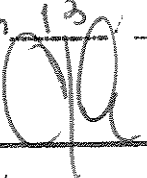


UNIVERSIDADE ESTADUAL DE CAMPINAS

Faculdade de Engenharia Elétrica

Departamento de Engenharia de Computação e Automação Industrial

RECONSTRUÇÃO 3D A PARTIR DE SEÇÕES TRANSVERSAIS DE OBJETOS

Este exemplar corresponde à redação final da tese	
defendida por	<u>Hélio Pedrini</u>
e aprovada pela Comissão	
Julgadora em	<u>13</u> / <u>12</u> / <u>1994</u>
	Orientador

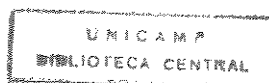
Autor : Hélio Pedrini

Orientador : Prof. Dr. Clésio Luís Tozzi

Dissertação submetida à Comissão de Pós-Graduação da Faculdade de Engenharia Elétrica da Universidade Estadual de Campinas - UNICAMP, como parte dos requisitos necessários para obtenção do Título de Mestre em Engenharia Elétrica.

Campinas-SP

Dezembro de 1994



*Dedico este trabalho aos meus pais e aos meus irmãos,
que sempre me apoiaram com amor, carinho e compreensão
em todos os momentos de minha vida.*

*Se não houver frutos
Valeu a beleza das flores*

*Se não houver flores
Valeu a sombra das folhas*

*Se não houver folhas
Valeu a intenção da semente*

(Henfil)

AGRADECIMENTOS

- Ao Prof. Dr. Clésio Luís Tozzi pela orientação, amizade e incentivo durante a realização deste trabalho;
- Aos Prof. Dr. Roberto de Alencar Lotufo e Dra. Wu Shin-Ting pelas sugestões e comentários realizados;
- À colega Tânia Martins Preto pela colaboração na visualização das imagens através do algoritmo Scanline Z-Buffer;
- Ao colega José Eduardo C. Castanho pelo auxílio no processo de aquisição e registro das imagens das plantas;
- À Prof. Dra. Marília de Moraes Castro do Departamento de Morfologia e Sistemática Vegetal do Instituto de Biologia cujas amostras dos cortes transversais das plantas contribuíram para a obtenção dos resultados experimentais;
- A todos os colegas do Laboratório de Computação e Automação (LCA) pelo apoio e agradável convívio que me proporcionaram nestes anos;
- A todo o pessoal do Laboratório de Processamento de Informações Georeferenciadas (LAPIG) do Instituto de Geociências pelos recursos computacionais que permitiram a digitalização e processamento das imagens das plantas;
- A todos os colegas do CPqD-Telebrás que tornaram possível a conclusão deste trabalho;
- A Hermínia Capel Clara pela paciência, apoio e carinho que me acompanharam durante a elaboração deste trabalho;
- A todos que, direta ou indiretamente, contribuíram para a realização deste trabalho.

RESUMO

O objetivo deste trabalho é propor uma solução para o problema de reconstrução da superfície externa de objetos tridimensionais definidos através de suas seções transversais.

As principais abordagens existentes na literatura são analisadas e comparadas, buscando-se compreender os conceitos fundamentais envolvidos no processo de reconstrução de objetos.

A metodologia proposta utiliza um algoritmo heurístico baseado em técnicas de triangulação, o qual foi implementado na forma de uma ferramenta computacional e aplicado na reconstrução de objetos sintetizados e reais complexos. As principais limitações encontradas nos métodos convencionais de triangulação foram eliminadas. As dificuldades em relação à ligação de contornos com formas e tamanhos não similares foram solucionadas. O problema de ramificação de contornos também foi tratado satisfatoriamente. O método pode operar automaticamente sem a necessidade de interação com o usuário durante o processo de reconstrução.

A avaliação do algoritmo de reconstrução foi realizada através da extração de medidas geométricas dos objetos tais como volume, área superficial e centro de massa. A visualização dos objetos foi realizada através de um módulo do ambiente PROSIM (Prototipação e Síntese de Imagens Foto-Realistas), desenvolvido pelo próprio grupo de Computação de Imagens do DCA/FEE.

ABSTRACT

The purpose of this work is to present a triangulation method for the external surface reconstruction of 3D objects defined by a sequence of two-dimensional cross sections.

Most of the algorithms proposed in the literature suffer from some limitations. Incorrect results can be obtained when the contours in two successive slices are very different. In the same way, the presence of multiple contours in a slice leads to ambiguities in defining the correct links. The algorithm proposed in this work provides a solution to these problems and it was implemented and applied to the reconstruction of the external surface of complex shaped objects.

Quantitative measurements were accurately estimated from the reconstructed objects, such as surface area and volume, in order to validate the reconstruction method proposed.

Conteúdo

1	Introdução	1
2	Métodos de Reconstrução	6
2.1	Métodos Baseados em Volume	7
2.2	Métodos Baseados em Superfície	10
2.2.1	Técnicas de Triangulação	13
2.2.1.1	Métodos Ótimos	15
2.2.1.1.a	Método de Fuchs, Kedem e Uzelton	17
2.2.1.2	Métodos Heurísticos	18
2.2.1.2.a	Aproximação de Keppel	18
2.2.1.2.b	Heurística de Christiansen e Sederberg	20
2.2.1.2.c	Heurística de Ganapathy e Dennehy	22
2.2.1.2.d	Heurística de Ekoule, Peyrin e Odet	24
3	Metodologia de Reconstrução de Objetos 3D	27
3.1	Reconstrução da Superfície entre Contornos	28
3.1.1	Caso 1:1	28
3.1.1.1	Caso 1:1 entre Contornos Convexos	29
3.1.1.2	Caso 1:1 entre Contornos não Convexos	30
3.1.1.2.a	Decomposição dos Contornos	32
3.1.1.2.b	Transformação dos Contornos	33
3.1.2	Caso 1:n	35
3.1.2.1	Interpolação dos Contornos	35
3.1.3	Caso n:m	38
3.1.3.1	Determinação da Sobreposição de Contornos	38
3.1.3.2	Decisão de Mapeamento	40
3.2	Comentários Finais	42
4	Determinação de Medidas e Visualização dos Objetos	45
4.1	Determinação de Medidas	45
4.1.1	Área Superficial	45
4.1.2	Volume	46
4.2	Visualização	48
4.2.1	Eliminação de Superfícies Escondidas	48
4.2.2	Modelos de Representação de Objetos	48
4.2.3	Modelos de Iluminação e Tonalização	49
4.2.4	Transformações Geométricas	50
4.2.5	Transparência	50
4.2.6	Sombra	50
5	Análise do Algoritmo e Resultados	52
5.1	Aplicação e Avaliação do Algoritmo	52
5.2	Imagens dos Objetos Reconstruídos	59

6 Conclusões	74
Bibliografia	76
Apêndice A Documentação da Ferramenta de Reconstrução	82

Lista de Figuras

1.1	Etapas envolvidas no processo de reconstrução 3D	2
1.2	(a) Contornos com formas diferentes(b) Contornos com orientações diferentes (c) Múltiplos contornos em cada seção transversal	3
1.3	Contornos identificados e conectados entre duas seções transversais	4
2.1	Cortes Transversais	7
2.2	Voxels formando uma fatia	8
2.3	Objeto	9
2.4	Convenção de numeração de um octante	9
2.5	Representação <i>octree</i> do objeto	10
2.6	(a) Objeto seccionado por planos paralelos(b) contornos formados pela intersecção da superfície do objeto com o plano	11
2.7	Pontos arbitrariamente distribuídos no espaço 3D	12
2.8	Pontos distribuídos ao longo de contornos planares paralelos	12
2.9	(a) Interpolação entre contornos não similares (b) Ramificação de um contorno em outros dois	13
2.10	Exemplo de Homotopia	14
2.11	Face triangular formada entre contornos adjacentes	15
2.12	Grafo representando todas as superfícies aceitáveis entre 2 contornos	15
2.13	Exemplo de (a) associação de triangulação com (b) teoria de grafos	19
2.14	Exemplo de contornos apresentando segmentos convexos e côncavos	19
2.15	Grafo associado ao exemplo dado	20
2.16	Triangulação através do critério da aresta de comprimento mínimo: face (P_i, P_{i+1}, Q_j) é escolhida em relação à (P_i, Q_j, Q_{j+1}) , pois $\overline{P_{i+1}Q_j} < \overline{P_iQ_{j+1}}$	21
2.17	Triangulação entre dois contornos não similares	21
2.18	Cilindro seccionado por planos paralelos	22
2.19	Exemplo em que o algoritmo da aresta de comprimento mínimo apresenta falhas	22
2.20	Janela retangular envolvendo o contorno	23
2.21	Contorno mapeado em um quadrado unitário	23
2.22	Passos do método proposto por Ganapathy e Dennehy	24
2.23	Dois possíveis caminhos para ligação dos contornos C_{k-1} e C_k	25
2.24	Procedimento de ligação entre contornos convexos	26
3.1	Ligação entre contornos (a) caso 1:1 (b) caso 1:n (c) caso n:m	29
3.2	Triangulação entre contornos não convexos	31
3.3	Contornos (a) convexo (b) côncavo	31
3.4	Exemplo de decomposição de contorno não convexo em concavidades elementares	32
3.5	Concavidades de 2a. ordem para o exemplo dado	33
3.6	Representação hierárquica da decomposição do contorno apresentado	33
3.7	Projeção de uma concavidade elementar	34
3.8	Ramificação de contornos	35

3.9	Renumeração de contornos da ramificação	36
3.10	Situação de ramificação (a) triangulação entre contornos (b) vista lateral da triangulação	37
3.11	Contornos apresentando centróides não colineares	37
3.12	Triangulação entre contornos	37
3.13	Avaliação da sobreposição de dois contornos adjacentes	39
3.14	Exemplo de mapeamento de contornos (a) Contornos (b) Matriz de ligação	40
3.15	Situações consideradas pelo algoritmo de reconstrução	43
4.1	Face triangular formada entre contornos adjacentes	46
4.2	Vetores formados entre vértices de contornos adjacentes	46
4.3	Objeto e sua representação B-Rep	49
5.1	Formas geométricas utilizadas (a) Cilindro (b) Esfera (c) Cone (d) Forma Combinada	53
5.2	Exemplos de cortes transversais de um pimentão	55
5.3	Contornos de cortes transversais de um pimentão	56
5.4	Exemplos de cortes transversais da planta	58
5.5	Gráfico de desempenho do algoritmo de triangulação	59
5.6(a)	Cilindro	60
5.6(b)	Rendering (Bouknight) do Cilindro	61
5.7(a)	Esfera	62
5.7(b)	Rendering (Bouknight) da Esfera	63
5.8(a)	Cone	64
5.8(b)	Rendering (Bouknight) do Cone	65
5.9(a)	Forma Combinada	66
5.9(b)	Rendering (Bouknight) da Forma Combinada	67
5.10(a)	Ramificação	68
5.10(b)	Rendering (Bouknight) da Ramificação	69
5.11(a)	Pimentão	70
5.11(b)	Rendering (Bouknight) do Pimentão	71
5.12(a)	Plantas	72
5.12(b)	Rendering (Bouknight) das Plantas	73
A.1	Arquitetura da Ferramenta de Reconstrução	85
A.2	Interface principal da ferramenta	86
A.3	Interface de parâmetros de visualização	87
A.4	Exemplo de arquivo de faces triangulares obtidas através da triangulação entre dois contornos adjacentes	89

Lista de Tabelas

I	Resultados da reconstrução de quatro formas geométricas	53
II	Resultados da reconstrução de um pimentão	54
III	Tempo de CPU amostrado para cada objeto reconstruído	59
IV	Exemplo de relatório de medidas de um objeto	84

INTRODUÇÃO

Em muitos ramos da ciência aplicada, há um grande interesse em se reconstruir objetos tridimensionais a partir de suas seções transversais. Estas seções são obtidas através da intersecção do objeto com uma coleção de planos não necessariamente paralelos entre si.

O processo de reconstrução consiste, basicamente, em se obter uma representação tridimensional do objeto possibilitando não apenas sua visualização, mas também a compreensão de sua estrutura através da extração e análise de parâmetros geométricos do objeto.

Dentre as diversas áreas de aplicação envolvendo a reconstrução computadorizada destacam-se a medicina, a biologia, a microscopia, a geologia. Em geologia, por exemplo, superfícies de terrenos podem ser construídas a partir de mapas topográficos, imagens de satélites ou através de aerofotogrametria [47]. Na medicina, atividades como planejamento de tratamentos através de radiação, estudos de características morfológicas de órgãos, planejamento cirúrgico e auxílio em diagnósticos têm recebido cada vez mais contribuições com a utilização das técnicas de reconstrução computadorizada [2, 11, 15, 28, 36, 80]. Em microscopia, o estudo de estruturas celulares, muitas vezes, envolve a análise de um grande número de imagens bidimensionais correspondendo a cortes seriados da região de interesse.

Um método ainda muito usual de se realizar a análise de estruturas 3D é a observação seqüencial das imagens bidimensionais, cujas formas e dimensões são individualmente avaliadas e *mentalmente* reconstruídas pelo especialista. A aparência tridimensional do objeto pode ser obtida simplesmente pelo *empilhamento* ou sobreposição dos cortes na ordem correta, quando as distâncias entre cortes sucessivos forem pequenas. Entretanto, se os cortes não estiverem proximamente dispostos, os espaços vazios entre cada corte devem ser *preenchidos* ou interpolados para se formar o objeto. Este procedimento apresenta diversas limitações ocasionadas, principalmente, pelo grande número de imagens amostradas, pela dificuldade inerente ao processo manual de análise de estruturas complexas, pelo fato do resultado da reconstrução depender fortemente da experiência do especialista e pela baixa confiabilidade e precisão na determinação de medidas dos objetos sob investigação.

A disponibilidade de um método automático de reconstrução e visualização tem como objetivo a eliminação das limitações mencionadas, reduzindo significativamente o tempo de reconstrução de objetos. O desenvolvimento de uma ferramenta computadorizada para a reconstrução de objetos 3D envolve o conhecimento e a integração de diversas áreas, algumas já bastante estudadas e disseminadas, outras apenas em estágio inicial. Um sistema de reconstrução, conforme ilustrado na figura 1.1,

deve considerar basicamente três etapas principais: aquisição, pré-processamento e segmentação das imagens; reconstrução a partir das imagens segmentadas; visualização e extração de medidas geométricas dos objetos. A seguir, cada um destes tópicos será discutido sucintamente de modo a introduzir os principais conceitos abordados neste trabalho.

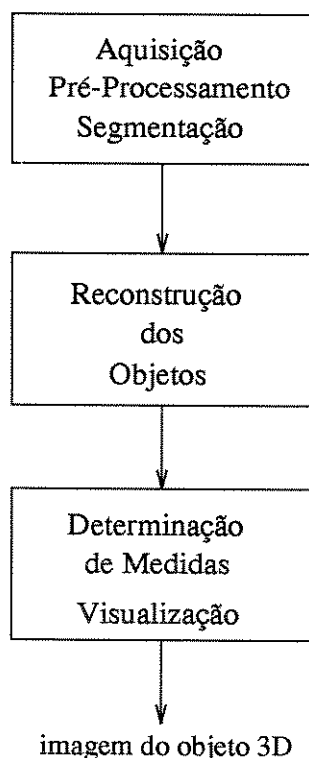


Figura 1.1 Etapas envolvidas no processo de reconstrução 3D.

A *aquisição* das imagens pode ser realizada através de diferentes métodos e depende da aplicação em questão. Os principais meios de aquisição empregam *scanners*, câmeras de vídeo, tomógrafos, satélites, radares, etc, cujas imagens são posteriormente digitalizadas numa forma apropriada para processamento em computadores.

Muitas vezes o método de aquisição adotado não garante um alinhamento preciso das imagens, tornando-se necessário um *pré-processamento* para verificar e, eventualmente, corrigir a correspondência entre as imagens. A *segmentação* é o processo no qual as regiões de interesse são identificadas nas imagens e está baseada nos princípios de descontinuidade e similaridade dos níveis de cinza que determinam as fronteiras entre cada objeto. As principais técnicas de segmentação são a detecção de bordas, a detecção de regiões e a limiarização [29].

Uma vez segmentadas as imagens, pode-se realizar a *reconstrução* dos objetos, cujo objetivo é determinar a representação tridimensional que melhor aproxime o objeto. Vários métodos de reconstrução têm sido propostos na literatura e, basicamente, podem ser classificados em duas categorias principais: métodos baseados em volume e métodos baseados em superfície. Ao longo deste trabalho serão apresentados os métodos mais relevantes para a solução do problema, buscando-se esclarecer as principais vantagens e desvantagens de cada um.

Os resultados mais importantes obtidos pela reconstrução, sem dúvida, são a *determinação de características* do objeto tais como volume, área superficial, centro de massa, etc, e a *visualização* tridimensional de suas estruturas.

Com o emprego cada vez mais freqüente das técnicas de processamento de imagens em áreas bastante diversificadas, surge a necessidade de uma intensa cooperação entre os grupos de profissionais envolvidos nestas atividades. O Departamento de Engenharia de Computação e Automação Industrial da Faculdade de Engenharia Elétrica da UNICAMP vem trabalhando nos últimos anos na experimentação e capacitação de seu grupo através de diversas linhas de pesquisa relacionadas à área de Computação de Imagens.

A grande motivação deste trabalho está no fato de que a maior parte dos algoritmos existentes na literatura não resolve satisfatoriamente o problema, restringindo consideravelmente o número de aplicações em diversas áreas. Resultados incorretos são obtidos, por exemplo, quando os contornos possuem formas e tamanhos muito diferentes. Do mesmo modo, a presença de múltiplos contornos numa mesma seção transversal pode gerar ambigüidades quanto à ligação dos vértices que formarão as faces triangulares. A figura 1.2 ilustra as principais situações onde os algoritmos apresentam estas deficiências.

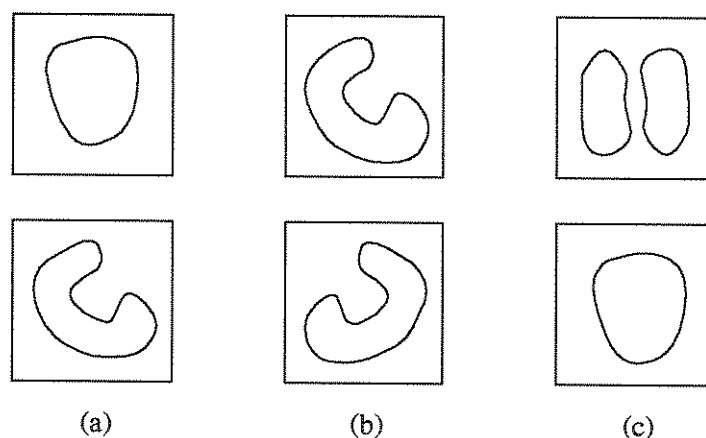


Figura 1.2 (a) Contornos com formas diferentes (b) Contornos com orientações diferentes (c) Múltiplos contornos em cada seção transversal.

O objetivo principal deste trabalho é apresentar os aspectos relevantes à reconstrução de objetos tridimensionais a partir de suas seções transversais e desenvolver uma metodologia que elimine os problemas mencionados acima, permitindo:

- fornecer uma solução para o problema de contornos com forma e orientação não similares;
- fornecer uma solução para o problema de ramificação de contornos, ou seja, presença de múltiplos contornos em cada seção transversal;
- operar automaticamente, ou seja, sem a necessidade de interação com o usuário durante o processo de reconstrução.

As principais contribuições deste trabalho em relação à reconstrução tridimensional estão na proposta de uma metodologia que resolva satisfatoriamente o problema.

Para tal, estabeleceu-se um novo critério de decisão de mapeamento de contornos pertencentes a diferentes seções transversais, onde realiza-se a identificação de quais contornos devem ser ligados entre si para a formação da superfície dos objetos. Um exemplo de possíveis situações que devem ser identificadas durante o mapeamento de contornos é ilustrado na figura 1.3.

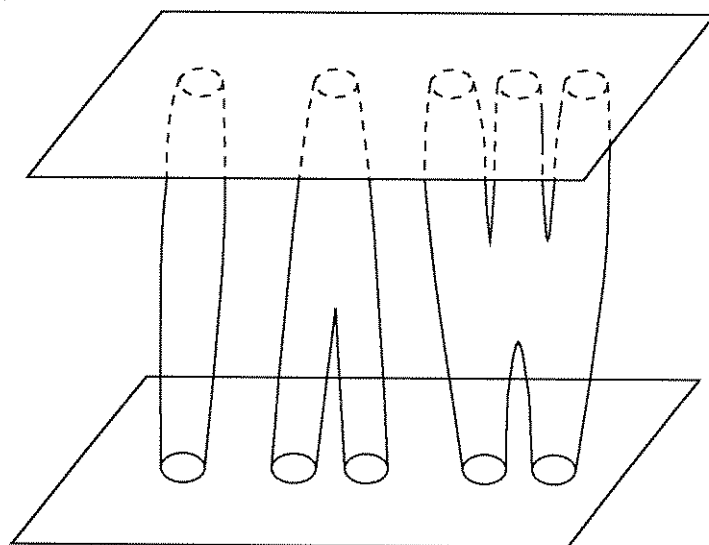


Figura 1.3 Contornos identificados e conectados entre duas seções transversais.

Outra contribuição realizada foi a implementação de uma ferramenta automática para a reconstrução 3D conforme a metodologia proposta, permitindo a avaliação dos resultados através da sua aplicação prática em objetos sintetizados e reais complexos. A plataforma utilizada foi uma estação de trabalho SUN - SparcStation 370 com sistema operacional UNIX, placa gráfica GXP com capacidade para representar cores em 24 bits e um monitor colorido com resolução de 1152 x 900 pixels. Os algoritmos foram escritos utilizando-se a linguagem C e as funcionalidades da ferramenta são ativadas através de uma interface gráfica operando sob o sistema X Window, o qual é um sistema gráfico multitarefa baseado em janelas, distribuído e independente do hardware.

Este trabalho está organizado como segue. O capítulo 2 apresenta as principais abordagens relativas à reconstrução encontradas na literatura enfatizando-se a técnica de triangulação, a qual é amplamente analisada e adotada ao longo deste trabalho.

O capítulo 3 apresenta uma proposta de metodologia para tratar satisfatoriamente o problema de reconstrução, eliminando as principais deficiências encontradas em grande parte dos métodos existentes.

No capítulo 4 é descrito um procedimento para a extração de medidas geométricas dos objetos reconstruídos através do algoritmo proposto, onde é feita uma estimativa da área superficial e do volume dos objetos. Os componentes relevantes à visualização dos objetos reconstruídos também são abordados, tais como iluminação, transparência, cor, eliminação de superfícies escondidas, entre outros, com o objetivo de se buscar imagens de alta qualidade.

No capítulo 5, a metodologia proposta é aplicada à reconstrução da superfície

externa de objetos sintetizados e reais, cujos resultados demonstram a validade dos estudos realizados. Imagens de cada objeto reconstruído são apresentadas utilizando diferentes técnicas de tonalização.

No capítulo 6, são feitos comentários finais a respeito dos aspectos principais que compõem este trabalho e são feitas algumas sugestões para futuros trabalhos de pesquisa na área.

A documentação da ferramenta de reconstrução implementada é descrita no apêndice A, onde encontram-se detalhes a respeito do hardware utilizado, interfaces com o usuário, funcionalidades disponíveis, organização dos dados, etc.

Finalmente, é fornecida uma extensa bibliografia referente aos métodos de reconstrução pesquisados e assuntos relacionados.

MÉTODOS DE RECONSTRUÇÃO

Os métodos de reconstrução encontrados na literatura podem ser classificados, basicamente, em duas abordagens principais: métodos baseados em volume e métodos baseados em superfície [79].

Os métodos baseados em volume estão, normalmente, associados à visualização de estruturas tridimensionais internas a um objeto e, devido a este fato, são frequentemente relacionados à área de Visualização Científica.¹ Estes métodos podem ser inseridos no contexto da reconstrução de objetos, à medida que também buscam uma representação volumétrica a partir de múltiplas fatias bidimensionais sucessivas. Um componente essencial na interpretação dos dados 3D é a habilidade em manipular e apresentar os dados de modo que características como forma, tamanho e cor dos dados possam ser facilmente visualizadas. Estes métodos demandam, em geral, grandes quantidades de memória para o armazenamento dos dados e cujas técnicas de *rendering* demandam alto tempo de processamento. Eles adquirem maior importância quando o espaçamento entre os cortes transversais é relativamente reduzido.

Os métodos baseados em superfície são melhor aplicados na reconstrução de superfícies externas dos objetos e há grande interesse em se determinar uma representação tridimensional dos objetos, possibilitando sua visualização bem como a extração de parâmetros geométricos dos objetos. Uma representação possível é obtida através da descrição das faces de cada objeto, cujo modelo é conhecido como Representação por Superfícies Limitantes (*Boundary Representation* ou simplesmente B-Rep). Os objetos são descritos não apenas por seus elementos geométricos mas também por sua topologia. Nos métodos baseados em superfície, a reconstrução dos objetos é realizada através da aproximação das superfícies entre contornos pertencentes às seções transversais dos objetos.

As técnicas de visualização mantêm uma estreita relação com o modelo utilizado para se representar o objeto e, portanto, diferem para os métodos de reconstrução baseados em volume e superfície.

As seções seguintes discutirão as características relevantes destas duas abordagens, porém com maior ênfase aos métodos baseados em superfície, já que a metodologia desenvolvida neste trabalho utiliza algoritmos de reconstrução pertencentes a esta classificação.

¹Classe de aplicações envolvendo o uso de técnicas de visualização de imagens para auxiliar o cientista a extrair informação a partir de dados experimentais ou simulados, facilitando a compreensão de sua estrutura.

2.1 Métodos Baseados em Volume

Dentre as principais áreas de aplicação utilizando os métodos baseados em volume, sem dúvida, a que teve um maior desenvolvimento foi a área médica. Graças ao aumento da velocidade de processamento e capacidade de armazenamento dos computadores e ao surgimento de técnicas de aquisição de dados como Tomografia Computadorizada por Raios-X, Ressonância Magnética, Tomografia por Emissão de Pósitrons e Ultrassonografia, estruturas internas complexas do corpo humano têm sido amplamente estudadas e melhor compreendidas ao longo das duas últimas décadas [10, 11, 28, 33, 34, 45, 48, 49, 53, 54, 62, 72].

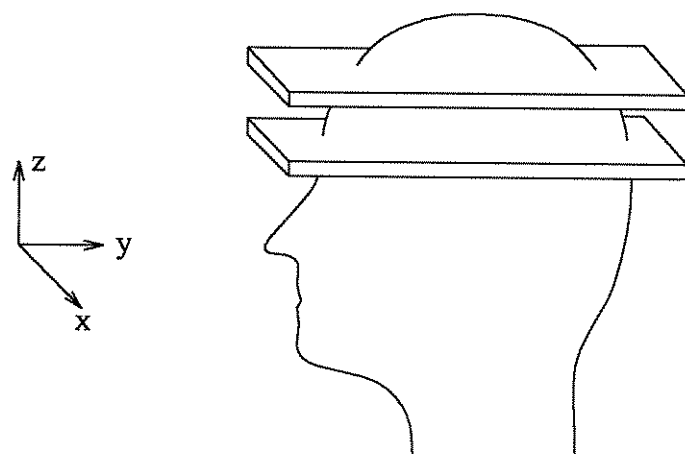


Figura 2.1 Cortes Transversais.

Os dados volumétricos obtidos pelos dispositivos (*scanners*) médicos são, em geral, um conjunto de fatias paralelas e uniformemente espaçadas, representando cortes transversais bidimensionais (2D) do órgão sob investigação. A figura 2.1 ilustra duas fatias paralelas ao plano xy , ou seja, os cortes transversais estão dispostos ao longo do eixo z . Cada fatia é formada por elementos de volume retangular, denominados *voxels*.²

A figura 2.2 ilustra voxels formando uma fatia, os quais são paralelepípedos retangulares. Associado a cada voxel há um valor, denominado densidade do voxel, representando uma propriedade física do objeto como, por exemplo, atenuação de raio-X, concentração de radiofármacos, índice de refração, velocidade, densidade do material, opacidade, etc.

Como já mencionado, um voxel é um elemento de volume retangular obtido quando o espaço é dividido por três conjuntos de planos paralelos, cada conjunto sendo ortogonal aos outros dois. É comum em Tomografia Computadorizada que estes planos não sejam igualmente espaçados, onde cada voxel possui igual tamanho nas dimensões x e y enquanto é ligeiramente maior na dimensão z , não constituindo, portanto, voxels cúbicos ou isotrópicos [28]. Isto pode causar distorções (em geral, *aliasing*) ao se tentar visualizar o objeto sob diferentes ângulos. Um modo de minimizar este problema é através da reamostragem das imagens originais [10], cujo

²O termo voxel é derivado de *volume element*, análogo ao termo pixel de *picture element* no caso 2D.

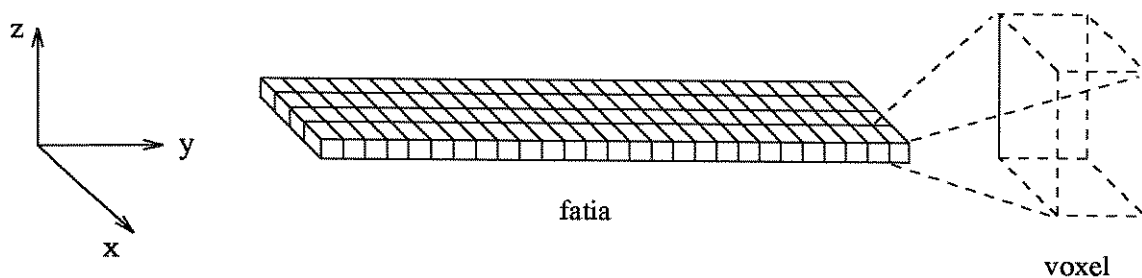


Figura 2.2 Voxels formando uma fatia.

objetivo é criar novas fatias intermediárias e, assim, melhorar a qualidade (maior precisão) das superfícies produzidas no processo de *rendering*. A técnica mais simples é conhecida como interpolação linear, a qual assume uma variação linear da densidade dos voxels na direção z .

Em análise de imagens 3D é importante a existência de uma representação geométrica ou modelo do objeto. Uma das representações que mais se adapta a este tipo de dados é obtida através da estrutura chamada *octree* [9, 12, 20, 44, 59, 64, 66, 74], a qual possibilita que o objeto possa ser eficientemente armazenado, facilitando sua posterior manipulação e visualização.

A característica essencial da representação através de *octrees* é a numeração hierárquica do espaço Euclidiano 3D em elementos de volume (voxels), os quais, sem perda de generalidade, podem ser considerados cúbicos. Deste modo, o espaço cúbico é subdividido em 8 subcubos (octantes) de igual volume e identificados segundo uma determinada ordem.

Cada um desses octantes representa ou uma região homogênea do espaço (densidade uniforme) ou uma região heterogênea. Os octantes heterogêneos são divididos novamente em suboctantes. Este procedimento é repetido até que hajam apenas octantes com densidade uniforme sendo que, no pior caso, um octante pode ser definido como um único voxel.

Este processo de subdivisão em octantes pode ser eficientemente representado através de uma árvore de grau 8, a qual é denominada *octree*. O nó raiz representa o objeto total; os nós terminais (nós folhas) representam os cubos nos quais nenhuma subdivisão foi necessária e os nós não terminais representam as regiões heterogêneas, ou seja, onde requereu-se uma subdivisão em 8 octantes.

Para melhor compreender a representação através de *octrees*, considere o objeto mostrado na figura 2.3.

Suponha que a convenção de numeração (ordem) dos octantes seja definida conforme ilustra a figura 2.4.

Considere também as densidades dos voxels como sendo VAZIO, PARCIAL ou CHEIO, dependendo se estes não pertencem, pertencem parcialmente ou pertencem inteiramente ao objeto, respectivamente.

A *octree* correspondente ao objeto anterior é ilustrada na figura 2.5.

As *octrees* exploram a coerência espacial inerentes aos objetos e mantêm uma descrição ordenada com respeito a um sistema de coordenadas de referência.

Existem diversas técnicas de visualização associadas aos métodos baseados em volume [7, 10, 23, 78, 79]. Nas técnicas de *rendering de volume binário*, a formação

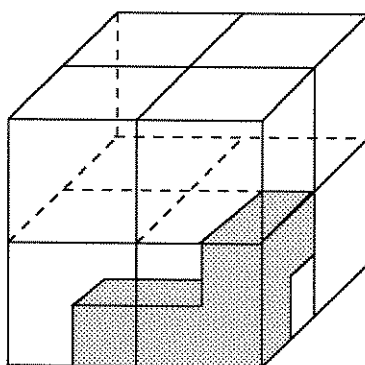


Figura 2.3 Objeto.

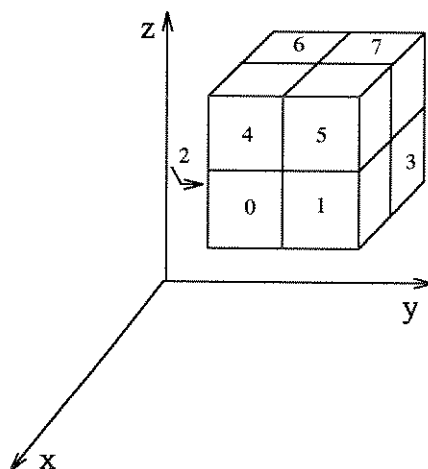


Figura 2.4 Convenção de numeração de um octante.

da imagem envolve, em geral, a projeção dos dados com remoção das superfícies escondidas e a tonalização. Nestas técnicas, o volume de dados é composto por voxels que podem ser opacos ou transparentes. A projeção procura identificar os voxels pertencentes à superfície do objeto que são visíveis ao observador. Métodos de *ray-tracing* [46, 65, 77] ou BTF [25] (*Back-to-Front*) podem ser utilizados para projetar cada voxel no dispositivo de saída. A tonalização tem como objetivo determinar a influência das fontes de luminosidade sobre as superfícies dos objetos no plano de visão. A imagem é gerada associando-se um tom de cinza a cada pixel, simulando a intensidade de luz na superfície do objeto.

Nas técnicas de *rendering de volume semi-transparente* [34], há a associação de uma cor e uma opacidade parcial a cada voxel, dando à imagem final um aspecto semi-transparente e colorido. São técnicas consideradas caras, pois consomem grandes quantidades de memória e o tempo de processamento cresce proporcionalmente ao número de voxels.

Muitas são as vantagens obtidas através dos métodos baseados em volume:

- correspondem mais proximamente ao formato nos quais os dispositivos médicos apresentam seus dados ao usuário;

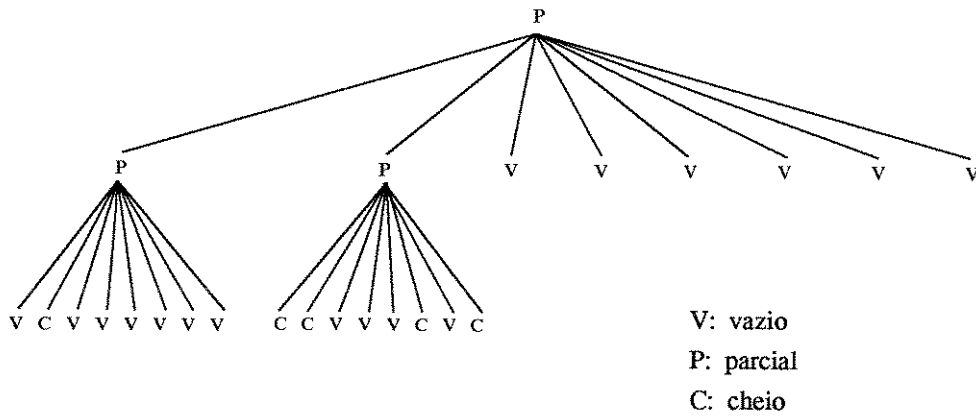


Figura 2.5 Representação *octree* do objeto.

- é fácil realizar medidas quantitativas dos objetos, por exemplo, volume, área da superfície, distância entre pontos;
- pode-se obter grande compressão de dados através da exploração de coerência espacial;
- muitos objetos podem ser suficientemente representados com a precisão de um pequeno voxel;
- permite operações simples tais como união, intersecção e diferença de objetos; translação, rotação e escalamento; localização de pontos no objeto;
- permite remoção de superfícies escondidas através de pesquisa segundo ordem especificada pelo ponto de observação [7]. Como exemplo, há o método BTF (*Back-To-Front*), onde a pesquisa é baseada na ordem decrescente de distância do observador [25].

2.2 Métodos Baseados em Superfície

Nos métodos baseados em superfície, a reconstrução dos objetos 3D também é realizada a partir de suas seções transversais. Estas seções transversais são obtidas interceptando-se o objeto por um número finito de planos, não necessariamente paralelos entre si, onde cada intersecção da superfície do objeto com os planos forma uma curva fechada, denominada de *contorno*.

A figura 2.6 mostra o seccionamento de um objeto através de dois planos e os contornos formados pela intersecção entre eles.

Estes contornos são, normalmente, definidos por uma sequência finita de pontos ou vértices segundo uma ordem arbitrária. A reconstrução do objeto pode ser realizada através da formação de superfícies entre os contornos, aproximando-se, assim, a superfície original do objeto.

Muitos métodos utilizam técnicas de interpolação não linear para reconstruir objetos, onde uma família de curvas paramétricas tais como Hermite-Coon, Bezier e B-Splines são utilizadas para representar as superfícies [8, 31, 47, 57]. Embora do ponto

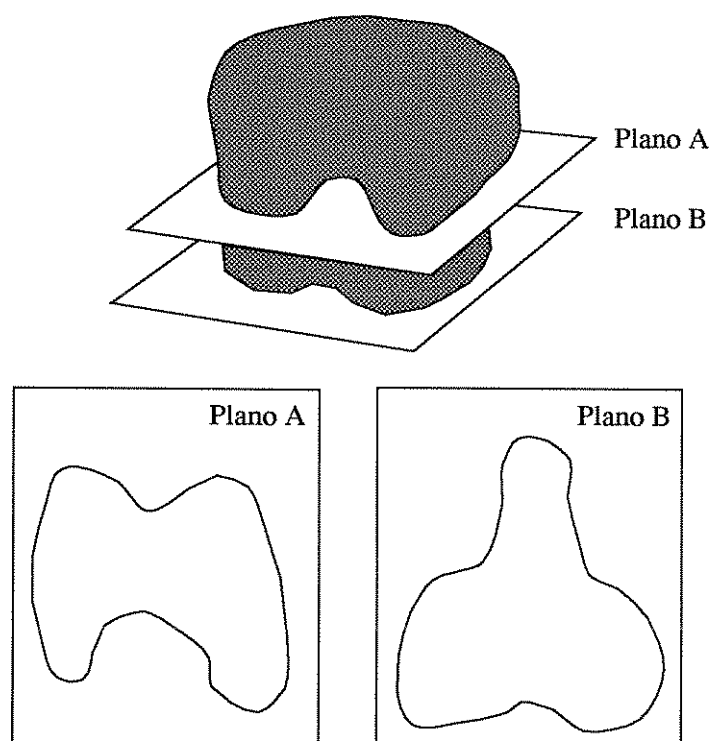


Figura 2.6 (a) Objeto seccionado por planos paralelos (b) contornos formados pela intersecção da superfície do objeto com o plano.

de vista matemático e computacional sejam muito complexos, estes métodos oferecem uma eficiente representação do objeto, fornecendo propriedades matemáticas muito úteis para análise, tais como localização e curvatura de pontos na superfície. Em Wu, Abel e Greenberg [83], por exemplo, cada contorno foi representado por B-splines uniformes e, posteriormente, as superfícies entre estes contornos foram aproximadas através de splines cardinais.

Em geral, as técnicas de interpolação linear são mais simples e um método muito conhecido é o de *triangulação*. O objetivo destes métodos é aproximar a superfície formada entre um conjunto de pontos 3D através de faces triangulares. A solução para o problema de triangulação pode ser abordada de duas maneiras distintas. Na primeira, um conjunto S é formado por m pontos distintos, $P_1, P_2, \dots, P_m$, os quais estão arbitrariamente distribuídos no espaço, sem qualquer conectividade assumida entre eles, como mostra a figura 2.7. Apesar de se mostrar mais genérica, a solução computacional deste problema é extremamente difícil de se obter. Em seu trabalho, Boissonnat [4, 5] baseou-se na triangulação de Delaunay [24], a qual possui a propriedade de formar triângulos cujos círculos que os circunscrevem não contêm nenhum outro vértice em seu interior.

Uma outra solução para o problema de triangulação ocorre quando os pontos em S estão distribuídos em contornos planares paralelos, como pode ser observado na figura 2.8. Esta particularização envolve um grande número de aplicações práticas e os algoritmos de triangulação a serem estudados ao longo deste trabalho buscam a aproximação da superfície entre pares de contornos consecutivos através de uma coleção de faces triangulares. A união destas faces triangulares, considerando-se

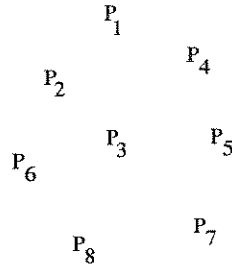


Figura 2.7 Pontos arbitrariamente distribuídos no espaço 3D.

todos os pares de contornos, resultará numa aproximação admissível da superfície original do objeto.

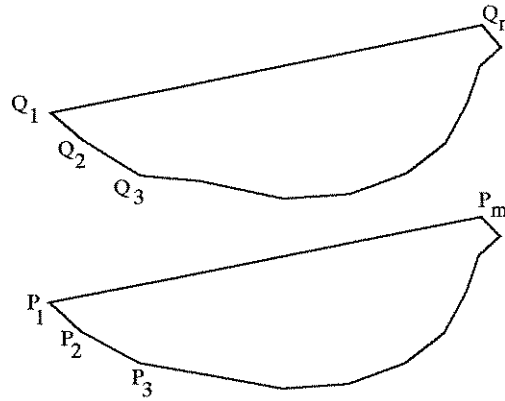


Figura 2.8 Pontos distribuídos ao longo de contornos planares paralelos.

O resultado obtido pelo método de triangulação de contornos depende fortemente do espaçamento entre as seções transversais e do processo de aquisição dos contornos. Este último compreende os problemas de seccionamento dos objetos, da detecção e afinamento das bordas e extração dos vértices dos contornos.

Outra técnica encontrada na literatura para realizar a reconstrução de superfícies de objetos é conhecida como interpolação elástica dinâmica [47, 48], onde um método de interpolação gera, iterativamente, uma seqüência de contornos intermediários entre dois contornos adjacentes. A idéia fundamental é determinar uma função que descreva a diferença posicional (distorção) dos vértices do contorno inicial em relação ao contorno final. Após a geração dos contornos intermediários, a reconstrução das superfícies pode ser realizada, por exemplo, através da aplicação de técnicas de triangulação. Dois exemplos ilustrando a geração de contornos intermediários são ilustrados na figura 2.9.

Outra técnica semelhante é conhecida como Modelo de Homotopia [70], onde as superfícies entre dois contornos adjacentes são formadas tomando-se como base uma função que transforma um dos contornos em outro. Sejam dois contornos representados por $f(x)$ e $g(x) : X \rightarrow \mathbb{R}^3$, onde $X = [x_0, x_1] \subset \mathbb{R}$. Por simplicidade, suponha o contorno inferior em $z = 0$ e o superior em $z = 1$. Além disto, $X = I = [0, 1]$. Então, a função F é denominada de homotopia entre f e g se $F(x, 0) =$

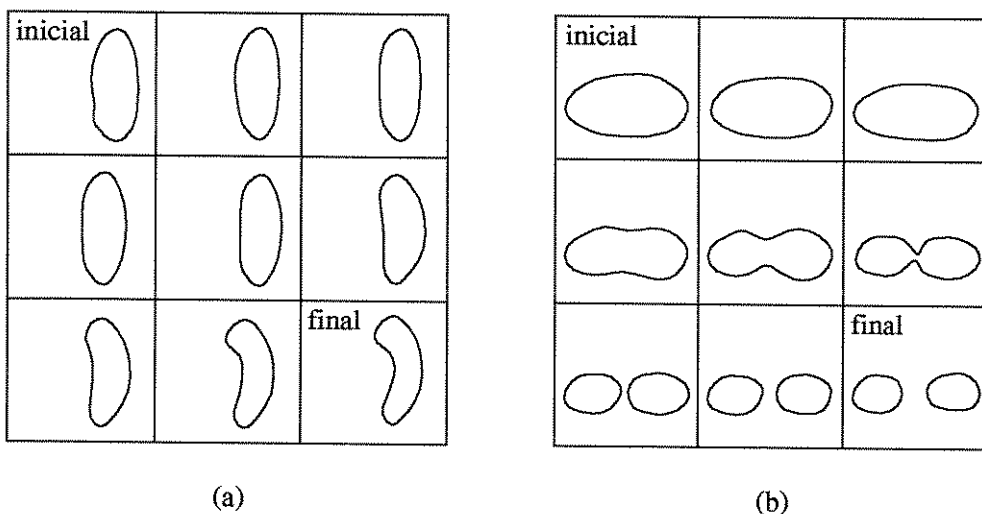


Figura 2.9 (a) Interpolação entre contornos não similares (b) Ramificação de um contorno em outros dois.

$f(x)$ e $F(x, 1) = g(x)$. Qualquer contorno em $z = t$ entre os contornos inferior e superior é dado por $F(x, t)$ para $x \in X$. Quando $F(x, t) = (1 - t)f(x) + tg(x)$, F é chamada de homotopia linear. A figura 2.10 mostra uma função F utilizada para transformar um contorno $f(x)$ em outro contorno $g(x)$. A geração da superfície entre contornos adjacentes pode ser formada conectando-se cada vértice em um contorno ao vértice correspondente em outro contorno através, por exemplo, de um algoritmo de triangulação.

Desde que os métodos de reconstrução desenvolvidos ao longo deste trabalho são baseados nas técnicas de triangulação, os principais conceitos envolvendo estas técnicas serão, a partir de agora, apresentados em detalhes.

2.2.1 Técnicas de Triangulação

O objetivo destas técnicas é encontrar uma superfície poliédrica aproximada entre pares de contornos consecutivos. Essa superfície é formada por uma coleção de faces triangulares, cuja união para todo par de contornos consecutivos resultará numa aproximação da superfície do objeto.

Considere dois contornos fechados C_1 e C_2 correspondendo à intersecção da superfície real do objeto com dois planos paralelos z_1 e z_2 , respectivamente. O contorno C_1 é descrito como uma seqüência de m pontos distintos $P_1, P_2, \dots, P_m$ e o contorno C_2 como uma seqüência de n pontos distintos $Q_1, Q_2, \dots, Q_n$. Ambos os contornos são ordenados no sentido anti-horário.

Um *segmento de contorno* é uma aproximação linear da curva conectando pontos consecutivos. Uma face triangular é composta de um único segmento de contorno e duas arestas conectando os pontos extremos do segmento de contorno a um ponto comum no contorno adjacente. A figura 2.11 ilustra as proposições acima.

O conjunto de faces triangulares representando uma superfície, por definição, satisfaz duas propriedades:

- cada segmento de contorno aparecerá em apenas uma face triangular. O con-

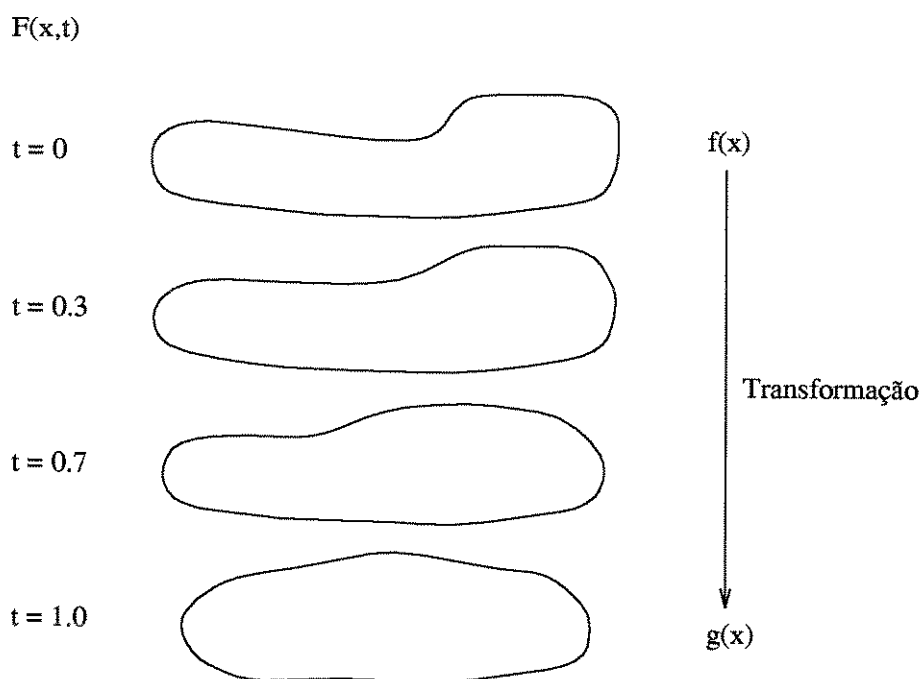


Figura 2.10 Exemplo de Homotopia.

junto conterá, portanto, exatamente $(m + n)$ faces;

- se uma aresta aparecer como aresta esquerda (direita) de alguma face triangular, ela aparecerá como aresta direita (esquerda) de exatamente uma outra face no conjunto.

Fuchs, Kedem e Uselton [26] mostraram que um conjunto de faces triangulares não consistente com estas propriedades não pode descrever completamente a superfície externa entre contornos planares adjacentes. Um conjunto de faces satisfazendo estas propriedades é chamado de *superfície aceitável*.

Existem, é claro, muitos conjuntos de faces triangulares $P_i P_{i+1} Q_j$ ou $P_i Q_j Q_{j+1}$ que satisfazem ambas as condições, sendo necessário, portanto, um critério adicional para se escolher as superfícies mais apropriadas.

Os algoritmos de triangulação existentes buscam encontrar a coleção de faces triangulares que resulte na melhor aproximação para a superfície segundo um determinado critério. Existem duas classes de métodos baseados na técnica de triangulação que podem ser destacadas conforme o tipo de critério adotado: os métodos ótimos e os métodos heurísticos.

Os métodos ótimos procuram determinar a melhor dentre todas as coleções de faces triangulares, baseando-se em critérios globais tais como maximização do volume ou minimização da área superficial. Entretanto, são métodos computacionalmente custosos.

Os métodos heurísticos baseiam-se em critérios locais que permitem aproximações admissíveis em relação ao caso ótimo, sendo computacionalmente mais baratos. Eles possuem grande aplicação quando a velocidade de computação é um elemento mais importante do que a obtenção de resultados ótimos.

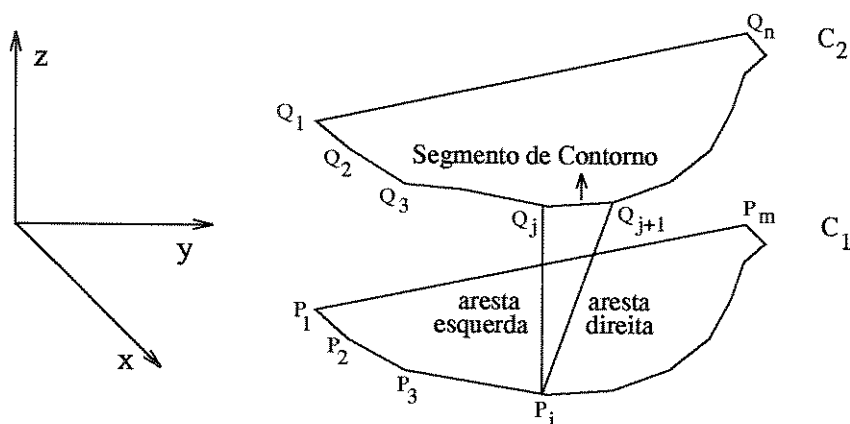


Figura 2.11 Face triangular formada entre contornos adjacentes.

2.2.1.1 Métodos Ótimos

Keppel [42] foi o primeiro autor a associar a técnica de triangulação e a teoria de grafos ao problema de reconstrução. Nesta abordagem, a construção de uma superfície entre dois contornos é equivalente a encontrar um caminho apropriado em um grafo dirigido. Através da associação de pesos às arestas deste grafo, um algoritmo clássico de busca de caminho mínimo [18, 38] pode produzir uma reconstrução ótima.

O conjunto de todas as superfícies aceitáveis para o par de contornos $P_1, P_2, \dots, P_m$ e $Q_1, Q_2, \dots, Q_n$ pode ser representado por um grafo dirigido $G[V, A]$. O conjunto de vértices de V corresponde a todas as possíveis conexões (arestas), $v_{i,j}$, entre dois pontos quaisquer P_i e Q_j dos contornos. O conjunto de arcos A corresponde a todas as possíveis faces triangulares. Assim, o arco $[v_{i,j}, v_{i,k}]$ de G , unindo dois vértices quaisquer $v_{i,j}$ e $v_{i,k}$, representa o triângulo $P_i P_j Q_k$.

Um arco $[v_{i,j}, v_{i,j+1}]$ unindo as colunas j e $j+1$ é chamado de horizontal. Um arco $[v_{i,j}, v_{i+1,j}]$ unindo as linhas i e $i+1$ é chamado de vertical. Tal grafo é apresentado na figura 2.12.

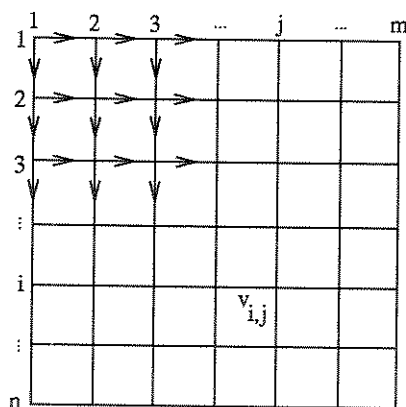


Figura 2.12 Grafo representando todas as superfícies aceitáveis entre 2 contornos.

Uma superfície aceitável é um caminho do vértice $v_{1,1}$ ao vértice $v_{m,n}$. O número de superfícies aceitáveis para um par de contornos contendo m e n pontos, respectivamente, é uma função $A[m, n]$, a qual pode ser recursivamente determinada como:

$$A[m, n] = A[(m-1), n] + A[m, (n-1)]$$

onde,

$$A[m, 1] = 1 \text{ para qualquer } m$$

$$A[1, n] = 1 \text{ para qualquer } n$$

Então,

$$A[m, n] = \frac{[(m-1) + (n-1)]!}{(m-1)!(n-1)!}$$

Pode-se observar que o valor da função acima cresce rapidamente com o aumento de m e n . Assim, se $m = n = 12$ vértices, por exemplo, há aproximadamente 10^6 possíveis combinações diferentes de superfícies triangulares. Devido a este elevado número de superfícies aceitáveis, torna-se proibitiva uma busca exaustiva para se encontrar a superfície *mais aceitável*, mesmo considerando-se pares de contornos simples.

Portanto, deve-se estabelecer um critério para escolha das superfícies ótimas dentre todas as existentes. Este critério define a *qualidade* de uma superfície aceitável, devendo se buscar a superfície de *melhor* qualidade.

Esta medida de qualidade pode ser descrita através da atribuição de custos ou pesos a cada arco do grafo G . Estes custos são escolhidos a partir de um determinado conjunto de valores.

Considere uma função $\varphi : A \rightarrow R$ definida como o mapeamento do conjunto de faces triangulares em números reais. Para cada arco a , associa-se um peso φ_a , que é o valor de φ para aquela face.

O caminho correspondente a uma superfície aceitável apresenta exatamente $(m+n)$ arcos $a_1, a_2, \dots, a_{m+n}$. O caminho escolhido é tal que a função:

$$\phi = \sum_{k=1}^{m+n} \varphi_{a_k} \quad (1)$$

satisfaça uma determinada restrição pré-estabelecida.

Keppel definiu a superfície ótima entre dois planos paralelos como sendo uma superfície aceitável tal que a função ϕ , conforme equação (1), seja máxima. Assim, segundo seu critério, a função ϕ corresponde ao volume englobado por uma superfície aceitável e a superfície ótima é aquela cujo volume é máximo.

Entretanto, Keppel não desenvolveu um algoritmo geral que encontrasse um caminho ótimo dentre todas as superfícies aceitáveis, mas apresentou uma solução baseada em critérios locais.

Por outro lado, Fuchs, Kedem e Uselton apresentaram uma solução geral para encontrar uma superfície ótima entre dois contornos. A função ϕ foi definida como a área lateral de uma superfície aceitável e a superfície ótima como sendo aquela cuja área é mínima.

2.2.1.1.a Método de Fuchs, Kedem e Uselton

Neste método foram introduzidas várias definições importantes. Um arco no grafo G é incidente *do* vértice correspondente à aresta esquerda da face e é incidente *ao* vértice correspondente à aresta direita da face triangular.

Qualquer conjunto de faces triangulares pode ser visto como um subgrafo de G , cujos arcos correspondem às faces entre contornos. Para qualquer subgrafo S de G , denota-se por $\text{indegree}_s(v)$ e $\text{outdegree}_s(v)$ o número de arcos em S que são incidentes *ao* vértice v e *do* vértice v , respectivamente.

Os subgrafos de G correspondendo às superfícies aceitáveis são referidos como subgrafos aceitáveis. Um subgrafo aceitável deve satisfazer um conjunto de condições necessárias e suficientes. Seja S um subgrafo de G . Então, S corresponde a uma superfície aceitável se, e somente se, S contém exatamente um arco horizontal entre duas colunas adjacentes e exatamente um arco vertical entre quaisquer duas linhas adjacentes e se S é Euleriano. Um grafo dirigido é Euleriano se, e apenas se, para qualquer vértice v de S , $\text{indegree}_s(v) = \text{outdegree}_s(v)$.

Lembrando que o número de superfícies aceitáveis é muito elevado, torna-se necessário estabelecer um critério para a escolha das superfícies que são ótimas. Este critério é descrito através da atribuição de custos a cada arco do grafo G e o custo de um caminho é definido como a soma dos custos dos arcos pesquisados no grafo. Uma superfície ótima é aquela correspondendo ao caminho de custo mínimo.

A busca de caminhos de custo mínimo em grafos tem sido amplamente estudada e pode ser encontrada em [38].

Dentre as principais soluções estão:

- (i) para um par de vértices (u, w) no grafo, encontrar um caminho de custo mínimo de u a w ;
- (ii) para todos os pares de vértices (u, w) no grafo, encontrar um caminho de custo mínimo de u a w .

No problema em questão, pretende-se encontrar um caminho de custo mínimo π , o qual inicia no vértice $v_{i,1}$ e termina no vértice $v_{m+1,n}$, $1 \leq i \leq m$. Um caminho de custo mínimo de $v_{i,1}$ a $v_{m+1,n}$ é definido como $\pi[i]$, $1 \leq i \leq m$. O caminho desejado π , correspondendo a uma superfície ótima, pode ser obtido primeiramente buscando-se $\pi[1], \pi[2], \dots, \pi[m]$ e, então, tomando-se um caminho de custo mínimo entre estes m caminhos.

Deste modo, uma terceira solução para o problema seria:

- (iii) para vários pares de vértices (u, w) no grafo, encontrar um caminho de custo mínimo entre u a w .

No método de Fuchs, a solução (iii) foi implementada resolvendo-se m instâncias da solução (i). O número de passos requeridos para encontrar um caminho ótimo segundo este método foi da ordem de:

$$(2mn + m) \log_2 m + 3mn + m^2$$

onde m e n são o número de vértices pertencentes a um par de contornos.

Os métodos ótimos exigem algoritmos eficientes de pesquisa global em grafos, apresentando alto custo computacional. Aproximações destes métodos ótimos têm

apresentado resultados satisfatórios, sendo baseadas em decisões heurísticas locais e nunca requerendo mais que $(m + n)$ passos de triangulação entre pares de contornos.

2.2.1.2 Métodos Heurísticos

Em muitos casos, algum conhecimento da natureza do objeto sendo modelado é disponível, facilitando a utilização dos métodos heurísticos de triangulação. Estes métodos podem ser descritos de modo similar aos algoritmos ótimos no sentido de que ainda buscam um caminho aceitável de $v_{1,1}$ a $v_{m,n}$, porém não requerendo que este caminho seja mínimo ou máximo com relação a um determinado critério. A função custo deve ser baseada apenas em restrições locais e ser computacionalmente barata.

Alguns métodos heurísticos serão apresentados nas seções seguintes.

2.2.1.2.a Aproximação de Keppel

Esta aproximação, já mencionada nos métodos ótimos, utiliza como critério de triangulação a maximização do volume entre pares de contornos.

Definiu-se uma função φ_h para os arcos horizontais e outra função φ_v para os arcos verticais. Associado a cada arco horizontal $[v_{i,j}, v_{i,j+1}]$ há um tetraedro,

$$T_{i,j}^h \{P_i P_{i+1} Q_j O_q\}$$

onde O_q é qualquer ponto situado no interior do contorno superior. O valor de φ_h para este arco é definido como sendo o volume de $T_{i,j}^h$.

Analogamente, cada arco vertical $[v_{i,j}, v_{i+1,j}]$ possui um tetraedro correspondente,

$$T_{i,j}^v \{P_i Q_j Q_{j+1} O_p\}$$

onde O_p é qualquer ponto situado no interior do contorno inferior. O valor de φ_v para este arco é definido como sendo o volume de $T_{i,j}^v$. Um exemplo mostrando as faces triangulares entre dois contornos adjacentes e seu grafo correspondente é apresentado na figura 2.13.

O caminho de uma superfície aceitável terá m arcos horizontais, $h_1, h_2, \dots, h_m$ e n arcos verticais, $v_1, v_2, \dots, v_n$.

Deste modo,

$$\phi = \sum_{i=1}^m \varphi_{h_i} + \sum_{j=1}^n \varphi_{v_j}$$

Para qualquer vértice $v_{i,j}$, deve-se calcular:

$$\varphi_h = \text{volume}(T_{i,j}^h) \quad \text{e} \quad \varphi_v = \text{volume}(T_{i,j}^v)$$

Se $\varphi_h > \varphi_v$, considerar o arco horizontal, ou seja, utilizar o vértice $v_{i,j+1}$. Caso contrário, considerar o arco vertical, através do vértice $v_{i+1,j}$. Desta maneira, a função ϕ pode ser localmente maximizada em exatamente $(m + n)$ passos.

Um fato importante é que o critério de triangulação de Keppel, o qual maximiza o volume formado entre pares de contornos adjacentes, deve ser aplicado apenas

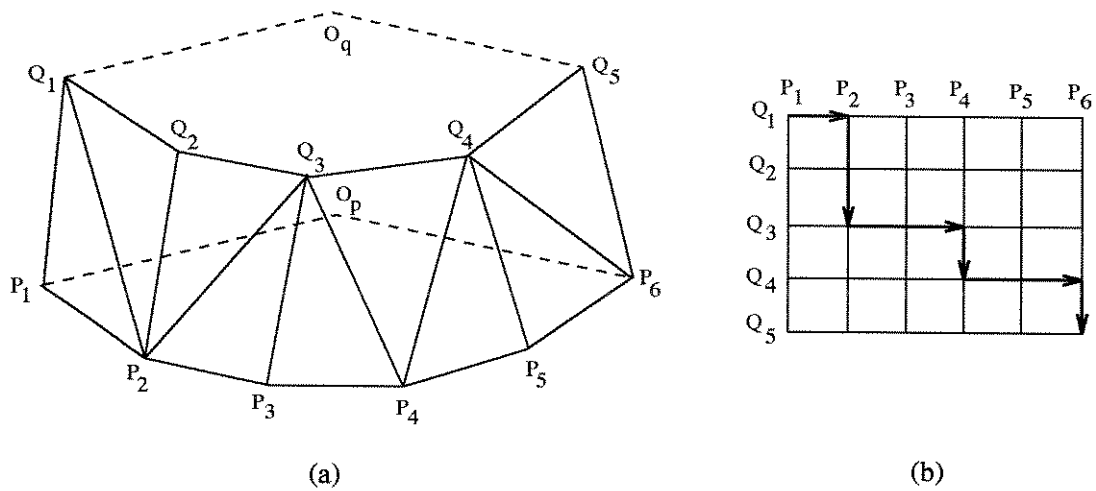


Figura 2.13 Exemplo de (a) associação de triangulação com (b) teoria de grafos.

em seqüências de pontos formando segmentos de contorno exclusivamente convexos. Para os segmentos côncavos, o algoritmo busca *minimizar* o volume formado pelos tetraedros. Conseqüentemente, deve haver uma decomposição dos contornos em conjuntos côncavos e convexos antes da aplicação do método.

Um exemplo é dado na figura 2.14 abaixo e discutido em seguida.

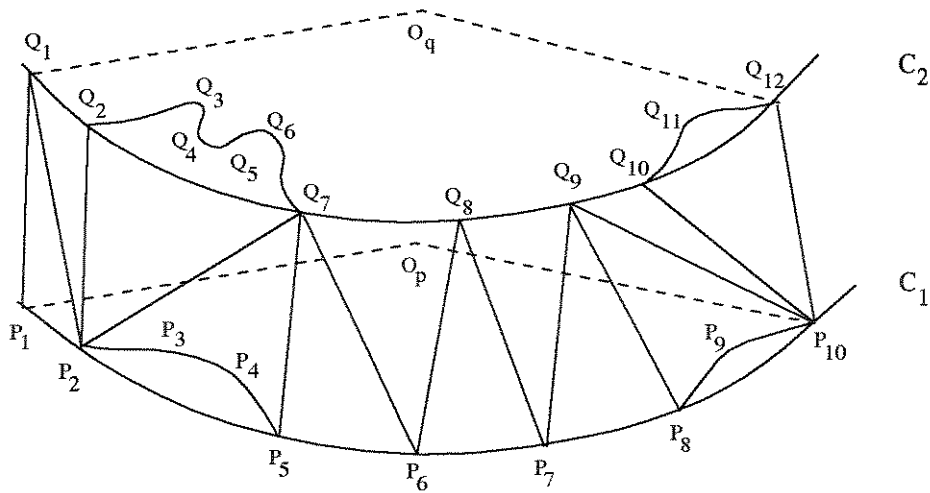


Figura 2.14 Exemplo de contornos apresentando segmentos convexos e côncavos.

O contorno C_1 é definido através do conjunto de vértices $S_0 = \{P_1, P_2, \dots, P_{10}\}$ e o contorno C_2 pelo conjunto de vértices $S_0 = \{Q_1, Q_2, \dots, Q_{12}\}$. O primeiro passo do algoritmo de triangulação considera, inicialmente, apenas os subconjuntos convexos $S_1^P = \{P_1, P_2, P_5, P_6, P_7, P_8, P_{10}\}$ e $S_1^Q = \{Q_1, Q_2, Q_7, Q_8, Q_9, Q_{12}\}$. O grafo G associado a estes conjuntos é apresentado na figura 2.15.

O segundo passo manipula os segmentos de contorno que foram omitidos anteriormente. A figura 2.14 ilustra os dois principais casos a serem distinguidos pelo algoritmo. No primeiro caso, os vértices P_2, P_5, Q_2 e Q_7 que delimitam os segmentos

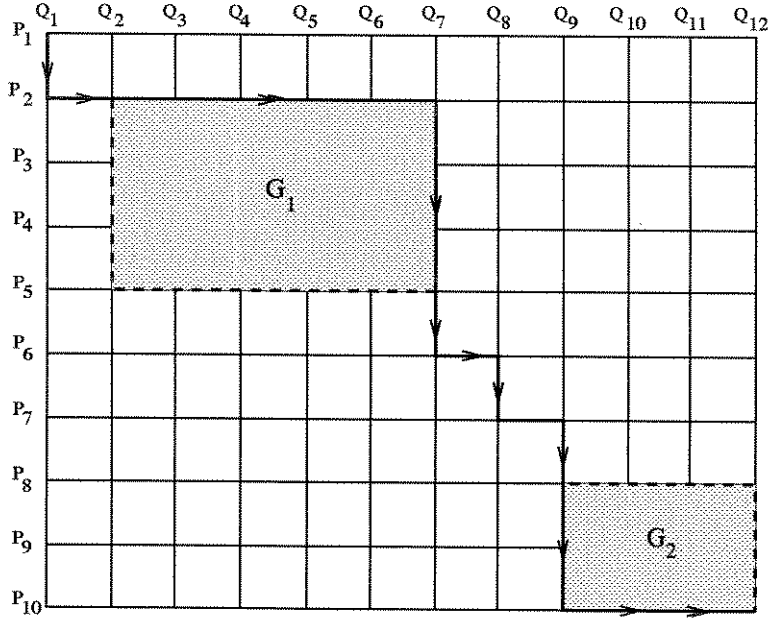


Figura 2.15 Grafo associado ao exemplo dado.

do contorno omitido são mutuamente ligados pelas arestas P_2Q_2 e P_5Q_7 . Assim, os subconjuntos $S_2^P = \{P_2, P_3, P_4, P_5\}$ e $S_2^Q = \{Q_2, Q_3, Q_6, Q_7\}$ devem ser triangulados. O caminho no grafo entre os vértices $v_{2,2}$ e $v_{5,7}$ delimita um subgrafo G_1 , no qual o poliedro $P_2P_3P_4P_5Q_2Q_3Q_6Q_7O_pO_q$ possui volume mínimo.

O segundo e menos intuitivo caso surge quando os segmentos entre Q_{10} e Q_{12} não estão conectados com os respectivos segmentos de P_8 a P_{10} . As ligações $P_{10}Q_9$ (ao invés de P_8Q_{10}) e $P_{10}Q_{12}$ delimitam os segmentos de contorno a serem manipulados no segundo passo de iteração do algoritmo. Os próximos subconjuntos côncavos são $S_3^P = \{P_8, P_9, P_{10}\}$ e $S_3^Q = \{Q_{10}, Q_{11}, Q_{12}\}$. O segundo subgrafo G_2 é delimitado pelos vértices $v_{8,9}$ (e não $v_{8,10}$ como no caso anterior) e $v_{10,12}$.

Finalmente, o último passo de iteração do algoritmo trata da triangulação do segmento convexo $S_4^Q = \{Q_3, Q_4, Q_5, Q_6\}$. Desde que não há nenhuma correspondência em S_0^P , a triangulação torna-se trivial neste caso.

2.2.1.2.b Heurística de Christiansen e Sederberg

Christiansen e Sederberg [13] observaram que os métodos heurísticos apresentavam melhor desempenho para pares de contornos com coerência de forma, tamanho e orientação. O critério estabelecido para realizar a triangulação entre contornos foi o critério da aresta de comprimento mínimo. Deste modo, dados dois contornos adjacentes C_{k-1} e C_k e assumindo que a aresta (P_i, Q_j) já está definida, a próxima aresta a ser selecionada será a de menor comprimento entre (P_{i+1}, Q_j) e (P_i, Q_{j+1}) . A figura 2.16 ilustra a utilização de tal critério de triangulação.

Esta classe de métodos fornece resultados satisfatórios quando os contornos C_{k-1} e C_k são convexos, ou quando eles possuem formas muito similares. Entretanto, quando C_{k-1} e C_k não satisfazem tais condições, a superfície resultante pode ficar distorcida. A figura 2.17 mostra dois contornos onde a triangulação baseada no

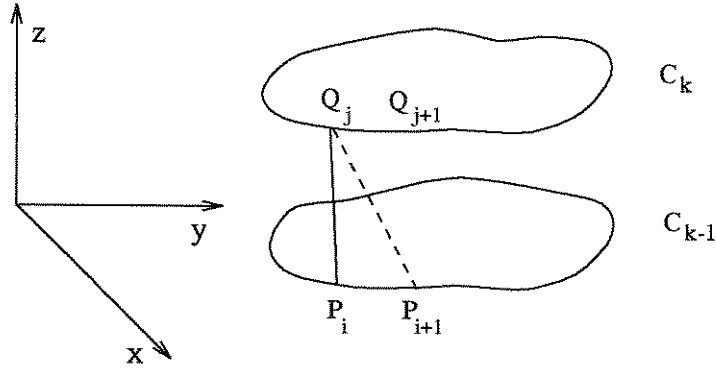


Figura 2.16 Triangulação através do critério da aresta de comprimento mínimo: face (P_i, P_{i+1}, Q_j) é escolhida em relação à (P_i, Q_j, Q_{j+1}) , pois $\overline{P_{i+1}Q_j} < \overline{P_iQ_{j+1}}$.

critério da aresta de comprimento mínimo apresenta resultados incorretos.

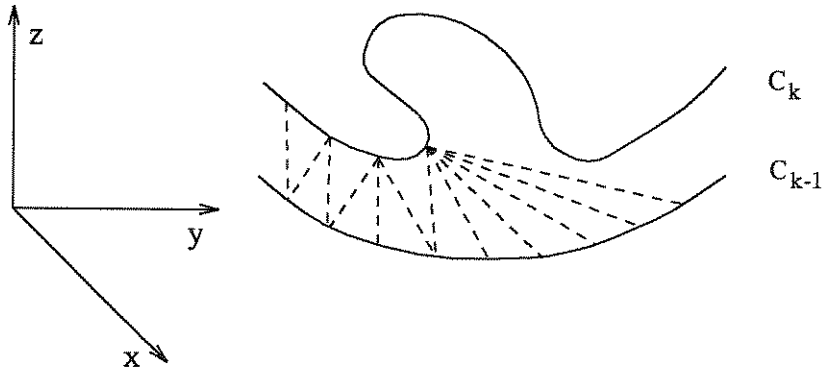


Figura 2.17 Triangulação entre dois contornos não similares.

Outra situação onde o critério da aresta de comprimento mínimo falha é quando os dois contornos não estão mutuamente centrados. Considere os contornos convexos C_{k-1} e C_k obtidos pela intersecção de dois planos paralelos com um cilindro circular disposto num ângulo diferente ao do eixo de simetria como mostra a figura 2.18.

A reconstrução da superfície entre os contornos C_{k-1} e C_k segundo este critério resultaria em um cone, como pode ser observado na figura 2.19.

Uma maneira de resolver este problema é mapear os contornos em um quadrado unitário antes do processo de triangulação. O mapeamento de um contorno é feito do seguinte modo:

1. definir uma janela retangular que inclua o contorno;
2. calcular ΔX , ΔY e os pontos centrais $\bar{X}$ e $\bar{Y}$, conforme ilustrado na figura 2.20;
3. mapear o contorno em um quadrado unitário centrado em $(0,0)$, através de escalamento e translação do contorno. As equações envolvidas são:

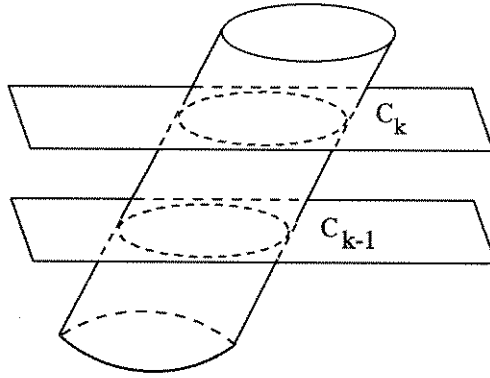


Figura 2.18 Cilindro seccionado por planos paralelos.

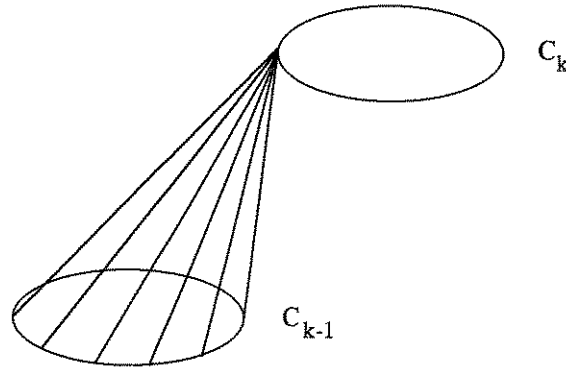


Figura 2.19 Exemplo em que o algoritmo da aresta de comprimento mínimo apresenta falhas.

$$X' = \frac{X - \bar{X}}{\Delta X} \quad \text{e} \quad Y' = \frac{Y - \bar{Y}}{\Delta Y}$$

O contorno resultante do mapeamento é mostrado na figura 2.21.

No critério de Christiansen e Sederberg, o peso φ_a associado a um arco a foi definido como o comprimento de cada aresta. Deste modo, a função ϕ definida na seção 2.2.1.1 é localmente minimizada a cada vértice selecionado durante o processo de triangulação.

2.2.1.2.c Heurística de Ganapathy e Dennehy

A coerência entre pares de contornos também pode ser obtida através de uma transformação de tal modo que o perímetro de cada contorno seja exatamente igual a 1.

Ganapathy e Dennehy [27] definiram o peso φ_a de uma face triangular como o comprimento do segmento de contorno pertencente à face dividido pelo perímetro do contorno no qual aquele segmento está situado.

A partir desta definição, dois fatos podem ser observados. Em primeiro lugar,

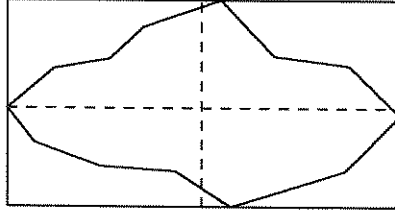


Figura 2.20 Janela retangular envolvendo o contorno.

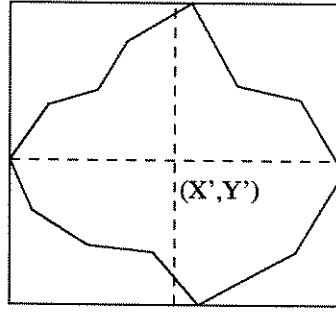


Figura 2.21 Contorno mapeado em um quadrado unitário.

para m arcos horizontais e n arcos verticais de uma superfície aceitável, tem-se que a soma dos custos horizontais ϕ_h e dos custos verticais ϕ_v são, respectivamente:

$$\phi_h = \sum_{i=1}^m \varphi_{a_{h_i}} = 1$$

$$\phi_v = \sum_{j=1}^n \varphi_{a_{v_j}} = 1$$

Além disto, $\phi = \phi_h + \phi_v$ é exatamente igual a 2 para todas as superfícies aceitáveis.

Enquanto ϕ tem valor igual para todas as superfícies aceitáveis, nem todas as superfícies são *igualmente aceitáveis*. A ordem na qual os segmentos de contorno são pesquisados é o fator determinante e tem significado físico. Para um par de contornos com coerência de forma e tamanho, a triangulação deveria proceder ao longo de cada contorno com aproximadamente a mesma taxa, eliminando a formação de superfícies exageradamente artificiais.

A heurística estabelece que faces triangulares devem ser adicionadas à superfície de tal modo que a diferença absoluta entre a soma dos custos horizontais e a soma dos custos verticais seja sempre minimizada. Quando uma superfície aceitável é obtida, a diferença é igual a zero. Entretanto, a recíproca não necessariamente é verdadeira.

A demonstração de um passo típico deste método é ilustrado na figura 2.22.

ϕ'_h representa a distância normalizada ao longo do contorno inferior já pesquisado. Analogamente, ϕ'_v é definido para o contorno superior. A próxima face triangular a ser adicionada à superfície pode ser ou $P_i P_{i+1} Q_j$ ou $P_i Q_j Q_{j+1}$.

$P_i P_{i+1} Q_j$ é escolhido se:

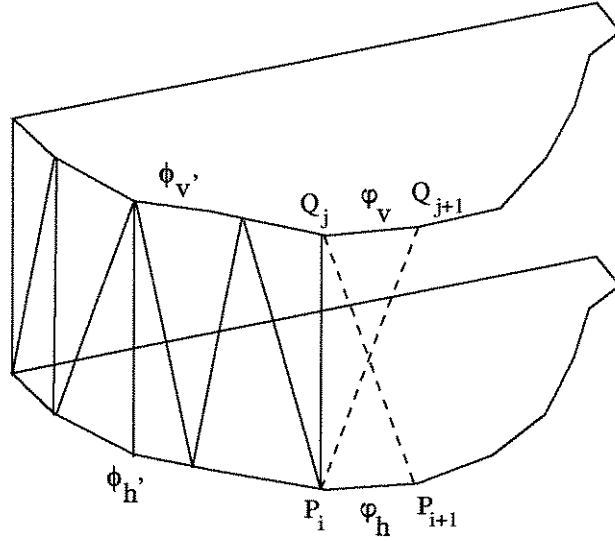


Figura 2.22 Passos do método proposto por Ganapathy e Dennehy.

$$|\phi'_h + \varphi_h - \phi'_v| < |\phi'_v + \varphi_v - \phi'_h|$$

Neste caso, i é incrementado e procede-se ao longo do contorno inferior. Caso contrário, $P_i Q_j Q_{j+1}$ é escolhido, j é incrementado e procede-se ao longo do contorno superior. Este método requer exatamente $(m+n)$ passos para realizar a triangulação completa entre cada par de contornos.

O método de Ganapathy e Dennehy possui algumas vantagens em relação aos de Fuchs e Keppel discutidos anteriormente. A primeira delas é que a decisão de triangulação local é baseada não apenas nos comprimentos dos segmentos de contornos envolvidos a cada passo, mas também nas distâncias já pesquisadas ao longo de cada contorno. Devido a este fato, esta heurística incorpora características de *memória* no processo de triangulação. Outra vantagem é que não necessita subdividir os contornos em segmentos côncavos e convexos antes de realizar a triangulação como no método de Keppel, além de requerer significativamente menos passos computacionais quando comparado com o método de Fuchs. Finalmente, esta heurística não impõe a restrição de que os contornos devem ser mutuamente centrados entre si ou mesmo localizados em planos paralelos.

2.2.1.2.d Heurística de Ekoule, Peyrin e Odet

O método de Ekoule, Peyrin e Odet [21] também baseou-se no critério da aresta de comprimento mínimo, a exemplo do método de Christiansen e Sederberg visto no item 2.2.1.2.b. Entretanto, não é necessário que cada face triangular seja criada *seqüencialmente* ao longo dos vértices dos contornos adjacentes, constituindo um critério de decisão de triangulação menos local.

Seja a seqüência ordenada de cortes transversais com níveis $z_1, z_2, \dots, z_N$. A triangulação é realizada entre o contorno C_{k-1} no nível z_{k-1} ao contorno C_k no nível z_k , onde C_{k-1} e C_k são contornos fechados. Suponha que o contorno C_{k-1}

seja definido por m pontos P_i , $1 \leq i \leq m$ e que o contorno C_k seja definido por n pontos Q_j , $1 \leq j \leq n$, ambos ordenados no sentido anti-horário. O conjunto de faces triangulares ligando C_{k-1} e C_k pode ser definido a partir de uma lista de arestas possuindo as seguintes propriedades:

- cada aresta tem um vértice em C_{k-1} e um vértice em C_k ;
- duas arestas consecutivas da lista têm um, e apenas um, vértice comum e definem uma face triangular (P_i, Q_j, Q_{j+1}) ou (P_i, P_{i+1}, Q_j) , como mostra a figura 2.23. A orientação da face pode ser definida, se necessário, por seu vetor normal $P_i P_{i+1} \otimes P_i Q_j$ para a face (P_i, P_{i+1}, Q_j) ou $P_i Q_{j+1} \otimes P_i Q_j$ para a face (P_i, Q_j, Q_{j+1}) , onde $\otimes$ denota o produto vetorial.

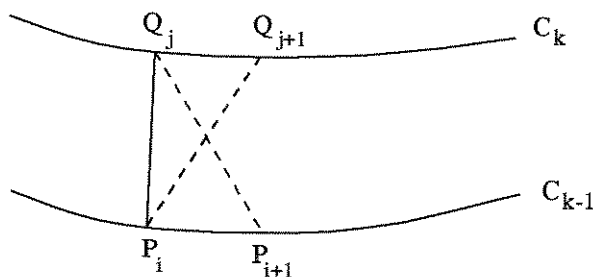


Figura 2.23 Dois possíveis caminhos para ligação dos contornos C_{k-1} e C_k .

A criação da lista ordenada de arestas pode ser decomposta em dois passos principais:

- cada vértice do contorno que possui o menor número de pontos é conectado ao vértice mais próximo no contorno adjacente. Por exemplo, se $m \leq n$, uma seqüência de pares de pontos é obtida pela ligação de cada ponto P_i ao mais próximo, denotado por $Q_{j(i)}$. Caso o par de pontos $(P_i, Q_{j(i)})$ seja selecionado, $Q_{j(i+1)}$ é pesquisado entre os sucessores de $Q_{j(i)}$, porém dentro de uma vizinhança local deste ponto.
- as arestas restantes são criadas pela ligação de cada ponto do conjunto $S_i = \{Q_j / j \in [j(i), j(i+1)]\}$ ou ao ponto P_i ou ao ponto P_{i+1} como segue. Seja Q_q o primeiro ponto de S_i tal que $d(P_i, Q_q) > d(P_{i+1}, Q_q)$. Os pontos livres Q_k de S_i , ou seja, os pontos Q_k ainda não ligados a P_i , para $j(i) < k < q$, são conectados a P_i . Os pontos Q_k restantes, $q \leq k < j(i+1)$, são conectados a P_{i+1} , se estes não estiverem ligados. Este procedimento é ilustrado na figura 2.24.

Quando os contornos adjacentes não são convexos ou não apresentam formas similares, as mesmas deficiências apresentadas na heurística de Christiansen e Sederberg podem ocorrer durante o processo de triangulação. Deste modo, é necessário também que haja um pré-processamento dos contornos antes da aplicação desta heurística.

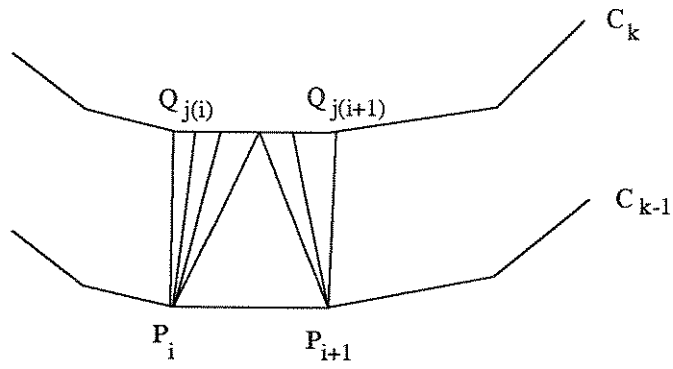


Figura 2.24 Procedimento de ligação entre contornos convexos.

Uma vantagem obtida em relação aos outros algoritmos de triangulação é que esta heurística não impõe a atribuição arbitrária dos pontos P_1 e Q_1 nos contornos inferior e superior, respectivamente, por onde o processo de triangulação deve necessariamente iniciar.

METODOLOGIA DE RECONSTRUÇÃO DE OBJETOS 3D

O desenvolvimento de uma metodologia para reconstrução de objetos 3D envolve o conhecimento e a integração de diversas áreas relativas à Computação de Imagens.

Infelizmente, a maior parte dos algoritmos encontrados na literatura possui limitações no tratamento de objetos complexos, sendo uma característica desejável o processamento preciso e totalmente automático, pois a interação humana durante a reconstrução pode ser proibitiva em muitas aplicações.

O objetivo deste capítulo é a proposição de uma metodologia de reconstrução de objetos 3D a partir de suas seções transversais baseada em técnicas de triangulação, cuja solução apresente resultados satisfatórios para grande número de aplicações desta natureza.

As deficiências mais comuns são encontradas quando os contornos possuem formas e tamanhos muito diferentes. A presença de múltiplos contornos numa mesma seção transversal também pode gerar dificuldades no processo de formação das faces triangulares.

Um sistema de reconstrução baseado nas técnicas de triangulação é composto de várias etapas. Inicialmente, as regiões de interesse do objeto devem ser identificadas durante a segmentação das imagens dos cortes transversais e os contornos bidimensionais que descrevem as fronteiras destas regiões devem ser extraídos. Na literatura são encontrados diversos trabalhos de extração de contornos utilizando técnicas de detecção de bordas [29]. Adicionalmente, algoritmos de afinamento de bordas podem ser aplicados de modo a reduzir o número total de pontos amostrados e, conseqüentemente, facilitar a reconstrução das superfícies.

O registro e alinhamento dos contornos devem também ser garantidos antes da reconstrução das superfícies, evitando possíveis casos de rotação e translação durante a aquisição das imagens dos cortes transversais. As técnicas convencionais especificam dois pontos homólogos localizados em posições pré-definidas ao longo de cada corte do objeto e realizam o registro baseando-se nestes pontos. Outras técnicas [1, 52] baseiam-se em regiões características de cada contorno, permitindo a determinação das transformações geométricas necessárias para estabelecer o registro de cada contorno do objeto. Em Kehtarnavaz, Simar e Figueiredo [40], os contornos são descritos através de uma representação sintático-semântica de alto nível, em contraste com a descrição convencional consistindo de um conjunto de vértices. Esta representação leva em conta as regiões de alta similaridade dos contornos, constituindo a base para o processo de triangulação.

Uma vez que os contornos de todos os cortes transversais tenham sido extraídos e registrados, a reconstrução da superfície entre eles pode ser realizada.

A metodologia descrita neste capítulo foi implementada e amplamente aplicada tanto em objetos sintetizados quanto em objetos reais complexos. A aquisição, pré-processamento e segmentação das imagens foram realizados através de pacotes desenvolvidos pelo próprio grupo de Computação de Imagens do DCA/FEE e do pacote Khoros [43], o qual é um ambiente aberto para tratamento e visualização de imagens desenvolvido pela Universidade do Novo México, Albuquerque, EUA. Facilidades para detecção de bordas, crescimento de regiões, edição de imagens, conversão entre formatos de imagens, rotinas de impressão estão disponíveis nestes pacotes. A visualização dos objetos reconstruídos foi realizada através do método ScanLine, o qual faz parte do ambiente PROSIM (Prototipação e Síntese de Imagens Foto-Realistas) também desenvolvido inteiramente pelo grupo do DCA/FEE.

As contribuições introduzidas à resolução do problema de reconstrução serão discutidas em detalhes ao longo deste capítulo.

3.1 Reconstrução da Superfície entre Contornos

A reconstrução do objeto é realizada através da aproximação das superfícies entre os contornos existentes nas seções transversais. O método utiliza um algoritmo heurístico baseado na técnica de triangulação, cuja superfície do objeto é formada por uma coleção de faces triangulares para todo par de contornos consecutivos.

Objetos complexos podem apresentar muitos contornos em cada seção transversal, havendo necessidade de se identificar quais contornos devem ser ligados entre si para a formação das superfícies dos objetos. Deste modo, precedendo a triangulação, o método de reconstrução deve realizar o chamado *mapeamento* dos contornos, no qual é feita esta identificação de ligação entre os contornos.

Três possíveis situações podem ocorrer durante o mapeamento de contornos, conforme ilustra a figura 3.1. O caso mais simples é a ligação de um contorno em cada seção transversal, denominado *caso 1:1*. A segunda situação, denominada *caso 1:n*, caracteriza-se pela ligação de um contorno em uma fatia com vários contornos em outra fatia. Finalmente, o *caso n:m* considera a ligação de múltiplos contornos em ambas as seções transversais.

A seguir, as considerações relativas à reconstrução de contornos para cada uma destas situações serão discutidas separadamente.

3.1.1 Caso 1:1

Esta situação é caracterizada pela ligação de apenas um contorno em cada seção transversal.

O algoritmo de triangulação utilizado neste trabalho é heurístico, pois a busca de uma solução ótima exigiria maior tempo de resposta e maior quantidade de memória. O algoritmo utilizado foi o de Ekoule, Peyrin e Odet [21], o qual é baseado no critério da aresta de comprimento mínimo. A escolha deste algoritmo deveu-se ao fato de que sua heurística não impõe a atribuição dos pontos iniciais nos contornos inferior e superior, por onde o processo de triangulação deve obrigatoriamente iniciar a geração das faces triangulares. Como já mencionado no capítulo 2, a classe de métodos baseada no critério da aresta de comprimento mínimo possui limitações

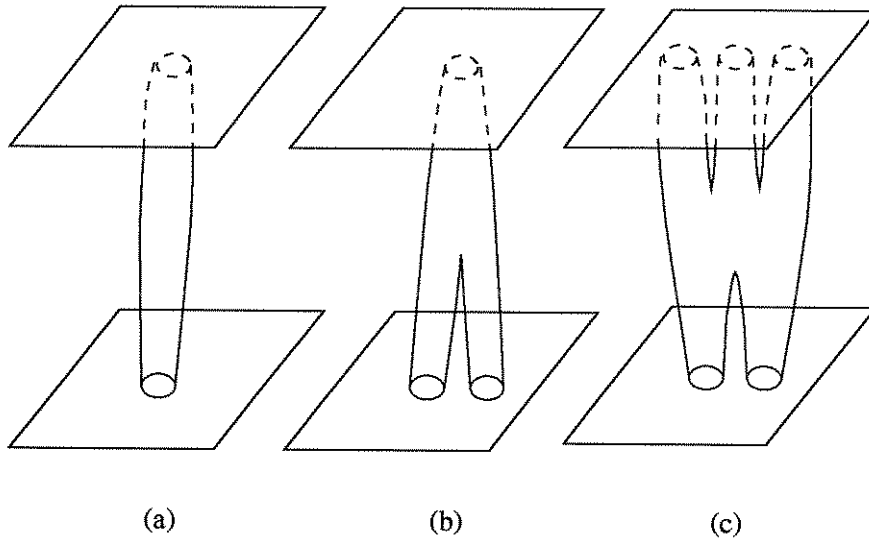


Figura 3.1 Ligação entre contornos (a) caso 1:1 (b) caso 1:n (c) caso n:m.

quanto à manipulação de contornos que não sejam convexos.

Primeiramente, será descrita uma técnica para resolver o problema de triangulação no caso simples de dois contornos convexos. Em seguida, é feita uma generalização de modo a atender qualquer tipo de contorno.

3.1.1.1 Caso 1:1 entre Contornos Convexos

O algoritmo de Ekoule, Peyrin e Odet pode ser diretamente aplicado quando os dois contornos são convexos. Entretanto, para garantir o correto funcionamento do critério da aresta de comprimento mínimo é necessário que os contornos estejam mutuamente centrados. A solução proposta neste trabalho difere da apresentada por Christiansen e Sederberg: os contornos também são mapeados em um quadrado unitário, porém com o centro do quadrado coincidindo com o centróide de cada contorno.

O cálculo do centróide dos contornos foi baseado no trabalho de Cook [15], o qual faz uso dos momentos de uma região delimitada por um arco sobre os eixos x e y . Estes momentos foram calculados através do teorema de Green onde, dado um contorno formado pelos vértices $P_{(x_1, y_1)}, P_{(x_2, y_2)}, \dots, P_{(x_m, y_m)}$, o momento em relação ao eixo x é dado por:

$$M_x = \sum_{i=1}^m \Delta y_i (x_i y_i + \frac{1}{2}(\Delta x_i y_i + \Delta y_i x_i) + \frac{1}{3} \Delta y_i \Delta x_i)$$

$$\text{onde } \Delta x_i = x_{i+1} - x_i \\ \Delta y_i = y_{i+1} - y_i$$

Analogamente,

$$M_y = - \sum_{i=1}^m \Delta x_i (x_i y_i + \frac{1}{2}(\Delta x_i y_i + \Delta y_i x_i) + \frac{1}{3} \Delta y_i \Delta x_i)$$

A área de cada contorno $P_1, P_2, \dots, P_m$ é calculada considerando-se a soma sucessiva dos $(m-2)$ triângulos formados pelos vértices P_1, P_{i+1}, P_{i+2} , $1 \leq i \leq m-2$, dada pela expressão:

$$A = \frac{1}{2} \sum_{i=1}^{m-2} \overline{P_1 P_{i+1}} \otimes \overline{P_1 P_{i+2}}$$

onde $\otimes$ denota o produto vetorial.

Deste modo, o cálculo do centróide (C_x, C_y) pode ser realizado por:

$$C_x = \frac{M_y}{A} \quad \text{e} \quad C_y = \frac{M_x}{A}$$

A partir do centróide acima, os demais vértices que formam o contorno podem ser mapeados no quadrado unitário e, em seguida, a triangulação do par de contornos pode ser realizada.

3.1.1.2 Caso 1:1 entre Contornos não Convexos

A estratégia utilizada no trabalho para garantir a manipulação de contornos côncavos foi evitar a introdução de critérios mais complexos ao método de triangulação. Em contrapartida, deve haver uma etapa de pré-processamento destes contornos de modo a reduzir o problema ao de triangulação entre contornos convexos já discutido.

O objetivo do passo de pré-processamento é, então, transformar qualquer contorno C apresentando concavidades em um contorno convexo C' , sem perda da informação relativa à distribuição de pontos no contorno original.

Deste modo, a ligação entre dois contornos arbitrários poderá ser realizada através do procedimento de triangulação entre contornos convexos já descrito na seção 3.1.1.1. O método baseia-se na hipótese de que os *convex hull's* dos contornos, ou seja, o menor conjunto convexo contendo uma sequência de vértices do contorno [60, 71, 76], são similares em forma e orientação. A transformação do contorno C em um contorno convexo é realizada através da projeção de cada vértice de C em seu *convex hull*.

Para ilustrar a idéia do procedimento de projeção, considere os dois exemplos a seguir. A figura 3.2(a) representa dois contornos que apresentam uma concavidade. Neste caso, o algoritmo de triangulação deveria ligar todos os pontos da concavidade de um contorno aos pontos da concavidade do outro contorno. Na figura 3.2(b), troca-se o contorno inferior por um mais distorcido. Em ambos os casos, parece natural ligar todos os pontos localizados entre P_i e P_j (em um contorno) aos pontos localizados entre Q_i e Q_j (em outro contorno). A criação deste mapeamento pode ser analisada em termos de uma projeção, a qual tem de satisfazer a seguinte propriedade: a projeção dos contornos em seus respectivos *convex hull's* transforma os pontos localizados entre P_i e P_j próximos às projeções dos pontos localizados entre Q_i e Q_j .

Deste modo, o procedimento de projeção deve considerar as deformações locais do contorno. Para este propósito, as seções seguintes apresentarão uma maneira de descrever contornos em termos de um conjunto de componentes elementares convexos. O procedimento utilizado na ferramenta de reconstrução para verificar se

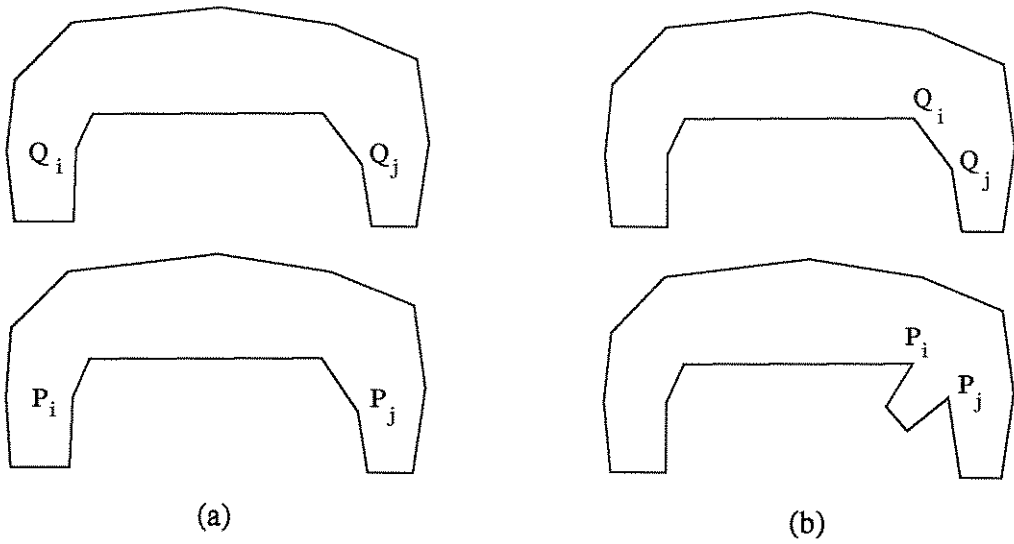


Figura 3.2 Triangulação entre contornos não convexos.

um determinado contorno é convexo ou côncavo baseou-se no algoritmo de Cyrus-Beck para recorte (*clipping*) bidimensional de objetos e pode ser encontrado em [63]. A análise deste algoritmo é realizada através do sinal obtido pelo cálculo do produto vetorial entre pares de segmentos de contorno (não colineares). As conclusões possíveis a partir desta análise são:

- a) o polígono é côncavo: alguns produtos vetoriais possuem sinal positivo enquanto outros possuem sinal negativo;
- b) o polígono é convexo: todos os produtos vetoriais possuem sinal positivo ou todos possuem sinal negativo.

A figura 3.3 ilustra um exemplo da aplicação do algoritmo acima em dois contornos.

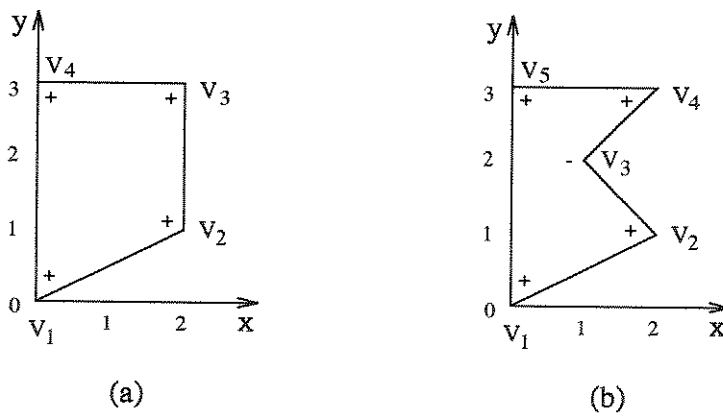


Figura 3.3 Contornos (a) convexo (b) côncavo.

3.1.1.2.a Decomposição dos Contornos

Um contorno fechado C é côncavo se ele difere de seu *convex hull* E . Seja $C^\circ = C$ um contorno definido pela seqüência ordenada de pontos $P_i, 1 \leq i \leq m$, e E° seu *convex hull* consistindo de uma subsequência ordenada de pontos de contorno $C^\circ - E^\circ = \{P_k/k \in \mathbf{K}\}$, onde $\mathbf{K}$ é um subconjunto ordenado de inteiros em $[1..m]$.

Se $E^\circ \neq C^\circ$, existem, no mínimo, dois pontos P_{j_1} e P_{i_1} que são sucessores em E° , mas não em C° . Conseqüentemente, o conjunto $\{P_k/k \in (j_1, i_1)\}$ não é vazio e está contido em $C^\circ - E^\circ$. A concavidade de 1ª ordem $C_{(i_1, j_1)}^1$ é o contorno definido pelo conjunto ordenado $(P_k/k = i_1, j_1)$. O exemplo a seguir ilustra uma decomposição de 1ª ordem. Seja C° o contorno $P_i, 1 \leq i \leq 15$, mostrado na figura 3.4.

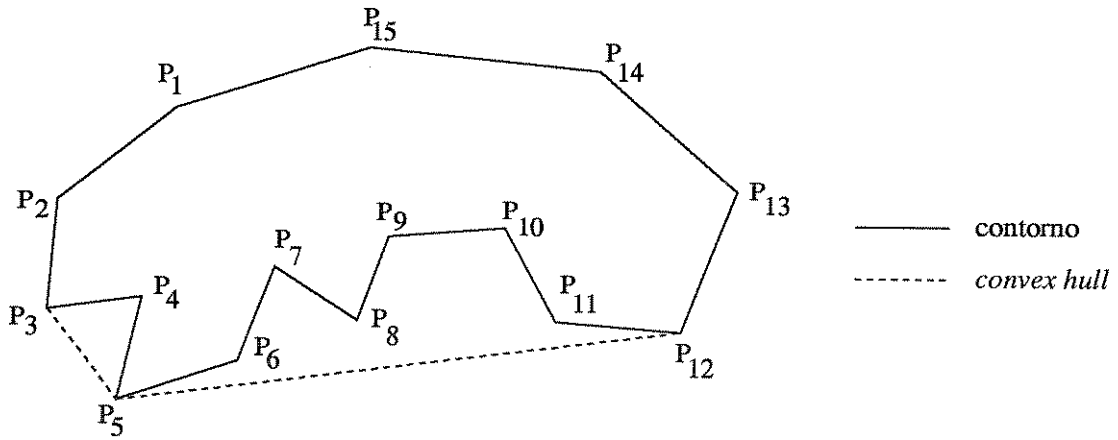


Figura 3.4 Exemplo de decomposição de contorno não convexo em concavidades elementares.

O algoritmo utilizado na determinação do *convex hull* baseia-se na eliminação sucessiva dos vértices responsáveis pelas concavidades do contorno, sendo estes identificados através do sinal resultante do produto vetorial mencionado anteriormente. Para o exemplo dado, o *convex hull* E° corresponde à seqüência de pontos $(P_1, P_2, P_3, P_5, P_{12}, P_{13}, P_{14}, P_{15})$. Desta forma, o contorno original C° possui duas concavidades de 1ª ordem, $C_{(3,5)}^1$ e $C_{(5,12)}^1$ definidas por:

$$C_{(3,5)}^1 = (P_3 P_4 P_5)$$

$$C_{(5,12)}^1 = (P_5 P_6 P_7 P_8 P_9 P_{10} P_{11} P_{12})$$

O processo pode ser repetido a fim de decompor cada concavidade de 1ª ordem assim definida. Sejam $E_{(i_1, j_1)}^1$ o *convex hull* de $C_{(i_1, j_1)}^1$. Da mesma maneira, se $E_{(i_1, j_1)}^1 = C_{(i_1, j_1)}^1$, a concavidade de 1ª ordem define um contorno convexo e, neste caso, sua decomposição termina; tal concavidade é chamada *elementar*. Caso contrário, se $E_{(i_1, j_1)}^1 \neq C_{(i_1, j_1)}^1$, então $C_{(i_1, j_1)}^1$ tem, no mínimo, uma concavidade de 1ª ordem de índice (i_2, j_2) . Tal concavidade, chamada de concavidade de 2ª ordem, é denotada por $C_{(i_1, j_1)(i_2, j_2)}^2$. Para o exemplo anterior, $C_{(5,12)}^1$ apresenta três concavidades de 2ª ordem, cujo *convex hull* $E^1 = (P_5 P_7 P_9 P_{10} P_{12})$:

$$C_{(5,12)}^2 = (P_5 P_6 P_7)$$

$$C_{(5,12)}^2 = (P_7 P_8 P_9)$$

$$C_{(5,12)}^2 = (P_{10}P_{11}P_{12}),$$

os quais são apresentados na figura 3.5.

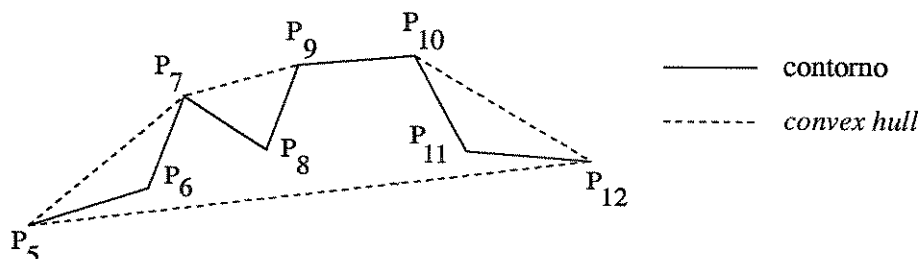


Figura 3.5 Concavidades de 2ª ordem para o exemplo dado.

Assim sucessivamente, realiza-se a decomposição de todas as concavidades de modo a resultar apenas em concavidades elementares. A decomposição final do contorno pode ser representada por uma estrutura hierárquica, cuja raiz é o contorno inicial de C^0 e cujos nós-folhas são as concavidades elementares de C^0 . No nível n da árvore, cada nó representa uma concavidade de ordem n . O contorno do exemplo anterior pode ser descrito hierarquicamente pela figura 3.6.

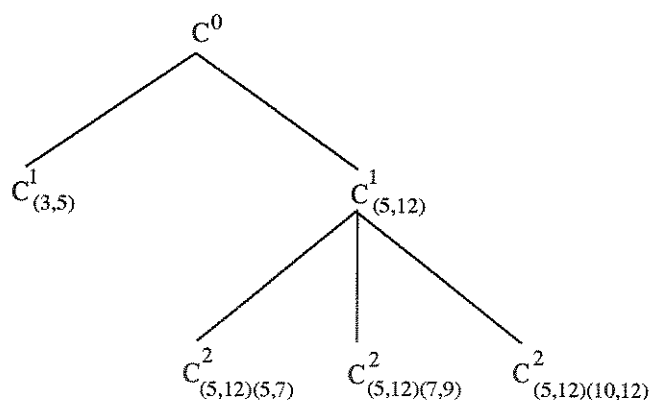


Figura 3.6 Representação hierárquica da decomposição do contorno apresentado.

Esta decomposição é associada de forma única a cada contorno, fornecendo uma análise sintética e exaustiva da deformação de um contorno. Ela será utilizada na seção seguinte para transformar um contorno arbitrário C em um contorno convexo.

3.1.1.2.b Transformação dos Contornos

A transformação do contorno consiste em se projetar cada ponto de C^0 em seu *convex hull* E^0 . Isto equivale a reduzir a árvore associada a C em um único nó que representa um contorno convexo. Para tal, usa-se um procedimento *bottom-up* a fim de eliminar recursivamente todos os nós folhas correspondendo às concavidades elementares. Passo a passo, um nó folha é eliminado pela projeção das concavidades elementares definidas por este nó no *convex hull* do contorno definido por seu nó pai.

A transformação dos contornos finalizará quando da eliminação de todos os ramos da árvore.

A projeção das concavidades elementares será agora definida. Seja $C_{(i_1, j_1) \dots (i_n, j_n)}^n$ uma concavidade elementar de ordem n . Cada ponto P_i , para $i \in (i_n, j_n)$ é projetado na linha unindo os pontos P_{i_n} a P_{j_n} , como mostra a figura 3.7.

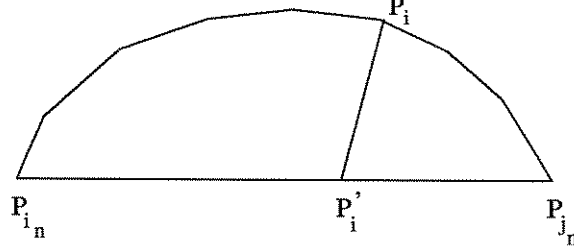


Figura 3.7 Projeção de uma concavidade elementar.

As coordenadas (x'_i, y'_i) de P'_i , isto é, a projeção de P_i , são dadas por:

$$x'_i = x_{i_n} + R_i(x_{j_n} - x_{i_n})$$

$$y'_i = y_{i_n} + R_i(y_{j_n} - y_{i_n})$$

onde R_i é definido como:

$$R_i = \frac{\sum_{k=i_n}^{i-1} d(P_k, P_{k+1})}{\sum_{k=i_n}^{j_n-1} d(P_k, P_{k+1})}$$

sendo d a distância Euclidiana em R^2

R_i é um fator peso normalizado que leva em conta a distância entre P_i e P_{i_n} (respectivamente P_{j_n}) no contorno. Deste modo, a definição da projeção permite manter a distância relativa entre os pontos na concavidade.

Quando o contorno C estiver completamente processado, o contorno convexo C' obtido terá o mesmo número de pontos. Além disso, cada ponto P_i de C é associado a um único ponto $P'_i = T(P_i)$ em C' .

Esta transformação pode ser implementada através do seguinte procedimento recursivo:

Transformação(C : contorno)

```
{
  se ( $C$  não é convexo) então
    {
      processar convex hull de  $C$ , denotado por  $E$ 
      processar as concavidades de 1ª ordem de  $C$ :  $C_1^1, C_2^1, \dots, C_n^1$ 
      para  $k = 1$  até  $n$  faça
        {
          Transformação( $C_k^1$ )
          Projeção de  $C_k^1$  em  $E$ 
        }
    }
```

}
}

A hipótese de ligação entre dois contornos não convexos possuindo deformações locais diferentes, porém com *convex hull's* similares não se mostrou tão restritiva para as aplicações práticas utilizadas neste trabalho.

Sejam C_{k-1} e C_k dois contornos não convexos a serem ligados e C'_{k-1} e C'_k seus respectivos contornos convexos associados, obtidos pela transformação T . O processo de triangulação descrito na seção 3.1.1.1 liga dois contornos convexos C'_{k-1} e C'_k e, então, estabelece relações entre os pontos de C'_{k-1} e C'_k . Pelo uso da transformação T , as mesmas relações podem ser transpostas entre os pontos de C_{k-1} e C_k . De fato, se (P'_i, Q'_j) constitui uma aresta entre C'_{k-1} e C'_k , então (P_i, Q_j) com $P'_i = T(P_i)$ e $Q'_j = T(Q_j)$ definem uma aresta entre C_{k-1} e C_k .

3.1.2 Caso 1:n

Muitos objetos complexos são compostos de várias ramificações e a seguinte situação pode ser encontrada: um contorno C_{k-1} no nível z_{k-1} tem de ser ligado a vários contornos $C_{k,1}, \dots, C_{k,r}$ no nível z_k . A situação dual, na qual um conjunto de contornos $C_{k-1,1}, \dots, C_{k-1,r}$ no nível z_{k-1} tem de ser ligado a um contorno C_k no nível z_k , é similar.

Este problema, denominado *caso 1:n*, será solucionado através da redução a um conjunto de problemas 1:1, cujos procedimentos de decomposição e transformação dos contornos e de triangulação já foram discutidos.

3.1.2.1 Interpolação dos Contornos

Considere um caso simples onde um contorno se ramifica em outros dois, como mostra a figura 3.8.

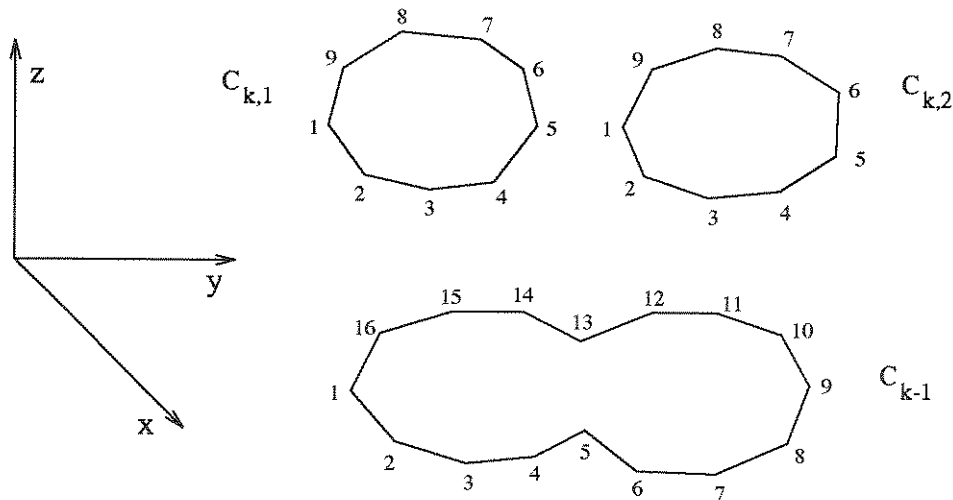


Figura 3.8 Ramificação de contornos.

A idéia do método é considerar todas as ramificações como sendo um único contorno fechado. Este processo envolve dois passos:

- introduzir um novo vértice entre os vértices mais próximos das ramificações. Cada novo vértice deve ser criado no nível intermediário $\bar{z}_k$ entre os níveis z_{k-1} e z_k :

$$\bar{z}_k = \frac{(z_{k-1} + z_k)}{2}$$

- renumerar os vértices de cada ramificação levando-se em conta a inclusão dos novos vértices, de modo que eles representem apenas um contorno (ver figura 3.9). Desta maneira, os vértices incluídos e os seus vizinhos (anterior e posterior) possuirão dois números distintos.

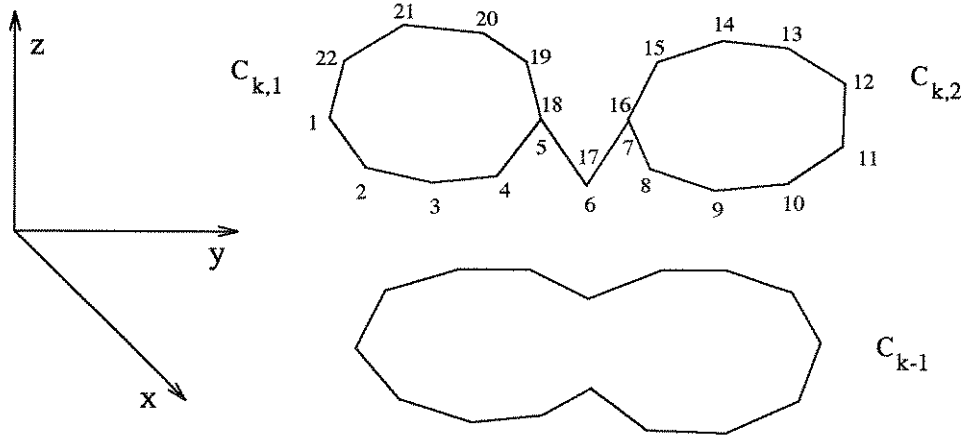


Figura 3.9 Renumeração de contornos da ramificação.

O procedimento 1:n é finalizado através da triangulação entre este contorno interpolado e o contorno de nível adjacente. O processo de triangulação aplicado é idêntico ao analisado no caso 1:1. Para o exemplo anterior, a triangulação correspondente é mostrada na figura 3.10.

Há casos em que o mecanismo de introdução de novos vértices pode não resultar numa solução correta. Isto pode ser ilustrado na figura 3.11, onde os contornos no nível C_k possuem centróides não colineares.

Para estas situações, ao invés de apenas dois vértices, eventualmente três novos vértices deveriam ser introduzidos para se alcançar a solução correta. O processo de triangulação também deve ser modificado para que a região interna dos contornos não fique aberta. A figura 3.12 apresenta uma vista superior da triangulação (região hachurada) de modo a completar a superfície do objeto.

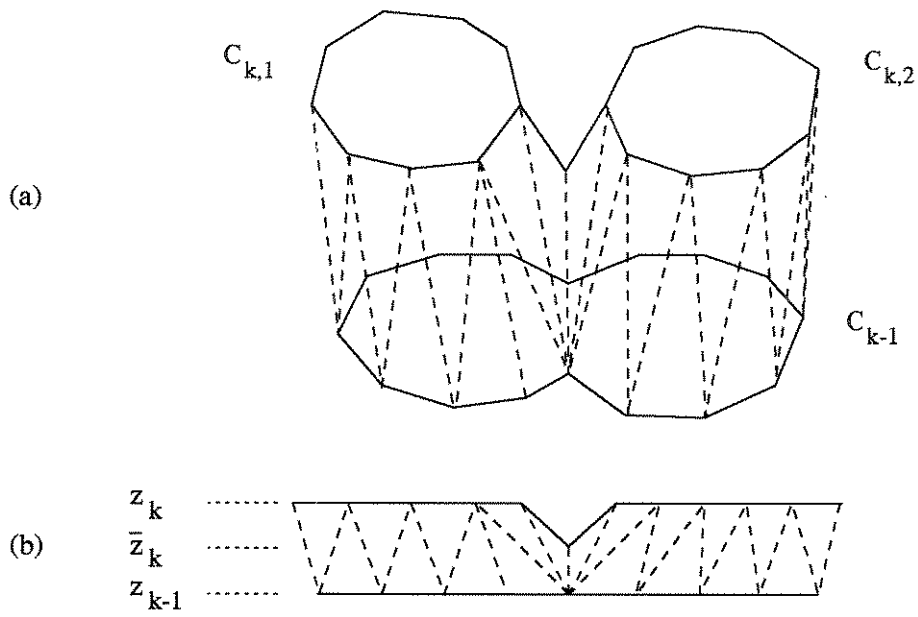


Figura 3.10 Situação de ramificação (a) triangulação entre contornos (b) vista lateral da triangulação.

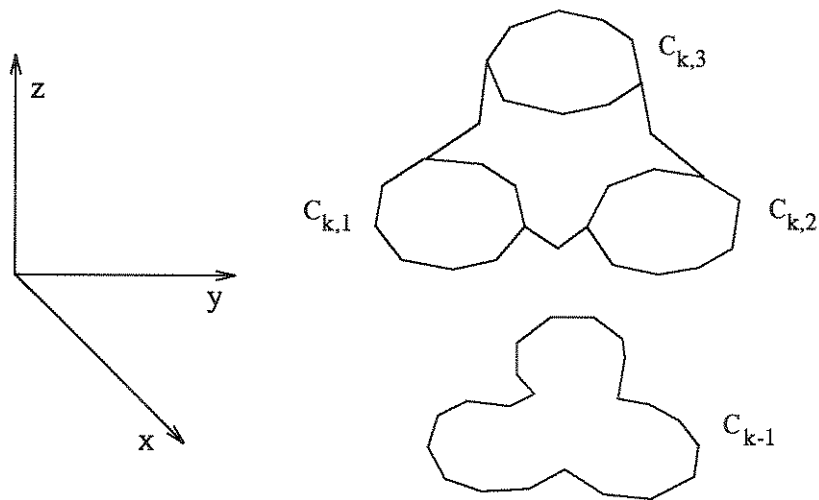


Figura 3.11 Contornos apresentando centróides não colineares.

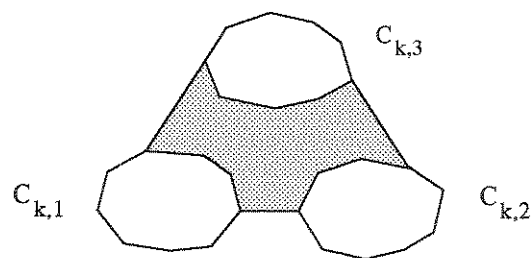


Figura 3.12 Triangulação entre contornos.

3.1.3 Caso n:m

O procedimento geral n:m deve permitir a ligação entre um conjunto de r contornos fechados $C_{k-1,1}, C_{k-1,2}, \dots, C_{k-1,r}$ no nível z_{k-1} e um conjunto de s contornos fechados $C_{k,1}, C_{k,2}, \dots, C_{k,s}$ no nível z_k . A idéia consiste em automaticamente se dividir o problema em procedimentos 1:1 e 1:n. O problema principal em se obter esta decomposição é decidir quais contornos $C_{k-1,i}$ têm de ser ligados a quais contornos $C_{k,j}$.

Quando a amostragem na direção z é suficientemente fina, dois contornos $C_{k-1,i}$ e $C_{k,j}$ possuirão aproximadamente a mesma posição relativa em suas respectivas fatias. Sob esta hipótese, se duas fatias forem sobrepostas, os dois contornos apresentarão regiões comuns.

Infelizmente, nem sempre uma amostragem fina ocorre entre fatias adjacentes. Um modo de decidir quais contornos devem ser ligados, com o mínimo de interação humana possível, é estimando-se o grau de sobreposição entre pares de contornos de fatias adjacentes. Este grau de sobreposição é, então, utilizado como um critério de decisão que estabelece se dois contornos devem ou não ser conectados. Para este propósito, um mapeamento entre múltiplos contornos é apresentado nas seções seguintes.

3.1.3.1 Determinação da Sobreposição de Contornos

Em situações onde os cortes transversais apresentam múltiplos contornos, é necessário se determinar quais os contornos possuem ramificações e, portanto, como devem ser ligados no algoritmo de triangulação. Este mapeamento é baseado na determinação do grau de sobreposição de contornos entre fatias adjacentes. Para dois contornos $C_{k-1,i}$ e $C_{k,j}$ pertencentes a seções adjacentes de níveis z_{k-1} e z_k , este valor poderia representar o resultado do cálculo da área de intersecção entre estes contornos. Entretanto, propõe-se uma aproximação desta área, a qual requer menos computação e que será suficiente na maior parte das aplicações. Esta aproximação é realizada a partir das áreas dos retângulos que englobam cada contorno em planos adjacentes.

Sejam $(x_{\min_{k-1,i}}, y_{\min_{k-1,i}})$ e $(x_{\max_{k-1,i}}, y_{\max_{k-1,i}})$ as coordenadas mínimas e máximas, respectivamente, de todos os pontos pertencendo ao contorno $C_{k-1,i}$. Os valores $x_{\min_{k,j}}, y_{\min_{k,j}}, x_{\max_{k,j}}$ e $y_{\max_{k,j}}$ são similarmente definidos para o contorno $C_{k,j}$. A figura 3.13 ilustra um exemplo de dois contornos C_1 e C_2 em fatias adjacentes.

Este conjunto de coordenadas define dois retângulos englobando os contornos $C_{k-1,i}$ e $C_{k,j}$ cujos lados são paralelos aos eixos x e y . O grau de sobreposição dos dois contornos é definido pela área de intersecção entre estes dois retângulos e pode ser obtido pelo seguinte algoritmo:

$$\begin{aligned} &\text{Área_de_Intersecção}(C_{k-1,i}, C_{k,j}: \text{contorno}) \\ &\{ \\ &\quad \text{se ambos retângulos englobando } C_{k-1,i} \text{ e } C_{k,j} \text{ não estão incluídos um ao outro} \\ &\quad \{ \\ &\quad \quad A_{i,j} = \min([x_{\max_{k,j}} - x_{\min_{k-1,i}}], [x_{\max_{k-1,i}} - x_{\min_{k,j}}]) \cdot \\ &\quad \quad \min([y_{\max_{k,j}} - y_{\min_{k-1,i}}], [y_{\max_{k-1,i}} - y_{\min_{k,j}}]) \\ &\quad \} \\ &\} \end{aligned}$$

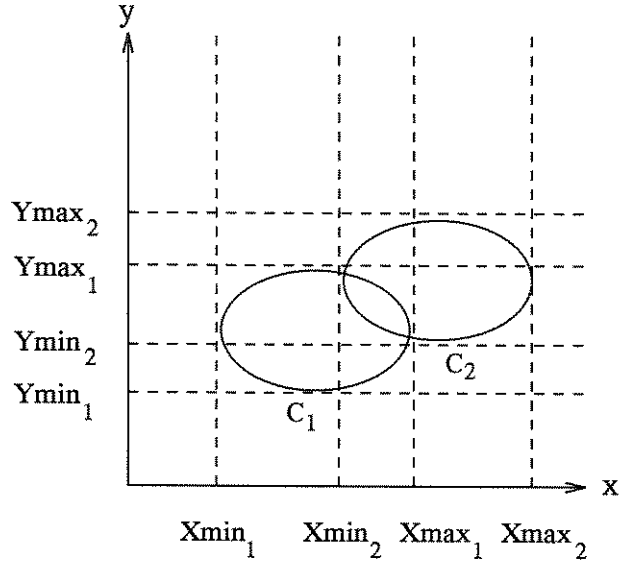


Figura 3.13 Avaliação da sobreposição de dois contornos adjacentes.

$$\begin{aligned}
 &\text{senão} \\
 &\quad \{ \\
 &\quad \quad A_{i,j} = \min([x_{max_{k-1,i}} - x_{min_{k-1,i}}], [x_{max_{k,j}} - x_{min_{k,j}}]), \\
 &\quad \quad \min([y_{max_{k-1,i}} - y_{min_{k-1,i}}], [y_{max_{k,j}} - y_{min_{k,j}}]) \\
 &\quad \} \\
 &\}
 \end{aligned}$$

onde,

$$\begin{aligned}
 \min(a, b) &= \begin{cases} a & \text{se } a \leq b \\ b & \text{se } a > b \end{cases} \\
 [a] &= \begin{cases} a & \text{se } a > 0 \\ 0 & \text{se } a < 0 \end{cases}
 \end{aligned}$$

De modo a se obter um valor normalizado, o grau de sobreposição de $C_{k-1,i}$ e $C_{k,j}$ pode ser dado por:

$$A'_{[i,j]} = \max\left(\frac{A_{[i,j]}}{A_{[k-1,i]}}, \frac{A_{[i,j]}}{A_{[k,j]}}\right) \quad (2)$$

onde,

$A_{[k-1,i]}$ é a área do retângulo englobando $C_{k-1,i}$

$A_{[k,j]}$ é a área do retângulo englobando $C_{k,j}$

$$\max(a, b) = \begin{cases} a & \text{se } a \geq b \\ b & \text{se } a < b \end{cases}$$

A expressão (2) acima estima o grau de sobreposição de dois contornos em níveis adjacentes, ou seja, determina se os dois contornos devem ou não ser ligados através da triangulação. Seu valor varia entre 0 e 1, indicando desde um mapeamento nulo até um mapeamento completo entre os contornos. Como forma de estabelecer

um limite mínimo de sobreposição de contornos, foi criado um limiar t tal que se $A'_{[i,j]} > t$, então haverá ligação entre $C_{k-1,i}$ e $C_{k,j}$. Um limiar de 80% mostrou-se adequado para os objetos submetidos à ferramenta de reconstrução implementada neste trabalho.

3.1.3.2 Decisão de Mapeamento

O grau de sobreposição $A'_{[i,j]}$ anteriormente definido é avaliado para cada par de contornos $C_{k-1,i}$ e $C_{k,j}$ de níveis z_{k-1} e z_k , respectivamente. Deste modo, pode-se criar uma matriz de mapeamento M de ordem $r \times s$ elementos tal que:

$$M_{i,j} = \begin{cases} 1 & \text{se } A'_{[i,j]} > t \text{ para } 1 \leq i \leq r \\ 0 & \text{se } A'_{[i,j]} \leq t \text{ para } 1 \leq j \leq s \end{cases}$$

O exemplo da figura 3.14 ilustra o mapeamento entre contornos pertencentes a duas fatias adjacentes de níveis z_1 e z_2 , onde cada uma possui, respectivamente, 6 e 4 contornos. A matriz de ligação associada ao mapeamento também é apresentada.

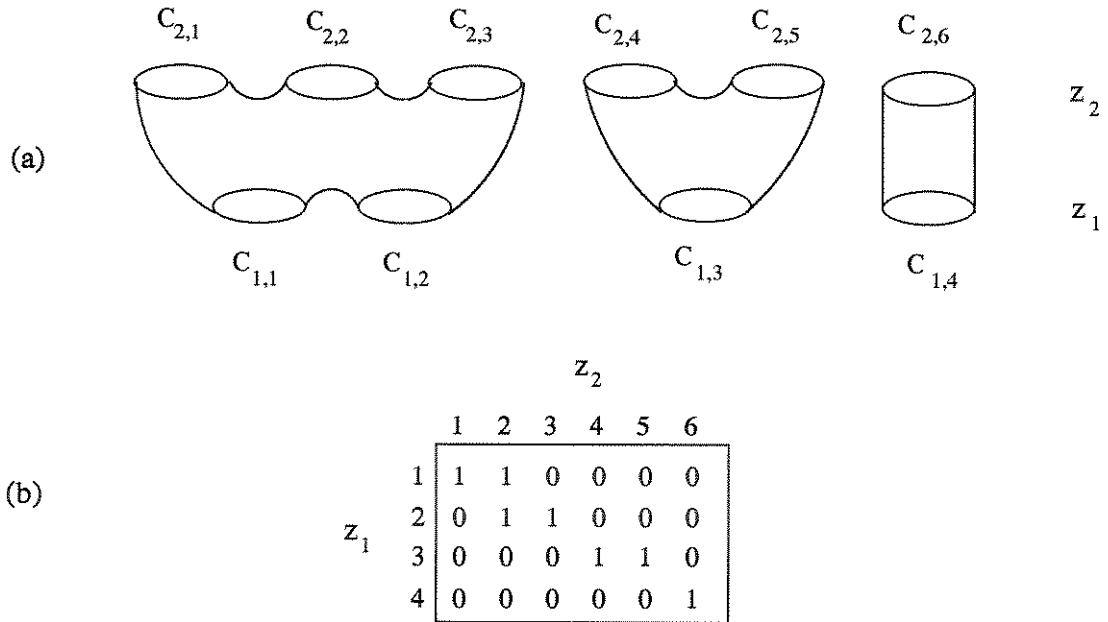


Figura 3.14 Exemplo de mapeamento de contornos (a) Contornos (b) Matriz de ligação.

Toda a informação referente ao mapeamento dos contornos entre dois níveis consecutivos pode ser obtida a partir da análise desta matriz M :

- o contorno $C_{k-1,i}$ não é ligado a nenhum outro contorno se:

$$\sum_{j=1}^s M_{i,j} = 0$$

		z_k					
		1	2	...	j	...	s
z_{k-1}	1	0	0	...	0	...	0

- o contorno $C_{k,j}$ não é ligado a nenhum outro contorno se:

$$\sum_{i=1}^r M_{i,j} = 0$$

		z_k						
			j					
	1	0						
	2	0						
	$\vdots$	$\vdots$						
z_{k-1}	i	0						
	$\vdots$	$\vdots$						
	r	0						

- uma ligação simples (caso 1:1) é criada entre $C_{k-1,i}$ e $C_{k,j}$ se, e apenas se:

$$\sum_{i=1}^r M_{i,j} = 1 \quad \text{e} \quad \sum_{j=1}^s M_{i,j} = 1$$

		z_k					
		1	2	...	j	...	s
	1				0		
	2				0		
	$\vdots$				$\vdots$		
z_{k-1}	i	0	0	...	1	...	0
	$\vdots$				$\vdots$		
	r				0		

De fato, se $\sum_{i=1}^r M_{i,j} = 1$, então, $C_{k-1,i}$ possui um único contorno correspondente no nível z_k , e se $\sum_{j=1}^s M_{i,j} = 1$, então $C_{k,j}$ possui um único contorno

correspondente no nível z_{k-1} . Assim, $C_{k-1,i}$ e $C_{k,j}$ são conectados através de um procedimento 1:1.

- uma ligação múltipla (caso 1:n) é criada de $C_{k-1,i}$ (respectivamente, $C_{k,j}$) se:

$$\sum_{j=1}^s M_{i,j} > 1 \text{ (respectivamente, } \sum_{i=1}^r M_{i,j} > 1)$$

		z_k					
		1	2	...	j	...	s
1					0		
2					0		
$\vdots$					$\vdots$		
z_{k-1}	i	0	0	...	1	...	1
$\vdots$					$\vdots$		
r					0		

Se

$$\sum_{j=1}^s M_{i,j} > 1 \text{ (respectivamente, } \sum_{i=1}^r M_{i,j} > 1)$$

então, o contorno $C_{k-1,i}$ (respectivamente $C_{k,j}$) possui mais de um contorno correspondente na fatia adjacente. Assim, um procedimento 1:n é realizado entre todos os contornos correspondentes.

Após a determinação do tipo de ligação entre os contornos, a triangulação final pode ser realizada utilizando-se o procedimento adequado àquela situação. A figura 3.15 resume todas as situações que o algoritmo de reconstrução deve considerar até a realização da triangulação entre os contornos.

3.2 Comentários Finais

A metodologia proposta neste capítulo teve como objetivo resolver sistematicamente as principais deficiências encontradas nos métodos de reconstrução existentes.

A aproximação das superfícies dos objetos é realizada através de um algoritmo heurístico de triangulação. Contornos apresentando complexas concavidades são corretamente manipulados. Dificuldades em relação à ligação de contornos com formas e tamanhos não similares são solucionadas. O problema de ramificação de contornos também é tratado satisfatoriamente. O método pode operar automaticamente sem a necessidade de interação com o usuário durante o processo de reconstrução.

A reconstrução das superfícies a partir de um conjunto de contornos pertencentes às seções transversais do objeto pode ser melhorada explorando-se as características ou estruturas locais de cada contorno. Desta forma, a triangulação poderia ser

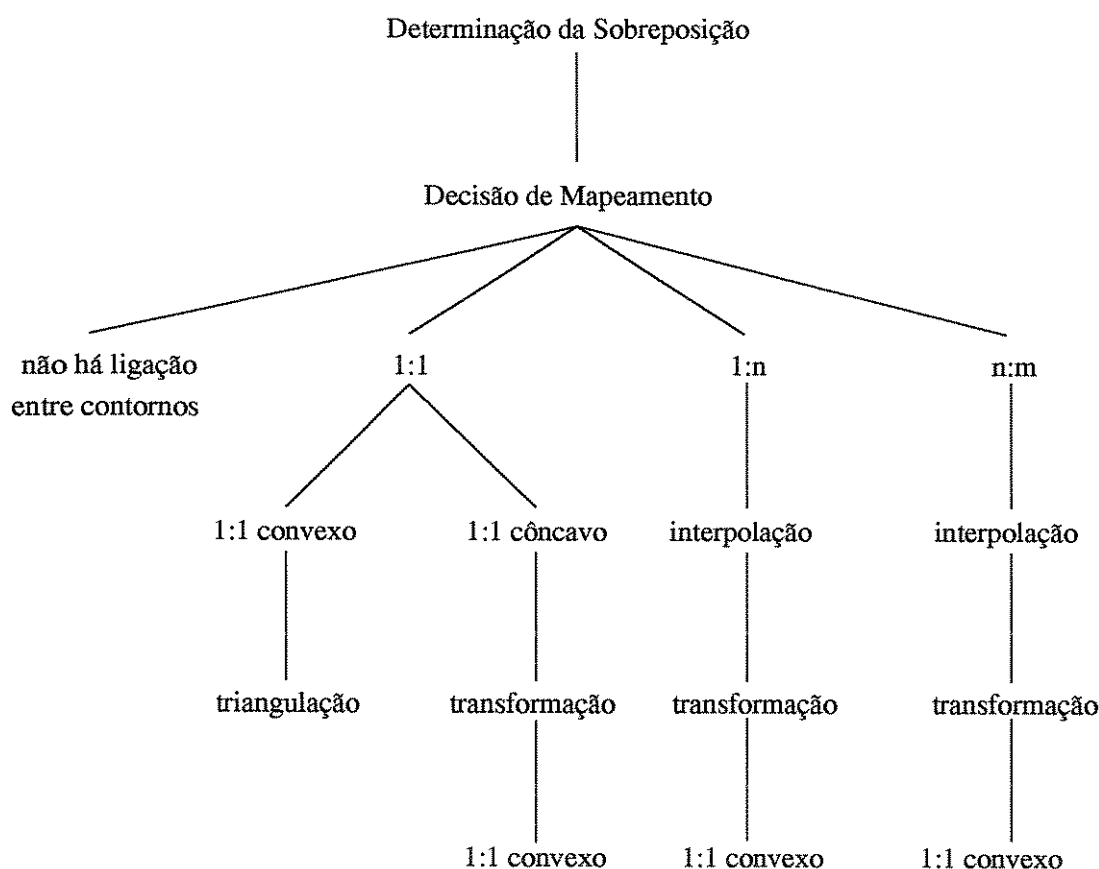


Figura 3.15 Situações consideradas pelo algoritmo de reconstrução.

realizada primeiramente nestas regiões de alta similaridade e, posteriormente, nas regiões restantes. A identificação destas regiões de alta similaridade envolveria uma análise prévia ou um pré-processamento dos contornos amostrados.

Técnicas baseadas em reconhecimento de padrões poderiam ser utilizadas na classificação de regiões similares. Uma abordagem é apresentada no trabalho de Keltarnavaz, Simar e Figueiredo [40, 41], onde cada contorno é descrito por uma representação sintático-semântica em contraste com a descrição convencional composta de um conjunto de vértices. A partir desta descrição alto nível, consistindo de primitivas baseadas na informação estrutural dos contornos e em atributos semânticos, são estabelecidas relações de similaridade entre contornos adjacentes. A reconstrução das superfícies entre contornos utiliza estas relações de similaridade, produzindo uma triangulação local nestas regiões. Num segundo passo, o algoritmo de triangulação é aplicado nas regiões ainda não conectadas.

Outra dificuldade muito comum dos algoritmos de reconstrução ocorre quando pares de contornos a serem triangulados apresentam número de vértices muito diferentes entre si. Devido a este fato, a superfície reconstruída pode apresentar distorções ocasionadas pela existência de muitas faces triangulares contendo um vértice comum. Uma maneira de resolver este problema seria através da reamostragem dos pontos pertencentes a cada contorno, buscando-se triangulações com taxas equivalentes nos contornos superior e inferior.

Uma sugestão de melhoria da capacidade do método proposto de reconstrução

seria a manipulação de contornos apresentando subcontornos internos (aberturas). Neste caso, a identificação e mapeamento dos contornos internos e externos também poderia ser realizada através da aplicação do grau de sobreposição analisado anteriormente.

Este trabalho propôs uma solução que operasse de forma a mais automática possível, porém em casos onde possam existir ambigüidades no mapeamento dos contornos é interessante que o processo de reconstrução possa realizar decisões através da interação com o usuário.

DETERMINAÇÃO DE MEDIDAS E VISUALIZAÇÃO DOS OBJETOS

4.1 Determinação de Medidas

Após a obtenção da representação tridimensional dos objetos através do processo de reconstrução descrito, pode-se determinar, com certa precisão, diversos parâmetros geométricos do objeto, tais como área superficial, volume, centro de massa, entre outros. A análise quantitativa destes parâmetros permite um melhor entendimento da estrutura tridimensional dos objetos, facilitando também sua posterior visualização.

Neste trabalho, esta análise restringiu-se às estimativas de cálculo da área superficial e do volume de objetos reconstruídos, cujos valores serviram de base para avaliação da precisão do processo de reconstrução.

4.1.1 Área Superficial

Uma vez que o objeto tenha sido reconstruído, a área da superfície lateral pode ser calculada simplesmente tomando-se a soma das áreas das faces triangulares que formam a superfície do objeto [15, 16].

Suponha que P_i , Q_i e R_i sejam as coordenadas dos vértices do i -ésimo triângulo, com P_i situado no contorno superior, Q_i no contorno inferior e R_i podendo ser ou o próximo vértice no contorno superior, ou o próximo vértice no contorno inferior, como mostra a figura 4.1.

Os seguintes vetores são determinados:

$$\overline{X_i} = \overline{P_i Q_i} \quad \text{e} \quad \overline{Y_i} = \overline{P_i R_i}$$

Então, a área da superfície é dada por:

$$\text{Área} = \frac{1}{2} \sum_{i=1}^N |\overline{X_i} \otimes \overline{Y_i}|$$

onde N é número total de faces triangulares

$\otimes$ denota o produto vetorial

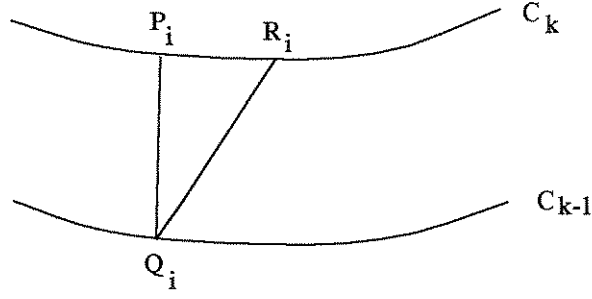


Figura 4.1 Face triangular formada entre contornos adjacentes.

4.1.2 Volume

Considere duas seções transversais adjacentes de um objeto reconstruído, onde o contorno superior possui m vértices e o contorno inferior possui n vértices. Portanto, existem exatamente $(m + n)$ faces triangulares que formam uma superfície entre os dois contornos.

Sejam P_i , Q_i e R_i as coordenadas dos vértices que definem a i -ésima face triangular (P_i no contorno superior, Q_i no contorno inferior e R_i sendo o vértice consecutivo no contorno superior ou inferior). Seja U um vértice qualquer do contorno superior. Então, os seguintes vetores podem ser formados, como mostra a figura 4.2.

$$\overline{A_i} = \overline{UP_i}, \quad \overline{B_i} = \overline{UQ_i} \quad \text{e} \quad \overline{C_i} = \overline{UR_i}$$

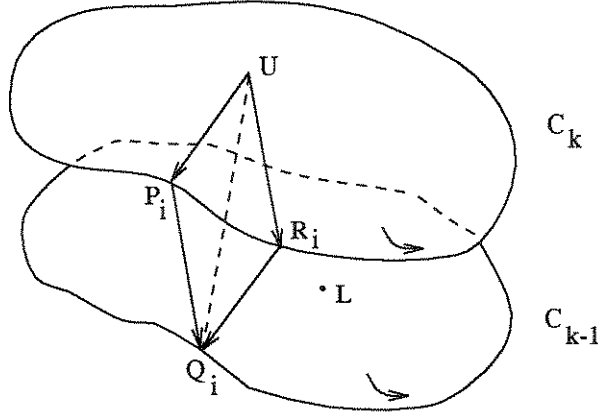


Figura 4.2 Vetores formados entre vértices de contornos adjacentes.

Redefinindo a notação dos vértices no contorno inferior para S_j , $j = 1, 2, \dots, n$, e considerando um ponto L neste mesmo contorno, os seguintes vetores podem ser formados:

$$\overline{D_j} = \overline{US_j} \quad \text{e} \quad \overline{E} = \overline{UL}$$

O volume entre estas duas seções transversais pode ser dado pela expressão:

$$\text{Volume} = \frac{1}{6} \left| \sum_{i=1}^{m+n} \overline{A_i} \odot \overline{B_i} \otimes \overline{C_i} + \sum_{j=1}^n \overline{D_{j+1}} \odot \overline{D_j} \otimes \overline{E} \right|$$

onde $\odot$ denota o produto escalar

$\otimes$ denota o produto vetorial

O volume total de um objeto é calculado levando-se em conta a soma de todas as seções transversais que compõem o objeto.

O algoritmo para estimativa do volume é simples do ponto de vista computacional. Entretanto, sua precisão depende de vários fatores, tais como a taxa de amostragem dos contornos e o espaçamento entre eles [14, 15, 16].

4.2 Visualização

A visualização dos objetos reconstruídos envolve o tratamento de diversos aspectos complexos como posição do observador, iluminação, transparência, cor, textura, eliminação de superfícies escondidas, entre outros.

As técnicas de visualização mantêm uma estreita relação com o modelo usado para se representar o objeto e, portanto, diferem para os métodos de reconstrução baseados em volume e superfície.

A ferramenta utilizada para a visualização dos objetos reconstruídos durante este trabalho faz parte de um projeto na área de Computação de Imagens denominado PROSIM (Prototipação e Síntese de Imagens Foto-Realistas), o qual foi inteiramente desenvolvido pelo Grupo de Computação de Imagens do Departamento de Engenharia de Computação e Automação Industrial (DCA) da FEE-UNICAMP e coordenado pelo Prof. Dr. Léo Pini Magalhães.

O algoritmo da ferramenta é baseado na técnica de Scanline Z-Buffer, a qual faz uso intenso da coerência espacial (principalmente a coerência de arestas) da imagem. Cada elemento do objeto é processado segundo a ordem de rastreamento do monitor.

As superfícies dos objetos consideradas no algoritmo Scanline são poligonais, sendo os objetos definidos por um conjunto de polígonos (faces) que por sua vez são definidos por arestas. Deste modo, as faces triangulares resultantes do método de reconstrução baseado no algoritmo de triangulação puderam ser diretamente utilizadas pela ferramenta de visualização.

A seguir serão apresentadas as principais características disponíveis na ferramenta de visualização.

4.2.1 Eliminação de Superfícies Escondidas

A determinação das superfícies dos objetos que são visíveis em relação a um observador é uma característica importante para a obtenção de imagens realísticas.

Os algoritmos Scanline operam no espaço da imagem [75], ou seja, determinam para cada pixel na imagem, o objeto cuja superfície é visível. Seu funcionamento é baseado em comparações de profundidade a cada linha de varredura processada, possibilitando a determinação das faces mais próximas do observador. Estes algoritmos exploram os princípios de ordenação e similaridade de modo a melhorar seu desempenho, em especial, a coerência de arestas [68, 75]. Esta similaridade ocorre, por exemplo, entre conjuntos de polígonos cujas projeções interceptam duas linhas sucessivas no dispositivo de visualização.

4.2.2 Modelos de Representação de Objetos

O modelo adotado é conhecido como modelo de superfícies limitantes (B-Rep), onde a descrição de um objeto é realizada não apenas por seus elementos geométricos como pontos, linhas e planos, mas também por sua topologia, ou seja, pelo relacionamento entre estes elementos geométricos na formação do objeto. Deste modo, cada face pode ser descrita pela superfície sobre a qual ela se encontra e pelo conjunto de arestas que delimitam esta superfície. Cada aresta, por sua vez, pode ser representada pela curva que a contém e pelos pontos que a delimitam.

Na figura 4.3 é apresentada uma pirâmide e sua representação B-Rep.

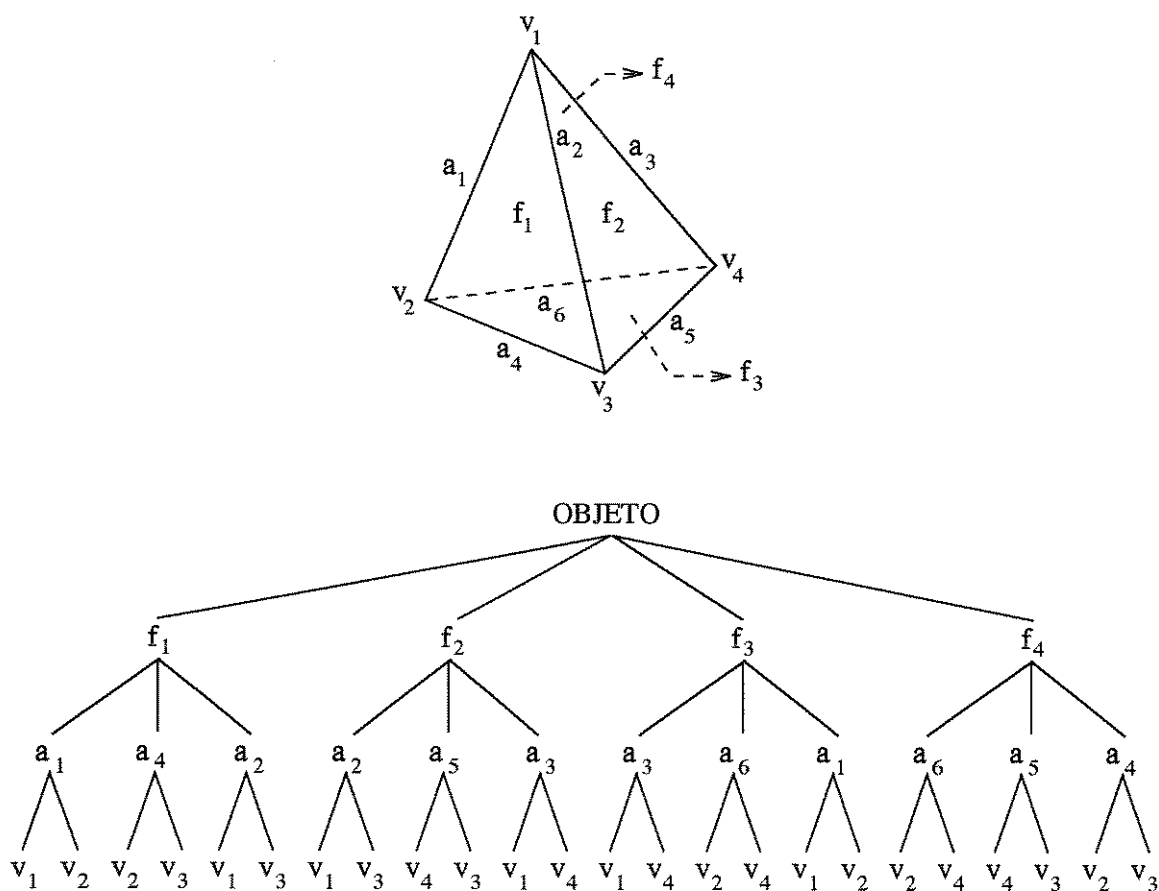


Figura 4.3 Objeto e sua representação B-Rep.

Para que um conjunto de superfícies limitantes represente um objeto real, as superfícies devem ser *limitadas* (ocupam um espaço finito) e *conectadas* (sempre existe um caminho sobre a superfície que une quaisquer dois de seus pontos).

4.2.3 Modelos de Iluminação e Tonalização

Um modelo de iluminação é uma formulação matemática que permite o cálculo da influência das fontes de luz sobre as superfícies dos objetos possibilitando um aumento do realismo da imagem.

A intensidade de luz recebida pelo observador é função da reflexão, transmissão e absorção da luz que incide sobre a superfície. O modelo implementado no Scanline considera, por simplificação, apenas os efeitos de reflexão, sendo esta composta pelas componentes ambiente, especular e difusa. O modelo de iluminação adotado foi o de Phong [58], sendo utilizadas algumas aproximações para a simulação da componente de reflexão, já que a simulação completa envolveria cálculos muito complexos e lentos sob o ponto de vista computacional.

A tonalização de superfícies pode ser definida como o tratamento da distribuição da luz sobre uma superfície iluminada, proporcionando a suavização das cores [61]. O modelo de iluminação calcula a intensidade de luz relativa a um ponto específico

da superfície, não fornecendo mecanismos suficientes para o cálculo da completa distribuição da luz ao longo de toda a superfície do objeto. As técnicas de tonalização implementadas foram a de Bouknight, Gouraud e Phong.

O modelo de Bouknight [6] produz uma única intensidade para cada polígono (face) proporcionando uma aproximação visual muito limitada, pois tende a realçar a aparência poliédrica, principalmente para objetos curvos. O modelo de Gouraud [30] calcula a intensidade nos vértices dos polígonos e, a partir da interpolação linear destes valores, obtém-se as intensidades dos pontos interiores. Este modelo fornece bons resultados, exceto em regiões de alta reflexão especular, ou seja, regiões contendo pontos de brilho. O algoritmo de Phong [58] calcula a intensidade para cada ponto do polígono e representa muito bem as regiões de alta reflexão especular. Entretanto, possui um custo computacional elevado em relação aos outros dois modelos e, normalmente, não é rápido o suficiente para apresentação em tempo real.

4.2.4 Transformações Geométricas

As transformações geométricas são operações matemáticas que permitem alterações uniformes em uma imagem definida sobre um sistema de coordenadas. Estas alterações não afetam a estrutura do objeto, mas sim o aspecto que ele vai assumir.

As principais transformações geométricas, as quais estão disponíveis no sistema Scanline, são: escalamento, translação e rotação.

Uma maneira sistemática de cálculo das coordenadas dos pontos transformados é através da utilização de uma matriz de transformação. Deste modo, uma transformação geométrica pode ser obtida pela multiplicação das coordenadas do ponto a ser transformado pela matriz de transformação e o resultado pode ser um escalamento, uma translação, uma rotação ou qualquer combinação destas transformações.

4.2.5 Transparência

A transparência é a propriedade de determinados materiais permitirem a passagem da luz. O efeito de transparência é observado quando a luz proveniente de objetos colocados atrás do corpo transparente atinge os olhos do observador.

Os algoritmos de transparência mais utilizados são os lineares [39] devido à sua simplicidade de implementação, pois consideram que a transmissividade (ou coeficiente de transparência) é constante, mesmo nas regiões onde o objeto é mais espesso. Com o objetivo de melhorar a qualidade oferecida pelos algoritmos lineares surgiram os algoritmos não lineares. Estes consideram que a quantidade de luz transmitida é proporcional à espessura da camada por onde o raio de luz atravessa. De maneira geral, o aumento de qualidade e a possibilidade de representar as propriedades ópticas do material que compõe o objeto justificam a utilização dos algoritmos não lineares, apesar do aumento do custo computacional envolvido.

4.2.6 Sombra

A simulação de sombras contribui grandemente para o aumento do realismo das imagens. O efeito das sombras proporciona a compreensão das relações entre os objetos e o melhoramento da percepção de profundidade.

As sombras são produzidas quando um objeto não é iluminado pela fonte, devido à existência de um outro objeto bloqueando o raio de luz. Se o objeto interceptor for opaco, o objeto localizado atrás deste estará em sombra. Se for transparente, o objeto interceptado será parcialmente iluminado em função do coeficiente de transparência. Para simular o efeito das sombras, é necessário determinar as partes da imagem que não estão sendo atingidas pelos raios de luz e que são visíveis ao observador. Os contornos da região de sombra são função do objeto interceptor e das fontes de luz responsáveis pela sombra.

O método implementado no Scanline-PROSIM baseia-se no trabalho de Begeon [3] e busca reduzir os cálculos requeridos na simulação de sombras.

ANÁLISE DO ALGORITMO E RESULTADOS

O objetivo deste capítulo é avaliar a metodologia proposta através da análise de determinados parâmetros extraídos dos objetos reconstruídos. Para este propósito, vários objetos sintetizados e reais foram submetidos à ferramenta de reconstrução, buscando-se estabelecer uma medida de comparação entre os resultados obtidos e os resultados teóricos esperados. Ao final, as imagens dos objetos reconstruídos são apresentadas utilizando-se os recursos disponíveis pela ferramenta de visualização Scanline Z-Buffer.

5.1 Aplicação e Avaliação do Algoritmo

O parâmetro utilizado na avaliação do algoritmo de reconstrução foi o volume dos objetos, cuja estimativa foi realizada através do procedimento discutido na seção 4.1.2.

As imagens utilizadas neste trabalho podem ser divididas em dois grupos: formas geométricas (simples e compostas) de comportamentos bem definidos analiticamente e imagens de objetos reais complexos.

As formas geométricas foram sintetizadas automaticamente através de algoritmos matemáticos, os quais geram as linhas de contorno correspondentes às seções dos objetos. As formas utilizadas foram:

- cilindro de raio e altura 10 cm;
- esfera de raio 10 cm;
- cone com base de raio 5 cm e altura 10 cm;
- subtração de um tronco de cone de altura 10 cm e bases de raios 5 e 3 cm com um outro tronco de cone de altura 10 cm e bases de raios 3 e 1 cm. Secciona-se longitudinalmente ao meio a forma resultante desta subtração.

A figura 5.1 ilustra as quatro formas geométricas acima definidas.

Para o cilindro, apenas duas seções foram utilizadas no processo de reconstrução, enquanto para as outras três formas foram utilizadas onze seções espaçadas entre si de 1 cm.

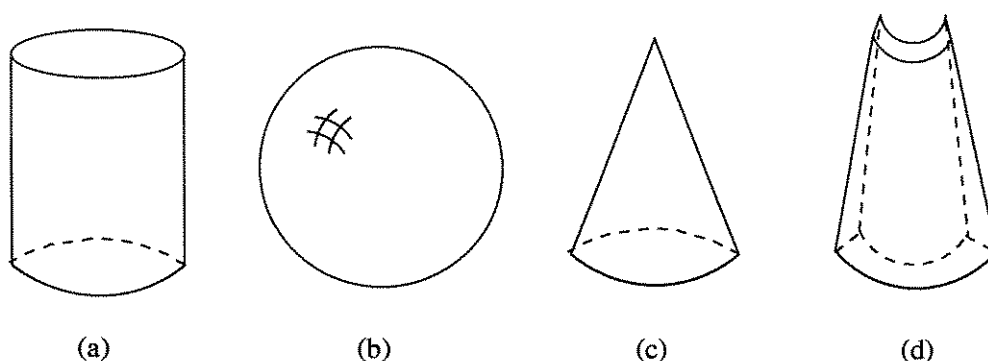


Figura 5.1 Formas geométricas utilizadas (a) Cilindro (b) Esfera (c) Cone (d) Forma Combinada.

Com o objetivo de estudar a influência de uma sub-amostragem dos vértices dos contornos no volume final dos objetos reconstruídos [55], o algoritmo de reconstrução foi aplicado variando-se o número de vértices em cada uma das três formas mencionadas. A relevância deste estudo está presente na possibilidade de obtenção de resultados com menor esforço computacional, embora mantendo uma precisão aceitável para as aplicações desejadas.

A tabela I apresenta o volume obtido pela reconstrução dos contornos originais das formas geométricas, bem como os resultados obtidos através da diminuição dos vértices desses contornos de 15%, 40% e 70%.

Formas Geométricas	Número de Contornos	Volume Real (cm^3)	Percentagem dos Vértices Originais	Volume Estimado (cm^3)	Erro (%)
Cilindro	2	3.141,59	100	3.124,70	0,54
			85	3.101,97	1,26
			60	3.070,05	2,28
			30	2.887,69	8,08
Esfera	2	4.188,79	100	4.164,49	0,58
			85	4.133,91	1,31
			60	4.092,86	2,29
			30	3.849,07	8,11
Cone	11	261,79	100	260,38	0,54
			85	258,44	1,28
			60	255,93	2,24
			30	240,54	8,12
Forma Combinada	11	188,50	100	186,26	1,19
			85	185,65	1,51
			60	182,77	3,04
			30	170,04	9,80

Tabela I Resultados da reconstrução de quatro formas geométricas.

Formas complexas obtidas através de imagens reais também foram submetidas ao algoritmo de reconstrução. Estas imagens foram digitalizadas no laboratório de

Computação de Imagens da UNICAMP e a aquisição dos vértices dos contornos foi efetuada no ambiente Khoros [43], o qual é um ambiente aberto de visualização de imagens desenvolvido na Universidade do Novo México, Albuquerque, EUA.

Primeiramente, utilizou-se imagens de seções transversais de um pimentão, o qual foi escolhido devido às formas côncavas e convexas de seus contornos e ao seu alto grau de deformação longitudinal. A figura 5.2 ilustra alguns cortes transversais do pimentão digitalizados através de um *scanner*. O espaçamento entre cada seção transversal foi de 5 mm, num total de 20 seções. Alguns dos contornos segmentados e registrados referentes a cada seção transversal são apresentados na figura 5.3.

A tabela II apresenta o volume obtido pela reconstrução do pimentão com uma variação percentual sobre os vértices dos contornos originais de 10%, 20%, 30%, 40%, 50% e 60%.

Analisando as tabelas I e II, pode-se observar que, tanto para as formas sintetizadas quanto para a forma real (pimentão), o erro na reconstrução dos objetos considerando os contornos originais foi muito pequeno, ou seja, menor que 2% e 5%, respectivamente. Com a diminuição do número de vértices dos contornos houve uma degradação do resultado obtido, como era esperado, porém dentro de um limite aceitável, mesmo para reduções de vértices de até 60%. O erro observado entre o volume medido e o volume estimado, no caso do pimentão, foi fortemente influenciado pela distância entre cortes, pela imprecisão do processo de seccionamento e da anotação dos pontos nos contornos. Contudo, o resultado pode ser considerado aceitável para grande número de aplicações.

Objeto Real	Volume Real (cm^3)	Porcentagem dos Vértices Originais	Volume Estimado (cm^3)	Erro (%)
Pimentão	131,50	100	125,59	4,49
		90	124,73	5,15
		80	124,45	5,36
		70	124,25	5,51
		60	123,81	5,85
		50	123,25	6,27
		40	121,44	7,65

Tabela II Resultados da reconstrução de um pimentão.

Outra área de aplicação que se constatou grandemente auxiliada pela reconstrução tridimensional automática é a Botânica, na qual há, em particular, interesse no estudo da anatomia e taxonomia vegetal de um vasto número de espécies. Um melhor entendimento da morfologia interna de plantas vasculares através do estudo comparativo dos tecidos e órgãos vegetativos (raiz, caule e folha) é fundamental para propósitos taxonômicos, fisiológicos e genéticos.

Estes estudos comparativos têm sido conduzidos pelos anatomistas de modo bastante trabalhoso. Cortes seriados são realizados nas regiões sob investigação através de micrótomos e fixados em lâminas. Através de corantes, os tecidos são diferenciados do sistema vascular. Com a utilização de um microscópio, as formas e as dimensões de cada corte são analisadas e avaliadas, de modo a se obter uma reconstrução tridimensional da região.

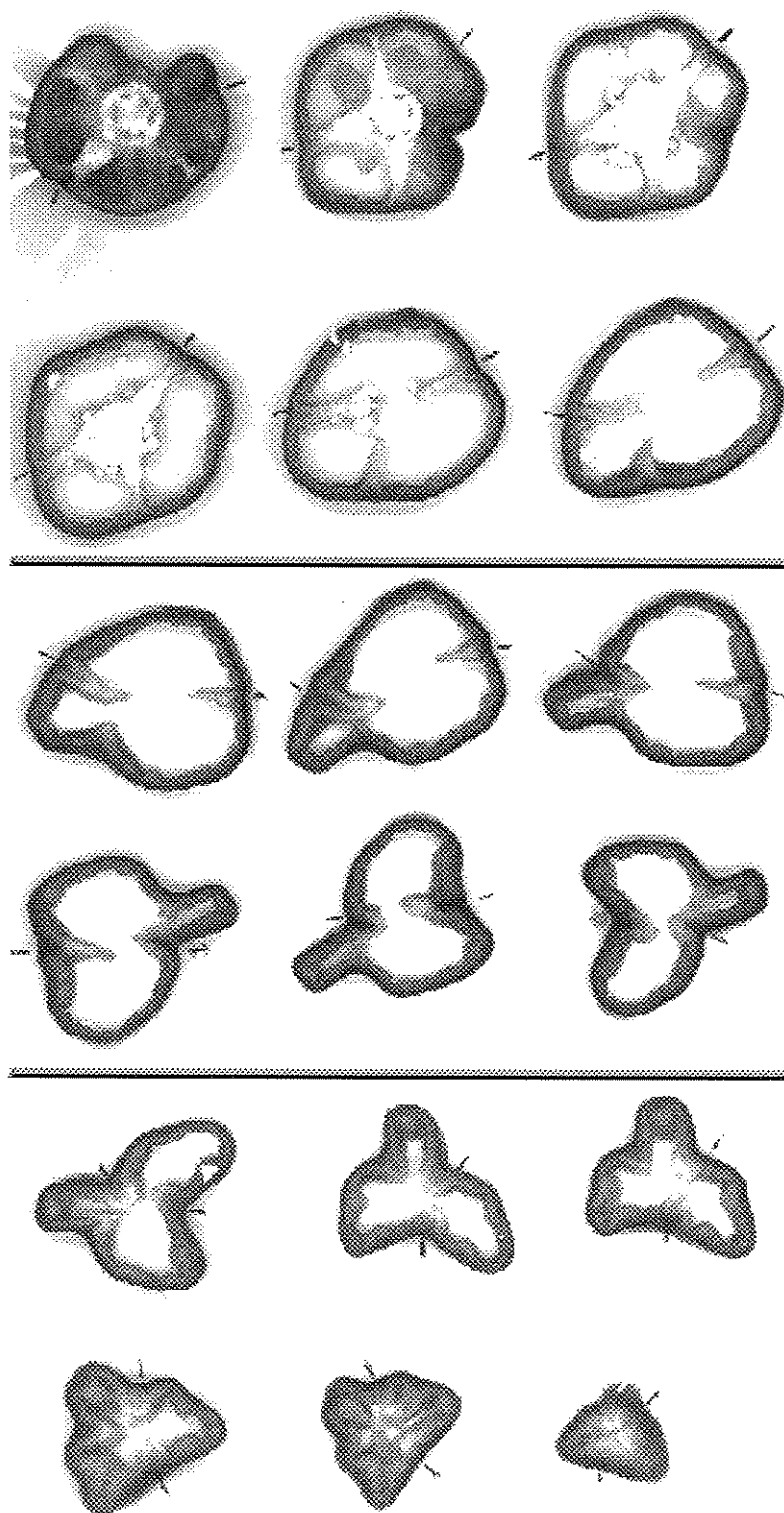


Figura 5.2 Exemplos de cortes transversais de um pimentão.

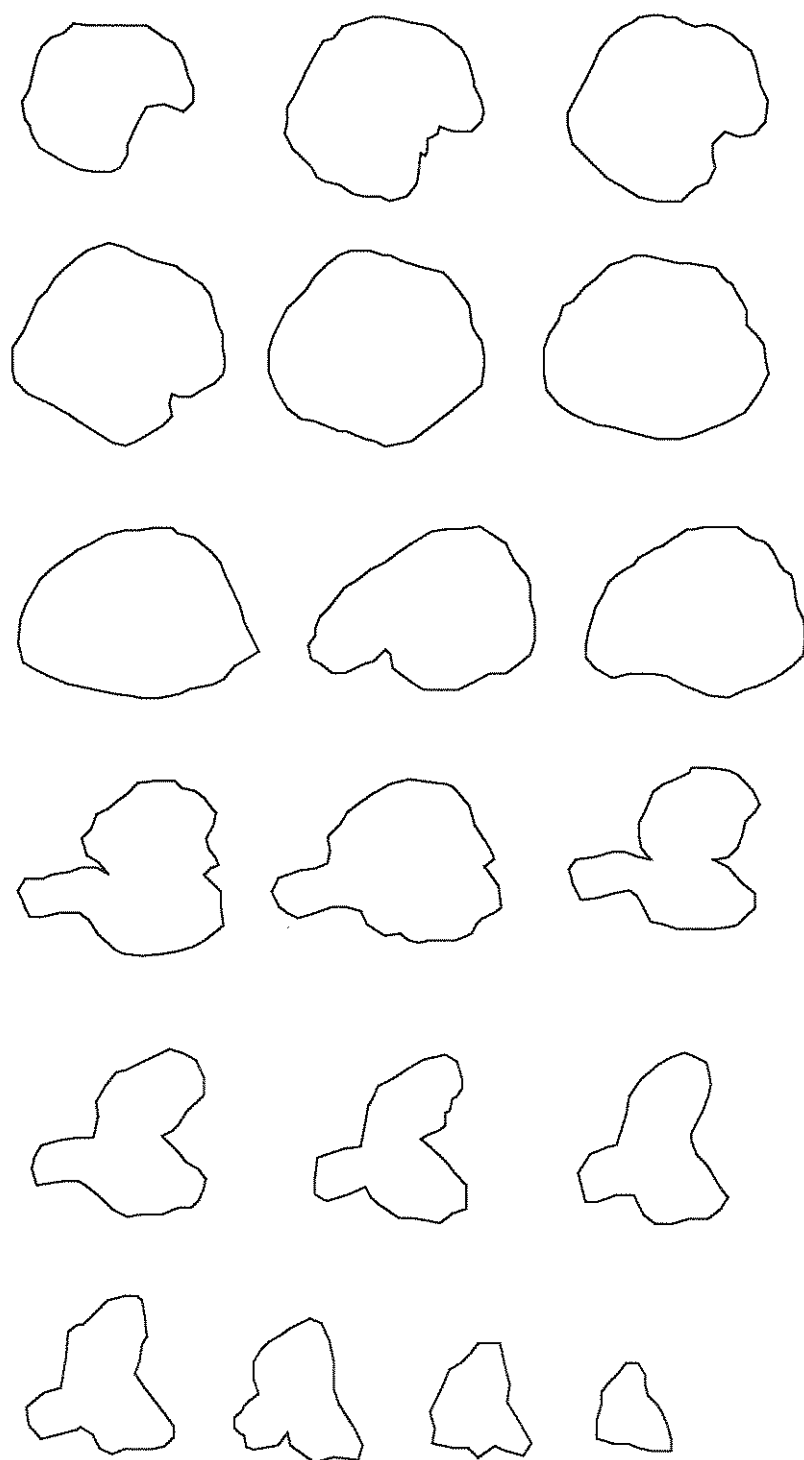


Figura 5.3 Contornos de cortes transversais de um pimentão.

Todo este processo é realizado manualmente e o resultado da reconstrução depende fortemente da experiência do anatomista. Em algumas situações, como por exemplo a determinação do volume de feixes do sistema vascular do caule, torna-se extremamente difícil ou impossível a realização precisa destas atividades mesmo para um anatomista experiente. Como consequência desta dificuldade inerente ao processo, um número muito limitado de plantas possui este aspecto anatômico estudado e compreendido.

A disponibilidade de uma ferramenta computadorizada para reconstrução tridimensional e determinação de medidas de regiões em espécies vegetais possibilitaria:

- maior confiabilidade e qualidade dos resultados;
- redução do tempo de estudo referente a cada espécie;
- o estudo comparativo entre diferentes espécies;
- a determinação de características anatômicas como elemento taxonômico.

As imagens das lâminas microscópicas correspondentes à sequência de cortes foram digitalizadas a partir de uma câmera EIKONIX 1412 acoplada a um microscópio (objetiva 2.5x e ocular 8x e processadas através do processador gráfico *I²S/IVAS*. A utilização destes recursos computacionais foi realizada no Laboratório de Processamento de Informações Georeferenciadas (LAPIG) do Instituto de Geociências da UNICAMP. O espaçamento entre os cortes foi de 8 μm . A figura 5.4 mostra alguns dos cortes transversais utilizados pelo sistema de reconstrução.

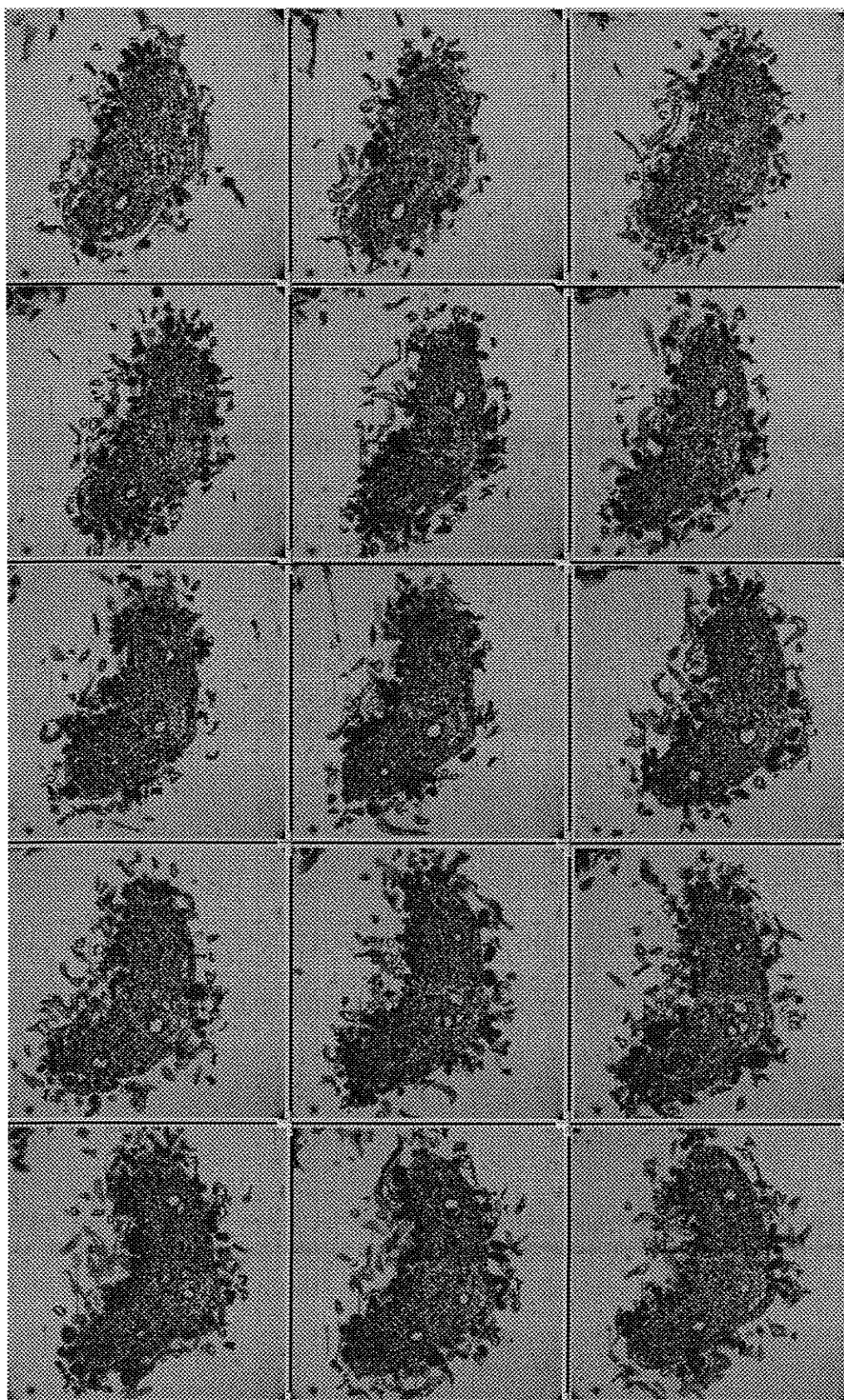


Figura 5.4 Exemplos de cortes transversais da planta.

5.2 Imagens dos Objetos Reconstruídos

As imagens obtidas pela reconstrução dos objetos submetidos à ferramenta são apresentadas a seguir. A partir destas imagens é possível observar as faces triangulares resultantes e o respectivo *rendering* para cada objeto reconstruído segundo a tonalização de Bouknight.

Verificou-se através de sucessivas reconstruções que o tempo de processamento requerido para o algoritmo de triangulação é aproximadamente linear em função do número de vértices dos objetos. É importante ressaltar que o tempo total de reconstrução leva em conta o pré-processamento (transformação) de cada contorno, no caso de contornos côncavos, e a triangulação propriamente dita. Evidentemente, contornos com concavidades complexas apresentarão maior tempo de pré-processamento. A tabela III mostra os tempos (em segundos) de CPU demandados na triangulação de cada objeto reconstruído. Em seguida, a figura 5.5 mostra um gráfico descrevendo o comportamento linear do algoritmo de triangulação.

Objetos	Nº de vértices	Tempo (s)
Cilindro	1020	3.5
Esfera	1118	4.1
Cone	961	3.2
Forma Combinada	1184	4.4
Ramificação	420	1.2
Pimentão	770	2.4
Planta	265	0.6

Tabela III Tempo de CPU amostrado para cada objeto reconstruído.

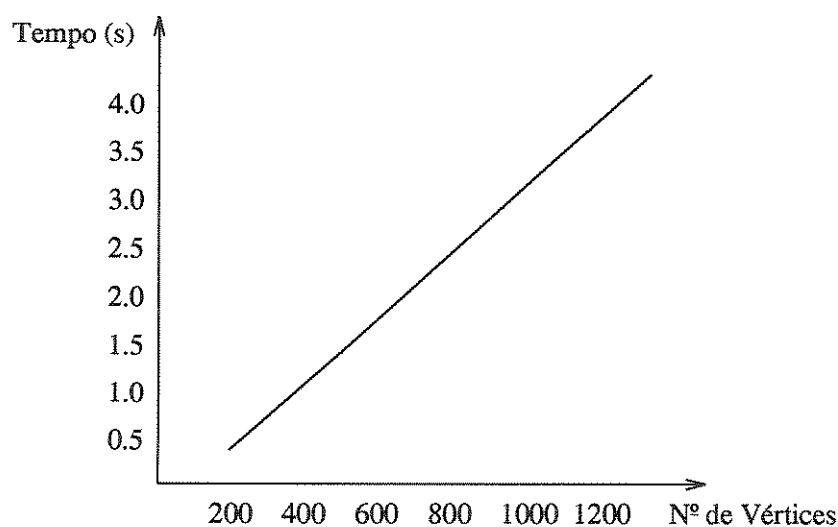


Figura 5.5 Gráfico de desempenho do algoritmo de triangulação.

a) CILINDRO	Raio	:	5 cm
	Altura	:	15 cm
	Espaçamento entre cortes	:	1 cm
	Nº total de faces triangulares	:	1920

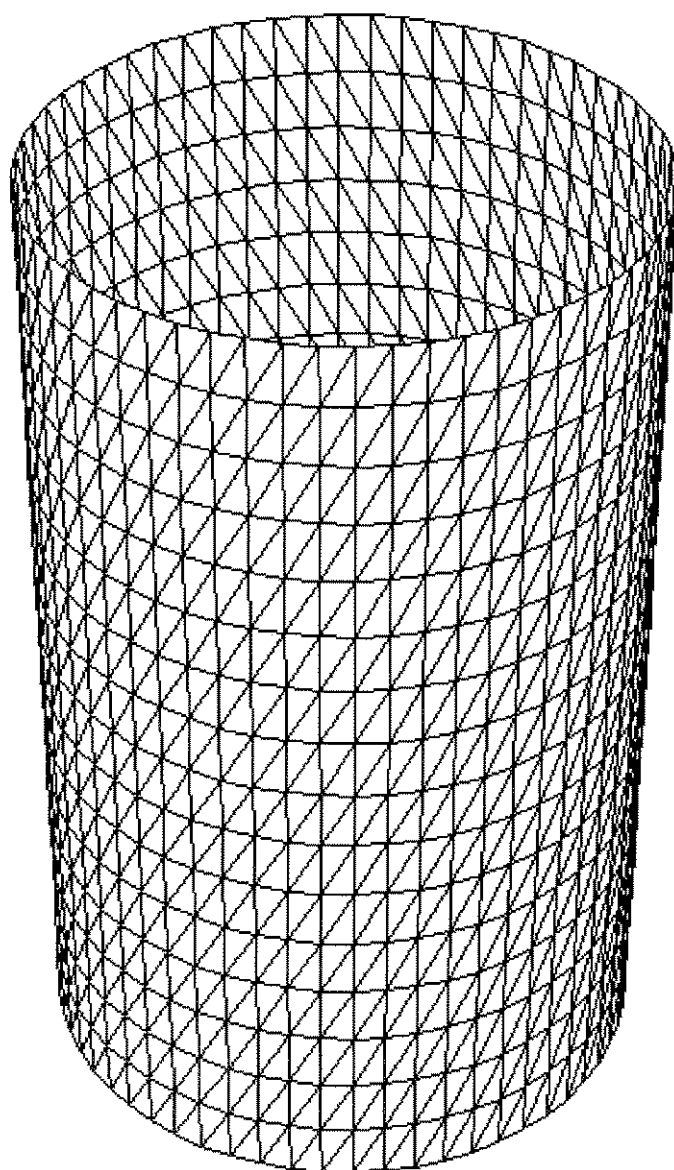


Figura 5.6(a) Cilindro.

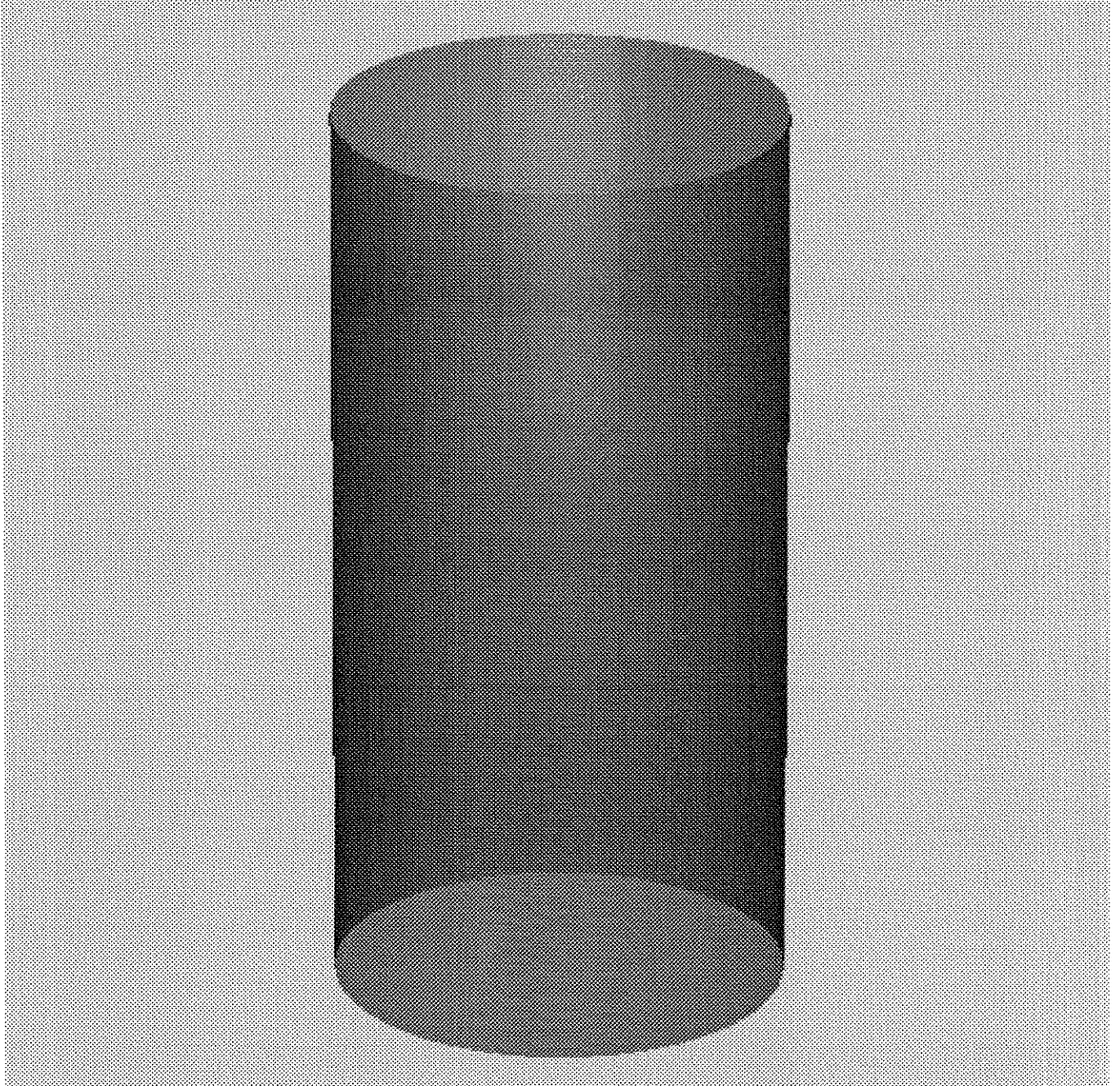


Figura 5.6(b) *Rendering* (Bouknight) do Cilindro.

b) ESFERA	Raio	: 7.5 cm
	Altura	: 33 cm
	Espaçamento entre cortes	: 0.5 cm
	Nº total de faces triangulares	: 2232

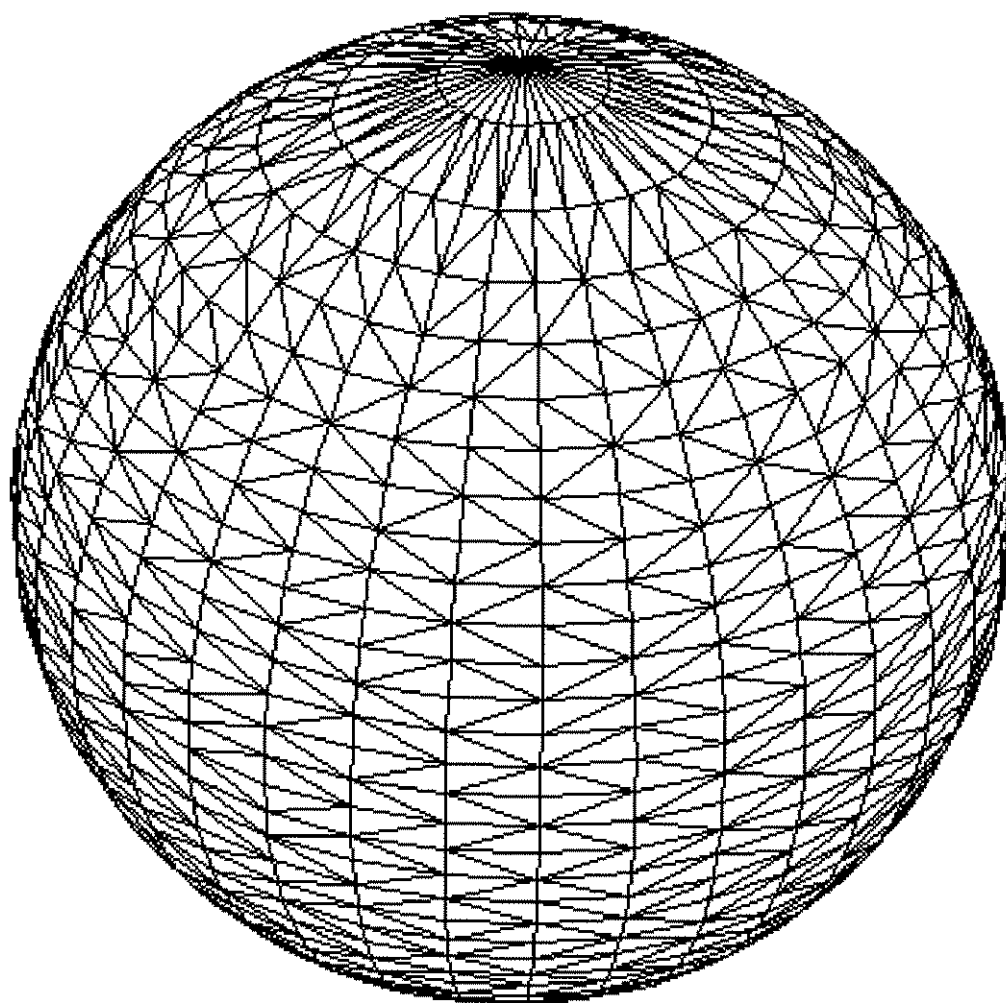


Figura 5.7(a) Esfera.

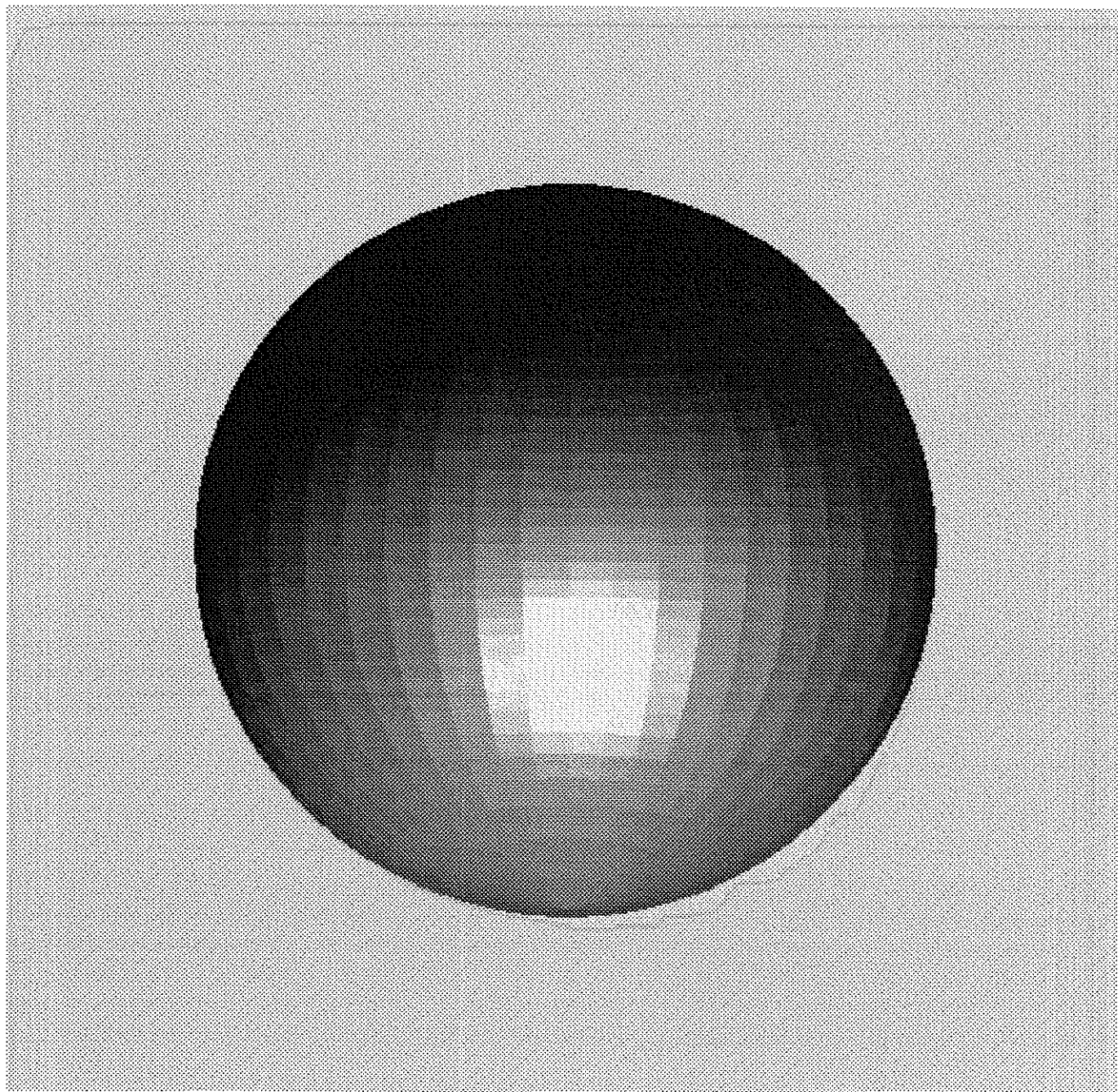


Figura 5.7(b) *Rendering* (Bouknight) da Esfera.

c) CONE

Raio	:	5 cm
Altura	:	15 cm
Espaçamento entre cortes	:	1 cm
Nº total de faces triangulares	:	1860

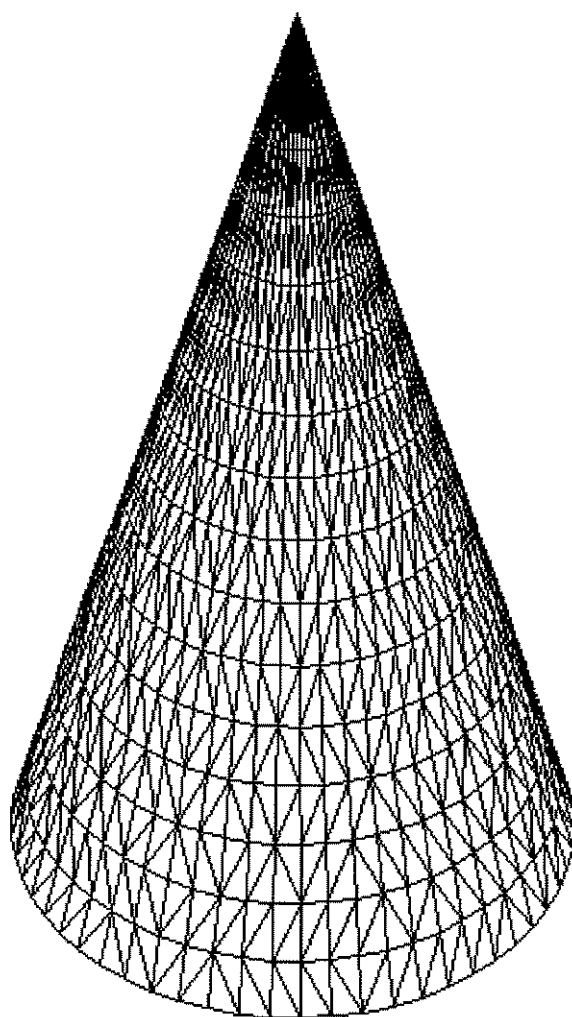


Figura 5.8(a) Cone.

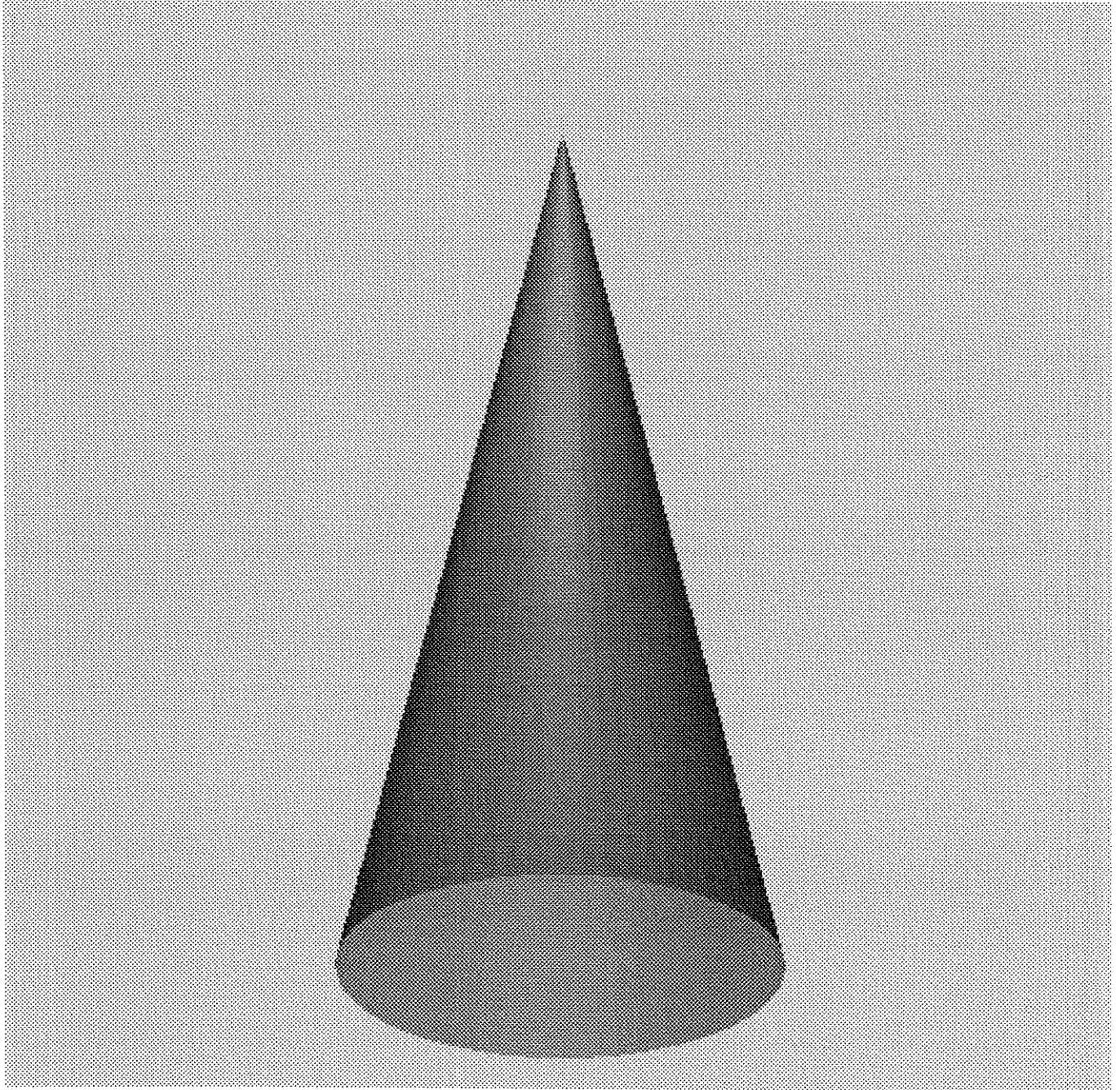


Figura 5.8(b) *Rendering* (Bouknight) do Cone.

d) FORMA COMBINADA

Raio externo da base maior	:	7 cm
Raio interno da base maior	:	5 cm
Raio externo da base menor	:	4 cm
Raio interno da base menor	:	2 cm
Altura	:	15 cm
Espaçamento entre cortes	:	1 cm
Nº total de faces triangulares	:	2220

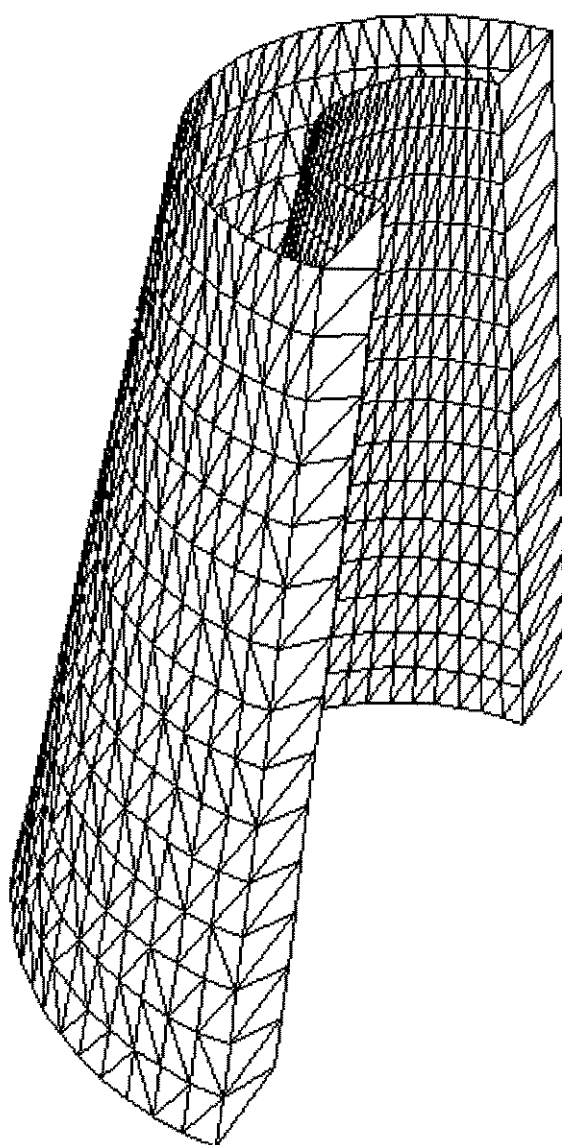


Figura 5.9(a) Forma Combinada.

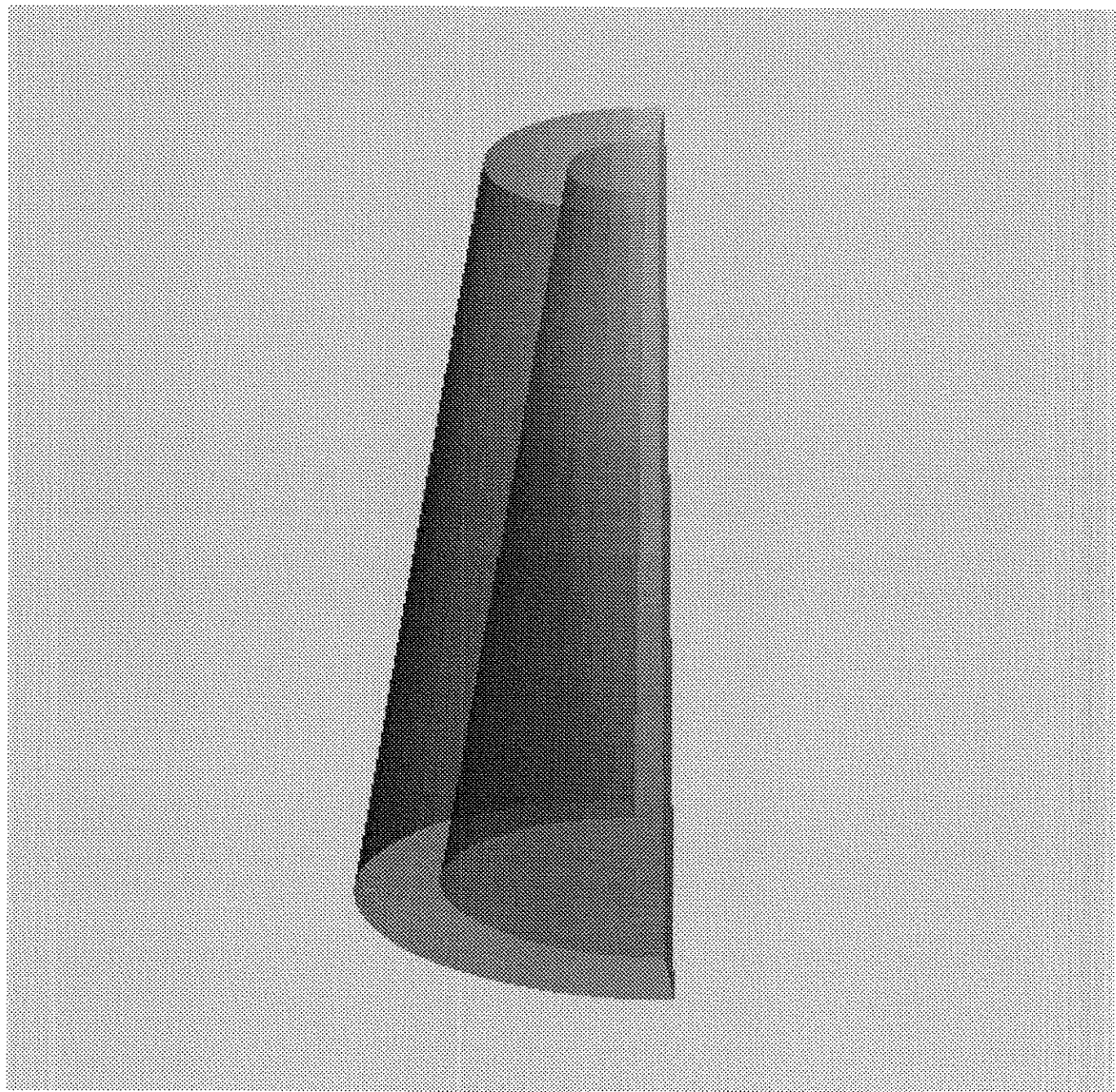


Figura 5.9(b) *Rendering* (Bouknight) da Forma Combinada.

e) RAMIFICAÇÃO

Número de cortes transversais : 19 cm
Espaçamento entre cortes : 2 cm
Nº total de faces triangulares : 736

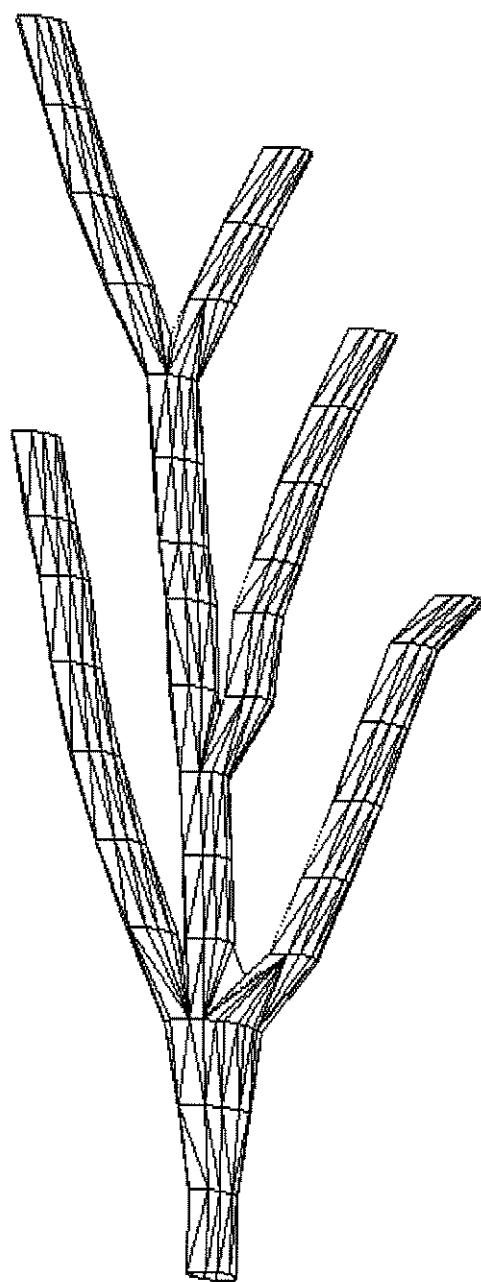


Figura 5.10(a) Ramificação.

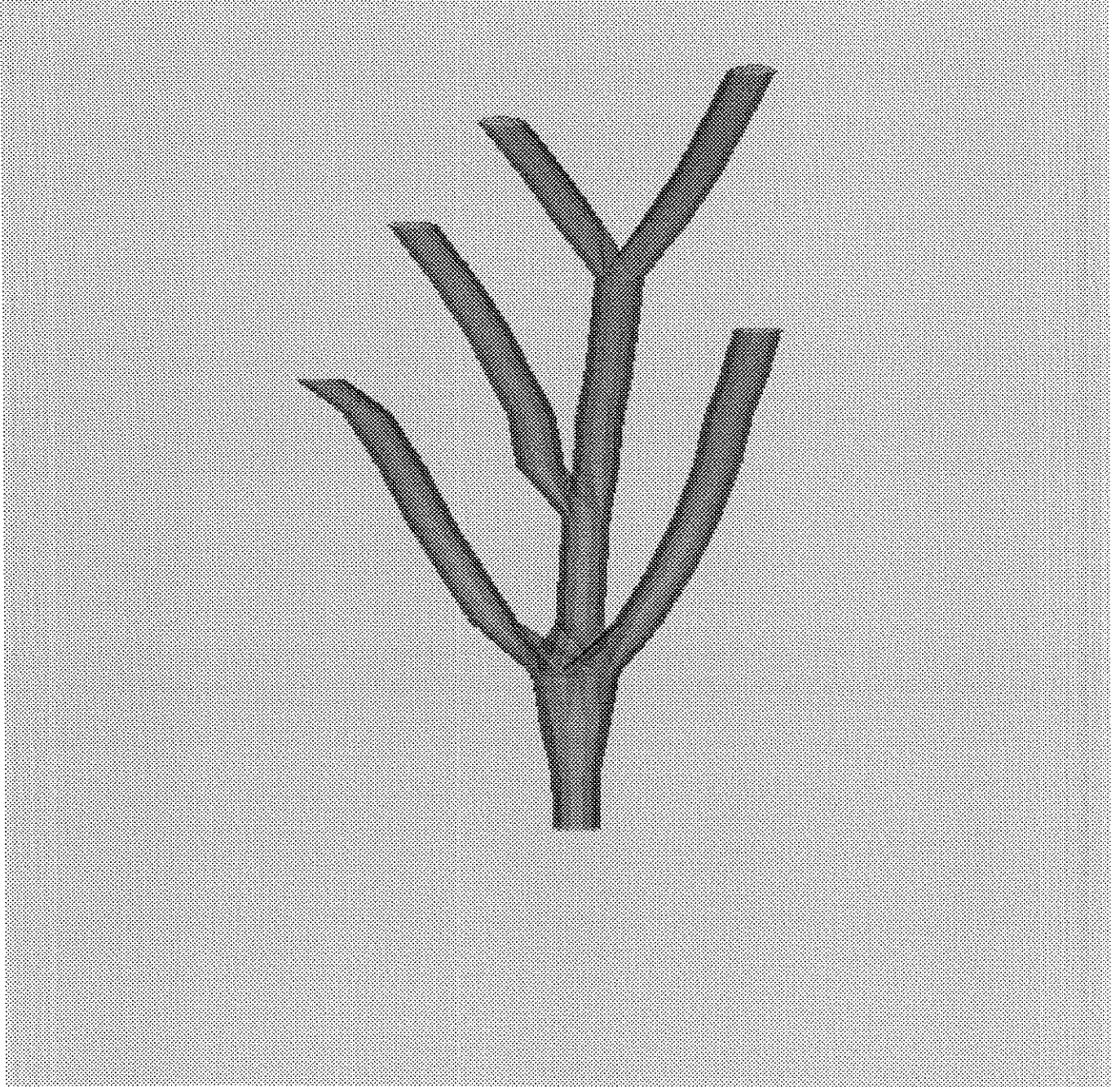


Figura 5.10(b) *Rendering* (Bouknight) da Ramificação.

f) PIMENTÃO

Número de cortes transversais	:	20
Espaçamento entre cortes	:	0.5 cm
Nº total de faces triangulares	:	1500

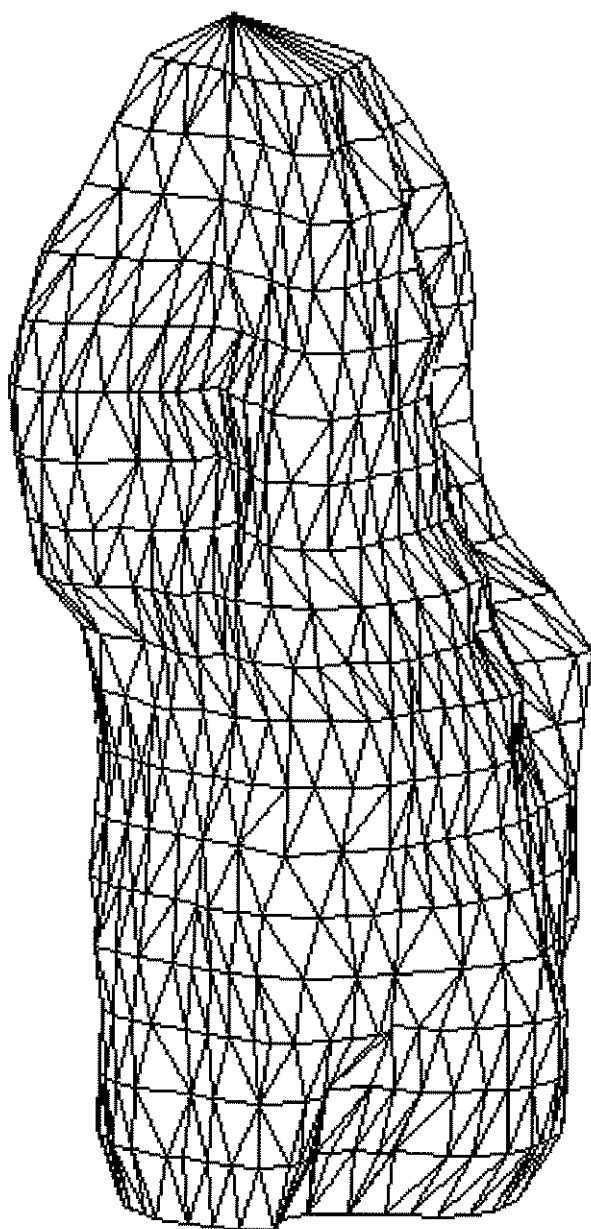


Figura 5.11(a) Pimentão.

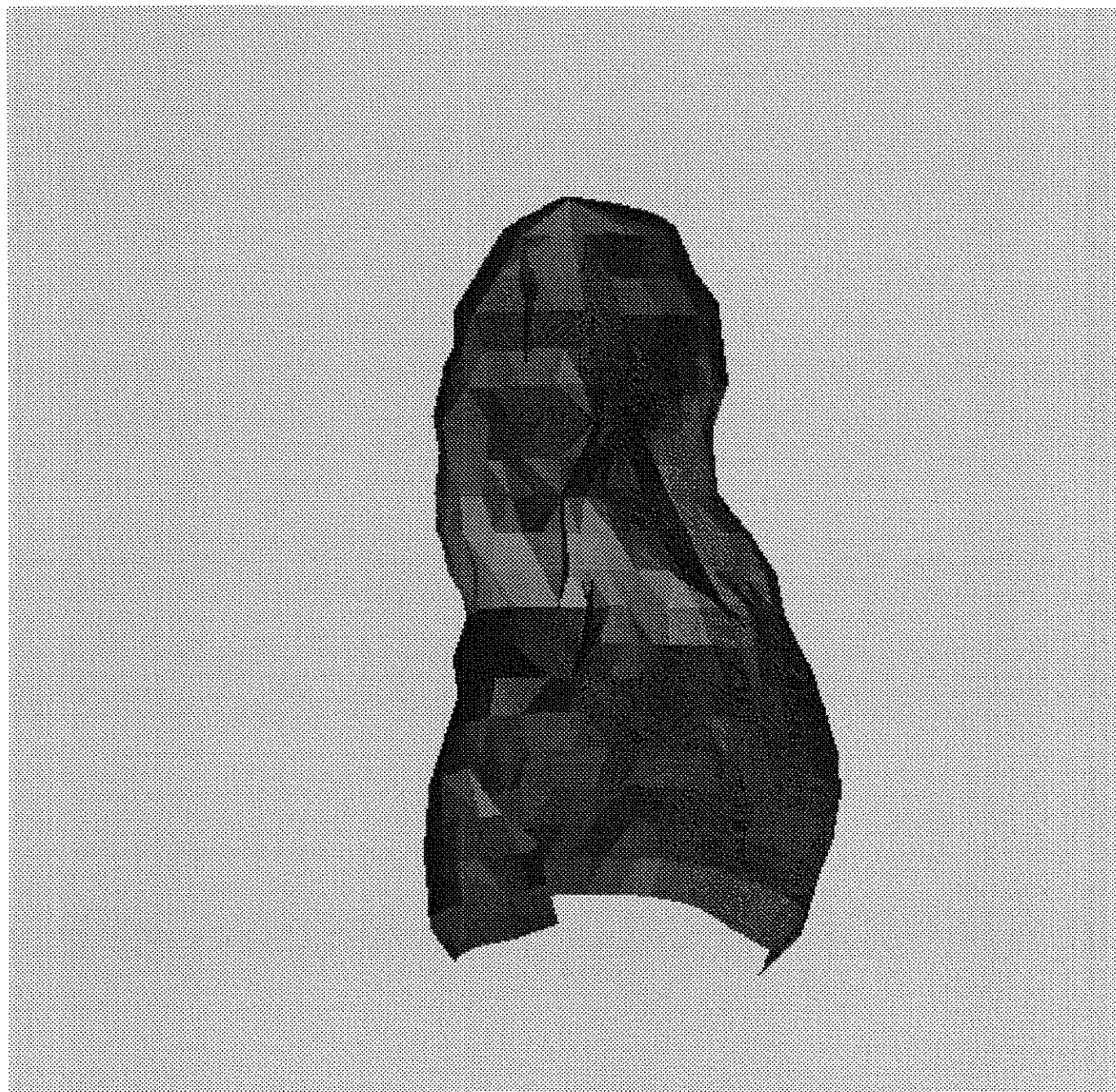


Figura 5.11(b) *Rendering* (Bouknight) do Pimentão.

g) PLANTA

Número de cortes transversais	:	21
Espaçamento entre cortes	:	0.5 mm
Nº total de faces triangulares	:	440

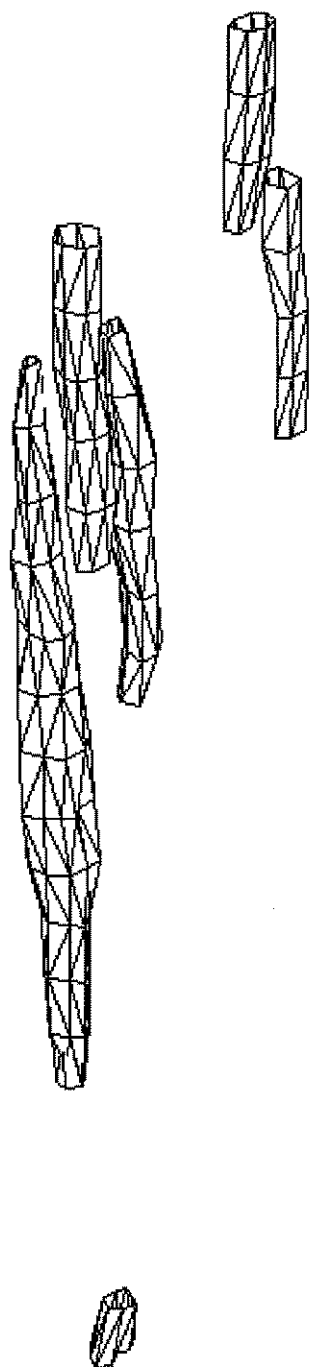


Figura 5.12(a) Plantas.

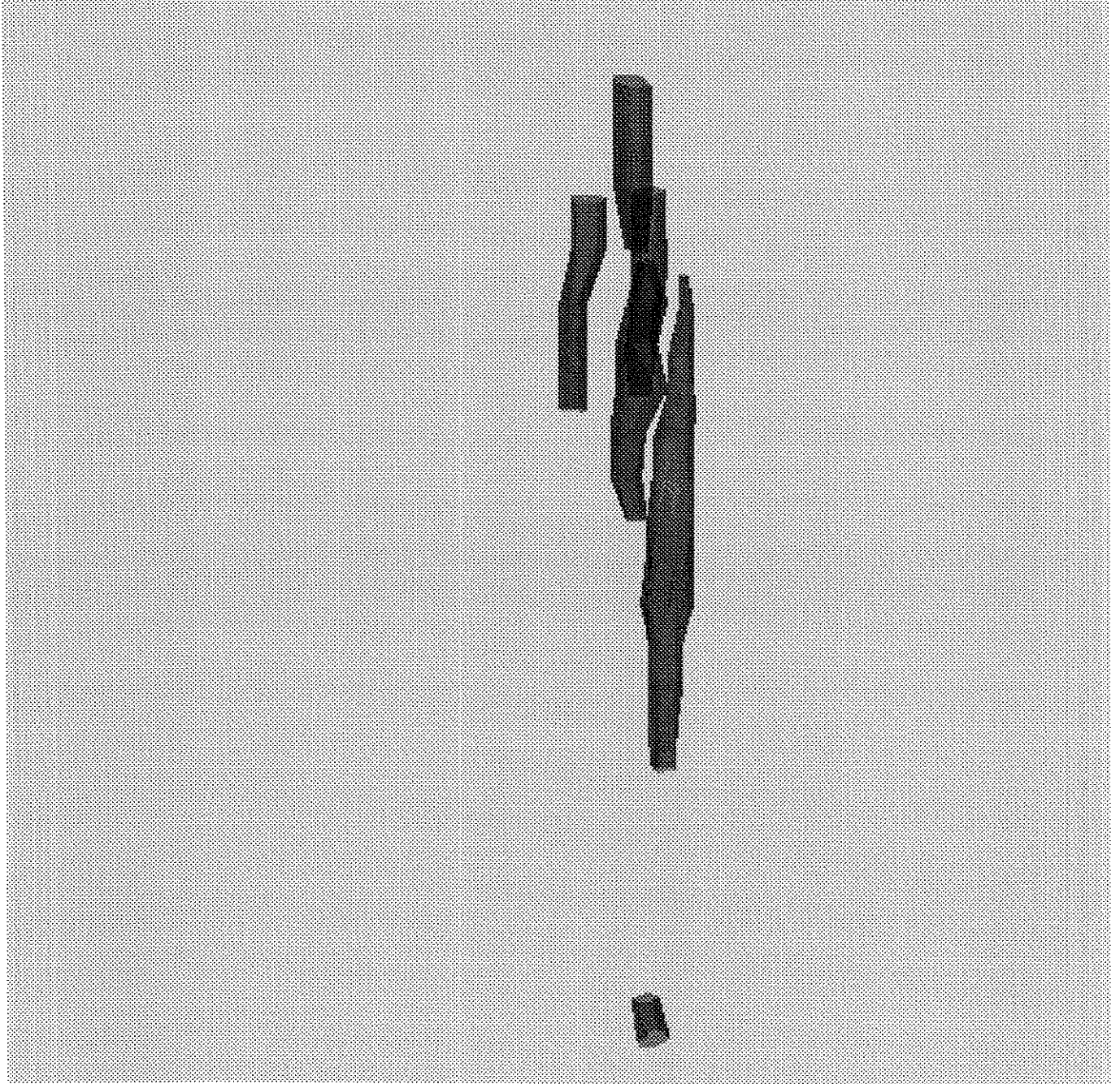


Figura 5.12(b) *Rendering* (Bouknight) das Plantas.

CONCLUSÕES

Este trabalho descreveu uma metodologia para resolução do problema de reconstrução da superfície externa de um objeto definido por suas seções transversais. A técnica utilizada foi a de triangulação e baseou-se em um algoritmo *heurístico*, pois a obtenção de uma solução ótima requereria maior tempo de resposta e maior quantidade de memória. Os resultados apresentados neste trabalho demonstraram ser satisfatórios para grande número de aplicações, sem contudo, exigirem alto esforço computacional.

O objetivo principal da reconstrução é fornecer uma representação tridimensional do objeto de modo a facilitar a compreensão de suas estruturas não apenas através de técnicas de visualização, mas também através da análise de medidas geométricas extraídas do objeto reconstruído.

Este trabalho iniciou-se com uma breve revisão dos principais métodos de reconstrução encontrados na literatura, buscando-se apresentar as características relevantes de cada um. As principais deficiências destes métodos foram eliminadas, de modo a fornecer uma solução para o problema de contornos com forma e orientação não similares; tratar o problema de ramificação de contornos, ou seja, realizar a identificação e ligação de múltiplos contornos em cada seção transversal e, finalmente, operar sem a necessidade de interação com o usuário durante o processo de reconstrução.

Uma ferramenta de reconstrução foi implementada a partir desta metodologia e observou-se que o algoritmo de triangulação comportou-se linearmente segundo o número de vértices do objeto em questão. Desta forma, a redução do número de vértices pode ser conveniente em situações onde se requer menor tempo computacional. Uma forma de se obter esta redução seria, por exemplo, através de taxas de amostragem mais baixas em regiões de menor curvatura nos contornos durante a etapa de pré-processamento (ver figura 1.1).

A avaliação do algoritmo de reconstrução foi realizada através da extração de medidas geométricas dos objetos sintetizados e reais submetidos à ferramenta. As medidas extraídas foram o perímetro e o centróide de cada contorno, além da área superficial e o volume dos objetos reconstruídos, cujos valores apresentaram um erro bastante pequeno em relação às suas formas originais.

A análise de estruturas tridimensionais ainda é, em muitas aplicações, realizada manualmente, envolvendo a aquisição dos cortes transversais, o registro das imagens, a segmentação das regiões de interesse, a reconstrução tridimensional e a extração de parâmetros característicos dos objetos. O resultado destas atividades exige estreita dependência em relação à experiência do pesquisador na área de atuação. A

disponibilidade de uma ferramenta computadorizada de reconstrução e visualização possibilita maior confiabilidade e qualidade dos resultados; redução do tempo referente à análise das estruturas tridimensionais dos objetos; maior abrangência das áreas de aplicação; facilidade em manipular grande quantidade de dados (exemplo, centenas de amostras bidimensionais); maior independência do usuário em relação aos processos inerentes à reconstrução de objetos (aquisição, registro, visualização, etc).

Como proposta de trabalhos futuros de pesquisa na área, uma melhoria dos resultados poderia ser obtida através de um procedimento automático que realizasse o registro entre os contornos baseando-se em pontos característicos de cada contorno. Estes pontos característicos permitem a determinação das transformações de escalamento, translação e rotação necessárias para realizar o registro (alinhamento) dos contornos do objeto. Diversas técnicas são encontradas na literatura. Cygansky e Orr [17] e Faber e Stokely [22] utilizaram a teoria de momentos tensoriais para encontrar as transformações de registro dos contornos. Merickel [52] e Arun, Huang e Blostein [1] basearam-se no critério de mínimos quadrados para encontrar a correspondência entre estes pontos característicos. Técnicas de reconhecimento de padrões [19] também podem ser aplicadas no registro de contornos.

A reconstrução também poderia ser significativamente melhorada caso o algoritmo de triangulação considerasse não apenas a informação referente à posição geométrica dos vértices entre fatias adjacentes, mas se levasse em conta também a distribuição global dos vértices ao longo do objeto. Técnicas de extração de características e de representação de dados 3D poderiam ser incorporadas com o objetivo de orientar o processo de reconstrução.

Outra extensão possível deveria permitir a manipulação de contornos apresentando aberturas em seu interior, sendo necessária a identificação e mapeamento dos contornos denominados internos e externos. Em muitos casos, também a manipulação de contornos constituindo uma seqüência aberta de vértices pode ser interessante. Uma observação a ser considerada, entretanto, é que o método desenvolvido neste trabalho não impõe a restrição de que as seções transversais sejam paralelas entre si.

A experiência obtida no desenvolvimento desta metodologia foi muito importante para estabelecer as dificuldades e necessidades envolvidas na reconstrução de objetos a partir de suas seções transversais. Como resultado deste trabalho, espera-se uma disseminação cada vez maior desta área de pesquisa, em particular nos meios acadêmicos, aumentando a abrangência das áreas de aplicações.

Bibliografia

- [1] K. S. Arun, T. S. Huang e S. D. Blostein. Least-Squares Fitting of Two 3D Point Sets. *IEEE Transactions on Pat. Anal. Mach. Intel*, vol. 9, nº 5, pág. 698-700 (setembro 1987).
- [2] Z. D. Bai, P. R. Krishnaiah, C. R. Rao, P. S. Reddy, Y. N. Sun e L. C. Zhao. Reconstruction of the Left Ventricle from Two Orthogonal Projections. *Computer Vision, Graphics and Image Processing*, vol. 47, pág. 165-188 (1989).
- [3] P. Begeron. A General Version of Crow's Shadow Volumes. *IEEE Computer Graphics and Applications*, pág. 17-28 (setembro 1986).
- [4] J. D. Boissonnat. Representing 2D and 3D Shapes with the Delaunay Triangulation. *Proceedings of IEEE ICASSP*, pág. 745-748 (1984).
- [5] J. D. Boissonnat. Shape Reconstruction from Planar Cross Sections. *Computer Vision, Graphics and Image Processing*, vol. 44, pág. 1-29 (1988).
- [6] J. Bouknight. A Procedure for Generation of Three-Dimensional Half-Toned Computer Graphics Representations. *Communications of the ACM*, vol. 13, nº 9, pág. 522-536 (setembro 1970).
- [7] S. Bright e S. Laflin. Shading of Solid Voxel Models. *Computer Graphics Forum*, vol. 5, pág. 131-137 (1986).
- [8] L. W. Chang, H. W. Chen e J. R. Ho. Reconstruction of 3D Medical Images: A Nonlinear Interpolation Technique for Reconstruction of 3D Medical Images. *Computer Vision, Graphics and Image Processing*, vol. 53, nº 4, pág. 382-391 (julho 1991).
- [9] H. H. Chen e T. S. Huang. A Survey of Construction and Manipulation of Octrees. *Computer Vision, Graphics and Image Processing*, vol. 43, nº 3, pág. 409-431 (setembro 1988).
- [10] L. S. Chen, G. T. Herman, R. A. Reynolds e J. K. Udupa. Surface Shading in the Cuberille Environment. *IEEE Computer Graphics and Applications*, vol. 5, nº 12, pág. 33-43 (dezembro 1985).
- [11] L. S. Chen e M. R. Sontag. Representation, Display and Manipulation of 3D Digital Scenes and their Medical Applications. *Computer Vision, Graphics and Image Processing*, vol. 48, pág. 190-216 (1989).
- [12] C. H. Chien e J. K. Aggarwal. Volume/Surface Octrees for the Representation of Three-Dimensional Objects. *Computer Vision, Graphics and Image Processing*, vol. 36, pág. 100-113 (1986).
- [13] H. N. Christiansen e T. W. Sederberg. Conversion of Complex Contour Line Definitions into Polygonal Element Mosaics. *Computers & Graphics*, vol. 12, nº 3, pág. 187-192 (Agosto 1978).

- [14] L. T. Cook, P. N. Cook, K. R. Lee, S. Batnitzky, B. Y. S. Wong, S. L. Fritz, J. Ophir, S. J. Dwyer III, L. R. Bigongiari e A. W. Templeton. An Algorithm for Volume Estimation Based on Polyhedral Approximation. *IEEE Transactions on Biomedical Engineering*, vol. 2, nº 9, pág. 493-500 (Setembro 1980).
- [15] P. N. Cook, S. Batnitzky, K. R. Lee, L. T. Cook, S. L. Fritz, S. J. Dwyer III e E. J. Charlson. Three-Dimensional Reconstruction from Serial Sections for Medical Applications. *Proceedings of the 14th Hawaii International Conference on System Sciences*, vol. 2, pág. 358-389 (1981).
- [16] T. Cook, S. J. Dwyer III, S. Batnitzky e K. R. Lee. A Three-Dimensional Display System for Diagnostic Imaging Applications. *IEEE Computer Graphics and Applications*, vol. 3, nº 5 (agosto 1983).
- [17] D. Cyganski e J. A. Orr. Applications of Tensor Theory to Object Recognition and Orientation Determination. *IEEE Transactions on Pat. Anal. Mach. Intel.*, vol. 7, nº 5, pág. 662-673 (setembro 1985).
- [18] N. Deo. Graph Theory with Applications to Engineering and Computer Science. *Prentice Hall, Inc., Englewood Cliffs*, New Jersey (1974).
- [19] R. O. Duda e P. E. Hart. Pattern Classification and Scene Analysis. *John Wiley & Sons*, NY (1973).
- [20] S. Dzik e J. Ezrielev. Representing Surfaces with Voxels. *Computer & Graphics*, vol. 16, nº 3, pág. 295-301 (1992).
- [21] A. B. Ekoule, F. C. Peyrin e C. L. Odet. A Triangulation Algorithm from Arbitrary Shaped Multiple Planar Contours. *ACM Transactions on Graphics*, vol. 10, nº 2, pág. 182-199 (abril 1991).
- [22] T. L. Faber e E. M. Stokely. Orientation of 3D Structures in Medical Images. *IEEE Transactions on Pat. Anal. Mach. Intel.*, vol. 10, nº 5, pág. 626-633 (setembro 1988).
- [23] A. X. Falcão, R. A. Lotufo e R. J. Gonçalves. Visualização de Volumes Aplicada à Área Médica. *Anais do V Simpósio Brasileiro de Computação Gráfica e Processamento de Imagens*, pág. 125-134 (novembro 1992).
- [24] T. P. Fang e L. A. Piegl. Delaunay Triangulation Using a Uniform Grid. *IEEE Computer Graphics and Applications*, vol. 5, nº 1, pág. 36-47 (maio 1993).
- [25] G. Frieder, D. Gordon, R. A. Reynolds. Back-to-Front Display of Voxel-Based Objects. *IEEE Computer Graphics and Applications*, vol. 5, nº 1, pág. 52-60 (janeiro 1985).
- [26] H. Fuchs, Z. M. Kedem e S. P. Uselton. Optimal Surface Reconstruction from Planar Contours. *Communications of the ACM*, vol. 20, nº 10, pág. 693-702 (outubro 1977).
- [27] S. Ganapathy e T. G. Dennehy. A New General Triangulation Method for Planar Contours. *Computer Graphics*, vol. 16, nº 3, pág. 69-75 (julho 1982).

- [28] D. Geist e M. W. Vannier. PC-Based 3D Reconstruction of Medical Images. *Computers & Graphics*, vol. 13, nº 2, pág. 135-143 (1989).
- [29] R. C. Gonzalez e P. Wintz. Digital Image Processing. *Addison-Wesley Publishing Company*. 2ª Edição (1987).
- [30] H. Gouraud. Computer Display of Curved Surfaces. *IEEE Transactions on Computers*, vol. 20, nº 6, pág. 623-629 (junho 1971).
- [31] V. G. Gujar, V. C. Bhavsar e N. N. Datar. Interpolation Techniques for 3-D Object Generation. *Computers & Graphics*, vol. 12, nº 3/4, pág. 541-555 (1988).
- [32] T. D. Haig, Y. Attikiouzel e M. Alder. Border Marriage: Matching of Contours of Serial Sections. *IEE Proceedings-I*, vol. 138, nº 5, pág. 371-376 (outubro 1991).
- [33] G. T. Herman. From 2D to 3D Representation. *Mathematics and Computer Science in Medical Imaging* pág. 197-220 (1988).
- [34] K. H. Höhne, M. Bomans, A. Pommert, M. Riemer, C. Schiers, U. Tiede e G. Wiebecke. 3D Visualization of Tomographic Volume Data Using the Generalized Voxel Model. *The Visual Computer*, vol. 6, pág. 28-36 (1990).
- [35] D. P. Huijsmans e G. J. Hense. Representation of 3D Objects Reconstructed from Series of Parallel 2D Slices. *Theoretical Foundations of Computer Graphics and CAD*, pág. 1031-1038 (1988).
- [36] K. A. Jaman, R. Gordon e R. M. Rangayyan. Display of 3D Anisotropic Images from Limited-View Computed Tomograms. *Computer Vision, Graphics and Image Processing*, vol. 30, pág. 345-361 (1985).
- [37] G. J. Jense e D. P. Huijsmans. Interactive Voxel-Based Graphics for 3D Reconstruction of Biological Structures. *Computers & Graphics*, vol. 13, nº 2, pág. 145-150 (1989).
- [38] D. B. Johnson. Efficient Algorithms for Shortest Paths in Sparse Networks. *Journal of The Associations for Computing Machinery*, vol. 24, nº 1, pág. 1-13 (janeiro 1977).
- [39] D. S. Kay. Transparency for Computer Synthesised Images. *Computer Graphics*, vol. 13, nº 2 (agosto 1974).
- [40] N. Kehtarnavaz, L. R. Simar e R. J. P. de Figueiredo. A Syntactic/Semantic Technique for Surface Reconstruction from Cross-Sectional Contours. *Computer Vision, Graphics and Image Processing*, vol. 42, pág. 399-409 (1988).
- [41] N. Kehtarnavaz e R. J. P. de Figueiredo. A Framework for Surface Reconstruction from 3D Contours. *Computer Vision, Graphics and Image Processing*, vol. 42, pág. 32-47 (1988).
- [42] E. Keppel. Approximating Complex Surfaces by Triangulation of Contour Lines. *IBM Journal of Research and Development*, vol. 19, pág. 2-11 (janeiro 1975).

- [43] The Khoros Group. Department of Electrical and Computer Engineering, University of New Mexico, Albuquerque (1991).
- [44] T. L. Kunii, T. Satoh e R. Yamaguchi. Generation of Topological Boundary Representations from Octree Encoding. *IEEE Computer Graphics and Applications*, pág. 29-38 (março 1985).
- [45] M. Levoy. Display of Surfaces from Volume Data. *IEEE Computer Graphics and Applications*, vol. 9, nº 3, pág. 29-37 (maio 1988).
- [46] M. Levoy. Efficient Ray Tracing of Volume Data. *ACM Transactions on Graphics*, vol. 9, nº 3, pág. 245-261 (julho 1990).
- [47] W. C. Lin, S. Y. Chen e C. T. Chen. A New Surface Interpolation Technique for Reconstructing 3D Objects from Serial Cross-Sections. *Computer Vision, Graphics and Image Processing*, vol. 48, pág. 124-143 (1989).
- [48] W. C. Lin, C. C. Liang e C. T. Chen. Dynamic Elastic Interpolation for 3D Medical Image Reconstruction from Serial Cross Sections. *IEEE Transactions on Medical Imaging*, vol. 7, nº 3, pág. 225-232 (setembro 1988).
- [49] G. Q. Maguire Jr. Graphics Applied to Medical Image Registration. *IEEE Computer Graphics and Applications*, pág. 20-28 (março 1991).
- [50] D. Meagher. Octree Generation, Analysis and Manipulation. *Image Processing Laboratory*, Renselaer Polytechnic Institute (1982).
- [51] R. R. Mercer, G. M. McCauley e S. Anjilvel. Approximation of Surfaces in Quantitative 3D Reconstructions. *IEEE Transactions on Biomedical Engineering*, vol. 37, nº 12, pág. 1136-1145 (dezembro 1990).
- [52] M. Merickel. 3D Reconstruction: The Registration Problem. *Computer Vision, Graphics and Image Processing*, vol. 42, pág. 206-219 (1991).
- [53] T. R. Nelson, T. T. Elvis. Visualization of 3D Ultrasound Data. *IEEE Computer Graphics and Applications*, vol. 13, nº 1, pág. 50-57 (novembro 1993).
- [54] D. R. Ney, E. K. Fishman e D. Magid. Volumetric Rendering of Computed Tomography Data: Principles and Techniques. *IEEE Computer Graphics and Applications* (1990).
- [55] H. Pedrini, L. M. A. Silva e C. L. Tozzi. Avaliação da precisão no processo de reconstrução 3D a partir de seções seriadas. *Anais do V Simpósio Brasileiro de Computação Gráfica e Processamento de Imagens*, pág. 155-161 (novembro 1992).
- [56] J. L. Pfaltz e A. Rosenfeld. Computer Representation of Planar Regions by their Skeletons. *Communications of the ACM*, vol. 10, nº 2, pág. 119-125 (fevereiro 1967).
- [57] B. Pham. Conic B-Splines for Curve Fitting: A Unifying Approach. *Computer Vision, Graphics and Image Processing*, vol. 45, pág. 117-125 (1989).

- [58] B. T. Phong. Illumination for Computer Generated Pictures. *Communications of the ACM*, vol. 18, nº 6, pág. 311-317 (junho 1975).
- [59] M. Potmesil. Generating Octree Models of 3D Objects from their Silhouettes in a Sequence of Images. *Computer Vision, Graphics and Image Processing*, vol. 40, pág. 1-29 (1987).
- [60] F. P. Preparata e S. H. Hong. Convex Hulls of Finite Sets of Points in Two and Three Dimensions. *Communications of the ACM*, vol. 20, nº 2, pág. 87-93 (fevereiro 1977).
- [61] T. M. Preto e L. P. Magalhães. *RT 28/91 - ProSim - ScanLine*, nº 28, UNICAMP-FEE-DCA (1991).
- [62] M. Rhodes. Computer Graphics in Medicine. *IEEE Computer Graphics and Applications* (1990).
- [63] D. F. Rogers. Procedural Elements for Computer Graphics. *McGraw-Hill Book Company* (1985).
- [64] H. Samet. Applications of Spatial Data Structures. *Computer Graphics, Image Processing and GIS*. Addison-Wesley Publishing Company, Inc (1990).
- [65] H. Samet. Implementing Ray Tracing with Octrees and Neighbor Finding. *Computer and Graphics*, vol. 13, nº 4, pág. 445-460 (1989).
- [66] H. Samet. The Design and Analysis of Spatial Data Structures. *Addison-Wesley Publishing Company, Inc* (1990).
- [67] J. Sandor. Octree Data Structures and Perspective Imagery. *Computers and Graphics*, vol. 19, nº 4, pág. 393-405 (1985).
- [68] ScanLine Methods for Displaying Parametrically Defined Surfaces. *Communications of ACM*, vol. 23, nº 1, pág. 468-479 (janeiro 1980).
- [69] C. A. Shaffer e H. Samet. Optimal Quadtree Construction Algorithms. *Computer Vision, Graphics and Image Processing*, vol. 37, pág. 402-419 (1987).
- [70] Y. Shinagawa e T. L. Kunii. The Homotopy Model: A Generalized Model for Smooth Generation from Cross Sectional Data. *Technical Report 90-016*, Universidade de Tóquio, Japão, pág. 1-37 (setembro 1990).
- [71] R. Silverman e A. H. Stein. Algorithms for the Decomposition of Convex Polygons. *Contemporary Mathematics*, vol. 119, pág. 19-167 (1991).
- [72] P. G. Spetsieris, V. Dhawan, S. Takikawa, D. Margouleff e D. Eidelberg. Imaging Cerebral Funcional. *IEEE Computer Graphics and Applications*, vol. 13, nº 1, pág. 15-26 (janeiro 1993).
- [73] S. N. Srihari. Representation of Three-Dimensional Digital Images. *Computing Surveys*, vol. 13, nº 4, pág. 399-424 (dezembro 1981).

- [74] S. K. Srivastava e N. Ahuja. Octree Generation from Object Silhouettes in Perspective Views. *Computer Vision, Graphics and Image Processing*, vol. 49, pág. 68-84 (1990).
- [75] I. E. Sutherland, R. F. Sproull e R. A. Schumacker. A Characterization of Ten Hidden-Surface Algorithms. *Computing Surveys*, vol. 6, nº 1, pág. 1-55 (março 1974).
- [76] G. T. Toussaint. Computational Geometry and Computer Vision. *Contemporary Mathematics*, vol. 119, pág. 213-224 (1991).
- [77] Y. Troussel e F. Schmitt. Active-Ray Tracing for 3D Medical. *Imaging Eurographics 1987*, pág. 139-150. (1987).
- [78] J. K. Udupa, D. Odhner. Shell Rendering. *IEEE Computer Graphics and Applications*, vol. 13, nº 1, pág. 58-67 (novembro 1993).
- [79] C. Upson. The Visualisation of Volumetric Data. *Computer Graphics* (1989).
- [80] S. B. Xu e W. X. Lu. Surface Reconstruction of 3D Objects in Computerized Tomography. *Computer Vision, Graphics and Image Processing*, vol. 44, pág. 270-278 (1988).
- [81] *Xlib Reference Manual for Version 11 of the X Window System*. O' Reilly & Associates, Inc.
- [82] *XView Programming Manual. An OPENLOOK Toolkit for X11*. O' Reilly & Associates, Inc.
- [83] S. C. Wu, J. F. Abel e D. P. Greenberg. An Interactive Computer Graphics Approach to Surface Representation. *Communication of the ACM*, vol. 23, nº 10, pág. 703-712 (outubro 1977).
- [84] M. Yau e S. N. Srihari. A Hierarchical Data Structure for Multidimensional Digital Images. *Communications of the ACM*, vol. 26, nº 7, pág. 504-515 (julho 1983).
- [85] M. J. Zyda, A. R. Jones e P. G. Hogan. Surface Construction from Planar Contours. *Computer & Graphics*, vol. 11, nº 4, pág. 393-408 (1987).

Apêndice A

DOCUMENTAÇÃO DA FERRAMENTA DE RECONSTRUÇÃO

O sistema de reconstrução 3D foi implementado em linguagem C sob o sistema operacional UNIX, empregando-se uma estação de trabalho SUN - SparcStation 370 com placa gráfica GXP com capacidade para representar cores em 24 bits e um monitor colorido com resolução de 1152 x 900 pixels.

A ferramenta foi desenvolvida utilizando-se o ambiente X Window, o qual é um sistema gráfico multitarefa baseado em janelas, distribuído e independente do hardware.

A construção das interfaces com o usuário foi significativamente simplificada através do *toolkit* XView [82], provendo objetos gráficos de alto nível tais como janelas, menus, botões, ícones, etc. Deste modo, apenas as rotinas referentes ao desenho de primitivas gráficas (pontos, segmentos de reta, polígonos) e aos atributos do contexto gráfico (espessura das linhas, cores) foram diretamente definidas através da biblioteca Xlib [81].

A arquitetura do sistema é descrita na figura A.1 a seguir. O sistema é dividido em módulos correspondendo a cada uma das etapas do processo de reconstrução, facilitando o seu entendimento e posteriores manutenções.

As funcionalidades que compõem o sistema são ativadas a partir da interface principal ilustrada na figura A.2, cujas características principais são:

- **Apresentar Imagens:** apresenta as imagens originais dos cortes transversais do objeto a ser reconstruído, as quais podem estar no formato GIF ou TGA. As imagens podem ser visualizadas no modo monocromático ou colorido.
- **Apresentar Contornos:** apresenta graficamente os contornos que descrevem a fronteira do objeto a ser reconstruído. Estes contornos são extraídos na etapa de segmentação das imagens e a correspondência entre eles deve estar garantida de modo a evitar, por exemplo, casos de rotação durante o processo de aquisição. Esta opção permite que o usuário visualize cada par de cortes transversais simultaneamente.
- **Analisar Mapeamento:** determina o tipo de ligação a ser realizada entre contornos pertencentes a pares de cortes adjacentes. A ligação entre contornos pode ser do tipo simples (caso 1:1) ou envolver ramificação de contornos (casos 1:n e n:m). Esta decisão de mapeamento é baseada no grau de sobreposição de contornos definido nas seções 3.1.3.1 e 3.1.3.2. Os contornos pertencentes

a um mesmo tipo de ligação são diferenciados dos demais através do uso de cores comuns.

- **Alterar Mapeamento:** permite que o usuário possa modificar o mapeamento dos contornos realizado automaticamente pela ferramenta. Esta interação está disponível de modo a resolver possíveis casos de ambigüidades durante o processo de mapeamento.
- **Reconstruir Objeto:** aproxima a superfície externa entre pares de contornos consecutivos através de uma coleção de faces triangulares, conforme algoritmo de triangulação descrito na seção 3.1.
- **Apresentar Medidas:** apresenta o número total de faces triangulares, a área superficial e o volume total do objeto reconstruído, cujas estimativas são realizadas a partir do algoritmo descrito na seção 4.1.
- **Apresentar Relatório:** apresenta uma tabela contendo, para cada contorno, o número de vértices, a área e o centróide calculados em cada *slice*. Ao final, é apresentado o número total de faces triangulares, a área superficial e o volume total do objeto reconstruído. A tabela IV ilustra um exemplo de relatório para um determinado objeto.
- **Visualizar Objeto:** faz o *rendering* do objeto reconstruído através do programa ScanLine Z-Buffer, o qual faz parte do ambiente PROSIM (Prototipação e Síntese de Imagens Foto-Realistas) também desenvolvido inteiramente pelo próprio grupo do DCA/FEE. A conversão entre o formato da triangulação resultante da reconstrução do objeto e o modelo de representação B-Rep utilizado pelo ScanLine é realizada de forma automática pela ferramenta de reconstrução. O usuário deve apenas especificar determinados parâmetros de visualização, tais como resolução da imagem, posição do observador, transformações geométricas, coeficiente de transparência e iluminação, cuja interface é mostrada na figura A.3.
- **Fim:** finaliza a execução da ferramenta de reconstrução.

RELATÓRIO REFERENTE AO ARQUIVO: OBJETO.DAT					
SLICE	CONTORNO	Nº VÉRTICES	PERÍMETRO	ÁREA	CENTRÓIDE
1.0	1	39	402	10205.8	(440.7;498.3)
2.0	1	48	465	14525.0	(433.2;488.4)
3.0	1	42	471	15339.4	(430.7;510.3)
	2	40	489	13291.7	(428.5;508.6)
	3	37	486	14321.5	(432.4;512.7)
4.0	1	44	488	16889.4	(429.4;511.6)
	2	40	493	17138.3	(417.3;506.2)
5.0	1	43	475	17971.4	(428.7;502.1)
6.0	1	45	495	18148.5	(426.8;517.9)
Número Total de Faces Triangulares					716
Área Superficial do Objeto					19786.3
Volume do Objeto					218715.7

Tabela IV Exemplo de relatório de medidas de um objeto.

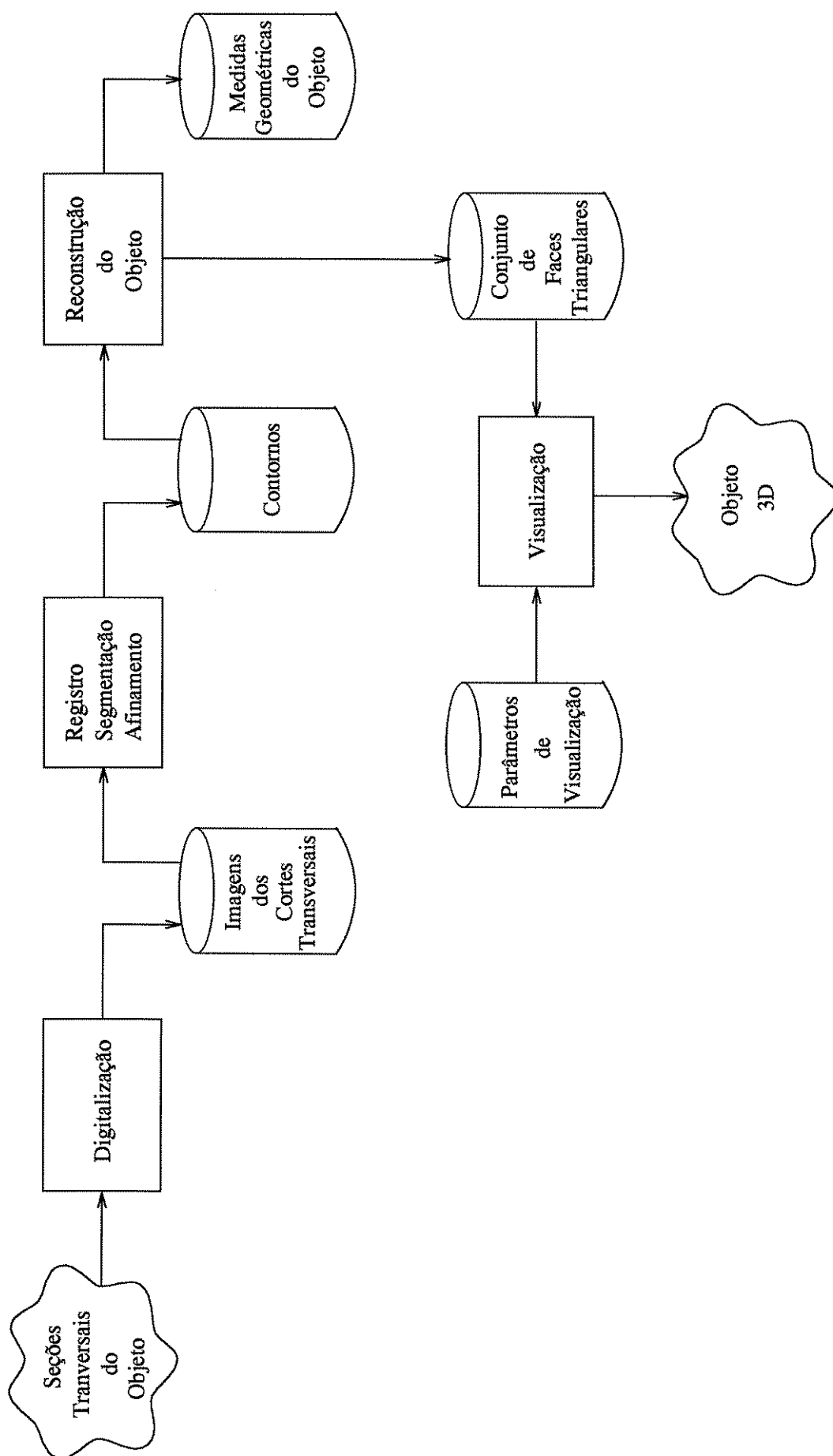


Figura A.1 Arquitetura da ferramenta de reconstrução

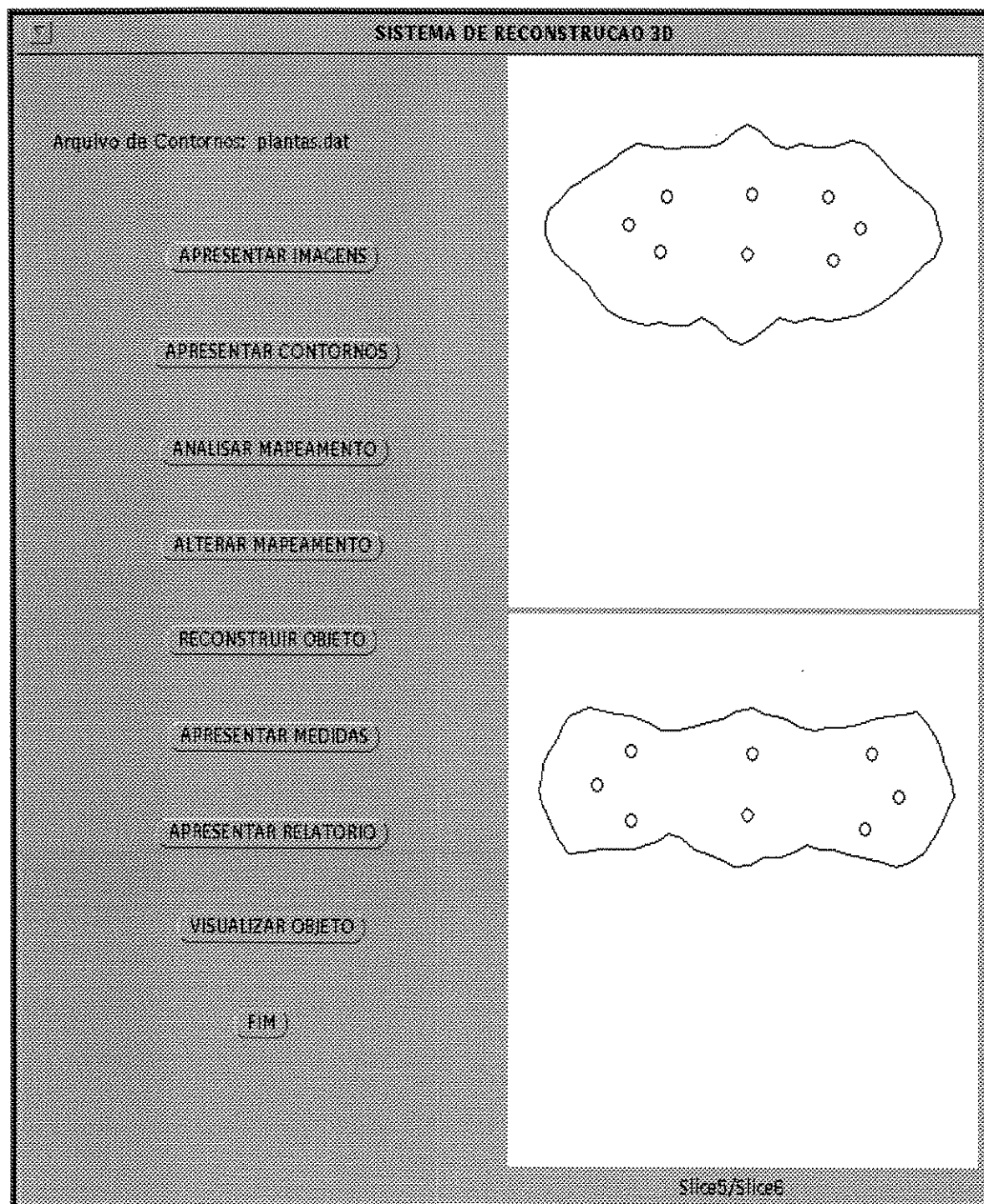


Figura A.2 Interface principal da ferramenta.

PARAMETROS DE VISUALIZACAO			
ILUMINACAO:	POSICIONAMENTO:	TRANSFORMACAO:	SHADING:
Ambiente: X: 55000 Y: 55000 Z: 55000	Observador: X: -2 Y: 5 Z: 4	Escalamto: X: 1.5 Y: 1.5 Z: 1.5	☒ Bouknight ☐ Gouraud ☐ Phong
Lamp_Light: X: 60000 Y: 60000 Z: 60000	Window: X1: -0.05 Y1: -0.05 X2: 0.1 Y2: 0.05	Translacao: X: 1 Y: 1 Z: 1	
Spot_Light: X: 60000 Y: 60000 Z: 60000	VRP: X: 0 Y: 0 Z: 0	Rotacao: X: 1 Y: 1 Z: 1	
Sun_Light: X: 60000 Y: 60000 Z: 60000	Resolucao: X: 800 Y: 900		OK

Figura A.3 Interface de parâmetros de visualização.

FORMATO DOS ARQUIVOS DE ENTRADA E SAÍDA

- **Arquivo de Contornos:** arquivo de entrada que contém os vértices dos contornos em cada corte transversal. O formato do arquivo é definido da seguinte forma:

$$\begin{array}{ccccccc}
 n_i & z_i & P_{1,1}^i(x, y) & P_{2,1}^i(x, y) & P_{3,1}^i(x, y) & \dots & P_{m,1}^i(x, y) \\
 & & P_{1,2}^i(x, y) & P_{2,2}^i(x, y) & P_{3,2}^i(x, y) & \dots & P_{m,2}^i(x, y) \\
 & & \vdots & & & & \\
 & & P_{1,n_i}^i(x, y) & P_{2,n_i}^i(x, y) & P_{3,n_i}^i(x, y) & \dots & P_{m,n_i}^i(x, y)
 \end{array}$$

onde,

n_i : número de contornos existentes no i -ésimo corte transversal

z_i : profundidade z correspondente ao i -ésimo corte transversal

$P_{j,k}^i(x, y)$: coordenadas x e y do j -ésimo vértice pertencente ao k -ésimo contorno do i -ésimo corte transversal

- **Arquivo de Faces Triangulares:** arquivo de saída que contém todas as faces triangulares do objeto reconstruído. É utilizado pelo programa de visualização ScanLine e possui o seguinte formato:

$$\begin{array}{l}
 0 \\
 n_p \\
 n_f \\
 P_1(x, y, z) \\
 P_2(x, y, z) \\
 \vdots \\
 P_{n_p}(x, y, z) \\
 3 \text{ Face}_1 \\
 3 \text{ Face}_2 \\
 \vdots \\
 3 \text{ Face}_{n_f}
 \end{array}$$

onde,

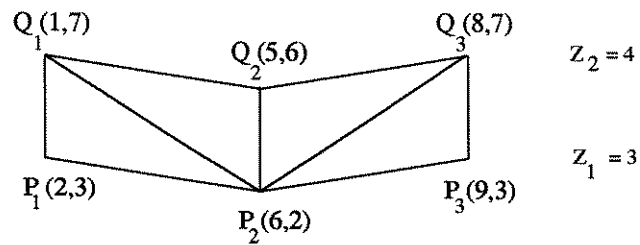
0: indica que o objeto possui faces poligonais planas

n_p : número total de pontos do objeto

n_f : número total de faces do objeto

$P_i(x, y, z)$: i -ésima face triangular do objeto, a qual é formada pelos índices dos pontos listados na ordem acima

O exemplo da figura A.4 mostra a triangulação entre dois segmentos de contornos e o respectivo arquivo de faces triangulares no formato ScanLine.



```

0
6
4
2 3 3
6 2 3
9 3 3
1 7 4
5 6 4
8 7 4
3 1 2 4
3 2 4 5
3 2 5 6
3 2 3 6

```

Figura A.4 Exemplo de arquivo de faces triangulares obtidas através da triangulação entre dois contornos adjacentes.

Os pseudocódigos relativos aos principais módulos da ferramenta de reconstrução são apresentados a seguir.

apresentar_imagens()

```
{
  atribuir modo monocromático ou colorido
  identificar formato da imagem (tga ou gif)
  para i=1 até n slices faça
    mostrar imagem do slicei (cor)
}
```

apresentar_contornos()

```
{
  para cada par de slice faça
    {
      separar contornos
      mostrar contornos do slicei na janela 1
      mostrar contornos do slicei+1 na janela 2
    }
}
```

analisar_mapeamento()

```
{
  para cada par de slice faça
    {
      inicializar matriz de mapeamento
      calcular grau de sobreposição entre contornos
      montar matriz de sobreposição
      identificar ligação segundo casos 1:1, 1:n e n:m
    }
}
```

alterar_mapeamento()

```
{
  para cada par de slice faça
    {
      atribuir grau de sobreposição desejado entre contornos
      alterar matriz com novo grau de sobreposição
      identificar ligação segundo casos 1:1, 1:n e n:m
      apresentar_contornos()
    }
}
```

```

reconstruir_objeto()
{
    analisar_mapeamento()
    ativar caso correspondente:
        . caso_1:1()
        . caso_1:n()
        . caso_n:m()

    extrair medidas do objeto
}

caso_1:1()
{
    transformar contorno do slicei
    mapear contorno em quadrado unitário

    transformar contorno slicei+1
    mapear contorno em quadrado unitário

    realizar triangulação entre contornos
}

caso_1:n()
{
    transformar contorno do slicei
    mapear contorno em quadrado unitário

    transformar contorno slicei+1
    mapear contorno em quadrado unitário
    interpolar os contornos

    realizar triangulação entre contornos
}

caso_n:m()
{
    transformar contorno do slicei
    mapear contorno em quadrado unitário
    interpolar os contornos

    transformar contorno slicei+1
    mapear contorno em quadrado unitário
    interpolar os contornos
}

```

```

    realizar triangulação entre contornos
}

transformar()
{
    se contorno  $C$  não é convexo então
    {
        processar convex hull do contorno  $C$ 
        processar as concavidades de 1ª ordem:  $C_1^1, C_2^1, \dots, C_n^1$ 
        para  $k = 1$  até  $n$  faça
        {
            transformar( $C_k^1$ )
            realizar projeção de  $C_k^1$  em seu convex hull
        }
    }
}

apresentar_medidas()
{
    apresentar (tela) as medidas extraídas:
    . nº de faces triangulares
    . área superficial
    . volume total
}

apresentar_relatório()
{
    apresentar (tela) relatório contendo:
    . nº de vértices, área, perímetro e centróide de cada contorno
    . área superficial e volume total dos objetos reconstruídos
}

visualizar()
{
    definir parâmetros de visualização:
    . iluminação
    . posição do observador
    . VRP (View Reference Point)
    . resolução da imagem
    . transformação (escala, translação, rotação)
    . coeficiente de transparência
    . cor

```

```
ler arquivos de vértices dos contornos
ler arquivos de faces triangulares obtidas pela reconstrução
converter formato para padrão Scanline
ativar módulo de visualização do Scanline
}
```

```
fim()
{
    fechar arquivos
    apagar arquivos temporários
    fechar interfaces gráficas
}
```