programação dinâmica e campos vetoriais, gerando contornos intermediários. Porém, nesse trabalho, trechos dos contornos inicial ou do meta podem ficar sem correspondência. Além disso, uma das restrições do método é que os contornos intermediários variem suavemente.

Uma das aplicações do casamento de contornos em reconhecimento de padrões é a identificação de objetos parcialmente escondidos. Uma vez que os contornos modelo dos objetos a serem reconhecidos foram segmentados, o contorno extraído da imagem de entrada é também segmentado, e esses segmentos são comparados com os dos modelos. Existem diversas formas para efetuar essa comparação, que utilizam segmentação por aproximação poligonal, primitivas sintáticas, descritores de Fourier e outros. O trabalho [72] apresenta um método que usa segmentação por aproximação poligonal e um algoritmo de busca em grafos, o A^* , para a identificação dos objetos mais prováveis na imagem.

Os trabalhos [52] e [61] usam o casamento de segmentos em processos de reconstrução 3D por superfícies. Para ilustrar melhor o funcionamento desses métodos, tome-se a figura 4.1.

Em [52], o casamento é feito por um processo *ad hoc*, com critério de menor

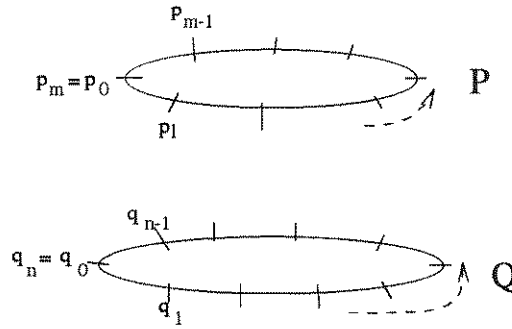


Figura 4.1: Caso geral do problema de casamento entre dois contornos.

distância euclidiana. Sejam dois contornos P e Q segmentados, $P = \{p_0, p_1, \dots, p_m\}$ e $Q = \{q_0, q_1, \dots, q_n\}$, como na figura 4.1. Primeiramente, o ponto p_0 é casado com o ponto q_i mais próximo. Como os pontos podem ser renomeados, sem perda de generalidade, esse primeiro ponto será chamado q_0 . O próximo ponto a ser testado é p_1 . As distâncias de p_1 aos pontos de Q , a partir de q_1 , são calculadas. O método assume que existe um ponto q_i tal que a distância $d(p_1, q_i)$ é um ponto de mínimo. Assim, o método calcula as distâncias até que essas comecem a aumentar em relação às anteriores. Nesse ponto, a busca é interrompida, e o ponto de menor distância é tomado. O método permite que junções sejam feitas em ambos os contornos. Para isso, uma vez fixado um ponto q_k , a mesma busca é feita no contorno P . Das duas menores distâncias, a menor define o casamento. O processo continua até que os contornos tenham sido percorridos completamente.

Em [61], uma vez que os contornos foram segmentados, toma-se aquele com menor número de segmentos, e realiza-se operações de junção ("merge") no outro, para que fique com o mesmo número de segmentos. Para cada segmento em cada contorno é atribuído um vetor de atributos, denominado "3-d-c-descriptor". Esse vetor tem como componentes alguns coeficientes ligados à curvatura e torção de cada segmento, uma vez que os contornos são espaciais. Calculados esses vetores, pode-se aplicar uma medida de similaridade a eles. Essa medida serve como função custo para o processo de junção e casamento entre os contornos. O método se baseia na técnica de busca em grafos denominada "best-first search", muito usada em problemas de inteligência artificial. Sejam novamente os contornos P e Q da figura 4.1. Um casamento inicial (p_0, q_0) é realizado. Supondo que $n > m$, então

as junções são feitas no contorno Q . A partir do nó (p_0, q_0) , os próximos nós possíveis são:

$$(p_1, q_1), (p_1, q_2), \dots, (p_1, q_k)$$

onde k depende do número de junções que são realizadas. Cada par de nós, o "pai" mais um dos "filhos", define um casamento entre dois segmentos. Tomando, por exemplo, os nós (p_0, q_0) e (p_1, q_1) , o casamento é definido entre os segmentos $\widehat{p_0 p_1}$ e $\widehat{q_0 q_1}$. Assim, para cada nó "filho" é calculada a distância, baseada na medida de similaridade, entre os segmentos casados. O nó que possui o menor custo é expandido da mesma forma. O processo prossegue escolhendo sempre o nó mais barato, até que os contornos sejam completamente percorridos.

O algoritmo implementado para casamento de segmentos aproveita idéias dos processos propostos em [52] e [61], e utiliza o algoritmo A^* para a busca do melhor casamento. A secção 4.2 apresenta uma introdução aos métodos heurísticos de busca em grafos, além de discutir particularmente o funcionamento do A^* . A secção 4.3 apresenta o método desenvolvido.

4.2 O Algoritmo A^*

4.2.1 Busca em um Espaço de Estados

A busca heurística é um tópico central em inteligência artificial (IA) [2]. Sua importância é devida ao conceito de *representação no espaço de estados*, onde diferentes problemas são transformados no de achar um caminho no espaço de estados, de um nó inicial a um meta. Para tornar essa busca eficiente, técnicas de busca que usam informação heurística sobre o problema foram desenvolvidas. De maneira geral, o espaço de estados é representado como um grafo, onde os nós são estados possíveis no processo de resolução de um determinado problema, e as arestas representam as passagens de um estado para outro. Por exemplo, em uma modelagem por espaço de estados de um jogo de xadrez, cada configuração possível das peças no tabuleiro é um estado, ou um nó do grafo. Uma jogada, ou um movimento de um dos jogadores, define uma passagem de um estado para outro, ou seja, uma aresta entre os nós. A representação por espaço de estados é a base de muitos métodos de IA. Sua estrutura reflete à da resolução de problemas em dois aspectos:

- permite uma definição formal do problema como sendo a conversão de uma dada situação, ou *estado inicial*, em uma situação desejada, ou *estado meta*, usando um conjunto de operações permitidas.

- faz com que o processo de resolução de um determinado problema seja uma combinação de técnicas conhecidas (cada uma representada como uma regra definindo uma aresta no espaço), e *busca*, uma técnica geral de explorar o espaço para tentar achar algum caminho entre os estados inicial e meta.

Voltando ao exemplo do xadrez, o estado inicial do espaço de estados corresponde ao tabuleiro em sua configuração de início de partida; o estado final define uma posição de xeque-mate.

Exemplos de métodos de busca são a busca em profundidade, em largura, "best-first" e o A^* [22]. Este será apresentado na próxima secção. Os conceitos e exemplos das secções que tratam do A^* são baseados principalmente nos trabalhos [11], [2], e no livro [22].

4.2.2 "Best-First Search" e o A^*

A busca em profundidade é boa porque permite que uma solução seja encontrada sem que todos os nós sejam expandidos. A busca em largura é boa porque não acaba em "becos sem saída". Uma maneira de combinar essas duas é seguir um caminho de cada vez, mas mudando de caminho toda vez que algum outro parecer mais promissor.

Em cada etapa da busca por "best-first", o nó gerado mais promissor é selecionado. Isso é feito aplicando-se uma heurística apropriada a cada um. Então, o nó selecionado é expandido para gerar seus sucessores. Se um deles for uma solução, o processo termina. Senão, esses sucessores são adicionados à lista de nós gerados anteriormente. Novamente, o nó mais promissor é selecionado e o processo continua. Geralmente, o que acontece é que uma busca em profundidade está sendo feita quando o nó mais promissor está sendo explorado. Mas eventualmente, se uma solução não for encontrada, a confiança nesse caminho pode diminuir, e em outros, aumentar.

Para ilustrar o processo, tome-se a figura 4.2.

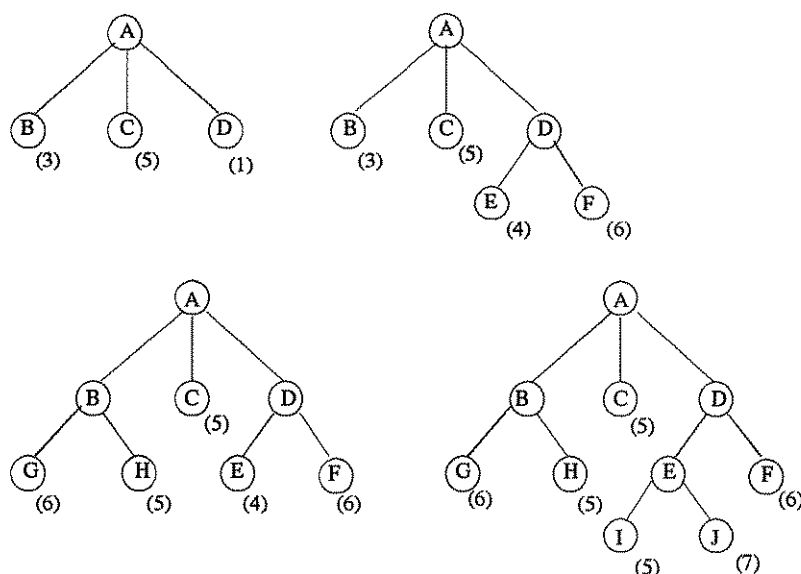


Figura 4.2: Exemplo do funcionamento do algoritmo "best-first search".

Inicialmente, tem-se o nó *A*, que como é único, é expandido, gerando *B*, *C* e *D*. Entre parênteses está indicado o custo de cada nó. O de menor custo é o nó *D*, que é expandido, gerando *E* e *F*. Dessa vez, o nó mais barato passa a ser *B*, que é expandido, gerando *G* e *H*. É importante notar que os nós examinados são sempre os terminais, ou folhas, ou ainda, os que não foram expandidos ou examinados anteriormente. Dessa vez, o nó *E* é o mais promissor, sendo expandido e gerando *I* e *J*. O processo continua até que um nó meta seja atingido.

Existem problemas que requerem a geração de grafos, ao invés de árvores, como a da figura 4.2. Como o método se comporta nesses casos? A figura 4.3 (a) mostra uma situação em que, a partir do nó *A*, os nós *B*, *C*, *D*, *E* e *F* foram gerados por processo análogo ao anteriormente explicado. O último nó gerado foi *F*, filho de *E*. O nó mais barato, e que deverá ser expandido, é *C*. Supondo que *C* também gere *F*, cria-se um ciclo no grafo (fig. 4.3 (b)). Para entender essa solução, deve-se interpretar a árvore gerada como sendo um grafo orientado valorado, onde cada aresta é dirigida do nó pai para o filho (fig. 4.3 (b)). O método para eliminar o ciclo procura identificar o caminho mais barato entre o nó inicial e aquele que criou o ciclo. No caso da figura 4.3, o caminho mais barato é aquele que passa por *C*. A ligação de *E* para *F* é cortada, e uma nova ligação, de *C* para *F*, criada. Além disso, o valor do nó *F* é atualizado.

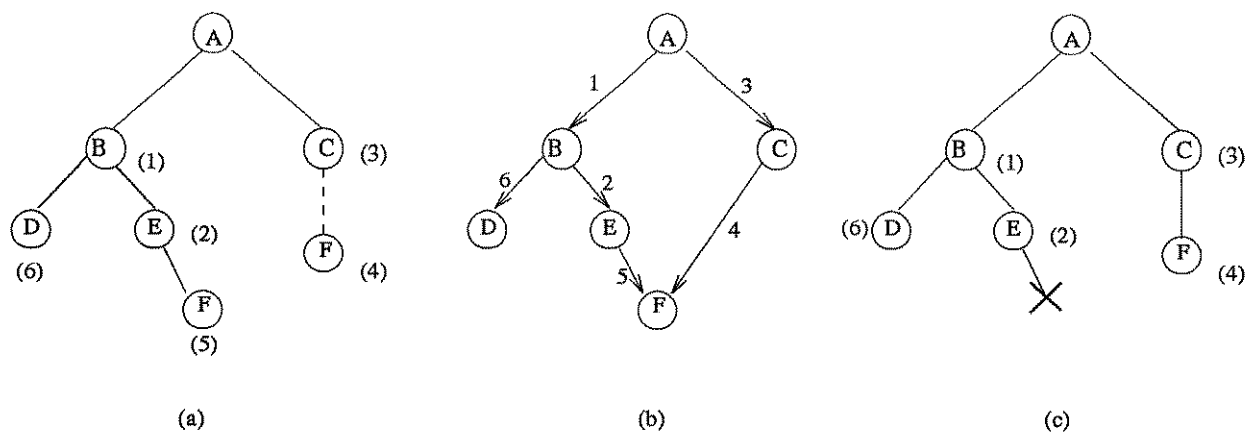


Figura 4.3: Espaço de estados e "best-first search".

O funcionamento do algoritmo é relativamente simples. Ele procede por passos, um nó é expandido a cada um, até que um nó meta seja gerado. Em cada passo, o algoritmo seleciona o nó mais promissor que ainda não tenha sido expandido, gera os sucessores desse nó, aplica a função custo para cada um, e os adiciona à lista de nós que são testados nos próximos passos, depois de checar se algum já foi gerado anteriormente. Esse teste garante que não existirão ciclos no grafo. Então o próximo passo começa. O grafo gerado é conhecido como "or graph".

O algoritmo "best-first" é uma simplificação do algoritmo A^* . Antes de apresentar uma versão desse algoritmo, alguns detalhes são discutidos.

4.2.3 As Listas OPEN e CLOSED

Durante o funcionamento do A^* , além do próprio grafo representando o espaço de estados, duas listas são mantidas e constantemente atualizadas. São as listas OPEN e CLOSED.

A lista OPEN serve para armazenar ponteiros para os nós que foram gerados e tiveram a função custo aplicada a eles, mas ainda não foram expandidos, ou seja, não tiveram seus sucessores gerados. Na verdade, ela funciona como uma lista de prioridades, onde os elementos com maior prioridade são justamente os nós mais promissores. Pode ser implementada como uma lista ordenada em ordem crescente de custo de cada nó.

A lista CLOSED indica os nós que já foram examinados. Essa lista deve ser implementada para o caso em que, durante o processo de busca, aparecerem ciclos no grafo. Dessa forma, pode-se identificar nós que já foram expandidos mas que, por ventura, estejam em condições de nós terminais no grafo. Isso ocorre toda vez que caminhos mais baratos para todos os seus sucessores sejam encontrados.

4.2.4 A Função Heurística

O A^* também necessita de uma função que estime o quão promissor é cada nó gerado. Isso permite que o algoritmo procure o (aparentemente) melhor caminho. Essa função será denotada por f' , em contraposição à função f que daria a verdadeira avaliação de um determinado nó. Essa função f' é definida como uma soma de duas componentes g e h' . A função g fornece o custo de se chegar do nó inicial até aquele que está sendo examinado. É importante notar que g não é uma estimativa, mas sim a soma exata dos custos de cada aresta. Mais ainda, do melhor caminho encontrado até então, entre o nó inicial e o corrente. O custo de cada aresta é calculado pela função de similaridade, denotada por $c(m, n)$, custo entre as arestas m e n . A função h' é uma estimativa do custo do melhor caminho entre o nó corrente e um estado final, ou seja, uma solução. É nessa função que o conhecimento heurístico sobre o problema deve ser explorado. A função f' , portanto, é uma estimativa do custo de chegar a uma solução, partindo do estado inicial, e passando pelo nó em questão. Se existir mais de um caminho desses, o algoritmo guarda o melhor deles. É importante notar que, uma vez que g e h' devem ser somadas, é importante que h' seja uma medida do custo de chegar a uma solução a partir do nó corrente, ou seja, bons nós devem ter baixo valor de h' , maus nós devem ter valores altos. Também é importante que g seja não negativa.

Com essas explicações, o algoritmo A^* pode ser apresentado.

4.2.5 O Algoritmo A^*

São apresentadas duas versões do algoritmo. A primeira, com um maior nível de abstração, é uma adaptação do algoritmo apresentado em [2]. A segunda, um pouco mais detalhada quanto ao aspecto de implementação, foi desenvolvida durante os estudos

realizados a partir da primeira.

Algumas considerações merecem ser feitas sobre o algoritmo. A função g permite que a escolha do próximo nó a ser expandido seja feita não só com base em quão próximo se está da solução, medido pela função h' , mas também quão bom foi o caminho até aquele nó. A incorporação de g em f' faz com que o nó escolhido não seja simplesmente aquele que está mais próximo da solução. Essa característica é desejável se o importante for o caminho encontrado, e não simplesmente uma solução. Como será mostrado, esse é o caso do método de casamento de segmentos desenvolvido.

Se a função h' fosse uma estimativa perfeita de h , então o A^* convergiria imediatamente ao estado meta. Quanto melhor for h' , mais perto de um processamento sem "backtracking" o algoritmo está. Por outro lado, se h' for sempre 0, então o A^* irá procurar o caminho mais barato, de acordo com g . Esse é o caso do algoritmo de casamento implementado em [61]. É possível provar que, se para qualquer nó n , $h'(n)$ for um limitante inferior para $h(n)$, então o A^* é *admissível*, isto é, ele sempre acha o melhor caminho. Além disso, é feita uma simplificação no A^* se a *restrição de consistência* for obedecida pela função h' . Segundo essa restrição, para qualquer nó m e qualquer de seus sucessores n , tem-se:

$$h'(m) - h'(n) \leq c(m, n)$$

A interpretação dessa restrição diz que a estimativa h' deve sempre melhorar quando se parte de um nó para seu sucessor. Se essa restrição for obedecida, pode ser provado que, quando o A^* expande um nó n , ele já encontrou o melhor caminho até n . Na prática, isso quer dizer que o A^* nunca irá reexpandir um nó.

Um exemplo pode ilustrar uma propriedade importante da admissibilidade do A^* . Seja a situação da figura 4.4 (a).

Inicialmente, o nó A gerou os nós B e C. O custo f' é indicado como a soma entre h' e g . B é o nó mais barato, e gera D. Como há um empate entre D e C, o algoritmo pode decidir arbitrariamente. Supondo que ele decida pelo nó corrente, então E é gerado. Note-se que fica uma impressão de "perda de tempo" nesse caminho, uma vez que sempre, em todos os nós que foram gerados a partir de B, a função h' estimou uma distância de 3 até uma solução. Ou ainda, sub-estimou, como no caso dos nós B e D. Agora o nó C deve

algoritmo 4.1 () :

- 1– *Algoritmo A*;*
- 2 – *Ponha o nó inicial s em OPEN. Faça:*

$$g(s) = 0; f'(s) = h'(s);$$
- 3 – *Se OPEN estiver vazia, então termine com fracasso;*
- 4 – *Remover de OPEN o nó n com menor f' , e colocá-lo em CLOSED;*
- 5 – *Se n for uma solução, então termine com sucesso;*
- 6 – *Expandir o nó n , gerando todos seus sucessores. Se não existir nenhum sucessor, vá para 3. Para cada sucessor n_i , calcular:*

$$g_i = g(n) + c(n, n_i)$$
- 7 – *Se um sucessor n_i não estiver nem em OPEN, nem em CLOSED, faça:*

$$g(n_i) = g_i; f'(n_i) = g_i + h'(n_i);$$
Coloque n_i em OPEN e no grafo.
- 8 – *Se um sucessor n_i estiver em OPEN ou em CLOSED, e se $g(n_i) > g_i$, então atualize o nó fazendo:*

$$g(n_i) = g_i; f'(n_i) = g_i + h'(n_i);$$
Atualize o grafo com o melhor caminho encontrado, e coloque n_i em OPEN, se ele estiver em CLOSED.
- 9 – *Vá para 3;*
- 10– *Fim.*

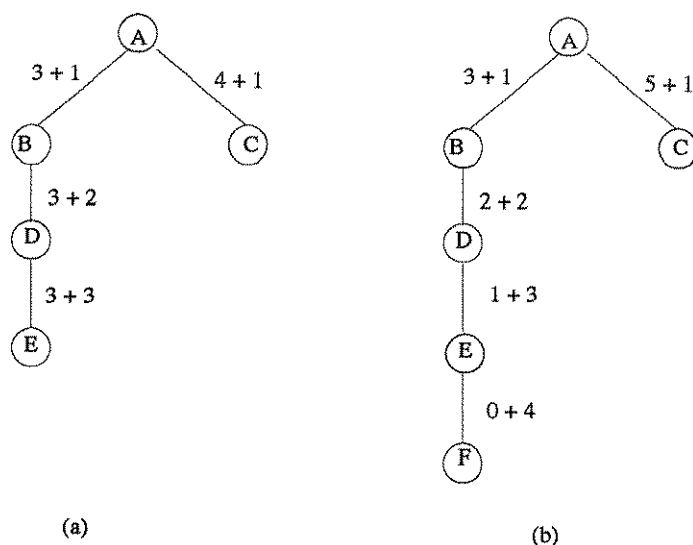


Figura 4.4: Avaliação de h' : (a) $h' \leq h$ e (b) $h' > h$.

ser expandido. Porém, além de um esforço inútil, não existem consequências mais sérias.

Agora, seja a figura 4.4 (b). O nó A gerou B e C. O B gerou D, que gerou E, que gerou F, que é uma solução. Mas, e se existir uma aresta entre C e F, ou seja, e se F for um sucessor de C? Nesse caso, existe um caminho melhor até a solução. Só que, como $h'(C)$ superestimou $h(C)$ ($=1$), essa melhor solução não foi encontrada.

Uma discussão mais completa sobre as condições que tornam o A^* ótimo, em relação à geração de um menor número de nós durante a busca, é encontrada em [11] e [2].

Será apresentada agora a segunda versão do A^* , desenvolvida a partir da primeira durante a implementação.

Embora o capítulo 6 vá discutir com maiores detalhes a implementação de todos os programas, cabe aqui apresentar detalhes do A^* para facilitar seu entendimento. Tome a figura 4.5. Nela estão representadas as principais estruturas de dados usadas pelo algoritmo. O grafo G , pelo próprio funcionamento do A^* , pode ser implementado como uma árvore. OPEN, CLOSED e n_i (algoritmo 4.2) podem ser implementadas como listas de ponteiros para nós do grafo. Além disso, OPEN deve ser uma lista ordenada, fazendo com que o primeiro nó seja sempre o próximo a ser expandido.

algoritmo 4.2 () :

```

1-  Algoritmo A* /* v. 2 */
    2 - temp = sgrcreate(G); /* cria e inicializa G */
    3 - sglcreate(OPEN,temp);
    4 - sglcreate(CLOSED,NIL);
    5 - Se empty(OPEN) então termine com falha;
    6 - n = firstnode(OPEN);
    7 - sglinsert(CLOSED,n);
    8 - Se goal(n) então termine com sucesso;
    9 - ni = explode(n);
    10 - Se empty(ni) então vá para o passo 5;
    11 - e = firstnode(ni);
    12 - Enquanto e ≠ NIL faça:
        g = n.g + c(n,e);
        e0 = belongs(e,OPEN);
        Se e0 == NIL então
            e0 = belongs(e, CLOSED);
        Se e0 == NIL então faça:
            e.h = h(e); e.g = g; e.f = g + e.h;
            temp = sgrinsert(G,n,e);
            sglinsert(OPEN,temp);
        Senão
            Se e0.g > g então
                atualize(G,OPEN,n,e0);
            e = firstnode(ni);
    13 - Fim_enquanto;
    14 - Vá para o passo 5;
15-  Fim_A*.

```

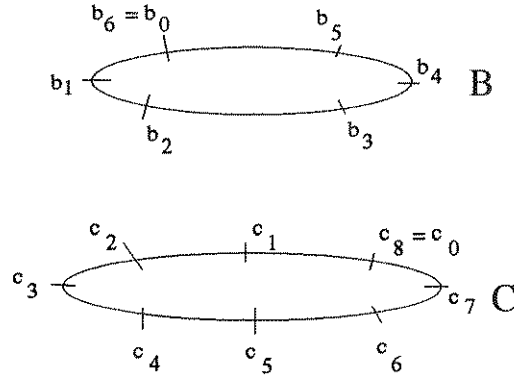



Figura 4.6: Situação inicial para o casamento dos contornos B e C .

que o conjunto-solução do casamento é dado por:

$$C = \{(b_i, c_j), i = 0, \dots, m, j \in \{0, \dots, n\}\} \quad (4.2)$$

Esta restrição pode ser facilmente relaxada no método desenvolvido, permitindo que pontos de B e C não sejam casados. Por outro lado, a restrição é necessária para o funcionamento do método de ajuste de superfícies (cap. 5). Além disso, ela permite que alguns fatos interessantes sejam usados. As duas implicações mais importantes que essa restrição traz são:

(1) o número de pontos do contorno C que deverão ser descartados pode ser calculado de antemão. Esses pontos definem o número de junções que existirão no processo. No caso dos contornos da equação 4.1, existirão $(n - m)$ junções.

(2) uma vez fixado o primeiro par, ou um casamento inicial, os outros casamentos são calculados ordenadamente, de acordo com os pontos $b_1, b_2, \dots, b_m$.

Sejam os contornos da figura 4.6, representados pelos conjuntos:

$$B = \{b_0, b_1, \dots, b_6\}$$

$$C = \{c_0, c_1, \dots, c_8\}$$

O primeiro passo é calcular todos os casamentos possíveis entre todos os segmentos. Como neste passo o interesse é por um casamento inicial, não são efetuadas as junções. Cada casamento inicial define um grafo de busca. O caminho mais barato encontrado entre

todos os grafos resulta no melhor casamento. Esse primeiro passo fornece o valor associado a cada casamento inicial com o auxílio da função de similaridade.

Uma outra forma de armazenar os segmentos é:

$$B = \{(b_0, b_1), \dots, (b_5, b_6)\}$$

$$C = \{(c_0, c_1), \dots, (c_7, c_8)\}$$

Assim, suponha que os segmentos que apresentarem menor valor de distância entre seus atributos sejam (b_1, b_2) e (c_3, c_4) . Então, o primeiro par a ser casado seria (b_1, c_3) . Eles podem ser renomeados como (b_0, c_0) , e todos os outros pontos reindexados (fig. 4.7).

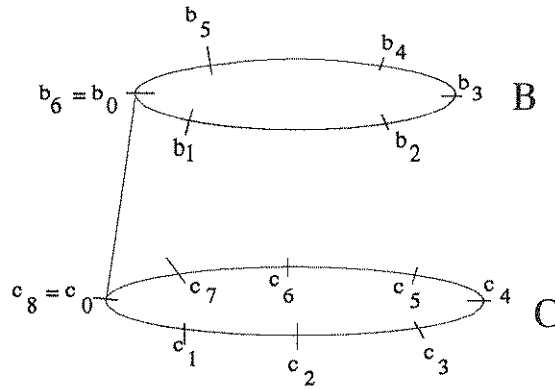


Figura 4.7: Casamento dos pontos b_0 e c_0 .

Tendo estabelecido o casamento inicial estabelecido, deve-se descobrir quem casará com b_1 . A princípio, b_1 pode casar com $c_1, c_2, \dots, c_7$. Se b_1 casar com c_1 , nada de novo aconteceu. Se b_1 casar com c_2 , então uma junção entre os segmentos $\widehat{c_0c_1}$ e $\widehat{c_1c_2}$ foi realizada, eliminando o ponto de segmentação c_1 . De acordo com a observação (1) realizada anteriormente, esse fato é usado para a definição de um limitante para o número de folhas a serem expandidas. Ou seja, como o número total de junções a serem realizadas é dada por $8 - 6 = 2$, o casamento inicial gerará 3 nós da árvore (fig. 4.8), pois se for realizado o casamento (b_1, c_4) , então 3 pontos são eliminados de C . Isso implica que algum ponto de B ficará "solteiro", violando a restrição (1). Supondo que o casamento mais barato foi (b_1, c_1) , o processo fica como na figura 4.9.

Explodindo novamente, tem-se a situação da figura 4.10.

Supondo que o nó mais barato é (b_2, c_3) , tem-se a situação da figura 4.11.

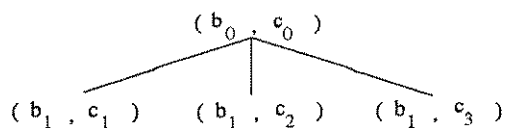


Figura 4.8: Primeira explosão no espaço de estados.

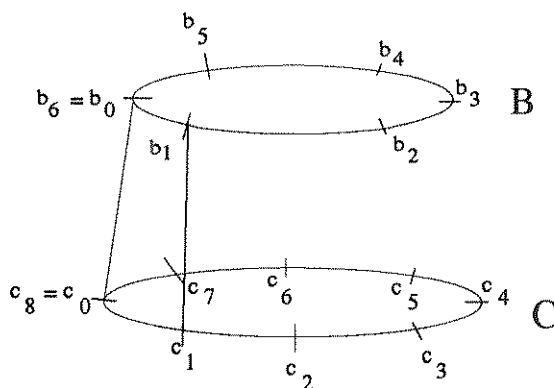


Figura 4.9: Casamento dos segmentos b_0b_1 e c_0c_1 .

A continuação do processo pode ser como na figura 4.12, com o resultado na figura 4.13.

O processo de casamento pode ser visualizado numa grade formada pelos pontos do produto cartesiano $B \times C$ (fig. 4.14).

Agora que o exemplo ilustrou os principais aspectos do funcionamento do processo de casamento, serão formalizados alguns conceitos, para que possam ser usados no A^* .

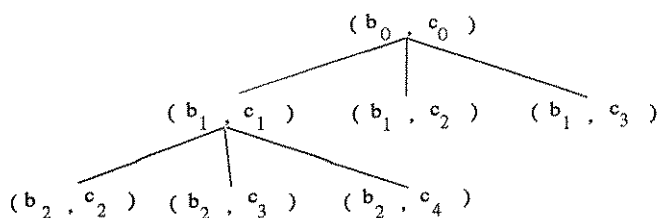


Figura 4.10: Segunda explosão no espaço de estados.

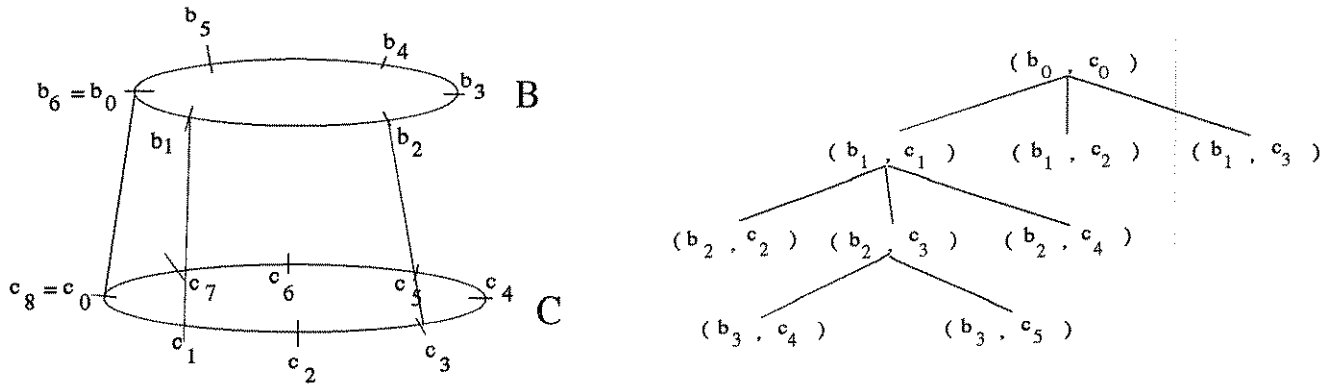


Figura 4.11: Situação de junção no casamento (entre os segmentos c_1c_2 e c_2c_3).

4.3.2 O Grafo de Busca

Como foi mostrado, uma vez que se saiba quem é o ponto de segmentação inicial de cada contorno, indexados por b_0 e c_0 , respectivamente, todos os outros pontos de segmentação são referenciados por números inteiros. Assim, uma grande simplificação pode ser feita armazenando-se, em cada nó, apenas um par de números inteiros, referenciados por b_i e c_j indicando o ponto de segmentação de cada contorno, além dos valores g , h' e f' . Suponha que os segmentos de cada contorno estão armazenados em listas de segmentos, obedecendo a ordem de cada contorno. As funções que fazem a "ligação" entre o A^* e as listas de segmentos são h' e $c(m, n)$. Assim, para usar essa indexação, basta que elas saibam como calcular os segmentos correspondentes. Além disso, não se pode perder de vista que o casamento desejado é definido pelo caminho da solução encontrada pelo A^* . Portanto, uma vez que a solução foi encontrada, o grafo deve permitir que um caminho de volta seja identificado, partindo do nó final e, através de cada predecessor, chegando ao nó ou estado inicial.

4.3.3 A Função Custo $c(m, n)$

Como já foi dito, a partir de um par de nós do grafo, um pai e um filho, pode-se identificar um par de segmentos. Uma vez identificados esse par, a partir dos nós indicados nos parâmetros da função custo, pode-se aplicar a função de similaridade da definição 3.3 para o cálculo do custo entre os dois nós. Os fatores peso podem ser ajustados de acordo

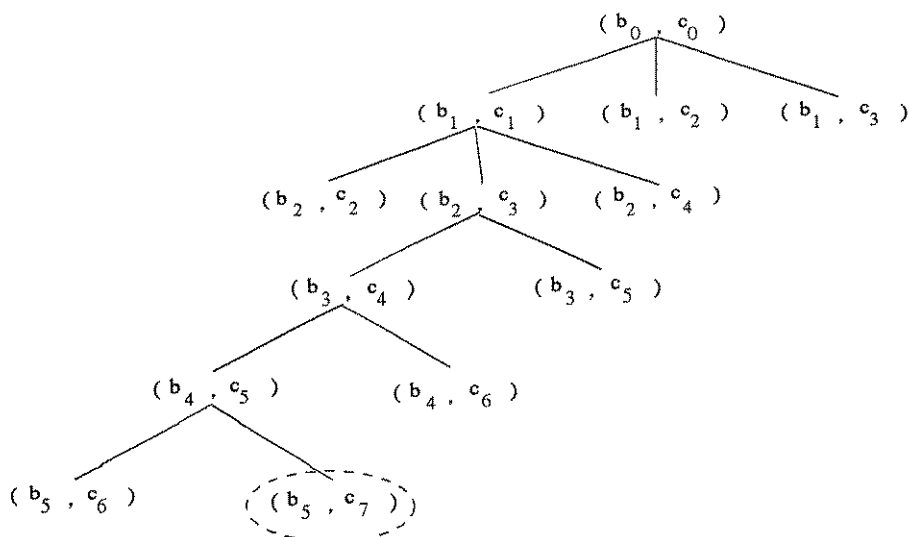


Figura 4.12: Situação de sucesso no espaço de estados.

com a aplicação. O capítulo 6 apresenta os valores utilizados em nossos testes.

4.3.4 As Funções h' , $\text{explode}(n)$ e $\text{goal}(n)$

A função $h'(e)$ depende da posição em que o processo se encontra em cada curva, ou seja, de quantos segmentos já foram casados, e quantos ainda faltam. A posição é dada pelo próprio ponteiro e . A informação sobre quantos segmentos faltam é obtida através de duas variáveis $bmax$ e $cmax$, que indicam o número de segmentos de cada contorno. No

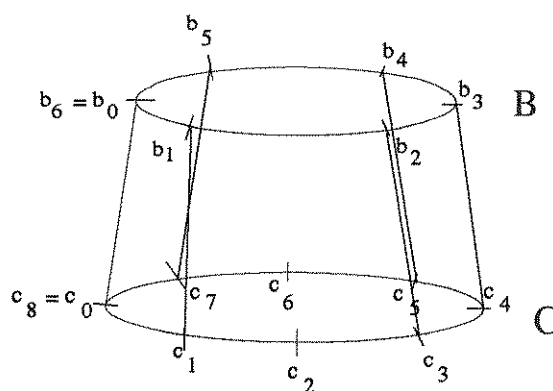


Figura 4.13: Os contornos B e C casados.

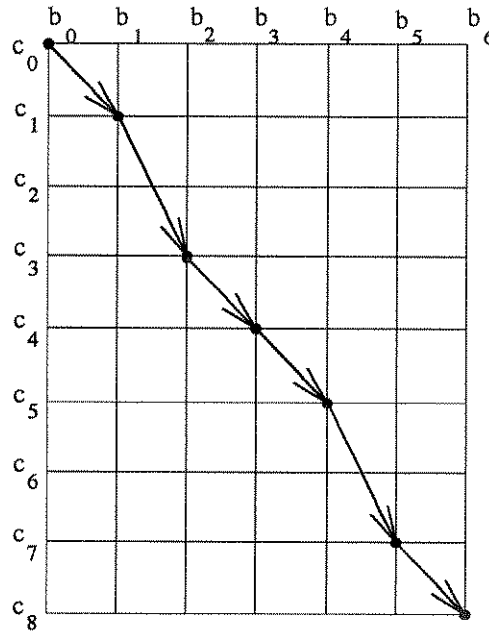


Figura 4.14: Processo de casamento visto em uma grade.

exemplo da figura 4.6 elas equivalem aos índices m e n , respectivamente, e assim tem-se $bmax = 6$ e $cmax = 8$.

Supondo que o ponteiro e está apontando para um nó do grafo com valores $bi = i_0$ e $cj = j_0$, então a função $h'(e)$ será da forma:

$$h'(e) = f(bmax, cmax, i_0, j_0)c_m(m, n)$$

onde a função c_m é uma estimativa do custo médio de uma aresta do grafo. Essa função deverá ser criada como uma estimativa, uma vez que não há como saber quanto vai custar cada aresta no caminho a ser encontrado. A função f deve estimar quantos nós ainda faltam, do nó corrente até o solução.

Utilizando a notação que, em determinado instante t_0 do processo, tem-se $bi = b_{i0}$ e $cj = c_{j0}$, então o número de segmentos que faltam ser casados é dado por:

$$\gamma = bmax - b_{i0} \quad (4.3)$$

É importante perceber que a profundidade da árvore que representa o espaço de estados até que uma solução seja encontrada é sempre a mesma, dependendo apenas do

número de segmentos do contorno B . Portanto, não importa quantas junções foram feitas até um certo nó da árvore, o número de segmentos que faltam ser casados é $\gamma - 1$. Logo, o custo estimado do caminho a ser percorrido até uma solução, seja ela qual for, é dado por:

$$(\gamma - 1)c_m(m, n) \quad (4.4)$$

Assim, a estimativa h' é ser calculada como:

$$h' = (\gamma - 1)c_m(m, n) \quad (4.5)$$

Um fato importante relacionado ao processo e à função "explode(n)" está ligado às junções de segmentos. Para explicar e melhor entender algumas implicações dessas junções e a incorporação dessa informação à função "explode(n)", são apresentados alguns pequenos exemplos.

Seja, primeiramente, o caso em que as junções só podem ocorrer no contorno com um número maior de segmentos (como foi assumido na secção passada, o contorno C). Então, o número de junções é dado pela diferença $c_{max} - b_{max}$. Logo, no exemplo da figura 4.6, ter-se-ia $8 - 6 = 2$ junções no contorno C . Portanto, essa é uma informação conhecida *a priori* sobre o futuro do processo. Se, em determinada etapa do processo, 1 junção já tiver sido feita, sabe-se que apenas 1 deverá ser realizada. Mais do que isso, 1 tem que ser realizada para o sucesso do processo. Por outro lado, se ambas junções tiverem sido realizadas, então sabe-se que *mais nenhuma junção pode ser realizada*. Essas informações possuem capital importância no processo de casamento, estando diretamente ligadas à etapa de expansão dos nós do grafo gerado pelo A^* .

Para saber quantas junções foram realizadas até certo ponto do processo, não é necessário contá-las em cada caminho. Não perdendo de vista que as junções estão sendo realizadas no contorno com maior número de segmentos, suponha que o processo se encontra no nó (b_5, c_7) . Como ambos contornos começaram a ser contados do 0 (b_0 e c_0 , respectivamente), então a única maneira de C estar no ponto de segmentação c_7 e B em b_5 é que já tenham ocorrido 2 junções nos segmentos de C .

Para o caso em que as junções podem ser feitas em ambos contornos, essas observações são parcialmente falsas. Por um lado, não se pode saber exatamente quantas junções serão realizadas. Pode-se sim afirmar qual o *número mínimo* de junções que serão

feitas de uma determinada etapa do processo até seu término. No caso do exemplo anterior, pode-se dizer *a priori* que serão feitas pelo menos 2 junções no contorno C ; isso porque, se ocorrer 1 junção no contorno B , então deverão ocorrer 3 junções no contorno C , e assim por diante. Por outro lado, no mesmo caso analisado anteriormente, se o processo se encontra no nó (b_5, c_7) , pode-se afirmar que, no mínimo, foram realizadas 2 junções nos segmentos do contorno C . Porém, mesmo assim, completa ou incompleta, a informação sobre as junções, em ambos os casos, é extraídas dos índice bi e cj .

Tome-se o caso em que as junções só ocorrem no contorno C . O número de junções que o processo deve realizar é dado por:

$$\hat{j} = cmax - bmax \quad (4.6)$$

O número de junções realizadas até um determinado nó n_s é dado por:

$$j(b_s, c_s) = c_s - b_s \quad (4.7)$$

A informação sobre o número de junções realizadas deve dirigir a explosão de cada nó. Seja a função μ definida como:

$$\mu(\hat{j}, j(b, c)) = \begin{cases} 0, j(b, c) > \hat{j} \\ 1, \text{caso contrário} \end{cases} \quad (4.8)$$

A função μ é usada como critério de explosão do nó. Se ela retornar 0, então aquele nó não é válido; caso contrário, o nó sucessor deve ser gerado.

Tomando novamente o processo, supondo que o nó a ser expandido é denotado por n_s , a função explode(n) é resumida no algoritmo 4.3.

Finalmente, a função goal(n), que representa a condição de parada com sucesso do algoritmo, é implementada usando a função γ . Se um nó meta for atingido, então $\gamma = 1$.

algoritmo 4.3 (Função explode(n)) :

1 - Função *explode*(n_s).

2 - $b_{new} = n_s - > bi + 1$;

3 - $c_{new} = n_s - > cj + 1$;

4 - Enquanto $(\mu(\hat{j}, j(b_{new}, c_{new})) == 1)$ faça

- gere o nó (b_{new}, c_{new}) ;

- $c_{new} = c_{new} + 1$;

5 - Fim.

Capítulo 5

Ajuste e Visualização de Superfícies Paramétricas

5.1 Introdução

Nas áreas que compõem a computação de imagens (síntese e processamento de imagens e visão computacional), as superfícies, sua descrição, representação e apresentação, cumprem um papel fundamental em diversos problemas. O projeto e desenvolvimento de automóveis, navios, aviões, garrafas e sapatos são exemplos de aplicações em síntese de imagens. Estudos de geografia e cartografia auxiliados por computador freqüentemente envolvem modelos que são expressados como superfícies paramétricas. Em processamento de imagens e reconhecimento de padrões, as superfícies entram na análise de imagens de pelo menos duas maneiras. Primeiro, pode-se tomar uma imagem como uma superfície, usando a intensidade de cada pixel como a terceira coordenada. Segundo, pode-se considerar as superfícies dos objetos que aparecem na imagem. Dessa maneira, o problema de segmentação de imagens é reduzido ao da identificação de grupos de pixels que formam uma determinada superfície na imagem.

Na maioria dessas aplicações, as superfícies envolvidas são demasiadamente complexas para serem descritas por apenas uma equação, levando à abordagem de aproximação por partes. As aproximações mais simples são as lineares por partes, particularmente a poliédrica triangular [82]. Seja $\{P\}$ um conjunto de pontos sobre uma superfície S . Então,

cada 3 pontos (não colineares) definem um plano, e uma escolha apropriada de triplas constitui uma aproximação poliédrica de S . Infelizmente, em muitos casos o número de triângulos necessário para uma aproximação razoável é muito grande, e as superfícies de grau mais elevado devem ser consideradas. Isso leva ao problema de como representar essas superfícies por partes.

Uma solução é definir um conjunto de curvas sobre a superfície e, associados a essas curvas, "patches" interpolantes. Outra solução é a utilização de pontos de controle, de maneira semelhante aos de curvas. Associadas ao primeiro tipo estão as superfícies de Coons, e similares, que procuram criar a superfície a partir de dados conhecidos (representados nas próprias curvas). Ao segundo tipo estão ligadas as superfícies de Bézier e outras semelhantes.

As superfícies formam um tópico bastante vasto, rico e diverso de estudo, onde as primeiras aplicações gráficas, segundo [56], datam da década de 40, com a utilização de cônicas para o auxílio do projeto do primeiro avião a jato, e chegando às superfícies racionais, Bézier e B-Splines, atualmente muito utilizadas em aplicações de CAD/CAM [26]. Os trabalhos [56, 26] apresentam históricos detalhados da evolução dos estudos dessas superfícies.

Associados à modelagem por superfícies, estão os problemas de sua geração e visualização, envolvendo temas como subdivisão, remoção de linhas escondidas, modelos de iluminação e outros. Este capítulo apresenta uma revisão de alguns tipos de superfícies paramétricas, bem como à geração e visualização das mesmas. Aspectos diretamente ligados à reconstrução 3D são explorados no decorrer do capítulo.

5.2 Superfícies Paramétricas

As superfícies de Hermite e de Coons são apresentadas em uma notação diferente das superfícies Bézier, B-Spline, e B-Spline racional, pois a notação utilizada nas primeiras é mais adequada ao método de reconstrução 3D desenvolvido, deixando explícitas as restrições necessárias para o ajuste de superfícies.

5.2.1 Curvas Paramétricas Cúbicas e Representação Matricial

Será introduzido um formalismo que é usado na definição de algumas superfícies paramétricas.

Os polinômios cúbicos que definem um segmento de curva $Q(t) = [x(t) y(t) z(t)]$, são dados por :

$$\begin{aligned} x(t) &= a_x t^3 + b_x t^2 + c_x t + d_x \\ y(t) &= a_y t^3 + b_y t^2 + c_y t + d_y \\ z(t) &= a_z t^3 + b_z t^2 + c_z t + d_z \end{aligned} \quad 0 \leq t \leq 1 \quad (5.1)$$

Tomando-se $T = [t^3 \ t^2 \ t \ 1]$, e definindo a matriz de coeficientes :

$$C = \begin{bmatrix} a_x & a_y & a_z \\ b_x & b_y & b_z \\ c_x & c_y & c_z \\ d_x & d_y & d_z \end{bmatrix} \quad (5.2)$$

pode-se escrever 5.1 como

$$Q(t) = [x(t) \ y(t) \ z(t)] = TC \quad (5.3)$$

A derivada de $Q(t)$ representa o vetor tangente paramétrico da curva. Aplicando essa idéia a 5.3, tem-se

$$\frac{dQ(t)}{dt} = Q'(t) = \left[\frac{dx(t)}{dt} \ \frac{dy(t)}{dt} \ \frac{dz(t)}{dt} \right] = \quad (5.4)$$

$$= \frac{dT}{dt} C = [3t^2 \ 2t \ 1 \ 0] C = \quad (5.5)$$

$$= [3a_x t^2 + 2b_x t + c_x \ 3a_y t^2 + 2b_y t + c_y \ 3a_z t^2 + 2b_z t + c_z] \quad (5.6)$$

Se dois segmentos se juntam, diz-se que a curva possui continuidade geométrica G^0 . Se as direções dos vetores tangentes dos dois segmentos no ponto de junção são as mesmas, então a curva tem continuidade geométrica G^1 .

Se os vetores tangente de dois segmentos de curva cúbica possuírem a mesma direção e norma no ponto de junção, diz-se que a curva tem continuidade de primeiro grau no parâmetro t , ou continuidade paramétrica, e é de classe C^1 .

Um segmento de curva $Q(t)$ é definido por restrições nos pontos finais (extremos), vetores tangentes e continuidade entre os segmentos de curva. Cada polinômio cúbico da equação 5.1 possui 4 coeficientes, e assim 4 restrições são necessárias para resolver o problema. As curvas de Hermite são definidas pelos pontos extremos e dois vetores tangentes. As curvas de Bézier, pelos dois pontos extremos e outros dois pontos que controlam os vetores tangentes.

Para ver como os coeficientes da equação 5.1 dependem de 4 restrições, seja a equação $Q(t) = TC$. Sejam ainda as matrizes M e G , com $C = MG$. M é uma matriz 4×4 chamada matriz base, e G é um vetor-coluna com 4 elementos, onde cada um possui uma componente para x , y e z . Esse vetor representa as restrições geométricas, como pontos finais e vetores tangentes, e por isso G é chamado vetor geométrico. As componentes que se referem à coordenada x são indicadas por G_x , e, da mesma forma G_y e G_z . Assim, tem-se :

$$Q(t) = [x(t) \ y(t) \ z(t)] = [t^3 \ t^2 \ t \ 1] \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \\ m_{41} & m_{42} & m_{43} & m_{44} \end{bmatrix} \begin{bmatrix} G_1 \\ G_2 \\ G_3 \\ G_4 \end{bmatrix} \quad (5.7)$$

Multiplicando $x(t) = TMG_x$, tem-se :

$$\begin{aligned} x(t) = & (t^3 m_{11} + t^2 m_{21} + t m_{31} + m_{41}) g_{1x} + \\ & (t^3 m_{12} + t^2 m_{22} + t m_{32} + m_{42}) g_{2x} + \\ & (t^3 m_{13} + t^2 m_{23} + t m_{33} + m_{43}) g_{3x} + \\ & (t^3 m_{14} + t^2 m_{24} + t m_{34} + m_{44}) g_{4x} \end{aligned}$$

As funções resultantes do produto $B = TM$ são conhecidas como funções base, funções peso, funções de mistura ou funções de concordância ("blending functions").

5.2.2 Curvas e Superfícies de Hermite

Sejam P_1 e P_2 os pontos extremos de um segmento, e R_1 e R_2 os vetores tangente nesses pontos. Para o cálculo da matriz base de Hermite M_h , que relaciona o vetor geométrico de Hermite G_h com os coeficientes polinomiais, deve-se escrever 4 equações, uma para cada restrição, em função dos coeficientes desconhecidos, e resolvê-las.

Seja G_{h_x} a componente x do vetor geométrico :

$$\begin{bmatrix} P_1 \\ P_2 \\ R_1 \\ R_2 \end{bmatrix}_x$$

e reescrevendo $x(t)$ das equações 5.1 e 5.7, tem-se :

$$x(t) = a_x t^3 + b_x t^2 + c_x t + d_x = TC_x = TM_h G_{h_x} = [t^3 \ t^2 \ t \ 1] M_h G_{h_x} \quad (5.8)$$

Calculando-se $x(0)$ e $x(1)$ por substituição direta em 5.8, tem-se :

$$\begin{aligned} x(0) &= [0 \ 0 \ 0 \ 1] M_h G_{h_x} \\ x(1) &= [1 \ 1 \ 1 \ 1] M_h G_{h_x} \end{aligned} \quad (5.9)$$

Diferenciando a equação 5.8 obtém-se $[3t^2 \ 2t \ 1 \ 0] M_h G_{h_x}$. Portanto, as restrições aos vetores tangente podem ser escritas como :

$$\begin{aligned} x'(0) &= [0 \ 0 \ 1 \ 0] M_h G_{h_x} \\ x'(1) &= [3 \ 2 \ 1 \ 0] M_h G_{h_x} \end{aligned} \quad (5.10)$$

As equações 5.9 e 5.10 podem ser colocadas em notação matricial da forma :

$$\begin{bmatrix} P_1 \\ P_2 \\ R_1 \\ R_2 \end{bmatrix} = G_{h_x} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 3 & 2 & 1 & 0 \end{bmatrix} M_h G_{h_x} \quad (5.11)$$

Para que a equação 5.11, assim como as respectivas equações para y e z , sejam satisfeitas, M_h deve ser a inversa da matriz acima :

$$\begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 3 & 2 & 1 & 0 \end{bmatrix}^{-1} = \begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

As superfícies polinomiais paramétricas bivariadas definem as coordenadas dos pontos que pertencem a essa superfície, usando 3 polinômios bivariados, para as coordenadas x, y e z . Assim como para curvas, polinômios de vários graus podem ser usados, mas um dos casos mais comuns é o de polinômios cúbicos. Superfícies de polinômios cúbicos são chamadas *superfícies bicúbicas*.

As superfícies paramétricas bicúbicas são generalizações das curvas cúbicas. Seja a equação $Q(s) = SMG$, onde G , o vetor geométrico, é uma constante. Se for tomado $G = G(t)$, função do parâmetro t , tem-se :

$$Q(s, t) = SMG(t) = SM \begin{bmatrix} G_1(t) \\ G_2(t) \\ G_3(t) \\ G_4(t) \end{bmatrix} \quad (5.12)$$

Para um dado t' fixado, $Q(s, t')$ é uma curva, uma vez que $G(t')$ é constante. Se t variar, o conjunto de curvas resultante é dito ser uma superfície paramétrica bicúbica. Cada $G_i(t)$ é uma curva cúbica, e é representada como :

$$G_i(t) = TM\mathcal{G}_i \quad (5.13)$$

onde

$$\mathcal{G}_i = [g_{i1} \ g_{i2} \ g_{i3} \ g_{i4}]^t \quad (5.14)$$

Reescrevendo a equação 5.13, usando a identidade $(A \ B \ C)^t = C^t B^t A^t$, o resultado é $G_i(t) = \mathcal{G}_i^t M^t T^t = [g_{i1} \ g_{i2} \ g_{i3} \ g_{i4}] M^t T^t$. Substituindo na equação 5.12, para cada G_i ,

tem-se :

$$Q(s, t) = SM \begin{bmatrix} g_{11} & g_{12} & g_{13} & g_{14} \\ g_{21} & g_{22} & g_{23} & g_{24} \\ g_{31} & g_{32} & g_{33} & g_{34} \\ g_{41} & g_{42} & g_{43} & g_{44} \end{bmatrix} M^t T^t \quad (5.15)$$

ou

$$Q(s, t) = SMGM^t T^t, \quad 0 \leq s, t \leq 1$$

ou ainda

$$x(s, t) = SMG_x M^t T^t$$

$$y(s, t) = SMG_y M^t T^t$$

$$z(s, t) = SMG_z M^t T^t$$

As *superfícies de Hermite* são definidas por uma matriz geométrica 4×4 G_h . Seja $x(s) = SM_h G_{h_x}$. Reescrevendo G_{h_x} como função do parâmetro t , tem-se :

$$x(s, t) = SM_h G_{h_x}(t) = SM_h \begin{bmatrix} P_1(t) \\ P_2(t) \\ R_1(t) \\ R_2(t) \end{bmatrix}_x \quad (5.16)$$

As funções $P_{1_x}(t)$ e $P_{2_x}(t)$ definem a componente x dos pontos extremos para a curva de parâmetro s . Analogamente, $R_{1_x}(t)$ e $R_{2_x}(t)$ definem a componente nos vetores tangentes. Para cada valor de t , existem 2 pontos extremos e dois vetores tangentes. Um "patch" de superfície é, essencialmente, uma interpolação cúbica entre $P_1(t) = Q(0, t)$ e $P_2(t) = Q(1, t)$ ou, alternadamente, $Q(s, 0)$ e $Q(s, 1)$.

No caso especial em que $Q(0, t)$, $Q(1, t)$, $Q(s, 0)$, $Q(s, 1)$ são linhas retas, o resultado é uma superfície regradada ("ruled surface").

Sejam $P_1(t)$, $P_2(t)$, $R_1(t)$, $R_2(t)$ representados na forma de Hermite :

$$P_{1_x}(t) = TM_h \begin{bmatrix} g_{11} \\ g_{12} \\ g_{13} \\ g_{14} \end{bmatrix}_x$$

$$P_{2_x}(t) = TM_h \begin{bmatrix} g_{21} \\ g_{22} \\ g_{23} \\ g_{24} \end{bmatrix}_x$$

$$R_{1_x}(t) = TM_h \begin{bmatrix} g_{31} \\ g_{32} \\ g_{33} \\ g_{34} \end{bmatrix}_x$$

$$R_{2_x}(t) = TM_h \begin{bmatrix} g_{41} \\ g_{42} \\ g_{43} \\ g_{44} \end{bmatrix}_x$$

que podem ser reescritas como

$$[P_1(t) P_2(t) R_1(t) R_2(t)]_x = TM_h \mathcal{G}_{h_x}^t \quad (5.17)$$

onde

$$\mathcal{G}_{h_x} = \begin{bmatrix} g_{11} & g_{12} & g_{13} & g_{14} \\ g_{21} & g_{22} & g_{23} & g_{24} \\ g_{31} & g_{32} & g_{33} & g_{34} \\ g_{41} & g_{42} & g_{43} & g_{44} \end{bmatrix}_x$$

Transpondo a equação 5.17, obtém-se :

$$\begin{bmatrix} P_{1_x}(t) \\ P_{2_x}(t) \\ R_{1_x}(t) \\ R_{2_x}(t) \end{bmatrix} = \begin{bmatrix} g_{11} & g_{12} & g_{13} & g_{14} \\ g_{21} & g_{22} & g_{23} & g_{24} \\ g_{31} & g_{32} & g_{33} & g_{34} \\ g_{41} & g_{42} & g_{43} & g_{44} \end{bmatrix}_x M_h^t T^t = \mathcal{G}_{h_x} M_h^t T^t \quad (5.18)$$

Substituindo a eq. 5.18 na equação 5.16, tem-se :

$$x(s, t) = SM_h \mathcal{G}_{h_x} M_h^t T^t$$

e, analogamente, para $y(s, t)$ e $z(s, t)$:

$$y(s, t) = SM_h \mathcal{G}_{h_y} M_h^t T^t$$

$$z(s, t) = SM_h \mathcal{G}_{h_z} M_h^t T^t$$

As 3 matrizes $\mathcal{G}_{h_x}, \mathcal{G}_{h_y}, \mathcal{G}_{h_z}$ cumprem o mesmo papel que a matriz G_h para curvas. O elemento g_{11_x} é $x(0, 0)$, uma vez que é o ponto inicial de $P_{1_x}(t)$, que por sua vez é o ponto inicial para $x(s, 0)$. Similarmente, g_{12_x} é $x(0, 1)$, pois é o ponto final de $P_{1_x}(t)$ que, por sua vez, é o ponto inicial de $x(s, 1)$. Além disso, g_{13_x} é $\frac{\partial x}{\partial t}(0, 0)$, uma vez que é o primeiro vetor tangente de $P_{1_x}(t)$. E g_{33_x} é $\frac{\partial^2 x}{\partial s \partial t}(0, 0)$, pois é o primeiro vetor tangente de $R_{1_x}(t)$ que, por sua vez, é a primeira inclinação de $x(s, 0)$.

Essas interpretações permitem que se escreva $\mathcal{G}_{h_x}$ como :

$$\mathcal{G}_{h_x} = \begin{bmatrix} x(0, 0) & x(0, 1) & \frac{\partial x}{\partial t}(0, 0) & \frac{\partial x}{\partial t}(0, 1) \\ x(1, 0) & x(1, 1) & \frac{\partial x}{\partial t}(1, 0) & \frac{\partial x}{\partial t}(1, 1) \\ \frac{\partial x}{\partial s}(0, 0) & \frac{\partial x}{\partial s}(0, 1) & \frac{\partial^2 x}{\partial s \partial t}(0, 0) & \frac{\partial^2 x}{\partial s \partial t}(0, 1) \\ \frac{\partial x}{\partial s}(1, 0) & \frac{\partial x}{\partial s}(1, 1) & \frac{\partial^2 x}{\partial s \partial t}(1, 0) & \frac{\partial^2 x}{\partial s \partial t}(1, 1) \end{bmatrix}$$

5.2.3 Superfícies de Coons

As superfícies de Coons são superfícies criadas a partir da interpolação de 4 curvas. Seja $P(u, v)$ uma função bivariada tal que, para s ou t constante, $P(u, v)$ é reduzida a uma curva espacial. Um "patch" de superfície é construído a partir dos lados $P(u, 0), P(u, 1), P(0, v), P(1, v)$, pela equação :

$$\begin{aligned} S(u, v) = & P(u, 0)(1 - v) + P(u, 1) + P(0, v)(1 - u) + P(1, v)u - \\ & P(0, 0)(1 - u)(1 - v) - P(0, 1)(1 - u)v - P(1, 0)u(1 - v) - \\ & P(1, 1)uv \end{aligned} \quad (5.19)$$

Os últimos 4 termos são necessários para que não se conte os cantos, resultantes da intersecção entre as curvas, duas vezes. Dessa forma, tem-se :

$$S(u, 0) = P(u, 0)$$

Será introduzida uma forma mais compacta para essas equações. Sejam :

$$\begin{aligned} P_1(v) &= \begin{bmatrix} P(0, v) \\ P(1, v) \end{bmatrix} \\ P_2(u) &= \begin{bmatrix} P(u, 0) \\ P(u, 1) \end{bmatrix} \\ b(u) &= \begin{bmatrix} 1 - u \\ u \end{bmatrix} \end{aligned} \quad (5.20)$$

A equação 5.19 pode ser reescrita como :

$$S(u, v) = b^t(u)P_1(v) + P_2^t(u)b(v) - b^t(u)Mb(v) \quad (5.21)$$

onde $M_{i,j} = P(i, j)$. As funções base, definidas dessa maneira, são lineares. Uma maneira mais geral que permite tomar outras funções é :

$$b(u) = \begin{bmatrix} b_0(u) \\ b_1(u) \end{bmatrix}$$

onde

$$b_i(u) = \delta_{iu}$$

exemplo 5.1 *Sejam*

$$\begin{aligned} b_0(u) &= \cos^2\left(\frac{\pi}{2}u\right) \\ b_1(u) &= \sin^2\left(\frac{\pi}{2}u\right) \end{aligned}$$

as funções base. Sejam ainda as curvas limitantes segmentos de retas com cantos ligando os pontos $P_1 = (0, 0, 0)$, $P_2 = (0, 1, 0)$, $P_3 = (1, 0, 0)$, $P_4 = (1, 1, 1)$. Pode-se obter as equações :

$$\begin{aligned} P(u, 0) &= uP_3, \quad P(u, 1) = uP_4 + (1 - u)P_2 \\ P(0, v) &= vP_2, \quad P(1, v) = vP_4 + (1 - v)P_3 \end{aligned}$$

Assim, pelas equações 5.20, tem-se :

$$P_1(v) = \begin{bmatrix} vP_2 \\ vP_4 + (1 - v)P_3 \end{bmatrix} \quad P_2(u) = \begin{bmatrix} uP_3 \\ uP_4 + (1 - u)P_2 \end{bmatrix}$$

e, pela equação 5.21

$$\begin{aligned}
 S(u, v) &= [\cos^2(\frac{\pi}{2}u) \sin^2(\frac{\pi}{2}u)] \begin{bmatrix} vP_2 \\ vP_4 + (1-v)P_3 \end{bmatrix} + \\
 &\quad [uP_3 \ uP_4 + (1-u)P_2] \begin{bmatrix} \cos^2(\frac{\pi}{2}u) \\ \sin^2(\frac{\pi}{2}u) \end{bmatrix} - \\
 &\quad [\cos^2(\frac{\pi}{2}u) \sin^2(\frac{\pi}{2}u)] \begin{bmatrix} P_1 & P_2 \\ P_3 & P_4 \end{bmatrix} \begin{bmatrix} \cos^2(\frac{\pi}{2}u) \\ \sin^2(\frac{\pi}{2}u) \end{bmatrix} = \\
 &= vP_2 \cos^2(\frac{\pi}{2}u) + [vP_4 + (1-v)P_3] \sin^2(\frac{\pi}{2}u) \\
 &\quad + uP_3 \cos^2(\frac{\pi}{2}v) + [uP_4 + (1-u)P_2] \sin^2(\frac{\pi}{2}v) \\
 &\quad - [P_1 \cos^2(\frac{\pi}{2}u) + P_3 \sin^2(\frac{\pi}{2}u)] \cos^2(\frac{\pi}{2}v) \\
 &\quad - [P_2 \cos^2(\frac{\pi}{2}u) + P_4 \sin^2(\frac{\pi}{2}u)] \sin^2(\frac{\pi}{2}v)
 \end{aligned}$$

Substituindo os valores de P_1, P_2, P_3, P_4 e fazendo os cálculos, pode-se mostrar que :

$$x(u, v) = u$$

$$y(u, v) = v$$

$$z(u, v) = u \sin^2(\frac{\pi}{2}v) + v \sin^2(\frac{\pi}{2}u) - \sin^2(\frac{\pi}{2}u) \sin^2(\frac{\pi}{2}v)$$

Para se garantir a continuidade do gradiente, deve ser usado na interpolação não apenas $P(u, v)$, mas também $P'(u, v)$. Isso é feito acrescentando-se duas componentes extras nos vetores das equações 5.20, representando as derivadas das funções e as funções base para essas derivadas. Também a matriz M deve ser expandida para 4×4 . Será

utilizada a notação :

$$g^u = \frac{\partial g(u, v)}{\partial u}$$

$$g^{uv} = \frac{\partial g(u, v)}{\partial u \partial v}$$

Define-se :

$$P_1(v) = \begin{bmatrix} P(0, v) \\ P(1, v) \\ P^u(0, v) \\ P^u(1, v) \end{bmatrix}$$

$$P_2(u) = \begin{bmatrix} P(u, 0) \\ P(u, 1) \\ P^v(u, 0) \\ P^v(u, 1) \end{bmatrix} \quad (5.22)$$

$$b(u) = \begin{bmatrix} b_{00}(u) \\ b_{01}(u) \\ b_{10}(u) \\ b_{11}(u) \end{bmatrix}$$

assumindo-se que

$$b_{ij}^k(u) = \delta_{kj} \delta_{iu}$$

Se $k = 0$, não foi feita uma diferenciação, $k = 1$ foi feita uma diferenciação. Assim :

$$b_{00}(0) = 1$$

$$b_{01}(1) = b_{00}^u(0) = b_{00}^u(1) = 0$$

Em outras palavras, $b_{0i}(u)$ é a função base para as curvas, enquanto que $b_{1,i}(u)$ é a função base para as derivadas. δ_{kj} é um para os casos : $k = j = 0$, correspondendo às duas primeiras componentes do vetor base, e $k = j = 1$ para as derivadas. A matriz M é dada por :

$$M = \begin{bmatrix} P(0, 0) & P(0, 1) & P^v(0, 0) & P^v(0, 1) \\ P(1, 0) & P(1, 1) & P^v(1, 0) & P^v(1, 1) \\ P^u(0, 0) & P^u(0, 1) & P^{uv}(0, 0) & P^{uv}(0, 1) \\ P^u(1, 0) & P^u(1, 1) & P^{uv}(1, 0) & P^{uv}(1, 1) \end{bmatrix}$$

Com essa notação, a eq. 5.21 continua válida.

5.2.4 Superfícies de Bézier

Esse método de construção de curvas e superfícies foi desenvolvido pelo engenheiro francês P. Bézier para utilização no projeto de carros da firma Renault. Uma superfície de Bézier é definida como :

$$P(u, v) = \sum_{i=0}^n \sum_{j=0}^m p_{ij} B_{i,n}(u) B_{j,m}(v)$$

onde as funções base $B_{i,n}(u)$ são definidas como :

$$B_{i,n}(u) = C(n, i) u^i (1 - u)^{n-i}$$

e

$$C(n, i) = \frac{n!}{i!(n-i)!}$$

As superfícies de Bézier gozam de propriedades como independência do sistema de coordenadas, controle global e outras.

5.2.5 Superfícies B-Spline

Uma superfície B-Spline é definida de forma similar à de Bézier :

$$P(u, v) = \sum_{i=0}^n \sum_{j=0}^m p_{ij} N_{i,n}(u) N_{j,m}(v)$$

onde a função base é a mesma definida no capítulo 3.

5.2.6 B-Splines Racionais

Um dos trabalhos mais conhecidos sobre B-Splines racionais é [90]. Segundo ele, as duas formas de representação de curvas e superfícies mais usadas em modelagem geométrica são a implícita e a paramétrica. A implícita possui a vantagem de representar precisa e concisamente círculos, cônicas e superfícies quádras como cilindros, esferas e cones. A principal desvantagem da forma implícita é a incapacidade de representar curvas e superfícies de formas livres ("free form"). As B-Splines, que fazem parte do conjunto das paramétricas, conseguem representar e manipular as formas livres, mas não círculos,

cônicas e quádricas. Segundo [90], as B-Splines racionais não uniformes ("non-uniform, rational B-Splines", NURBS) sintetizam, em uma só forma de representação, as vantagens das duas formas, paramétrica e implícita.

Para a definição da B-Spline racional, são usadas coordenadas homogêneas [88]. Se $P = (x, y, z)$ for um ponto do espaço, a representação correspondente usando coordenadas homogêneas é dada por :

$$P^h = (hx, hy, hz, h), h > 0$$

Analogamente, pode-se definir uma curva B-Spline usando coordenadas através da equação vetorial :

$$C^h(t) = \sum_{i=1}^n p_i^h N_{i,k}(t)$$

Para $a \leq t \leq b$, $C^h(t)$ forma um conjunto de pontos no espaço 4D. $C(t)$ denota a curva 3D associada à $C^h(t)$, e é obtida dividindo-se as coordenadas homogêneas de cada ponto pela sua coordenada homogênea, e projetando-se os pontos resultantes no espaço 3D original. Uma curva B-Spline racional é definida por :

$$C(t) = \frac{\sum_{i=1}^n h_i p_i N_{i,k}(t)}{\sum_{i=1}^n h_i N_{i,k}(t)}$$

Analogamente, pode-se definir uma superfície B-Spline racional como :

$$S(s, t) = \frac{\sum_{i=1}^n \sum_{j=1}^m h_{ij} p_{ij} N_{i,k}(s) N_{j,l}(t)}{\sum_{i=1}^n \sum_{j=1}^m h_{ij} N_{i,k}(s) N_{j,l}(t)}$$

As B-Splines racionais possuem propriedades comuns às B-Splines, como invariância a transformações lineares, está contida no fecho convexo e outras. Além disso, da mesma forma são definidas curvas de Bézier racionais. Maiores detalhes quanto às propriedades dessas curvas e superfícies são encontrados em [90, 16, 26].

5.3 Geração de Superfícies

Existem basicamente duas formas de traçar curvas e superfícies em dispositivos discretos : avaliação iterativa das equações e subdivisão.

5.3.1 Avaliação Iterativa das Equações

Uma vez que as equações que descrevem as superfícies geralmente envolvem muitos cálculos, é importante desenvolver algoritmos eficientes que gerem os pontos ao longo das curvas sobre a superfície, definidas pelas direções de parametrização.

Um método para reduzir o número de operações aritméticas na equação polinomial é conhecido como regra de Horner [13], que especifica uma ordem de fatoração dos termos de um polinômio. Seja o polinômio cúbico :

$$f(u) = a_0u^3 + a_1u^2 + a_2u + a_3$$

que, sendo avaliado segundo a ordem :

$$f(u) = [(a_0u + a_1)u + a_2]u + a_3$$

necessita de 3 multiplicações e 3 adições. Embora essa técnica de fatoração reduza o número de operações aritméticas, ela ainda envolve multiplicações em ponto flutuante.

Um método mais eficiente de avaliar equações polinomiais é gerar recursivamente cada valor sucessivo da função, incrementando-se o valor anterior :

$$f_{i+1} = f_i + \Delta f_i \quad (5.23)$$

onde Δf_i é chamado "forward difference". A função f_i é avaliada em u_i , e f_{i+1} é avaliada em $u_{i+1} = u_i + \delta$, onde δ é o passo do parâmetro u . Por exemplo, a equação linear $f(u) = a_0u + a_1$ pode ser avaliada usando a eq. 5.23 e $\Delta f = a_0\delta$. Em polinômios de maior ordem, Δf é uma função polinomial de u . Para uma curva cúbica, tem-se :

$$\Delta f_{i+1} = \Delta f_i + \Delta^2 f_i$$

Essas equações são usadas para obter os pontos da curva para $u \in [0, 1]$. Os valores iniciais para $i = 0$ e $u = 0$ são :

$$f_0 = a_3$$

$$\Delta f_0 = a_0\delta^3 + a_1\delta^2 + a_2\delta$$

$$\Delta^2 f_0 = 6a_0\delta^3 + 2a_1\delta^2$$

Para superfícies, o cálculo incremental é aplicado aos parâmetros u e v .

5.3.2 Técnicas de Subdivisão

A técnica de subdivisão para superfícies é uma extensão da análoga para curvas. A subdivisão pára adaptativamente quando o segmento de curva em questão pode ser aproximado por uma reta. Os detalhes variam para cada tipo de curva, uma vez que o processo de subdivisão e o teste de aproximação são diferentes para cada tipo.

Para ilustração do processo, sejam as curvas de Bézier. Uma propriedade importante delas é que, se uma curva de Bézier de grau n for dividida em seu ponto médio, as duas curvas obtidas são também de Bézier de grau n [42]. Para determinar o polígono de controle de cada uma das curvas resultantes, devem ser feitas 3 subdivisões sucessivas pelo ponto médio de cada lado do polígono original. Cada um dos pontos dessa subdivisão é um vértice dos novos polígonos de controle (fig. 5.1).

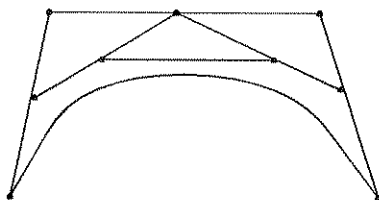


Figura 5.1: Bézier e polígono de controle

Como a curva de Bézier está contida no fecho convexo do polígono de controle, então cada polígono resultante está mais próximo da curva de Bézier original. O algoritmo de geração por subdivisão funciona obtendo sucessivamente polígonos de controle mais próximos da curva original, até que ela possa ser substituída pelo próprio polígono de controle, segundo um erro de aproximação. A geração de superfícies é uma extensão do método para curvas [40].

5.4 Visualização

Muitas aplicações de computação gráfica envolvem a visualização de objetos 3D. Modelagem de moléculas, CAD, animação e simulação de vôo são exemplos de visualização de objetos e cenas espaciais. Essas diferem de imagens 2D não apenas em relação à terceira dimensão, mas também quanto à idéia de criação de realismo dos objetos. Produzir uma

imagem realística de um objeto 3D em um dispositivo bidimensional apresenta problemas como noção de profundidade, identificação de partes escondidas, tratamento de iluminação, cor, sombras e textura.

Segundo [43], existem basicamente 2 problemas a serem resolvidos para a visualização de superfícies, um relacionado à geometria, e o outro ao significado dos dados a serem visualizados. O primeiro tem como objetivo determinar as áreas visíveis de cada objeto na imagem final. O segundo corresponde à iluminação e cor dos elementos da imagem. Três tarefas gerais estão envolvidas na resolução desses problemas : geração dos objetos, determinação da iluminação e da visibilidade. O problema de geração de superfícies foi discutido na secção anterior.

Um modelo de iluminação é usado no cálculo da intensidade de luz que deve ser observado em cada ponto de uma superfície. Esse cálculo de intensidade é baseado em propriedades ópticas da superfície, e sua posição em relação às fontes de luz.

A questão de visibilidade é extremamente importante na geração de imagens realísticas, estando ligadas à identificação e remoção de partes escondidas da imagem em relação a um observador.

As diferentes arquiteturas dos sistemas de visualização diferem principalmente em relação à ordem em que as operações de geração, iluminação e visibilidade são efetuadas, e na maneira pela qual elas se relacionam. Ainda segundo [43], pode-se identificar 3 arquiteturas básicas de sistemas de visualização :

- (1) Geração - Iluminação - Visibilidade : sistemas baseados no "scanline".
- (2) Geração - Visibilidade - Iluminação : sistemas baseados no "ray-tracing"
- (3) Iluminação - Geração - Visibilidade : sistemas baseados no radiossidade.

Capítulo 6

Implementação e Resultados Obtidos

6.1 Introdução

Neste capítulo são apresentados detalhes da implementação de um processo de reconstrução 3D por ajuste de superfícies de Coons, utilizando técnicas descritas nos capítulos anteriores. Os detalhes dizem respeito, particularmente, às estruturas de dados, ambientes e bibliotecas utilizadas. De maneira geral, pode-se dividir os detalhes de implementação quanto às etapas de reconstrução 3D, apresentadas nos capítulos 3, 4 e 5 em :

(1) Segmentação e representação de contornos : foi criado um programa de segmentação e representação de contornos por B-Splines, além de outros que auxiliam a análise e visualização dos resultados dessa etapa.

(2) Casamento de segmentos entre contornos : existem 2 programas centrais para esta etapa, um que realiza o casamento entre dois contornos, e outro que organiza o casamento entre n contornos, usando o primeiro programa. Também foram criadas ferramentas que auxiliam a análise e visualização dos resultados.

(3) Ajuste de superfícies : a etapa de geração e visualização da superfície final foi resolvida através da integração de dois ambientes : o de modelagem de curvas e superfícies "model", criado por Luiz Gonzaga Silveira Jr. e Ulisses de Mendonça Braga durante a disciplina de "Introdução à Computação Gráfica", ministrada pelos professores Wu Shin Ting e Léo Pini Magalhães, junto ao programa de pós-graduação da FEE-UNICAMP;

e o programa de visualização de superfícies "scanline", do projeto Prosim [84]. Para a integração dos dados resultantes da etapa de casamento de segmentos, foi desenvolvido um programa que, a partir de um conjunto de contornos casados, gera um arquivo de pontos de controle no formato de entrada do "model". A saída do "model" foi modificada para o formato de entrada do "scanline". O mesmo arquivo de entrada para o "scanline" é usado por um programa de visualização de estruturas do tipo "wireframe" que foi desenvolvido pela profa. Wu Shin Ting.

Além desses, o ambiente "Khoros" foi utilizado para o pré-processamento das imagens, além da utilização de sua interface de programação visual, o "Cantata", e rotinas da biblioteca "PBMPLUS". Os programas foram implementados na linguagem C. Maiores detalhes desses ambientes e bibliotecas, além de sua integração num processo de reconstrução 3D, são apresentados na secção 6.3. A secção 6.2 apresenta aspectos das principais estruturas de dados criadas nos diversos programas. A secção 6.4 apresenta e discute os resultados obtidos nas 3 etapas da reconstrução 3D.

6.2 Estruturas de Dados

Existem 3 estruturas principais que são manipuladas pelos programas envolvidos nas etapas de segmentação e casamento de contornos : vetores de pontos, lista circular de segmentos e árvore do espaço de estados do A^* . Além dessas, todas as outras são estruturas padrão, com implementação baseada nas referências [21] e [44].

Primeiro, tem-se a definição do registro *Point*, que armazena as coordenadas de um ponto da imagem. São tomadas coordenadas inteiras, referenciadas por $p.x$ e $p.y$, se p for do tipo *Point*.

Os contornos são armazenados em vetores de pontos, em uma declaração do tipo :

```
Point *contour;
```

permitindo que contornos abertos ou fechados sejam facilmente representados. A principal vantagem do uso dessa estrutura reside em sua simplicidade. Os principais problemas

encontrados se referem a operações mais complexas que porventura devam ser realizadas nos contornos, exigindo funções e procedimentos mais elaborados. Optou-se por representar contornos fechados fazendo o último elemento do vetor ser igual ao primeiro. Caso contrário, o contorno é aberto.

A lista de segmentos foi criada devido à necessidade de representar o contorno segmentado, o polígono de controle de cada segmento, seus vetores e curvaturas associados e os atributos de cada segmento. Essa lista foi definida como circular, duplamente ligada [44]. Sua declaração é dada por :

```
typedef struct seglist *Segptr;

typedef struct seglist {
int np;
Point *p_i;
Vector *v_i;
double *c_i;
double attributes[N_ATTRIBUTES];
Segptr prior;
Segptr next;
} Segnode;
```

A variável np da declaração acima indica a cardinalidade do segmento. Os vetores p_i, v_i e c_i armazenam os pontos do polígono de controle, os vetores e curvaturas associadas, respectivamente. Essa estrutura pode ser melhorada em relação à memória através da eliminação do vetor v_i, exigindo que seu cálculo seja feito a partir de p_i quando necessário. Os pontos extremos de p_i são de segmentação do contorno, e o segundo ponto de segmentação de um segmento coincide com o primeiro do segmento consecutivo. Dessa maneira, curvas segmentadas abertas e fechadas são representadas testando-se o segundo ponto de segmentação do último segmento e o primeiro do segmento inicial. O número de atributos de um segmento pode ser alterado através da variável N_ATTRIBUTES, permitindo que essa estrutura seja usada com outros descritores, desde que as funções sejam devidamente modificadas.

Finalmente, o grafo de espaço de estados do A^* , pelo funcionamento do algoritmo implementado, pode ser definido como uma árvore n-ária, uma vez que não existem ciclos. Por sua vez, a definição dessa árvore fica simplificada se sua representação for feita por uma árvore binária. A representação de uma árvore n-ária por uma binária é feita através da diferenciação dos dois possíveis filhos de um nó, um sendo filho, e o outro irmão (fig. 6.1).

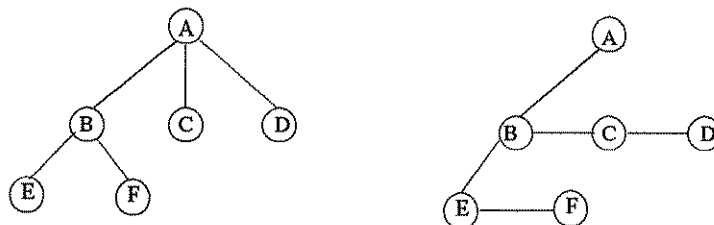


Figura 6.1: Representação do espaço de estados por uma árvore binária.

Duas observações importantes influenciaram na definição da árvore : ela deve permitir manipulações intensas, com muitas operações de inserção, retirada e atualização dos nós, e também permitir a identificação do caminho de um nó meta ao inicial. Com base nessas observações, e nas características do funcionamento do A^* , foi definida a estrutura :

```
typedef struct seggraph *Sgraphptr;

typedef struct seggraph {
    int bi, cj;
    double g, h, f;
    Sgraphptr lbrother; /* left brother */
    Sgraphptr rbrother; /* right brother */
    Sgraphptr father;
    Sgraphptr sons;
} Sgraphnode;
```

Dessa maneira, o caminho de uma solução é facilmente obtido voltando pelos ponteiros *father* de cada nó. Uma configuração possível da árvore é apresentada na figura 6.2 (obs: esta e as próximas figuras estão colocadas no final do capítulo).

A principal desvantagem dessa representação reside no fato de que, infelizmente,

é raro o caso em que algoritmos de manipulação de grafos e árvores mais complexas sejam simples de implementar corretamente, e mais raro ainda de se garantir a exatidão da implementação desses algoritmos. Por outro lado, a grande vantagem está na simplificação que essa estrutura permite no processo pelo qual o A^* elimina os ciclos, atualizando caminhos e propagando melhorias recursivamente.

6.3 Etapas da Reconstrução 3D e Ambientes Utilizados

Dadas duas ou mais imagens, correspondendo às fatias do objeto, a primeira etapa a se cumprir é a segmentação do objeto em cada fatia. O resultado é a obtenção de uma imagem binária que, para os programas desenvolvidos, deve ser armazenada no formato PBM, que se caracteriza pela sua simplicidade de estrutura, e pode ser lido por diversos pacotes de visualização de imagens, como o XV, o Khoros, e outros.

O pré-processamento que leva à segmentação e transformação da imagem no formato PBM foi feito no Khoros, pacote aberto de processamento e visualização de imagens desenvolvido na Universidade do Novo México, EUA. Além das facilidades de processamento de imagens que o Khoros oferece, ele também permite a utilização de seu ambiente de programação visual baseado na conexão de módulos que representam programas ou conjuntos deles. Esse ambiente é conhecido como Cantata, e sua utilização é feita de duas maneiras : integração de programas ao próprio Khoros ou definição de "panes", que funcionam como módulos de interface entre o Cantata e os programas desenvolvidos. As diversas configurações que são ser programadas no Cantata são conhecidas como "workspaces".

Uma vez obtida a imagem binária no formato PBM, o próximo passo é a extração do contorno. Isso é feito por 2 programas, *pbmTORun*, que fornece uma representação intermediária através do código de corrida [68], e *runtocc*, que extrai os contornos e os representa pelo código da cadeia. Esses dois programas foram desenvolvidos pelo professor Roberto A. Lotufo -FEE - UNICAMP-, e utiliza rotinas do PBMPLUS, uma biblioteca de funções e programas para conversão de imagens.

O próximo passo é realizar a segmentação e representação dos contornos. Isso é feito pelo programa *bspliseg*, que toma um contorno no código da cadeia, e gera um arquivo contendo sua representação segmentada. Os arquivos de saída desse programa possuem

extensão *.sp*. Esse arquivo, além de representar os pontos de segmentação e de controle de cada segmento, também possui informações como curvatura e deslocamento de cada ponto, que são usadas na extração de atributos de cada segmento. A partir dos arquivos *.sp*, uma série de facilidades de visualização foi criada, como programas que geram os contornos B-Spline, aproximação poligonal pelos pontos de segmentação, de controle, marcação desses pontos e outros.

O casamento de segmentos é feito de duas maneiras : entre cada dois contornos ou, dado um conjunto de contornos, casamento entre todos eles. Cada contorno deve estar armazenado num arquivo *.sp*. O programa que faz o casamento entre dois contornos é o *segmatch*.

O programa *allmatch* toma um conjunto de dois ou mais arquivos contendo os contornos, identifica aquele com o menor número de segmentos e casa todos os outros em função deste. Para esse programa, cada arquivo contendo um contorno deve possuir um nome comum, o prefixo, e um número de identificação do contorno. Assim, um exemplo seria um conjunto de arquivos "fatia0.sp, fatia1.sp, ...". Esse programa permite que apenas um subconjunto desses contornos sejam casados, através da especificação de um índice inicial e outro final.

Os contornos casados devem ser armazenados num conjunto de arquivos de forma semelhante aos não casados. Esses arquivos casados são usados pelo programa *coons*, que os toma como entrada e gera um arquivo com extensão *.gdi*. Esse arquivo está no formato de entrada do modelador de superfícies *model*. Esse ambiente, além de superfícies de Coons, permite a modelagem de curvas de Hermite, Bézier, B-Spline, superfícies de Bézier, B-Spline, varredura e rotação.

Uma vez modelado, a superfície é armazenada num arquivo de extensão *.brp* que, através do programa *create_brp*, é transformado em outro arquivo, também de extensão *.brp*, mas do tipo registro. Esse segundo arquivo é usado pelo programa *scanline* [84] para a visualização da superfície final. O *scanline* utiliza um arquivo de entrada *.in*, onde são especificados os arquivos contendo as superfícies, além dos parâmetros de visualização.

6.4 Resultados Obtidos

6.4.1 Pré-processamento

Todo o pré-processamento necessário para a obtenção da imagem binária e conseqüente segmentação do objeto de interesse foi feito através do Khoros. Foram usadas imagens criadas no XFIG, imagens de letras de um cartaz digitalizado por um "scanner", e por um processo de tomografia computadorizada de raio-x. Todas as segmentações foram feitas por limiar ("threshold"). Em alguns casos, como nas imagens de tomografia, foram usados métodos de morfologia matemática durante o pré-processamento para, por exemplo, fechar buracos nos objetos.

6.4.2 Representação e Segmentação

A notação dos parâmetros dos programas usados nesta secção é a mesma do algoritmo 3.2. T_s indica o limiar de segmentação, T_c o limiar para pontos de controle, e α é o cosseno do ângulo usado no ajuste global, $-1 \leq \alpha \leq 1$, e o ângulo está mais próximo de 180 graus quando α estiver mais próximo de -1.

Nos 2 casos, das letras digitalizadas e da imagem de tomografia, os parâmetros do algoritmo foram os mesmos : $T_s = 0.39$, $T_c = 0.32$ e $\alpha = -0.65$. Apenas no caso das letras foi tomado $\alpha = -0.9$.

A primeira coluna da tabela 6.3 mostra as letras digitalizadas. A segunda mostra o contorno extraído de cada letra. A terceira coluna mostra uma aproximação poligonal, onde os vértices, marcados com quadradinhos pretos, são os pontos de segmentação calculados pelo algoritmo implementado. A quarta coluna mostra as letras representadas por B-Splines. Na quinta estão as aproximações poligonais geradas pelos polígonos de controle de cada letra. A coluna de nome "ncont" mostra o número de pontos de cada contorno. A coluna "nseg" mostra o número de segmentos de cada letra. Por fim, a última coluna mostra o número total de pontos necessários para nossa representação, somando os pontos de segmentação e os de controle.

A tabela 6.4 mostra um exemplo de aplicação em imagens médicas. Esse con-

torno possui 666 pontos. Ele foi segmentado por 21 pontos, e foram necessários 48 pontos de controle. Os pontos de segmentação são marcados com quadradinhos pretos na última imagem dessa tabela, enquanto que os de controle foram marcados com quadradinhos brancos. A taxa de compressão é de aproximadamente 10:1.

A figura 6.5 mostra a segmentação pelo método original descrito em [27]. A figura 6.6 mostra uma ampliação do resultado do método modificado. Finalmente, a figura 6.7 mostra uma ampliação do contorno representado, onde os quadradinhos pretos denotam os pontos de segmentação, e os brancos os de controle.

6.4.3 Casamento de Contornos

O primeiro exemplo, mostrado na figura 6.8, mostra o contorno de duas letras "c" que foram geradas manualmente no XFIG. Note-se que o número de segmentos é diferente. O primeiro possui 8 pontos de segmentação, e o segundo 13. Depois que o casamento foi feito, o segundo "c", que ficou com 8 pontos de segmentação, também é mostrado na tabela. O último quadro mostra as duas imagens sobrepostas, com um segmento de reta ligando os pontos de segmentação. A figura 6.9 mostra os dois contornos casados, lado a lado, com as retas indicando o casamento entre os pontos de segmentação. Neste exemplo, os fatores peso da função custo são iguais a 1.

O segundo exemplo ilustra a independência de escala. Na tabela 6.10, o segundo contorno foi obtido ampliando-se o primeiro. O primeiro contorno possui 8 pontos de segmentação, e o segundo 11. Também são mostrados o segundo contorno já casado, após as operações de junção, e o casamento, representado por segmentos de reta. Para este exemplo, o fator peso dos atributos que representam o vetor diretor é igual a 2, e os demais iguais a 1.

O terceiro exemplo mostra o processo de casamento em duas imagens obtidas por tomografia computadorizada de raio-x. A figura 6.11 mostra dois contornos segmentados de uma cabeça que pertencem a duas fatias consecutivas. O primeiro possui 16 pontos de segmentação, e o segundo 23. Os contornos casados são representados na figura 6.12. Neste exemplo, todos os fatores peso são iguais a 1.

6.4.4 Ajuste de Superfícies

A figura 6.13 mostra o ajuste de superfícies aos contornos casados da tabela 6.10. A figura 6.14 mostra o ajuste aos contornos da tabela 6.8. Nos dois casos, foi usado o modelo de iluminação de Phong.

A figura 6.15 mostra a estrutura "wireframe" da reconstrução de uma superfície a partir de dois contornos, um na forma de uma letra "b", e outro na forma de uma letra "D". Pode-se perceber os contornos intermediários gerados, mostrando a transformação de um contorno no outro. A figura 6.16 mostra a reconstrução 3D de um trecho do joelho de uma pessoa. Foram usadas 8 secções, obtidas a partir de tomografia computadorizada de raios-x. Para a segmentação das imagens, foi usado limiar para pele.

6.5 Análise dos Resultados e Principais Problemas Encontrados

A modificação proposta no algoritmo de representação por B-Splines melhorou consideravelmente o método de [27] nos dois aspectos que a motivaram, ou seja, diminuição do número de segmentos e obtenção de uma segmentação que reflita melhor a estrutura do contorno. As taxas de compressão para imagens experimentais ficaram por volta de 10:1, e são consideradas boas em relação aos métodos levantados na literatura. Testes feitos em imagens sintéticas apresentaram taxas melhores. Em geral, o método se mostrou bastante robusto, inclusive para contornos complexos. O principal problema está na identificação de um número excessivo de pontos de controle em cada segmento, o que faz decair a taxa de compressão. Trabalhos como [59] afirmam obter taxas de até 20:1 em imagens de impressões digitais. Além disso, a identificação dos limiares para a obtenção de resultados satisfatórios nem sempre é intuitiva e direta, variando muito, mesmo em contornos do mesmo tipo. Isso dificulta o desenvolvimento de um processo de segmentação de um conjunto de n contornos de forma automática, para a reconstrução 3D. Isso porque, tanto para o casamento de segmentos, como para o ajuste de superfícies, é interessante que o número de segmentos em cada contorno não varie muito. No caso do casamento, porque a largura da árvore do espaço de estados está diretamente ligada à diferença do número de segmentos entre

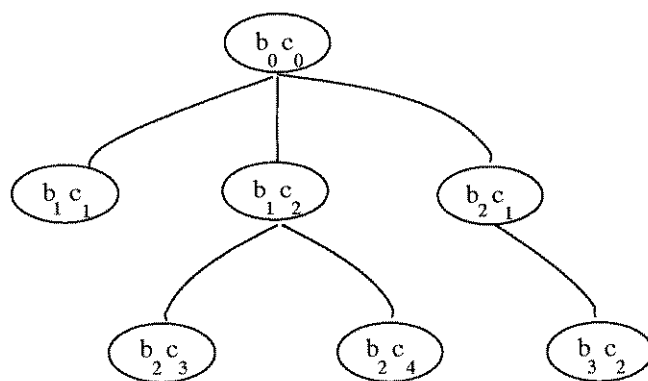
cada par de contornos. No caso do ajuste de superfícies porque, geralmente, o número de segmentos de um contorno está diretamente ligado a sua complexidade. Suponha que um contorno com muitos segmentos, provavelmente possuindo muitos detalhes, deva ser casado com outro com poucos segmentos. Como no final do processo de casamento ambos ficarão com o mesmo número de segmentos, igual ao do contorno mais simples, os detalhes do mais complexo poderão ser perdidos.

O método de casamento entre dois contornos apresentou resultados satisfatórios, principalmente se o número de segmentos entre os contornos não variar muito. A possibilidade de ajuste dos pesos na função de similaridade usada, definida pelo método dos mínimos quadrados, conferiu robustez ao método, permitindo inclusive sua aplicação para o caso em que os contornos eram transladados na imagem. Porém, como o casamento é realizado localmente, isto é, para cada par de contornos, problemas aparecem no caso de um número grande de contornos que variem razoavelmente entre si. Isso pode ser observado na fig. 6.16, seguindo-se o caminho de cada ponto do primeiro ao último contorno. Além disso, como foi dito, se um contorno for muito mais detalhado que o outro, e por isso possuir um número maior de segmentos, ele será penalizado, com parte de informação estrutural acabando perdida no processo. Uma prática que foi adotada durante os testes, e que acabou se mostrando útil em muitas situações, foi a de identificar dentre os contornos a serem casados, aquele que fosse mais detalhado. Esse contorno é então usado para guiar todos os casamentos. Isso é conseguido fazendo-se com que todos os outros contornos possuam um número maior de segmentos. O tempo de processamento, assim como as restrições de memória, foram considerados bons para o caso em que o número de segmentos não varia muito.

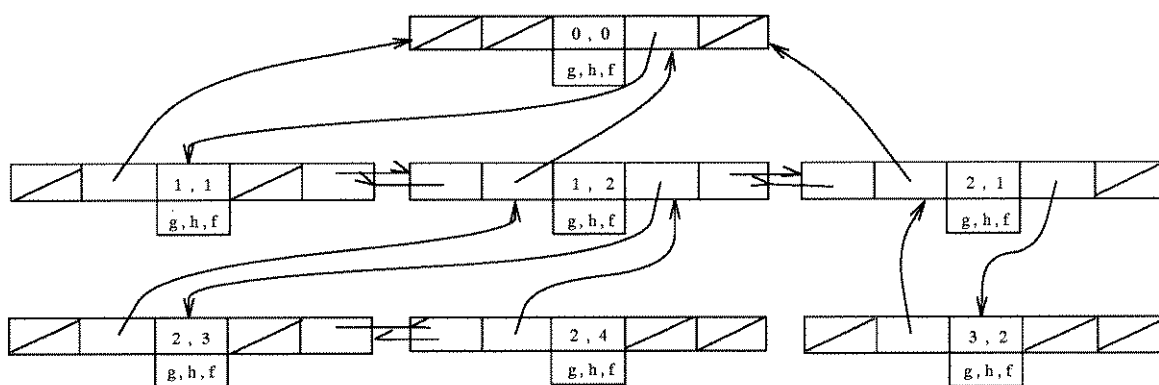
A principal dificuldade encontrada no ajuste de "patches" de Coons reside no fato de que, para esse tipo de superfícies, além dos 4 pontos que definem o "patch", mais 12 valores, correspondentes aos vetores tangente e torção, devem ser especificados para o controle da suavidade da superfície. A especificação desses valores não é trivial e a solução adotada foi definir todos eles como sendo iguais a zero. Existem métodos, como [52], que apresentam fórmulas heurísticas para a definição desses valores.

lbrother	father	bi , cj	sons	rbrother
		g , h , f		

(a)



(b)



(c)

Figura 6.2: (a) representação de um nó do espaço de estados; (b) e (c) exemplo de uma configuração do espaço de estados e sua estrutura correspondente.

letra	contorno	ap. pol. 1	ap. pol. 2	B-Spline	ncont	nseg	nrep
U					195	12	25
Z					203	12	25
I					86	4	9
C					202	17	32
A					133	7	23
M					246	14	39
P					130	9	18

Figura 6.3: Tabela de resultados com letras digitalizadas (legenda : ap. pol. 1 - aproximação poligonal pelos pontos de segmentação; ap. pol. 2 - aproximação poligonal por pontos de segmentação e de controle; ncont - número de pontos do contorno; nseg - número de pontos de segmentação; nrep - número total de pontos (segmentação + controle).

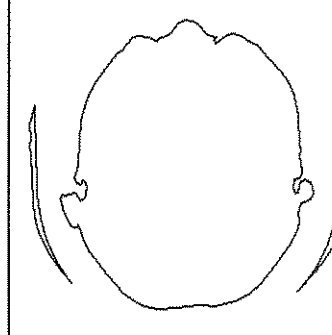
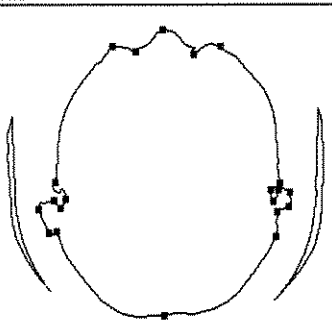
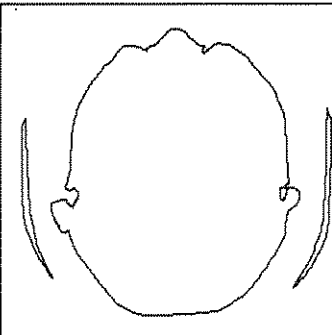
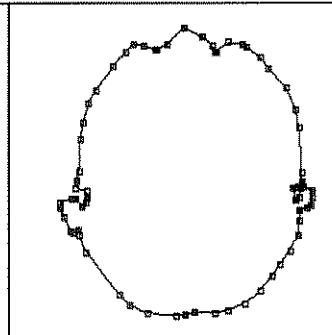
Original	Binária	Contorno
		
Segmentação	B-Spline	Pontos de Controle
		

Figura 6.4: Tabela de resultados com imagem de tomografia.

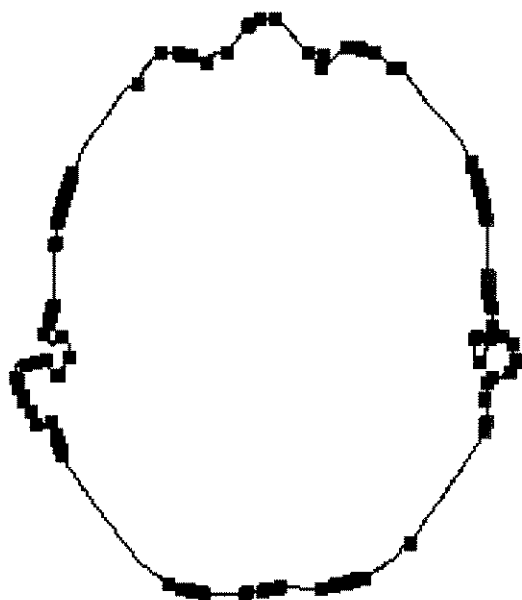


Figura 6.5: Contorno representado por B-Spline pelo método original.

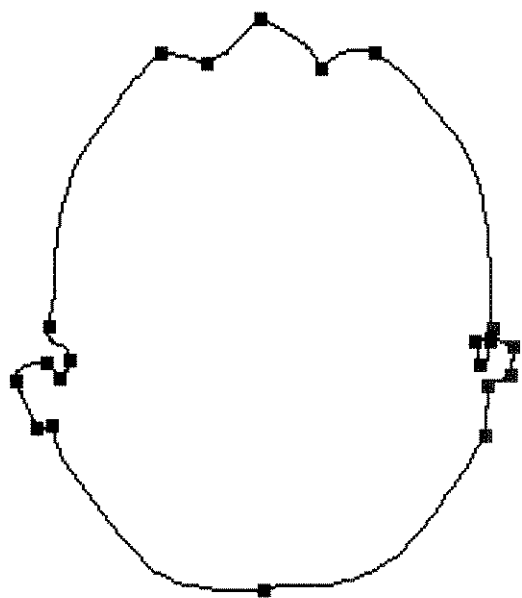


Figura 6.6: Contorno representado por B-Splines com ajuste global.

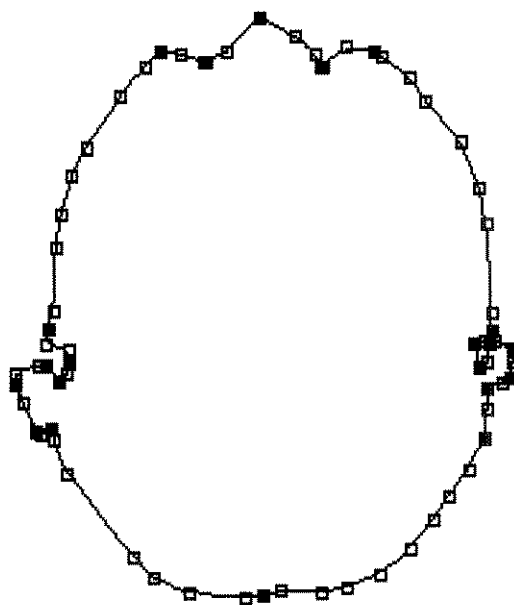


Figura 6.7: Representação por B-Splines com pontos de segmentação e de controle ($\square$).

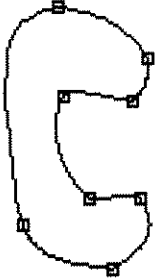
Primeiro contorno	Segundo contorno
	
Contorno casado	Contornos sobrepostos
	

Figura 6.8: Tabela de resultados do casamento de segmentos.

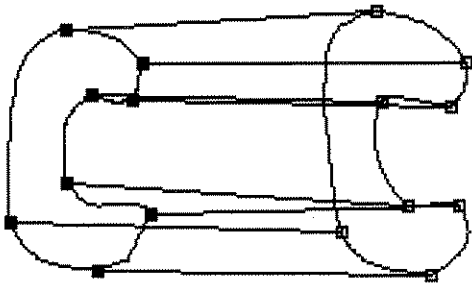


Figura 6.9: Representação do casamento de segmentos entre os contornos.

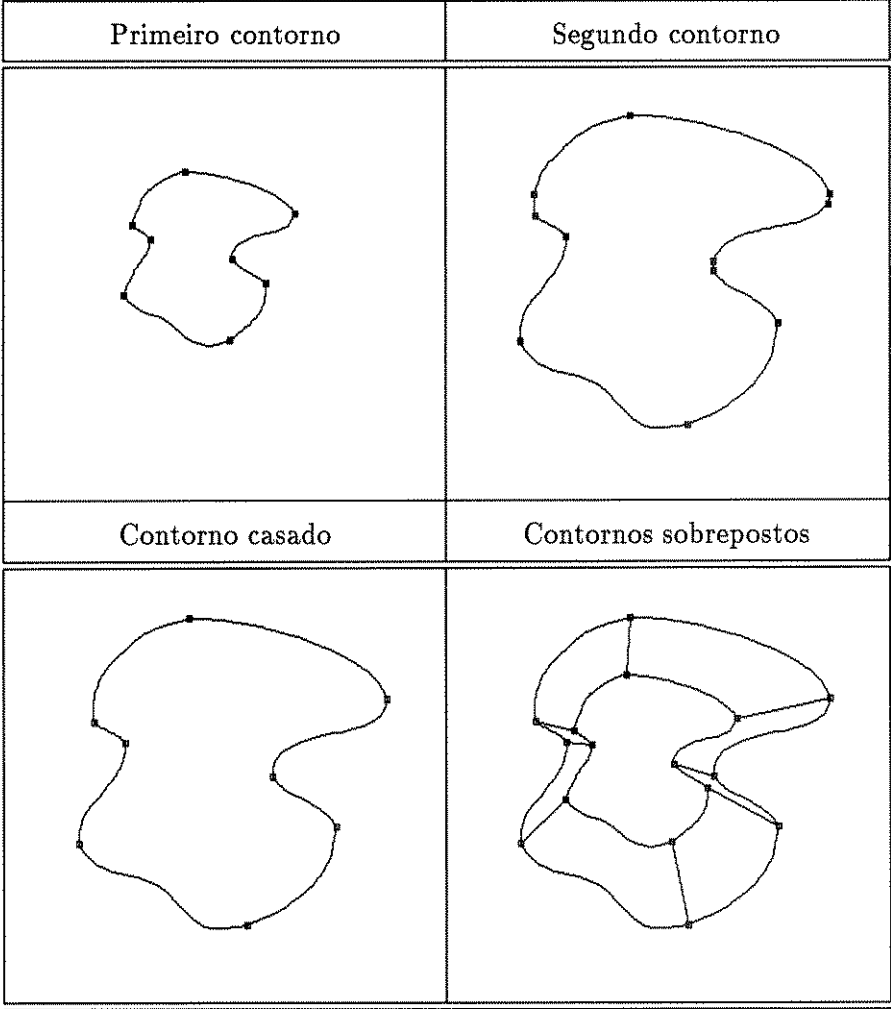


Figura 6.10: Casamento de segmentos.

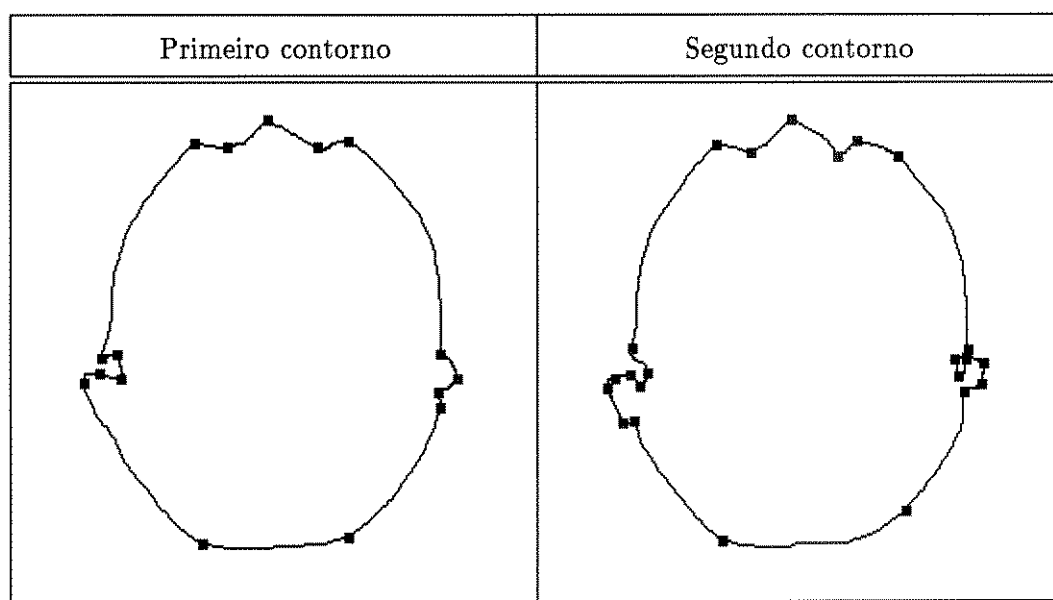


Figura 6.11: Contornos segmentados obtidos a partir de imagem de tomografia.

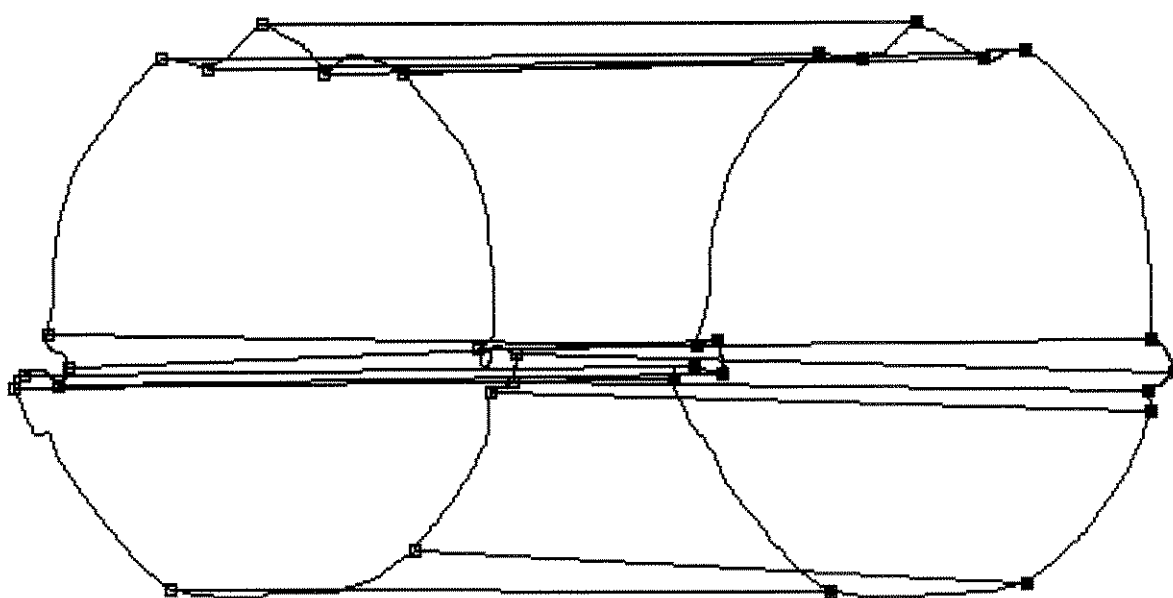


Figura 6.12: Representação do casamento de segmentos dos contornos da cabeça.

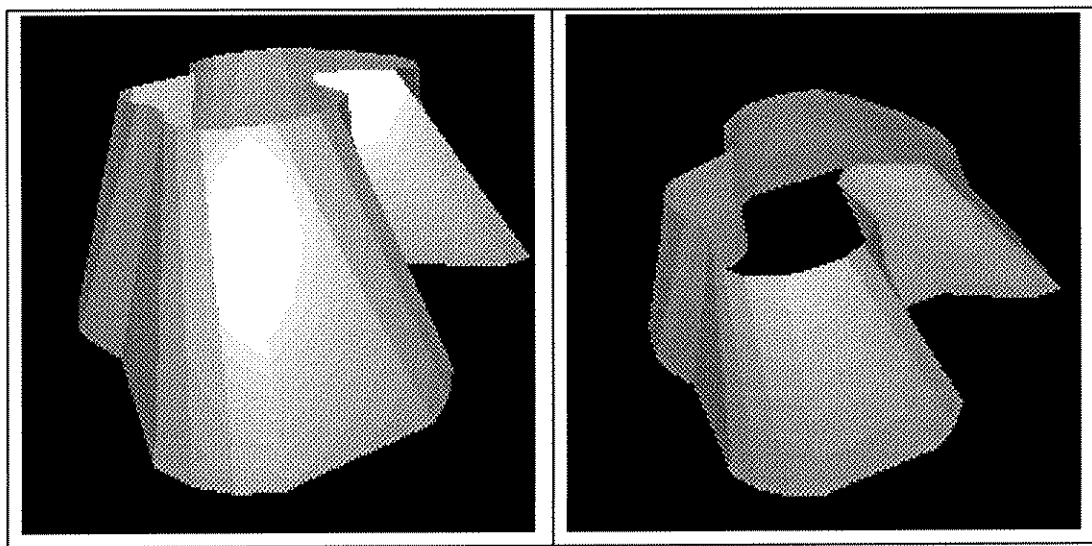


Figura 6.13: Reconstrução 3D por ajuste de superfícies de Coons dos contornos da fig. 6.10.

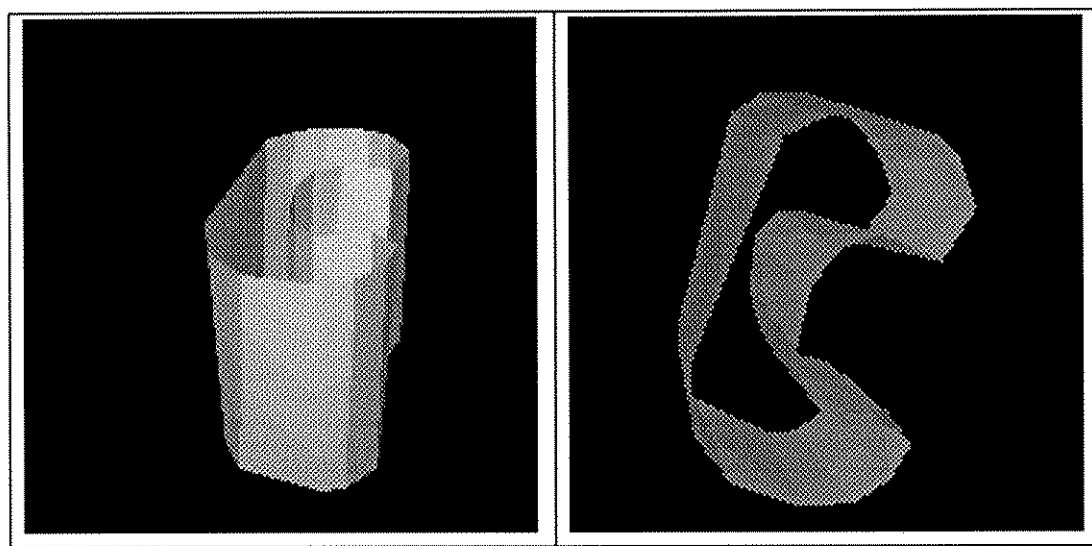


Figura 6.14: Reconstrução 3D por ajuste de superfícies de Coons dos contornos da fig. 6.8.

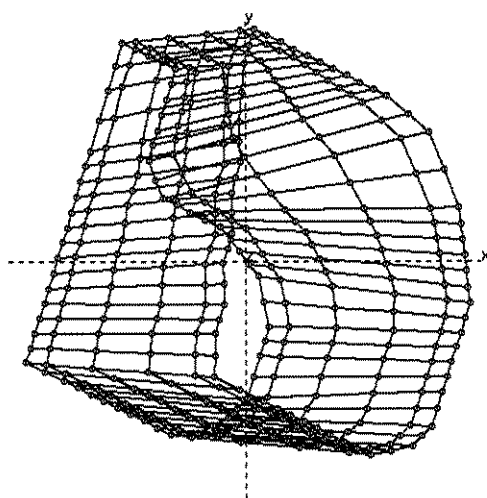


Figura 6.15: Estrutura "wireframe" de uma reconstrução 3D.

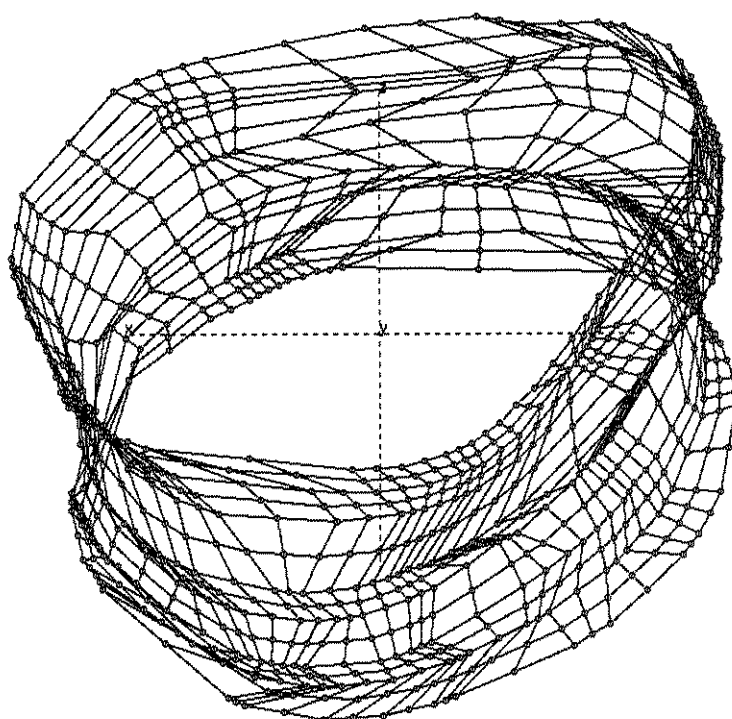


Figura 6.16: Estrutura "wireframe" de uma reconstrução 3D de um joelho.

Capítulo 7

Conclusão

7.1 Sobre o que foi feito: Comentários finais

O presente estudo sobre reconstrução 3D por ajuste de superfícies paramétricas incluiu uma introdução às principais modalidades de reconstrução tridimensional, aspectos e referências sobre o pré-processamento para a abordagem de superfícies, e, com maiores detalhes, diversas técnicas e métodos ligados às 3 etapas básicas para reconstrução 3D por superfícies paramétricas por "patches".

Foi apresentada uma revisão de métodos para segmentação, representação e descrição de contornos, problema importante para muitas aplicações de visão computacional. O método de representação desenvolvido e implementado pode ser usado para segmentação geral de contornos, representação por B-Splines e aproximação poligonal em 2 níveis de curvas. O método implementado é uma modificação do algoritmo proposto em [27], e melhora o método original em dois aspectos principais :

- 1 - representação mais compacta do contorno, necessitando um número menor de pontos e realizando uma maior filtragem de ruídos.

- 2 -melhor caracterização estrutural dos segmentos, permitindo que informações mais coerentes em relação ao contorno total sejam extraídas.

Além disso, todos os atributos apresentados podem ser usados na descrição de contornos, permitindo a definição de outros descritores mais adequados segundo outras

aplicações.

Foi desenvolvido e implementado um método de casamento de segmentos entre contornos baseado no algoritmo A^* . Esse algoritmo é uma generalização do método proposto em [61] e utiliza informações heurísticas que não são usadas pelo método original. Ele foi usado para os problemas de reconstrução 3D, e da forma pela qual está definido, pode ser modificado para utilização em outras aplicações, permitindo que outros descritores sejam utilizados, ou que as operações de junção sejam feitas em ambos contornos. O problema de casamento de contornos aparece em diferentes áreas de processamento de imagens e visão computacional, como visão estereoscópica e análise de imagens que se modificam no tempo. Acredita-se que o mesmo algoritmo desenvolvido possa ser usado nessas aplicações.

Finalmente, os métodos implementados foram integrados a dois ambientes de geração e visualização de superfícies, respectivamente, permitindo a obtenção de modelos 3D de objetos por ajuste de superfícies de Coons.

Além dos estudos reunidos nesta dissertação e dos programas criados, foram publicados os trabalhos [69] e [70], que tratam da etapa de segmentação, representação e descrição de contornos.

7.2 Sobre o que se pode fazer: Sugestões para trabalhos futuros

Além das melhorias que deverão ser feitas no método implementado, resultantes principalmente de um período mais longo de aprendizado e utilização dos programas em outras aplicações, existem diversos pontos que podem ser explorados e incorporados a este trabalho, bem como abordados como pesquisas à parte. Esses pontos são apresentados e brevemente discutidos nas próximas secções, divididas segundo a etapa a que se refere.

7.2.1 Pré-processamento: Correspondência entre Contornos

Esta fase é de importância fundamental para qualquer tentativa de automação do processo total de reconstrução 3D por ajuste de superfícies, e é a que menos foi estudada

na literatura levantada. O estudo dessas técnicas é importante não só para a reconstrução 3D, mas também para as aplicações apresentadas no capítulo sobre casamento de contornos. Na reconstrução 3D, qualquer tentativa de resolver o problema de subdivisão de superfícies tem que passar por essas técnicas.

7.2.2 Segmentação e Representação de Contornos

Além dos ajustes estruturais por vizinhança mínima e ângulo entre vizinhos adjacentes, aplicados aos pontos de segmentação, existe um terceiro teste que pode ser aplicado aos pontos de controle de cada segmento, o da distância de cada um desses pontos à reta suporte do segmento (fig. 7.1).

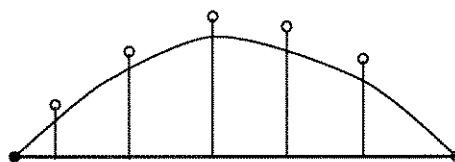


Figura 7.1: Distâncias dos pontos de controle à reta suporte do segmento.

Porém, esse teste seria feito não com o intuito de eliminar pontos de controle, uma vez que essa eliminação pode alterar a estrutura original do contorno (fig. 7.2), mas sim para identificar outros pontos de segmentação. Esse teste permite, de certa forma, um controle do processo sobre o fecho convexo do polígono de controle.

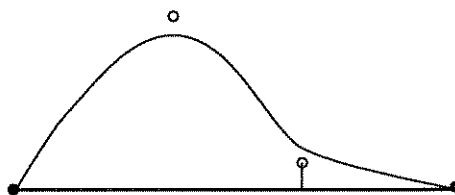


Figura 7.2: Situação em que o teste de distância pode alterar a estrutura do segmento.

O método de segmentação pode ter seu desempenho melhorado através de um pré-processamento do contorno por métodos rápidos de aproximação poligonal, fazendo com que o cálculo da curvatura seja feito em um número menor de pontos. Existem métodos extremamente rápidos que fazem essa aproximação poligonal a partir do próprio código da cadeia [65].

Muito interessante seria a incorporação dos programas ao Khoros. As principais mudanças que devem ser realizadas se referem à entrada e saída de cada programa. Atualmente, o principal impedimento para essa incorporação é a não existência de formatos de arquivo que representem o código da cadeia, o código dos arquivos *.sp*, e outros do gênero. Assim, esforços para incorporação ao Khoros devem se concentrar nesses pontos.

Em relação à teoria apresentada, a principal linha que pode ser seguida é a da extensão dos conceitos apresentados às curvas espaciais (3D). Como as B-Splines podem facilmente ser estendidas, acredita-se que também o restante do método possa ser. Talvez a mudança mais substancial seja a da utilização do conceito de torção, e não só o de curvatura. Esses conceitos são encontrados em livros de geometria diferencial.

7.2.3 Descrição e Casamento de Contornos

As principais diferenças dos problemas que envolvem o casamento de contornos dizem respeito à dependência ou não de escala, rotação e translação dos objetos. A análise e definição de outros descritores de segmento deve levar em consideração esses pontos. O descritor *Bs*, na forma pela qual foi definido, procura ser independente de escala, mas não de translação (devido ao atributo ponto médio) e nem de rotação (devido ao atributo vetor diretor). A maioria das aplicações em que o registro das imagens não garante sua coerência espacial entre as fatias, requer métodos que identifiquem, a partir do casamento, transformações que, se aplicadas aos contornos de cada fatia, procuram recuperar essa coerência. Um exemplo é a reconstrução 3D de um objeto que foi fatiado em lâminas para observação em microscopia. Uma das formas de tomar o descritor independente de rotação e translação é através do uso de apenas atributos estruturais. Nesse sentido, um maior aprofundamento nos estudos das características do histograma criado no capítulo 3 é desejável. O maior problema associado à forma pela qual ele foi definido é a amostragem desse histograma, ou seja, o número de pontos que dividem seu domínio, determinado pela cardinalidade do polígono de controle. Duas soluções são propostas para resolver esse problema. A primeira é o estudo do problema de similaridade entre histogramas, que é estudado em teoria estatística, e analisa várias medidas de similaridade. A segunda é através da utilização da curva B-Spline representada, ou até do próprio polígono de controle, para fazer uma re-amostragem dos pontos e com que os histogramas tenham o mesmo número

de pontos.

Uma outra forma de comparar os segmentos é através das equações das B-Splines representadas em cada um. Existe um teorema de geometria diferencial que garante a unicidade estrutural de duas curvas, caso suas curvaturas e torções coincidam. Nesse sentido, curvatura e torção de cada segmento tornam-se atributos promissores para pesquisas.

Associada à segmentação e representação de curvas espaciais está a definição de descritores 3D, a partir da extensão dos conceitos 2D, tema que também pode ser explorado em pesquisas futuras. Pouco existe sobre essa extensão na literatura levantada.

Em relação ao casamento de segmentos, existem 3 pontos que parecem promissores para pesquisas futuras, a medida que complementarão o método desenvolvido. O primeiro é a redefinição do método para que ele permita que operações de junção sejam feitas em ambos contornos. Duas coisas terão que ser modificadas no método original. A principal se refere à função h' , e suas implicações foram discutidas na secção 4.3.4. A segunda diz respeito à explosão de cada nó, que gera seus sucessores. Tome, por exemplo, a situação do casamento da figura 7.3. Suponha que o último segmento casado foi aquele definido por (b_{i-1}, c_{i-1}) . Pelo método original, supondo que C possui mais segmentos que B , então o próximo ponto do contorno C que deveria ser casado, obrigatoriamente seria par de b_i . Ou seja, os próximos nós a serem gerados seriam do tipo $(b_i, c_i), (b_i, c_{i+1}), \dots, (b_i, c_{i+a})$. Uma forma de permitir que junções sejam feitas no contorno B é incluir no conjunto dos nós gerados, casamentos do tipo $(b_{i+1}, c_i), (b_{i+2}, c_i), \dots$. No caso da figura 7.3, pode-se perceber, intuitivamente, que o casamento (b_{i+1}, c_i) é melhor do que $(b_i, c_i), (b_i, c_{i+1}), \dots$. Assim, as principais mudanças se referem à definição de novos critérios de explosão.

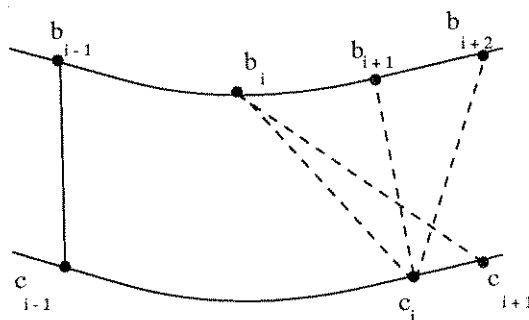


Figura 7.3: Casamento com junção em ambos contornos.

O segundo ponto está ligado a um estudo teórico mais aprofundado abordando a complexidade do algoritmo desenvolvido, e critérios que tornam o método ótimo. Um problema diretamente ligado a isso é o de estimar o custo médio de cada aresta, presente na função h' .

Por último, seria interessante o estudo de casamentos mais complexos, envolvendo mapeamentos parciais entre contornos e casos de subdivisão de superfícies, ou mapeamento de um para muitos contornos.

7.2.4 Ajuste de Superfícies

O ajuste de superfícies, talvez por estar ligado ao objetivo final do processo de reconstrução 3D, acaba fazendo com que o desdobrar de suas pesquisas dependa diretamente da aplicação. Por exemplo, em uma situação de simulação de uma cirurgia, suponha que o cirurgião deve cortar e puxar um pedaço da pele. A superfície que está representando a pele deve se comportar de forma mais semelhante a uma situação real. Nesse sentido, um tópico interessante a ser estudado é o da utilização de outros tipos de superfícies paramétricas, bem como suas propriedades.

A manipulação e análise do objeto 3D final também possui grande importância em diversas aplicações [5]. O estudo de técnicas de modelagem geométrica para incorporação de ferramentas que auxiliem nessas operações pode ser extremamente útil.

BIBLIOGRAFIA

- [1] Goshtasby A. "Gaussian decomposition of two-dimensional shapes: a unified representation for CAD and vision applications". *Pattern Recognition*, 25(5):463–472, 1992.
- [2] Martelli A. "On the Complexity of Admissible Search Algorithms". *Artificial Intelligence*, 8(1):1–13, 1977.
- [3] Wallin A. "Constructing Isosurfaces from CT Data". *IEEE Computer Graphics and Applications*, pages 52–62, november 1988.
- [4] Wilson A.J.S., J.G.Mills e M.G.Norman. "Experiments in tracking and rendering surfaces in 3D seismic images". Technical report, Edinburgh Parallel Computer Centre, Edinburgh, 1990.
- [5] Falcão A.X. *Visualização de Volumes aplicada à Área Médica*. dissertação de mestrado, FEE-UNICAMP, Campinas, 1993.
- [6] Soroka B.I. "Generalized Cones from Serial Sections". *Computer Graphic and Image Processing*, 15:154–166, 1981.
- [7] Boor C. "Calculating with B-splines". *Journal of Approximation Theory*, 6:50–62, 1972.
- [8] Levinthal C. e Ware R. "Three-dimensional reconstruction from serial sections". *Nature*, 236:207–210, march 1972.
- [9] Geiger D. "Tutorial em Visão Computacional". In *Anais: Jornada EPUSP/IEEE em Computação Visual*, pages 363–383, 1990.
- [10] Geist D. e Vannier M.W. "PC-Based 3D Reconstruction of Medical Images". *Computer & Graphics*, 13(2):135–143, sep 1989.

- [11] Gelperin D. "On the Optimality of A*". *Artificial Intelligence*, 8(1):69–76, 1977.
- [12] Gordon D. e Udupa J.K. "Fast Surface Tracking in Three-Dimensional Binary Images". *Computer Vision, Graphics and Image Processing*, 45(2):196–214, feb 1989.
- [13] Hearn D. e Baker M.P. *Computer Graphics*. Prentice-Hall, Englewood Cliffs, NJ, 1986.
- [14] Meyers D., Skinner S. e Sloan K. "Surfaces from Contours". *ACM Transactions on Graphics*, 11(3):228–258, jul 1992.
- [15] Rogers D.F. e Adams J.A. *Mathematical Elements for Computer Graphics*. McGraw-Hill, NY, 1976.
- [16] Rogers D.F. e Adams J.A. *Mathematical Elements for Computer Graphics*. McGraw-Hill, NY, 2nd edition, 1990.
- [17] Ballard D.H. e Brown C.M. *Computer Vision*. Prentice-Hall, Englewood Cliffs, NJ, 1982.
- [18] Burr D.J. "Elastic Matching of Line Drawings". *IEEE Transactions on Pat. Anal. Mach. Intel.*, 3(3):708–713, november 1981.
- [19] Filip D.J. e Ball T.W. "Procedurally representing lofted surfaces". *IEEE Computer Graphics and Applications*, pages 52–62, november 1989.
- [20] Ney D.R., Fishman E.K., Magid D. e Drebin A.R. "Volumetric Rendering of Computed Tomography Data: Principles and Techniques". *IEEE Computer Graphics and Applications*, pages 24–32, march 1990.
- [21] Horowitz E. e Sahni S. *Fundamentos de Estruturas de Dados*. Campus, RJ, 1984.
- [22] Rich E. e Knight K. *Artificial Intelligence*. McGraw-Hill, NY, 1991.
- [23] Ruiz E.E.S. *Comparação de técnicas e Métodos para Visão Computacional em Ambientes Industriais*. dissertação de mestrado, FEE-UNICAMP, Campinas, 1987.
- [24] Stein F. e Medioni G. "Structural Indexing Efficient 2-D Object Recognition". *IEEE Transactions on Pat. Anal. Mach. Intel.*, 14(12):1198–1204, december 1992.
- [25] L.H. Figueiredo e P.C.P. Carvalho. *Introdução à Geometria Computacional*. publicado pelo XVIII Colóquio Brasileiro de Matemática, 1991.

- [26] Farin G. *Rational Curves and Surfaces in: Mathematical Methods in Computer Aided Geometric Design - Lyche, T. and Schumaker, L.L.(eds.)*. Academic Press, San Diego, 1989.
- [27] Medioni G. e Yasumoto Y. "Corner detection and curve representation using cubic B-splines". *Computer Vision, Graphics and Image Processing*, 39:267–278, 1987.
- [28] Tavares G. *Funções de Morse e Classificação de Superfícies*. publicado pelo XII Colóquio Brasileiro de Matemática, 1979.
- [29] West G.A.W. e Rosin P.L. "Techniques for Segmenting Image Curves into Meaningful Descriptors". *Pattern Recognition*, 24(7):643–652, 1991.
- [30] Herman G.T. e Liu H.K. "Dynamic Boundary Surface Detection". *Computer Graphic and Image Processing*, 7:130–138, 1978.
- [31] Herman G.T. e Gordon R. "Reconstruction of pictures from their projections". *Communications of the ACM*, 14(12):759–768, dec 1971.
- [32] Freeman H. "On the Encoding of Arbitrary Geometric Configurations". *IEEE Trans. Electr. Comp.*, EC-10:260–268, 1961.
- [33] Freeman H. e Davis L. "A corner-finding algorithm for chain-coded curves". *IEEE Transactions on Computers*, C-26:297–303, march 1977.
- [34] Fuchs H., Levoy M. e Pizer S.M. "Interactive Visualization of 3D Medical Data". *IEEE Computer*, 22(8):46–50, aug 1989.
- [35] Fuchs H., Kedem Z.M. e Uzelton S.P. "Optimal Surface Reconstruction from Planar Contours". *Communications of the ACM*, 20(10):693–702, oct 1977.
- [36] Nishida H. e Mori S. "Algebraic Description of Curve Structure". *IEEE Transactions on Pat. Anal. Mach. Intel.*, 14(5):516–533, may 1992.
- [37] Pitas I. e Venetsanopoulos A.N. "Morphological Shape Representation". *Pattern Recognition*, 25(6):555–565, 1992.
- [38] Boissonnat J.D. "Shape Reconstruction from Planar Cross Sections". *Computer Vision, Graphics and Image Processing*, 44:1–29, 1988.
- [39] Cappelletti J.D. e Rosenfeld A. "Three-Dimensional Boundary Following". *Computer Vision, Graphics and Image Processing*, 48:80–92, 1989.

- [40] Foley J.D., Van Dam A., Freiner S.K. e Hughes J.F. *Fundamentals of Interactive Computer Graphics*. Addison Wesley Publishing Company, 2nd edition, 1990.
- [41] Angulo J.M. e Madrigal R.I. *Vision Artificial por Computador*. Paraninfo, Madrid, 1986.
- [42] Gomes J.M. “Representação de Arcos por Curvas de Bézier”. In *Anais: II SIBGRAPI*, pages 519–523, 1989.
- [43] Gomes J.M. e Velho L.C. *Conceitos Básicos de Computação Gráfica*. publicado pela VII Escola de Computação, IME - USP, São Paulo, 1990.
- [44] Tremblay J.P. e Sorenson P.G. *An Introduction to Data Structures with Applications*. McGraw-Hill, Singapore, 2nd edition, 1984.
- [45] Frenkel K.A. “Volume Rendering”. *Communications of the ACM*, 32(4):426–435, april 1989.
- [46] You K.C. e Fu K.S. “A Syntactic Approach to Shape Recognition Using Attributed Grammar”. *IEEE Transactions on Sys., Man and Cybern.*, SMC-9(6):334–345, june 1979.
- [47] Toennies K.D., Udupa J.K., Herman G.T., Wornom III I.L. e Buchman S.R. “Registration of 3D Objects and Surfaces”. *IEEE Computer Graphics and Applications*, pages 52–62, may 1990.
- [48] Fu K.S. *Syntactic Pattern Recognition and Applications*. Prentice-Hall, Englewood Cliffs, NJ, 1982.
- [49] Fu K.S. e Basu S. “Image Segmentations by Syntactic Method”. *Pattern Recognition*, 20(1):33–44, 1987.
- [50] Chen L.-S. e Sontag M.R. “Representation, Display and Manipulation of 3D Digital Scenes and Their Medical Applications”. *Computer Vision, Graphics and Image Processing*, 48(2):190–216, nov 1989.
- [51] Costa L.A.R. e D. Geiger. “Matching Deformable Contours for Medical Imaging”. In *Anais: I Simpósio Brasileiro de Automação Inteligente - I SBAI -*, pages 195–202, 1993.

- [52] Chang L.W., Chen H.W. e Ho J.R. "Reconstruction of 3D Medical Images: a non-linear interpolation technique for reconstruction of 3D medical images". *CVGIP: Graph. Models. and Image Proces.*, 53(4):382–391, july 1991.
- [53] Levoy M. "Display of Surfaces from Volume Data". *IEEE Computer Graphics and Applications*, 8(3):29–37, may 1988.
- [54] Levoy M. "A Hybrid Ray Tracer for Rendering Polygon and Volume Data". *IEEE Computer Graphics and Applications*, 10(2):33–40, mar 1990.
- [55] Rhodes M. "Computer Graphics in Medicine". *IEEE Computer Graphics and Applications*, pages 20–23, march 1990.
- [56] Sabin M. *Sculptured Surface Definition - A Historical Survey in: Computer Graphics Techniques: Theory and Practice - Rogers, D.F. and Earnshaw, R.A.(eds.)*. Springer-Verlag, NY, 1990.
- [57] Richard M.J., Allard J., Ghosh S.K. e Bougouss M. "Three-Dimensional Reconstruction of Human Limbs from Tomographic Views". *Computer and Biomedical Research*, 22:26–35, 1989.
- [58] Zyda M.J., Jones A.R. e Hogan P.G. "Surface Construction from Planar Contours". *Computer & Graphics*, 11(4):393–408, 1987.
- [59] Chong M.M.S., Gay R.K., Tan H.N. e Liu J. "Automatic Representation of Fingerprints for Data Compression by B-spline Functions". *Pattern Recognition*, 25(10):1199–1210, 1992.
- [60] Kehtarnavaz N., Simar L.R. e de Figueiredo R.J.P. "Syntactic/Semantic Technique for Surface Reconstruction from Cross-Sectional Contours". *Computer Vision, Graphics and Image Processing*, 42:399–409, 1988.
- [61] Kehtarnavaz N. e de Figueiredo R.J.P. "A Framework for Surface Reconstruction from 3D Contours". *Computer Vision, Graphics and Image Processing*, 42:32–47, 1988.
- [62] Baruch O. e Murray H.L. "Segmentation of Two-dimensional Boundaries using Chain-code". *Pattern Recognition*, 21(6):581–589, 1988.
- [63] Srinivasan P., Liang P. e Hackwood S. "Computational Geometric Methods in Volumetric Intersection for 3D Reconstruction". *Pattern Recognition*, 23(8):843–857, 1990.

- [64] T. Pavlidis e Ali F. "Hierarchical Syntactic Shape Analyser". *IEEE Transactions on Pat. Anal. Mach. Intel.*, PAMI-1(1):2-9, january 1979.
- [65] Cortez P.C. e Carvalho J.M. "Avaliação de 3 Algoritmos de Modelagem Poligonal de Contornos Bidimensionais". In *Comunicações: VI Simpósio Brasileiro de Computação Gráfica e Processamento de Imagens - VI SIBGRAPI- 19 a 22 de outubro de 1993, Recife, PE*, pages 21-24, 1993.
- [66] Schalkoff R. *Pattern Recognition: Statistical, Structural and Neural Approaches*. John Wiley & Sons, Singapore, 1992.
- [67] Duda R. O. e Hart P.E. *Pattern Classification and Scene Analysis*. John Wiley & Sons, NY, 1973.
- [68] Gonzalez R.C. e Wintz P. *Digital Image Processing*. Addison Wesley Publishing Company, Reading, Mass., 2nd edition, 1987.
- [69] Cesar Jr. R.M. e Lotufo R.A. "Segmentação e Descrição de Contornos por B-Splines, e sua Aplicação no Casamento de Contornos.". In *VI Simpósio Brasileiro de Computação Gráfica e Processamento de Imagens - VI SIBGRAPI- 19 a 22 de outubro de 1993, Recife, PE*, pages 55-64, 1993.
- [70] Cesar Jr. R.M. e Lotufo R.A. "Segmentação e Representação de Contornos por B-Splines para Aplicações em Visão Computacional.". In *I Simpósio Brasileiro de Automação Inteligente - I SBAI - 8 a 10 de setembro de 1993, Rio Claro, SP*, pages 223-232, 1993.
- [71] Cesar Junior R.M. e Linhares O.L. "Métodos Numéricos de Aproximação Poligonal: uma análise comparativa de 3 métodos". In *Anais: Jornada EPUSP/IEEE em Computação Visual*, pages 191-197, 1990.
- [72] Chaudhury S., Acharyya A. e Subramanian S. "Recognition of Occluded Objects with Heuristic Search". *Pattern Recognition*, 23(6):617-635, 1990.
- [73] Wu S.C., Abel J.F. e Donald P.G. "An Interactive Computer Graphics Approach to Surface Representation". *Communications of the ACM*, 20(10):703-712, october 1977.
- [74] Hung S.H. e Kaswand T. "Critical Points on a Perfectly 8-or-6 Connected Thin Binary Line". *Pattern Recognition*, 16(3):297-306, 1983.

- [75] Davis S.L. "Understanding Shapes: Angles and Sides". *IEEE Transactions on Computers*, C-26(3):236–242, march 1977.
- [76] Horowitz S.L. "A Syntactic Algorithm for Peak Detection in Waveforms with Applications to Cardiography". *Communications of the ACM*, 18(5):281–285, may 1975.
- [77] Liou S.P. e Jain R. "An Approach to Three-dimensional Image Segmentation". *CVGIP: Image Understanding*, 53(3):237–252, may 1991.
- [78] Chen S.Y., Lin W.C. e Chen C.T. "Split-and-Merge Image Segmentation Based on Localized Feature Analysis and Statistical Tests". *CVGIP: Graph. Models. and Image Proces.*, 53(5):457–475, september 1991.
- [79] Elvins T. "Tutorial on Volume Visualization". Transparências do SIBGRAPI'91 - IV Simpósio Brasileiro de Computação Gráfica e Processamento de Imagens, jul 1991.
- [80] Pavlidis T. "Waveform Segmentation through Functional Approximation". *IEEE Transactions on Computers*, C-22(7):689–697, july 1973.
- [81] Pavlidis T. *Structural Pattern Recognition*. Springer-Verlag, NY, 1977.
- [82] Pavlidis T. *Algorithms for graphics and image processing*. Computer Science, Rockville, 1982.
- [83] Pavlidis T. e Horowitz S.L. "Segmentation of plane curves". *IEEE Transactions on Computers*, C-23(8):860–870, august 1974.
- [84] Preto T. M. *SCANLINE: Um Sistema para Visualização de Imagens Foto-Realistas*. dissertação de mestrado, FEE-UNICAMP, Campinas, 1992.
- [85] Ramer U. "An Iterative Procedure for the Polygonal Approximation of Plane Curves". *Computer Graphic and Image Processing*, 1:244–256, 1972.
- [86] Tiede U., Hoehne K.H., Bomans M., Pommert A., Riemer M. e Wiebecke G. "Surface Rendering - Investigation of Medical 3D-Rendering Algorithms". *IEEE Computer Graphics and Applications*, pages 41–53, mar 1990.
- [87] Lin Wei-Chung, Chen Shiuh-Yung e Chen C.T. "A New Surface Interpolation Technique for Reconstruction 3D Objects from Serial Cross-Sections". *Computer Vision, Graphics and Image Processing*, 48(1):124–143, oct 1989.

- [88] Newman W.N. e R.F.Sproull. *Principles of interactive computer graphics*. Mc Graw-Hill, Singapore, 2nd edition, 1979.
- [89] Chou W.S., Wu C.M., Chen Y.C. e Hsieh K.S. "Detecting Myocardial Boundaries of Left Ventricle form a Single Frame 2D Image". *Pattern Recognition*, 23(7):799-806, 1990.
- [90] W.Tiller. "Rational B-splines for Curve and Surface Representation". *IEEE Computer Graphics and Applications*, pages 61-69, september 1983.
- [91] Shinagawa Y., Kunii T. e Kergosien Y.L. "Surface Coding Based on Morse Theory". *IEEE Computer Graphics and Applications*, pages 66-77, september 1991.