

**Guilherme Rossetti Anzolin**

***Interpolação de Curvas com Mudança Topológica***

Joinville / SC

Junho de 2006

**Guilherme Rossetti Anzolin**

***Interpolação de Curvas com Mudança Topológica***

Orientador: Prof. Marcelo da Silva Hounsell, PhD

UDESC - UNIVERSIDADE DO ESTADO DE SANTA CATARINA  
CENTRO DE CIÊNCIAS TECNOLÓGICAS  
DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO

Joinville / SC

Junho de 2006

Monografia sob o título “*Interpolação de Curvas com Mudança Topológica*”, defendida por Guilherme Rossetti Anzolin e aprovada em Junho de 2006, em Joinville, Santa Catarina, pela banca examinadora constituída pelos professores:

---

Prof. Marcelo da Silva Hounsell, PhD  
Departamento de Ciéncia da Computaçao - UDESC  
Orientador

---

Prof. Alexandre Gonçalves Silva, MSc  
Departamento de Ciéncia da Computaçao - UDESC

---

Prof. Roberto Silvio Ubertino Rosso Jr, PhD  
Departamento de Ciéncia da Computaçao - UDESC

---

Prof. Rogério Eduardo da Silva, MSc  
Departamento de Ciéncia da Computaçao - UDESC

*Aos meus pais e minha irmã, que sempre me deram carinho e incentivo,  
durante toda minha vida.*

*À minha namorada por todo carinho e apoio que recebi.*

## *Agradecimentos*

- Inicialmente à Deus por tudo que tenho;
- À meu orientador por seu apoio em todo momento;
- Aos meus caros amigos que me apoiaram durante todo o curso;
- Aos professores que contribuíram para o desenvolvimento desta monografia.

*“A mente que se abre a uma nova idéia jamais volta ao seu tamanho original”*

*Albert Einstein*

## ***Resumo***

Neste trabalho, uma técnica de reconstrução tridimensional baseada na interpolação de curvas com mudança de topologia será apresentada. Para resolver este problema, que tem, por exemplo, aplicações na área médica e na representação computacional de terrenos, conceitos e técnicas de modelagem geométrica serão pesquisados, como a criação e representação de modelos tridimensionais a partir de fatias bidimensionais. Um software será implementado para demonstrar a solução proposta para este problema de maneira eficiente.

## *Abstract*

In this work a three-dimensional reconstruction technique based on interpolation of curves with changing topology will be presented. To solve this problem, that has applications, for example, in the medical area and in the computational representation of terrains, concepts of geometric modeling will be studied, as the synthesis and representation of three-dimensional models through the bidimensional slices. A software will be implemented to demonstrate the solution proposal for this problem in an efficient way.

# ***Lista de Figuras***

1.1	Reconstrução de um crânio. Fonte: Bajaj et al (1996) . . . . .	14
1.2	Reconstrução de vasos sanguíneos. Fonte: Nonato et al (2005) . . . . .	14
1.3	Interpolação de curvas . . . . .	15
2.1	Representação Matricial e Poligonal. Fonte: Gatass e Peixoto (2000) . . . . .	18
2.2	Diferentes cálculos de centróide . . . . .	19
2.3	Curvas fechadas convexa e côncava . . . . .	20
2.4	Curvas intersectantes e auto-interceptante . . . . .	20
2.5	Representação por superfícies. Fonte: Castelo Filho (1998) . . . . .	21
2.6	Geometria e Topologia . . . . .	22
2.7	Documento VRML . . . . .	23
2.8	Documento XML . . . . .	25
3.1	Diferentes Correspondências . . . . .	27
3.2	Bifurcações - <i>Branching</i> . Fonte: Gatas e Peixoto (2000) . . . . .	28
3.3	Triangulação - <i>Tiling</i> . Fonte: Meyers et al (1994) . . . . .	29
3.4	Diferentes Malhas - <i>Tiling</i> . Fonte: Bajaj et al (1996) . . . . .	30
3.5	Reconstrução a partir de deformações. Fonte: McInerney e Terzopoulos (1996)	31
3.6	Reconstrução a partir de abordagen implícita . . . . .	32
3.7	Geração de malha auxiliada por um Grafo. Fonte: Meyers et al (1994) . . . . .	33

3.8	Superfícies geradas por algoritmos diferentes. Fonte: Meyers et al (1994) . . . . .	33
3.9	Definição da correspondência . . . . .	36
3.10	Objeto reconstruído sem bifurcação. Fonte: Gatass e Peixoto (2000) . . . . .	37
3.11	Definição dos discos para as curvas. Fonte: Treece et al (1999) . . . . .	37
3.12	Comparação de distâncias entre centróides. Fonte: Treece et al (1999) . . . . .	38
3.13	Resultado da correspondência e objeto final. Fonte: Treece et al (1999) . . . . .	39
3.14	Diferentes valores para $\beta$ . Fonte: Vargas (2001) . . . . .	40
3.15	Classificação e eliminação de tetraedros. Fonte: Vargas (2001) . . . . .	41
4.1	Diferentes abordagens para reconstrução. Fonte: Meyers et al (1992) . . . . .	43
4.2	Matriz de distâncias . . . . .	44
4.3	Curva sem nenhuma correspondência . . . . .	45
4.4	Representação <i>B</i> -rep em VRML . . . . .	46
4.5	Diferentes geração de malha . . . . .	47
4.6	Solução para a geração de malha . . . . .	48
4.7	Mapeamento dos pontos . . . . .	48
4.8	Local da bifurcação - <i>Branching</i> . . . . .	49
4.9	Arquitetura do $\Delta$ -connection . . . . .	49
4.10	GUI-2D de edição das curvas . . . . .	51
4.11	Armazenamento em XML . . . . .	52
4.12	Pseudo-código do $\Delta$ -connection . . . . .	53
4.13	Função <i>Tiling</i> . . . . .	53
A.1	Interpolação de curvas . . . . .	62

A.2 Reconstrução de um fêmur. Fonte: Gattass e Peixoto (2000) . . . . .	63
A.3 Cronograma . . . . .	65

# ***Sumário***

<b>1</b>	<b>Introdução</b>	<b>13</b>
1.1	Objetivos . . . . .	15
1.2	Estrutura do Trabalho . . . . .	16
<b>2</b>	<b>Fundamentação Teórica</b>	<b>17</b>
2.1	Curvas . . . . .	17
2.1.1	Representação de curvas . . . . .	17
2.1.2	Centróide . . . . .	18
2.1.3	Tipos de curvas . . . . .	19
2.2	Criação e Representação de Sólidos . . . . .	20
2.2.1	Representação B-rep . . . . .	20
2.3	VRML - <i>Virtual Reality Modeling Language</i> . . . . .	22
2.4	XML - <i>eXtensible Markup Language</i> . . . . .	24
<b>3</b>	<b>Reconstrução 3D</b>	<b>26</b>
3.1	Fundamentos . . . . .	26
3.1.1	Correspondência . . . . .	26
3.1.2	Bifurcação . . . . .	28
3.1.3	Geração de Malha . . . . .	29

3.1.4	Abordagens da Reconstrução 3D . . . . .	30
3.2	Estado da Arte . . . . .	35
3.2.1	Algoritmo de Barequet e Sharir (1996) . . . . .	36
3.2.2	Algoritmo de Treece, Prager, Gee e Berman (1999) . . . . .	37
3.2.3	Algoritmo $\beta$ -Connection (2001) . . . . .	39
<b>4</b>	<b>Algoritmo <math>\Delta</math>-connection</b>	<b>42</b>
4.1	Correspondência no $\Delta$ -connection . . . . .	43
4.2	Geração de Malha no $\Delta$ -connection . . . . .	45
4.3	Bifurcação no $\Delta$ -connection . . . . .	48
4.4	Arquitetura do $\Delta$ -connection . . . . .	49
4.5	Editor de curvas . . . . .	50
4.6	Representação da curva . . . . .	52
4.7	Reconstrução 3D: $\Delta$ -connection . . . . .	53
<b>5</b>	<b>Considerações Finais</b>	<b>56</b>
<b>Referências</b>		<b>58</b>
<b>Apêndice A – Plano de TCC</b>		<b>60</b>

# 1    *Introdução*

A Computação Gráfica é a área da computação destinada à geração de imagens em geral, tanto na forma de representação de dados quanto na forma de recriação do mundo real. A área de aplicação é enorme e vem crescendo a cada dia. Uma das áreas que evoluiu foi a Visualização Científica. Esta área visa resumir dados científicos (fluxo de fluidos, relatividade, reações químicas e nucleares, etc) e identificá-los através de representações gráficas a fim de facilitar a interpretação desses dados, que geralmente são em grande quantidade [Traina e Oliveira, 2004].

Segundo Vargas (2001), a aparição de dispositivos de medição não invasivos, como Tomografia por Ressonância Magnética e Tomografia Computadorizada, possibilitou a visualização de uma seqüência de seções planares de objetos tridimensionais. Este fato motivou o desenvolvimento de várias técnicas de Reconstrução Tridimensional.

A Reconstrução Tridimensional então passou a ser uma técnica de bastante interesse e pesquisa, pois a partir de imagens bidimensionais extraídas por tais dispositivos é possível determinar o modelo 3D do objeto que foi analisado. A Figura 1.1 ilustra uma das fatias obtidas por Tomografia Computadorizada (a); o processo de reconstrução executado sobre duas dessas fatias (b) e o processo finalizado (c) de reconstrução de um crânio.

Um outro exemplo de reconstrução a partir de fatias é a representação de canais, onde a visualização do modelo como um todo é mais importante que a visualização de detalhes entre as fatias. A Figura 1.2 mostra seções bidimensionais que representam vasos sanguíneos (a) e em seguida a interpolação destas fatias gerando todos os canais (b).

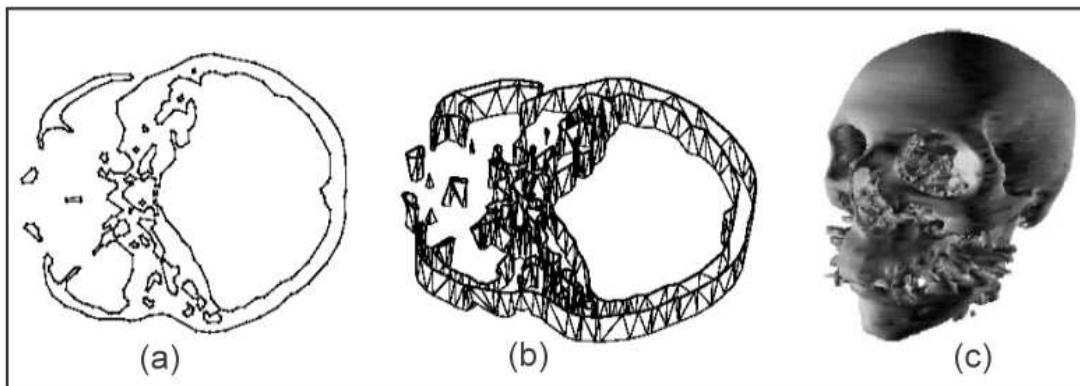


Figura 1.1: Reconstrução de um crânio. Fonte: Bajaj et al (1996)

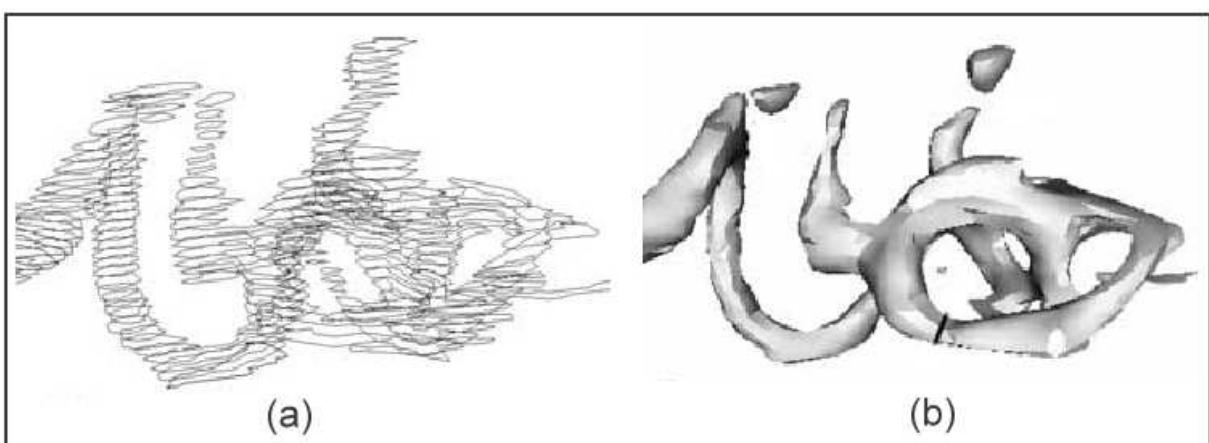


Figura 1.2: Reconstrução de vasos sanguíneos. Fonte: Nonato et al (2005)

Neste Trabalho de Conclusão de Curso serão estudadas técnicas de modelagem geométrica para reconstrução tridimensional através da interpolação de curvas considerando mudança de topologia.

Uma definição formal para o problema seria:

Dado um conjunto de curvas  $c_{ij}$ , onde  $i=1, 2, \dots, n$  e  $n$  é o número de curvas diferentes no plano  $x_j$ , criar um objeto  $O$  sendo que as curvas  $c_{ij}$  sejam a interseção de  $O$  com os planos  $x_j$ . Cada curva  $c_{ij}$  é representada por uma seqüência de  $p_i$  pontos.

O que se diz aqui sobre mudanças topológicas é o fato de que a reconstrução não é feita apenas de uma curva para uma outra, mas pode ocorrer que uma curva se relacione a duas outras, ou seja, como se estivesse ocorrendo uma metamorfose da curva para uma situação topológica diferente, em um plano diferente. A Figura 1.3 ajuda a ilustrar o problema, mostrando

à esquerda as curvas fechadas em seus respectivos planos, e na ilustração à direita a superfície do objeto gerado a partir da interpolação.

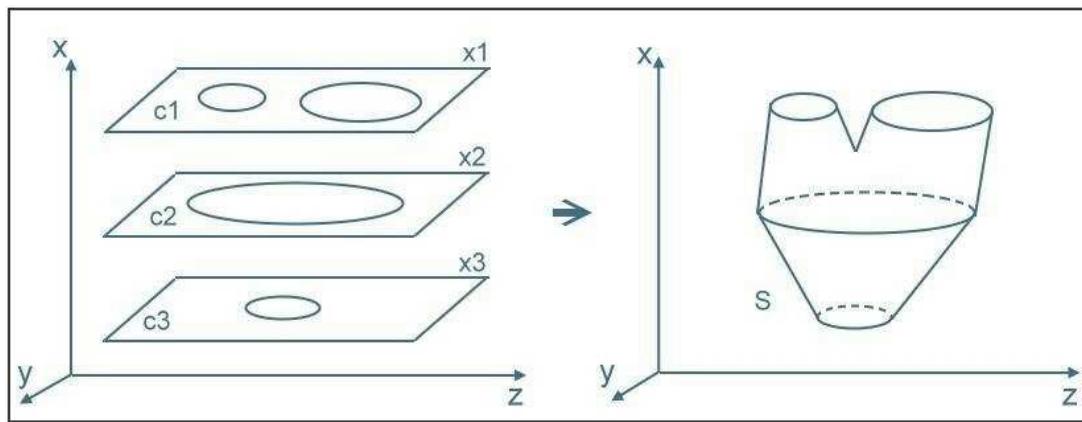


Figura 1.3: Interpolação de curvas

Como a Figura 1.3 pode mostrar, uma fatia  $c_i$  pode conter uma ou mais curvas fechadas independentes no mesmo plano, gerando assim uma mudança topológica na interpolação entre as fatias  $c_i$  e  $c_{i+1}$ . Como nada é definido para o comportamento das curvas no entre-planos, percebe-se que podem existir vários objetos que satisfaçam ao enunciado, ou seja, através de seções com mudança topológica é possível realizar diferentes interpolações, gerando diferentes objetos. Este trabalho visa obter uma solução algorítmica para o caso de interpolações lineares, a fim de estabelecer critérios para a interpolação quando ocorrer mudança topológica entre as seções.

## 1.1 Objetivos

O objetivo geral deste Trabalho de Conclusão de Curso (TCC) é realizar a reconstrução tridimensional através do desenvolvimento de uma solução eficiente para a interpolação entre curvas com mudança topológica. Os objetivos específicos são os seguintes:

- Entender os conceitos e técnicas de modelagem geométrica para a criação e representação de objetos;
- Identificar técnicas existentes na literatura que sejam apropriadas à interpolação de curvas (reconstrução 3D);

- Estabelecer uma solução para a interpolação de curvas considerando critérios de eficiência (a serem levantados e estabelecidos);
- Projetar e implementar a solução proposta.

Ao final deste TCC será obtido um melhor entendimento do problema da interpolação de curvas e reconstrução tridimensional; será apresentada a descrição detalhada de uma solução; estará disponível um *software* que demonstre a solução adotada gerando objetos exportados para VRML<sup>1</sup> (*Virtual Reality Modeling Language*) e haverá um sitio com a descrição dos principais aspectos do projeto, o *software*, exemplos de objetos gerados e referências.

## 1.2 Estrutura do Trabalho

Este Trabalho de Conclusão de Curso foi organizado da seguinte maneira:

- O capítulo 2 apresenta uma fundamentação teórica sobre modelagem geométrica e algumas tecnologias que serão utilizadas neste trabalho. Inicialmente são mostrados alguns conceitos sobre curvas e a técnica de representação de sólidos utilizada. Posteriormente serão descritos alguns fundamentos em relação às tecnologias escolhidas para a modelagem dos objetos e para o armazenamento de dados;
- O capítulo 3 apresenta um estudo feito sobre reconstrução 3D. Primeiro são descritos alguns fundamentos e abordagens utilizadas para a reconstrução 3D, seguidas de um comparativo. Posteriormente são apresentados, em ordem cronológica, três trabalhos correlatos ao tema proposto, desenvolvidos a partir de uma mesma abordagem de reconstrução;
- O capítulo 4 vai detalhar a solução proposta para este trabalho, apresentando as etapas de reconstrução que serão desenvolvidas e um projeto da implementação, demonstrando a arquitetura e um pseudo-código do algoritmo proposto;
- No capítulo 5 serão apresentadas as considerações finais do texto.

---

<sup>1</sup>Linguagem que permite descrever elementos geométricos e ambientes virtuais

## 2 Fundamentação Teórica

Os objetos reconstruídos devem ser armazenados de forma a atenderem requisitos de Modelagem Geométrica. A Modelagem Geométrica é uma área da Computação Gráfica que estuda, através de métodos matemáticos, mecanismos de criação de modelos geométricos, tratando da síntese, manipulação e topologia dos objetos gráficos no computador [Dieguez, 1989; Foley et al, 1996].

### 2.1 Curvas

Na Modelagem Geométrica, a representação de curvas é a base para a descrição de vários modelos, desde formas simples até objetos complexos, e tem um papel importante também para a visualização de fenômenos científicos [Azevedo e Conci, 2003]. A seguir serão revistos alguns conceitos de curvas que serão utilizados neste trabalho.

#### 2.1.1 Representação de curvas

Existem basicamente duas maneiras de representar as curvas (contornos): a representação matricial e a representação poligonal. A Figura 2.1 apresenta uma mesma curva sendo representada das duas maneiras.

Na representação matricial, a curva é descrita por cada ponto que é armazenado em um arranjo bidimensional (matriz) que é formada por  $i \times j$  elementos, esses elementos são retângulos denominados *pixels*. Um elemento  $(i,j)$  pode estar sobre, dentro ou fora do contorno que representa a curva, Os contornos são definidos por todos os elementos (*pixels*) situados sobre

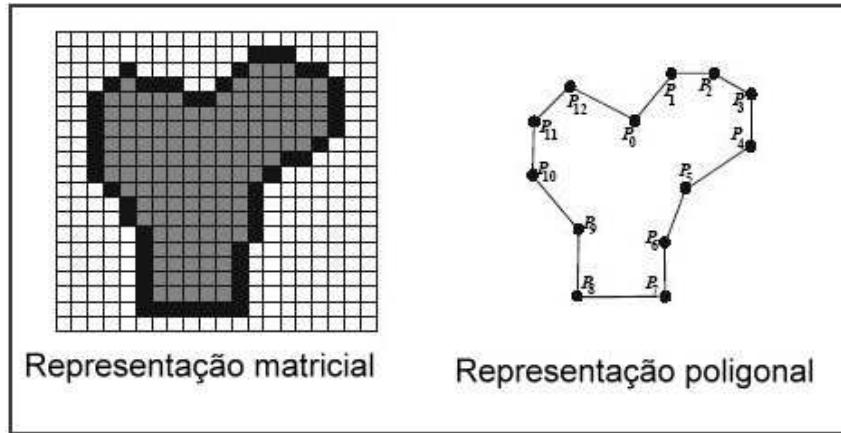


Figura 2.1: Representação Matricial e Poligonal. Fonte: Gatass e Peixoto (2000)

eles. A forma matricial de representação é utilizada para formar as imagens nas telas da maioria dos dispositivos de saída gráficos, como impressora e vídeos [Gattass e Peixoto, 2000]. Na representação poligonal, também denominada vetorial, a curva pode ser representada por uma seqüência de retas, sendo que cada reta é definida pelas coordenadas de seus pontos extremos. Assim a curva é descrita pelas coordenadas de seus respectivos vértices [Azevedo e Conci, 2003].

### 2.1.2 Centróide

A partir da representação poligonal de curvas pode-se obter algumas características como, por exemplo, a região central aproximada da mesma, que é calculada com base nas coordenadas dos vértices, e é uma informação necessária em muitas situações práticas como, por exemplo, para definir o ponto de lançamento automático de textos gráficos para identificação de elementos em tela. Esta região central é também chamada na literatura de “centróide” e pode ser calculada de diferentes maneiras, como através do centro de um círculo inscrito ou circunscrito à curva ou pela média aritmética de todos os vértices da curva (que é uma forma freqüentemente usada) [Figueiredo e Carvalho, 1991]. A Figura 2.2. apresenta três resultados de cálculo do centróide para uma mesma curva fechada: em (a) o centróide é calculado pela média aritmética dos pontos da curvas; em (b) é feita a média aritmética apenas dos pontos x e y extremos da curva; e em (c) é utilizado um círculo inscrito a partir dos pontos da curva e então calculado o centro deste

círculo.

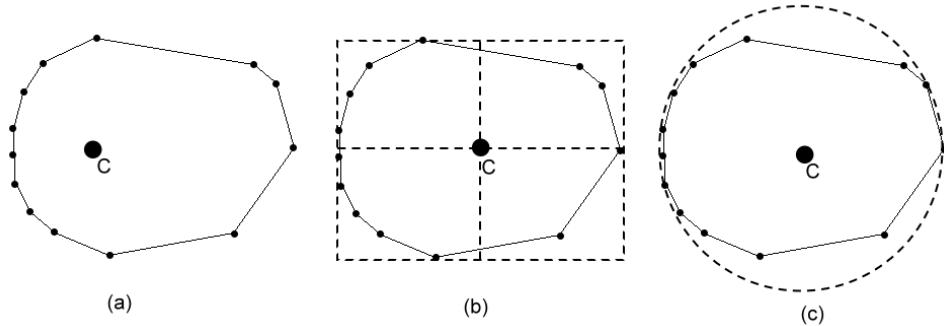


Figura 2.2: Diferentes cálculos de centróide

Outra técnica para calcular o centróide de uma curva é através do “centro de gravidade”, esta técnica consiste em dividir a curva em triângulos e fazer uma relação do centro de gravidade de todos eles, já que o centro de gravidade de um triângulo pode ser obtido facilmente pela média aritmética das coordenadas de seus vértices, esta técnica é precisa, porém de grande custo computacional. O cálculo do centróide a partir da média dos vértices da curva, ilustrado em (a) na Figura 2.2, é uma solução de baixo custo computacional, porém pode ter o resultado afetado por características do objeto, por exemplo, a concentração de vértices em uma região da curva causa um deslocamento indesejável no centróide em direção a essa região [Figueiredo e Carvalho, 1991], é o que pode ser observado pelo posicionamento do centróide na Figura 2.2 (a) em relação à (b) e (c). Por outro lado, a técnica representada em (b) na Figura 2.2, onde é feita a média dos pontos extremos da curva (técnica de *Bounding Box*), é de baixo custo computacional e não depende da distribuição dos pontos na curva, ou seja, resolve o cálculo do centróide de maneira satisfatória.

### 2.1.3 Tipos de curvas

Este trabalho irá tratar de curvas fechadas para a interpolação. A curva poderá ser convexa ou côncava, a Figura 2.3 ilustra essas duas situações, sendo que a diferença está que nas curvas côncavas um segmento de reta traçado a partir de dois pontos quaisquer ( $p_1, p_2$ ) do interior da curva pode passar por fora desta mesma curva.

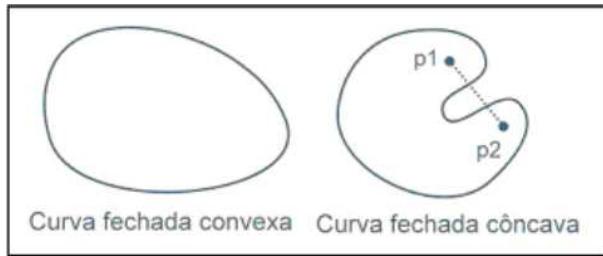


Figura 2.3: Curvas fechadas convexas e côncavas

Existem casos em que duas ou mais curvas do mesmo plano possuam pontos de seu interior em comum, tais curvas são denominadas intersectantes. Outro caso que merece ser mencionado é se uma curva possuir pontos de seu contorno em comum, neste caso ela é denominada auto-interceptante. A Figura 2.4 ilustra estes casos demonstrando os pontos  $p$  que identificam a intersecção e a auto-interceptação.

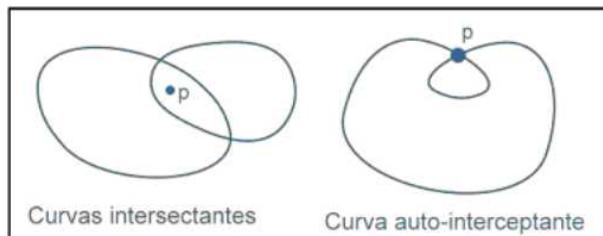


Figura 2.4: Curvas intersectantes e auto-interceptante

Neste trabalho serão tratadas para a interpolação as curvas fechadas côncavas e convexas, sendo elas não intersectantes nem auto-interceptantes

## 2.2 Criação e Representação de Sólidos

Os objetos 3D reconstruídos são armazenados utilizando-se uma técnica de representação de sólidos denominada *B-rep* em arquivo VRML, para possibilitar a sua visualização posteriormente. A seguir é descrita a técnica *B-rep* de representação de sólidos.

### 2.2.1 Representação B-rep

Através de superfícies, podem ser representados objetos 3D, esta é uma forma de representação denominada *Boundary Representation (B-rep)*, e consiste na descrição de um objeto

em termos de seus limites de superfícies: vértices, arestas e faces [Foley et al, 1996]. Então, nesta técnica representação, o sólido é descrito como um conjunto de faces, estas limitadas por arestas, que por fim são limitadas por vértices, como representado na Figura 2.5. É através desta relação que é definida a topologia que, juntamente com informações geométricas (coordenadas dos vértices), representa o objeto final [Castelo Filho, 1998].

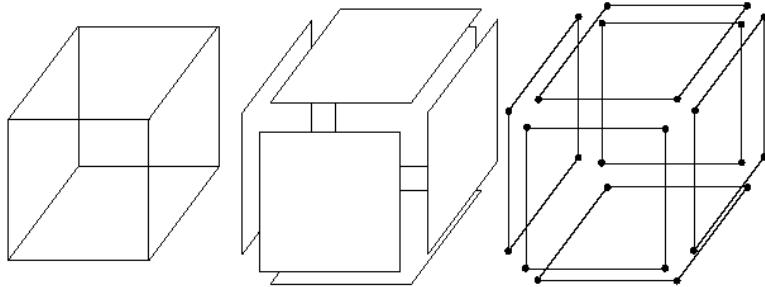


Figura 2.5: Representação por superfícies. Fonte: Castelo Filho (1998)

A representação de sólidos por superfícies possui vantagens sobre a representação por volume, que descreve o objeto varrendo todo o volume de dados que o representam. A representação volumétrica de modelos 3D é muito útil para a representação de modelos que não tem forma definida (como gases ou líquidos), pois não seria possível definir com facilidade sua superfície. A principal vantagem na utilização da técnica de representação de superfícies, em relação a métodos de representação volumétrica é a quantidade de informação armazenada, pois para gerar as faces (superfícies) de um objeto são necessárias bem menos informações do que para gerar seu volume [Gattass e Peixoto, 2000].

Existem diferentes estruturas de dados para a representação por superfícies, sendo que a principal delas, que é utilizada neste trabalho, é a baseada em arestas. Neste tipo de estrutura as faces são descritas através de um ciclo de arestas, determinando a topologia do objeto, e os vértices são obtidos indiretamente a partir dessas arestas, definindo a geometria. A geometria descreve a posição (coordenadas) de cada vértice do objeto no espaço. A topologia representa a relação dos vértices, ou seja, a seqüência em que eles se conectam formando então as faces do objeto, sempre descritos em um mesmo sentido (horário ou anti-horário) para todas as faces [Azevedo e Conci, 2003]. A Figura 2.6 ilustra a definição da geometria (as coordenadas dos

quatro vértices) e a topologia (seqüência horária, representada pelas setas, dos vértices definindo as quatro faces) de um tetraedro.

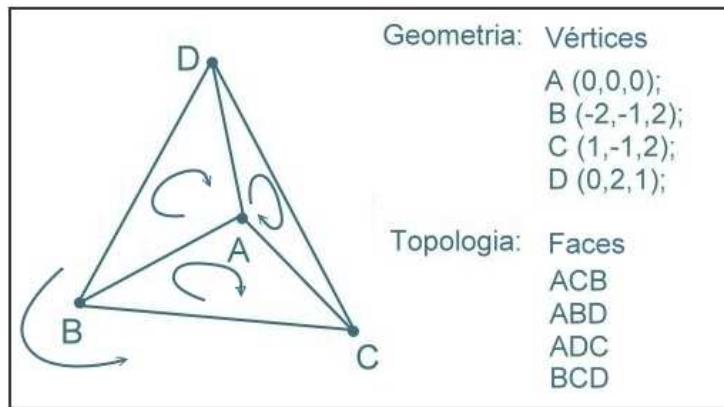


Figura 2.6: Geometria e Topologia

Segundo Speck (2001), a técnica *B-rep* possui vantagens sobre os outros métodos de modelagem, principalmente na geração de modelos complexos e na velocidade de verificação de relações topológicas. Isto ocorre pela maneira como o *B-rep* registra as informações do modelo, armazenando os parâmetros das arestas de forma explícita. Uma vez construída a superfície, aplica-se então técnicas de *rendering* conhecidas para visualizar o objeto.

## 2.3 VRML - Virtual Reality Modeling Language

VRML é uma linguagem de marcação para modelagem em Realidade Virtual e serve para a descrição de objetos e mundos tridimensionais interativos. A versão atual da linguagem (VRML 2.0) trabalha principalmente com a geometria 3D dos objetos que são elaborados a partir de primitivas geométricas (cones, esferas, cilindros e esferas) ou da definição de faces. Também é permitido o uso de objetos multimídia, como sons e filmes, para a interação com o usuário e a programação de eventos para gerar animações. Para a visualização de arquivos VRML é necessário configurar o navegador para que ele interprete o conteúdo do arquivo que está recebendo. Adiciona-se então um *plug-in*, que é um programa que adapta o navegador para que este suporte arquivos de tipos diferenciados. A linguagem VRML utiliza o sistema de coordenadas cartesiano para a modelagem, ou seja, um sistema baseado em três eixos de deslocamento

vetorial, usando a unidade de medidas “metros”, por integrar facilmente os ambientes virtuais escritos por usuários diferentes sem problemas de proporção [Carrard, 2001].

O Arquivo VRML tem a extensão *.wrl* e pode ser editado por qualquer editor de texto. A estrutura do arquivo é formada basicamente pelo cabeçalho VRML e os nós onde são descritos os modelos geométricos. O cabeçalho é obrigatório e representa a versão da linguagem e o conjunto de caracteres utilizado. Os nós são estruturas sintáticas utilizadas para representação dos objetos, a composição de um nó contém o tipo do nó, um par de chaves e um conjunto de atributos do objeto que esta sendo descrito. O arquivo VRML pode contar também com comentários, um comentário de linha é iniciado pelo símbolo “#”.

A Figura 2.7 apresenta um exemplo de código VRML: a primeira linha é o cabeçalho e depois é descrito um nó raiz chamado *shape*, utilizado para descrever os objetos modelados e formado por dois nós internos: o nó *appearance* e o nó *geometry*. O nó *appearance* têm a função de descrever a aparência do objeto a ele associado, o campo *material* define os atributos de aparência do material que compõe o objeto, no caso da Figura 2.7, é aplicada a cor verde através do parâmetro *diffuseColor*. O nó *geometry* define a forma tridimensional do objeto, normalmente ele define uma das primitivas geométricas, como na Figura 2.7, onde é desenhado um cubo através da primitiva *Box*, que define um prisma regular de quatro lados definidos pelo parâmetro *size*.

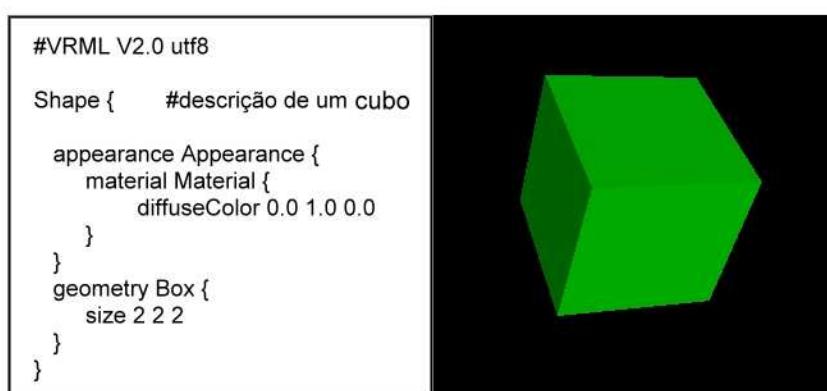


Figura 2.7: Documento VRML

O VRML permite, além de instanciar primitivas geométricas para modelagem, definir a geometria de qualquer tipo de objeto, descrevendo as faces do mesmo. Este trabalho utiliza

deste recurso que é descrito no capítulo 4, onde a proposta é detalhada.

## 2.4 XML - eXtensible Markup Language

XML é uma linguagem de marcação desenvolvida pela W3C (*World Wide Web Consortium*) que permite ao programador estruturar dados de uma maneira flexível adaptando de acordo com a aplicação. Este padrão foi desenvolvido para permitir o armazenamento, recuperação e transmissão de dados de forma simples e estruturada [Décio, 2000].

Um documento XML pode ser editado e visualizado por qualquer editor de texto simples. É formado basicamente por *tags* e dados, os *tags* são *strings* limitadas pelos caracteres “<” e “>” e os dados são as *strings* limitadas pelos *tags* que representam o conteúdo armazenado no documento. Cada elemento pertencente ao documento do XML começa com uma *tag* de abertura e termina com uma de fechamento, a *tag* de fechamento é diferenciada da de abertura por uma barra “/” que aparece antes do seu nome. Esta marcação dos dados permite introduzir informações sobre os dados, pois é o programador que define as *tags* que irá utilizar em seu documento [Graves, 2003].

A Figura 2.8 demonstra um exemplo de arquivo XML, formado inicialmente por uma declaração que especifica a versão do XML e o conjunto de caracteres utilizado, na segunda linha é escrito o primeiro elemento do documento, chamado também de elemento raiz, que termina na última linha com o *tag* de fechamento. Os demais elementos são descritos dentro do elemento raiz e podem ser formado por sub-elementos ou então pelo conteúdo que está sendo armazenado.

Para a organização da estrutura do documento XML podem ser utilizados recursos como o DTD (*Document Type Definition*), que é uma especificação para definir a estrutura do documento (elementos, atributos e notações que serão permitidos), ou o XSD (*XML Schema Definition*), recurso mais recente que funciona de forma análoga ao DTD, porém contando com algumas características a mais para tratar alguns problemas apresentados pelo DTD. Através destes recursos é possível, então, definir um tipo de estrutura, que deverá ser respeitada, para

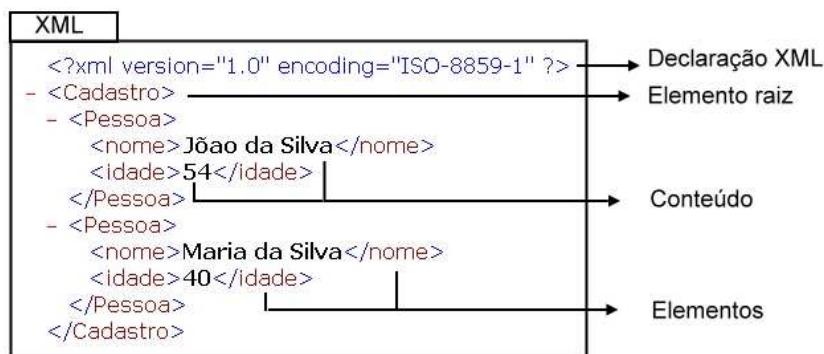


Figura 2.8: Documento XML

organizar os elementos de um documento XML [D  cio, 2000].

A XML    muito   til para transfer  ncia de dados entre banco de dados e aplicativos, e tem sido utilizado muito tamb  m para o armazenamento de dados. Uma das principais vantagens da utiliz  o de XML como banco de dados    pelo fato de sua estrutura ser mais expressiva do que a estrutura de bancos de dados relacionais, isso por que ela pode conter diferentes elementos, variar a ordem e a quantidade de atributos e possuir elementos iguais, o que torna mais f  cil de representar dados complexos [Graves, 2003].

Neste trabalho ser   utilizada XML para o armazenamento das informa  es relativas   s curvas que ser  o geradas. A forma como ser   utilizada a tecnologia para o armazenamento desses dados, sem a utiliz  o de recursos para a defini  o de estruturas (DTD ou XSD),    apresentada no cap  tulo 4.

# 3 *Reconstrução 3D*

## 3.1 Fundamentos

O problema de reconstrução tridimensional através de seções bidimensionais conta com alguns algoritmos que foram desenvolvidos no decorrer dos anos e, diante dos estudos levantados, nota-se que todos esses algoritmos levam em consideração as seguintes etapas para gerar o modelo geométrico desejado: a correspondência, a geração de malha e a ramificação. A seguir serão explicadas e exemplificadas essas etapas.

### 3.1.1 Correspondência

O problema de correspondência (*correspondence*) acontece quando existe mais de uma curva fechada em um dos planos paralelos, ou nos dois, o que pode gerar uma mudança topológica. A etapa de definição da correspondência é considerada o principal problema da reconstrução 3D a partir de seções planares [Vargas, 2001]. Neste caso é preciso decidir qual será a correspondência das curvas entre os planos, ou seja, quais curvas de um plano que vão ser conectadas com quais outras do outro plano. A Figura 3.1 apresenta três das várias soluções de correspondência que podem ser geradas a partir das duas seções iniciais ilustradas no canto superior esquerdo. Em um dos casos cada curva de um plano é conectada com uma única mais próxima do plano seguinte (a), em um outro caso uma das curvas do plano inferior conecta-se com as duas do plano superior, enquanto a outra conecta-se apenas com a mais próxima (b) e na ultima ilustração apenas uma das curvas do plano inferior se conecta com as curvas do plano superior (c).

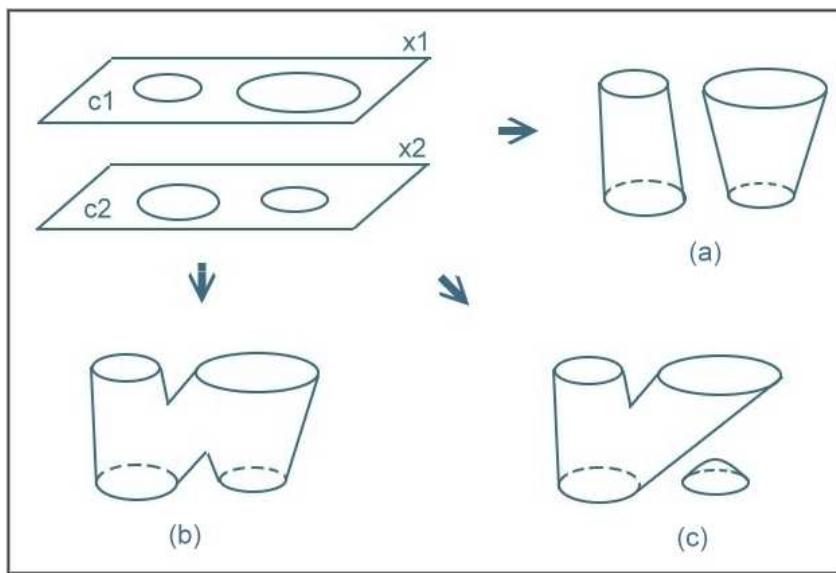


Figura 3.1: Diferentes Correspondências

O problema de correspondência só se torna trivial quando a relação entre as curvas nos planos é de 1 para 1 ( $1 \leftrightarrow 1$ ), ou seja, uma curva em cada plano, onde as duas curvas se conectam (superfície contínua) ou não (superfície descontínua). No caso de um dos planos conter uma curva e o outro mais de uma ( $1 \leftrightarrow n$ ) o problema de correspondência já deve levar em consideração algum critério para a conexão, e o mesmo acontece quando são tratadas mais de uma curva nos dois planos ( $m \leftrightarrow n$ ), como foi ilustrado na Figura 3.1.

Segundo Gattass e Peixoto (2000), a decisão da correspondência pode ser tomada com base nas informações das distâncias entre as curvas. De um modo geral, se os contornos se encontram muito distantes entre si, não há conexão entre eles. Mas se o espaçamento entre fatias adjacentes não é muito grande, é possível estabelecer a conexão destes contornos.

Porém uma solução somente baseada em distância pode não gerar todos os objetos desejados a partir de um determinado conjunto de fatias. Para escolha da correspondência, então, seria interessante contar com técnicas flexíveis permitindo que as várias soluções de conexão possam ser realizadas.

### 3.1.2 Bifurcação

O problema da bifurcação (*branching*) ocorre quando a relação entre as curvas no entre-planos é de  $1 \leftrightarrow n$  ou  $m \leftrightarrow n$ , ou seja, quando existe mudança topológica entre as duas seções. As bifurcações representam os pontos de sela na superfície gerada, e torna o problema de geração de malha mais delicado, pois a definição do local em que será feita a bifurcação não é uma solução trivial. A partir de duas seções com mudança topológica podem ser feitas bifurcações em diferentes regiões do entre-planos. A Figura 3.2 apresenta soluções diferentes de *branching* dados dois conjuntos de curvas iniciais (a).

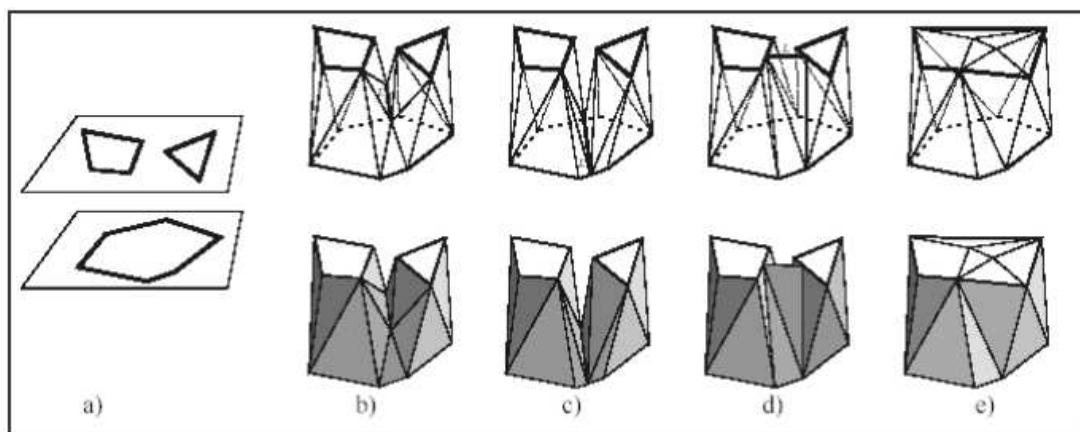


Figura 3.2: Bifurcações - *Branching*. Fonte: Gatas e Peixoto (2000)

Das diferentes soluções ilustradas na Figura 3.2, cada uma representa a bifurcação em um nível do entre-planos. Uma considera a bifurcação justamente no ponto médio entre um plano e o outro (b), o que seria a solução mais próxima do real. Outra solução divide o contorno que não sofre mudança topológica em duas regiões e então efetua o *branching* de cada região com o contorno correspondente (c). A solução (d) efetua o *branching* logo que começa o entre-planos, unindo os contornos com mudança topológica a fim de aumentar a superfície do sólido final. E a ultima solução (e) simplesmente não considera a bifurcação, gerando um contorno que abrange as duas curvas do plano superior, e realizando a interpolação de  $1 \leftrightarrow 1$  com a curva do plano inferior.

Alguns algoritmos trazem soluções eficientes para resolver a etapa de *branching* através de reconstrução volumétrica, outros a consideram apenas como uma consequência da interpolação

com mudança topológica, não tratando detalhadamente a bifurcação. Na seção 3.2 serão demonstrados alguns desses algoritmos que trataram desta etapa.

### 3.1.3 Geração de Malha

A geração de malha *tiling* é o processo de interpolação entre as curvas correspondentes. É nesta etapa que são definidos quais vértices de uma curva serão conectados a quais vértices da curva “correspondente”, gerando assim uma malha triangular que representa a superfície do objeto final.

Com as curvas representadas na forma poligonal o processo de geração de malha consiste na conexão de todos os vértices das curvas fechadas. Tal processo é tratado na literatura como triangulação [Bajaj et al, 1996; Gattass e Peixoto, 2000; Vargas, 2001], pois cada face da superfície final é composta por 3 vértices conectados, sendo dois destes vértices pertencentes à uma curva e um deles à outra. A Figura 3.3 representa o processo de triangulação entre duas curvas, uma representada pelos pontos Q e outra pelos pontos P, e uma das faces formada pelos pontos P<sub>1</sub>, Q<sub>1</sub> e P<sub>2</sub> da superfície que será gerada.

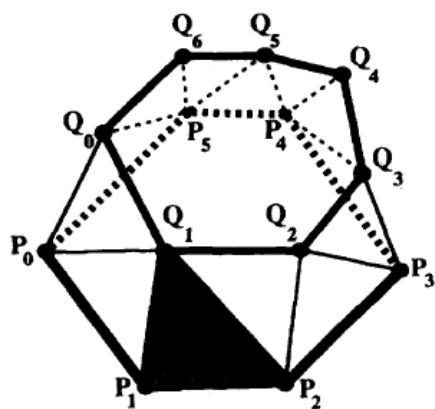


Figura 3.3: Triangulação - *Tiling*. Fonte: Meyers et al (1994)

A geração de malha nem sempre conta com uma solução trivial. A Figura 3.4 ilustra uma projeção de duas curvas (a) estando cada uma em um plano e o processo de geração de malha resultando em duas soluções diferentes. Em uma das soluções (b) a geração de malha é baseada na construção de uma superfície mínima, sem interseção, conectando os pontos mais próximos

de cada curva, na outra solução (c) é feita a geração de malha sem levar em consideração a relação espacial dos pontos das duas curvas, ou seja, um ponto da extrema esquerda da curva superior pode se conectar com um ponto da direita da curva inferior, construindo uma superfície intersectante.

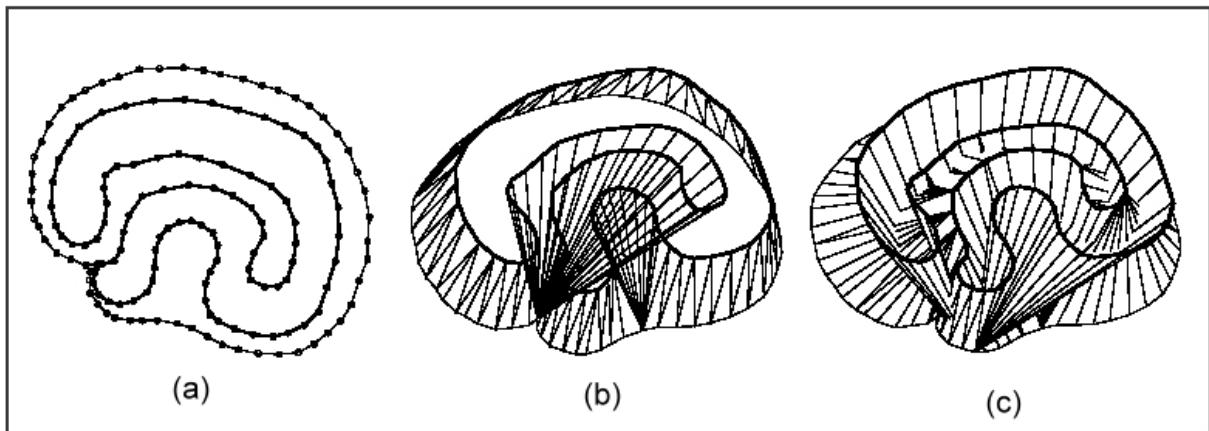


Figura 3.4: Diferentes Malhas - *Tiling*. Fonte: Bajaj et al (1996)

A consideração da topologia dos contornos é essencial para gerar uma boa triangulação. Se entre as duas seções a topologia for muito diferente e não houver uma relação entre os pontos das curvas que serão interpoladas, a superfície gerada pode não ser um dos resultados esperados, como no exemplo (c) da Figura 3.4.

### 3.1.4 Abordagens da Reconstrução 3D

Tendo em vista que o problema de reconstrução tridimensional a partir de seções planares tem inúmeras aplicações e a área de pesquisa é grande para o desenvolvimento de algoritmos que tratem da reconstrução, várias técnicas já foram desenvolvidas para tratar da interpolação de curvas. As diversas técnicas podem ser classificadas conforme o modo como abordam os problemas de reconstrução citados anteriormente (correspondência, bifurcação e geração de malha). Os quatro tipos de abordagens básicas de reconstrução são descritos a seguir.

A abordagem de **modelos deformáveis** utiliza geometria, física e teoria da aproximação para a reconstrução. A geometria é utilizada para representar a forma do objeto, a física impõe confinamentos em como a forma pode variar no espaço e tempo, e a teoria da aproximação

provê mecanismos e técnicas para aproximar os modelos reconstruídos aos dados originais medidos. Neste método são feitas deformações em um modelo inicial, que no caso são as fatias bidimensionais, para chegar ao objeto final. McInerney e Terzopoulos (1996) apresentaram um trabalho de reconstrução aplicado à medicina que utiliza modelos deformáveis, e provaram ser eficaz em segmentar, combinar, e representar estruturas anatômicas. A Figura 3.5 mostra um processo de reconstrução através de modelos deformáveis onde a partir de uma esfera (a) cortada por três planos são feitas deformações e aproximações até se chegar a um modelo desejado (b).

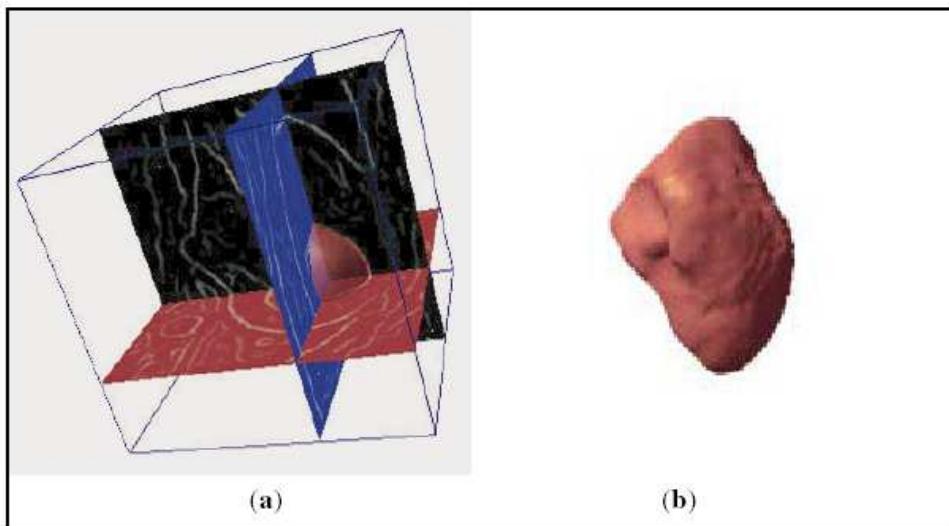


Figura 3.5: Reconstrução a partir de deformações. Fonte: McInerney e Terzopoulos (1996)

McInerney e Terzopoulos (1996) afirmam que os modelos deformáveis superam muitas das limitações de técnicas de baixo nível de processamento de imagens, fornecendo representações compactas e analíticas da forma do objeto. Uma vantagem desta técnica é que o processo de segmentação de imagens, onde a partir da imagem original é efetuada a representação poligonal das curvas, faz parte do processo de reconstrução, porém as etapas de reconstrução não são detalhadas. Pode-se perceber então que a técnica de reconstrução através de modelos deformáveis utiliza mais de conceitos de processamento de imagens, do que de modelagem geométrica.

Nas abordagens **implícitas** é utilizada uma função implícita para interpolar as curvas e gerar o objeto, de modo que a superfície do objeto (borda do objeto) que se quer reconstruir está no conjunto zero desta função, ou seja, em  $f(x, y)=0$ . Esta função é determinada a partir da

interpolação das funções de cada seção paralela (fatia) que contem as curvas a serem interpoladas. Gatass e Peixoto (2000), descrevem as abordagens implícitas através de duas etapas: a definição das funções que representam as curvas das fatias, chamadas de *field functions*, e a interpolação dessas funções para formar a função implícita que representará a superfície do objeto final. A Figura 3.6 ilustra a relação das funções com o objeto resultante.

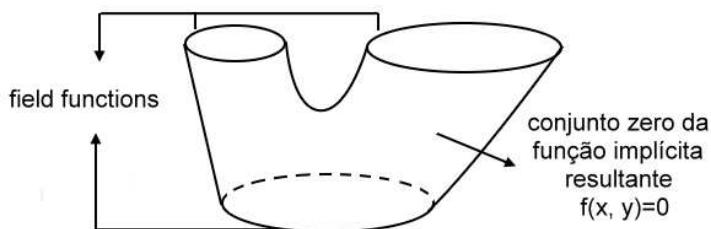


Figura 3.6: Reconstrução a partir de abordagens implícitas

Para esta abordagem a representação matricial de curvas é a mais adequada, pois existe uma correspondência natural entre a representação matricial e a função implícita, ou seja, uma curva representada matricialmente pode ser definida como o conjunto de pontos  $(x, y)$  da fatia, tal que  $f(x, y)$  representa cada *field function* utilizada para gerar a função implícita. As etapas de definição das correspondências e tratamento de bifurcação não contam com muita flexibilidade nas abordagens implícitas, pois são automaticamente definidas pela função, o que resulta em uma solução única de interpolação para um determinado conjunto de curvas inicial [Gatass e Peixoto, 2000]. Pode-se concluir que as abordagens implícitas tratam o problema de reconstrução de uma forma automática, porém não gera todos os modelos possíveis a partir de um conjunto de curvas.

As abordagens de **otimização** são baseadas em buscas ótimas em estruturas de dados definidas a partir das curvas iniciais para a determinação de critérios para reconstrução, como o cálculo da distância mínima entre os vértices de uma curva para a conexão dos mesmos. Existem vários trabalhos que utilizam otimização, pois a maioria dos algoritmos sempre procura gerar um modelo ótimo final. O trabalho de Meyers et al (1994) utiliza esse método para a reconstrução a partir de fatias, onde o problema da geração de malha entre duas curvas é expresso a partir de uma busca em uma estrutura de dados onde são mapeados os vértices das

duas curvas gerando um grafo que representa o caminho mínimo de conexão entre os vértices. A Figura 3.7 ilustra este processo: em (a) as duas curvas e seus respectivos vértices e em (b) a estrutura de dados utilizada para o mapeamento dos vértices e o grafo gerado representando o ciclo de menor caminho entre todos os vértices.

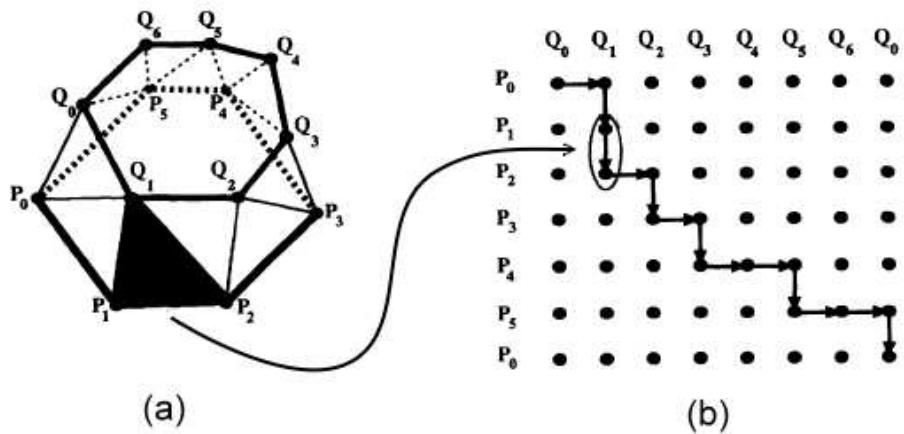


Figura 3.7: Geração de malha auxiliada por um Grafo. Fonte: Meyers et al (1994)

Um fato que deve ser mencionado quanto às abordagens de otimização é que nem sempre a superfície gerada a partir de um caminho ótimo entre os vértices das curvas interpoladas é a superfície final desejada [Meyers et al, 1994]. A Figura 3.8 mostra dois modelos gerados a partir de um mesmo conjunto inicial de curvas, em (a) a superfície do objeto é gerada por um algoritmo de otimização que calcula a superfície mínima entre as curvas, e em (b) o objeto é construído através de um algoritmo com regras definidas para a reconstrução.

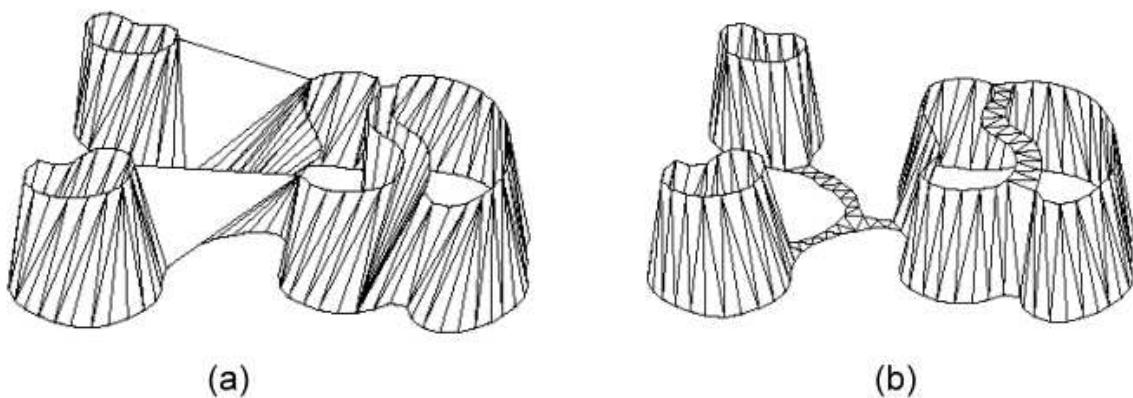


Figura 3.8: Superfícies geradas por algoritmos diferentes. Fonte: Meyers et al (1994)

Pela Figura 3.8 pode-se notar que algoritmos que consideram apenas critérios de otimização podem gerar superfícies indesejáveis. As abordagens de otimização podem tratar com eficiência do critério de geração de malha (*tiling*), mas é preciso alguma heurística para tratar o problema de bifurcação (*branching*), e também de correspondência (*correspondence*) [Vargas, 2001].

Abordagens que utilizam **heurísticas** tratam o critério de correspondência com flexibilidade. Nessas abordagens a reconstrução geralmente é realizada considerando as três etapas (*correspondence, branching e tiling*) contando com um conjunto de regras definidas (heurísticas), que são especificadas no algoritmo de reconstrução e decidem a geometria e topologia do modelo final. Bajaj et al (1999) demonstram que a maioria das técnicas de reconstrução que não utilizam heurísticas também não tratam das três etapas de reconstrução separadamente. Então é proposto um algoritmo, onde um conjunto de regras é definido e, a partir dessas regras, as três etapas de reconstrução 3D são realizadas para a construção do modelo final. Um exemplo de heurísticas utilizadas no trabalho de Bajaj et al (1999) é na etapa de geração de malha (*tiling*) que conta com um parâmetro  $w$ , o qual influencia no cálculo da superfície resultante e, de acordo com esse parâmetro, o objeto final pode ter uma superfície regular ou irregular. Além do parâmetro  $w$ , a heurística também utiliza os parâmetros  $D$ , que expressa uma distância entre os pontos de duas curvas paralelas, e  $H$ , que expressa a distância entre pontos de uma mesma curva. Então são definidas as seguintes regras para o cálculo deste parâmetro a partir de outros parâmetros já definidos:

- Regra 1: Se o parâmetro inicial  $D$  for menor ou igual a  $0.5H$ , então  $w$  assume o valor de  $2(1 - D/H)$ ;
- Regra 2: Se  $D$  for maior que  $0.5H$  e menor que  $H$ , então  $w$  assume o valor 1;
- Regra 3: Se  $D$  for maior ou igual a  $H$ , então  $w$  assume o valor  $H/D$ .

A partir dessas regras é definido então o valor de  $w$ , e consequentemente a superfície do objeto final. Assim que funciona a abordagem heurística para a reconstrução 3D, as regras definidas influenciam diretamente no objeto reconstruído.

Com base no estudo das principais técnicas de reconstrução 3D foi feito um comparativo para decidir qual técnica será utilizada para a solução proposta neste trabalho, a Tabela 3.1 apresenta este comparativo citando as quatro abordagens estudadas.

Tabela 3.1: Comparativo entre as abordagens de reconstrução 3D

Abordagem	Do que utilizam	Como consideram as três etapas de Reconstrução
<b>Modelos Deformáveis</b>	Processamento de Imagens, teoria de aproximação e física.	Não detalha nenhuma das três etapas
<b>Implícitas</b>	Funções implícitas	A correspondência e bifurcação não são tratadas, geradas automaticamente.
<b>Otimização</b>	Busca ótima em grafos	A geração de malha é tratada com eficiência, mas a correspondência e bifurcação não são detalhadas.
<b>Heurísticas</b>	Definição de regras	Geralmente as três etapas são tratadas com flexibilidade, principalmente a definição da correspondência.

Considerando o comparativo da Tabela 3.1, nota-se que a única abordagem que pode tratar das três etapas de reconstrução separadamente são as que utilizam de heurísticas, ou seja, que partem de regras definidas. Este tipo de abordagem também torna o algoritmo de reconstrução mais flexível, ou seja, o modo como é realizada a reconstrução pode ser controlado manipulando apenas as regras que foram definidas. Este trabalho utilizará essa abordagem para a criação do algoritmo que será proposto. A seção seguinte mostrará alguns trabalhos que utilizam a abordagem heurística.

## 3.2 Estado da Arte

Foi visto na seção anterior que a reconstrução tridimensional através da interpolação de curvas com mudança topológica pode gerar diferentes modelos geométricos. Para o modelo final ser próximo ao desejado, é necessário que durante a reconstrução sejam pré-estabelecidos critérios, ou seja, heurísticas que decidirão o processo. Como foi já mencionado neste trabalho, a principal etapa de reconstrução é a definição da correspondência entre as curvas quando existe mudança de topologia, e é justamente nesta etapa que é necessária a definição de regras para decidir quais curvas irão se conectar e como será feita essa conexão. A seguir serão descritos

alguns trabalhos que utilizaram heurísticas, focando na etapa de correspondência.

### 3.2.1 Algoritmo de Barequet e Sharir (1996)

No trabalho descrito por Barequet e Sharir (1996), são tratados os planos com curvas com mudança topológica, isto é, mais de uma curva em um plano ou nos dois. Para decidir sobre a correspondência das curvas é feita uma projeção xy de dois planos consecutivos. Nesta projeção são analisadas as curvas que se sobrepõe e para elas é definida a conexão, ou seja, a heurística é a seguinte: se na projeção, houver intersecção da área das curvas, elas são conectadas, caso contrário não. A Figura 3.9 demonstra as fatias com as curvas (a), a projeção xy dos dois planos, a intersecção das curvas para definir a correspondência (b), e a correspondência resultante (c).

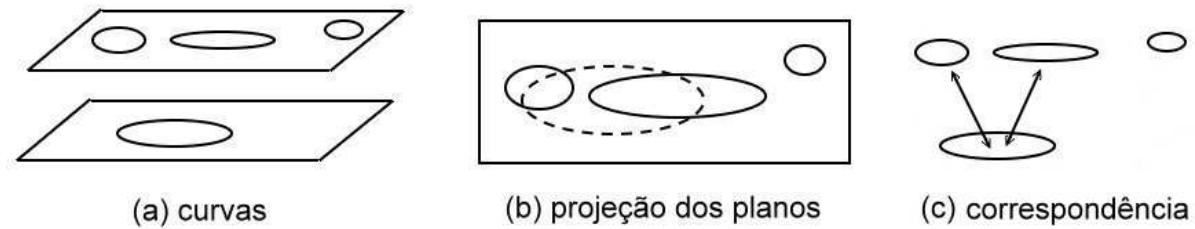


Figura 3.9: Definição da correspondência

Para a etapa de geração de malha (*tiling*), é realizada uma triangulação 2D a partir da projeção feita e depois feita a triangulação 3D. A etapa de tratamento de bifurcações (*branching*) não é detalhada nesta técnica, e considerando um exemplo em que existe mudança de topologia entre as curvas, o algoritmo apenas efetua a união das curvas com mudança topológica no mesmo plano, gerando uma só curva que é então interpolada com a curva correspondente do plano seguinte. A Figura 3.10 ilustra como seria o resultado final de uma reconstrução através deste trabalho, onde na fatia superior existisse mudança topológica.

No trabalho de Barequet e Sharir (1996) a heurística só é utilizada pra definir a correspondência das curvas e o tratamento de bifurcações não é considerado, sendo um dos poucos trabalhos que utilizam heurísticas e não consideram uma das etapas de reconstrução 3D.

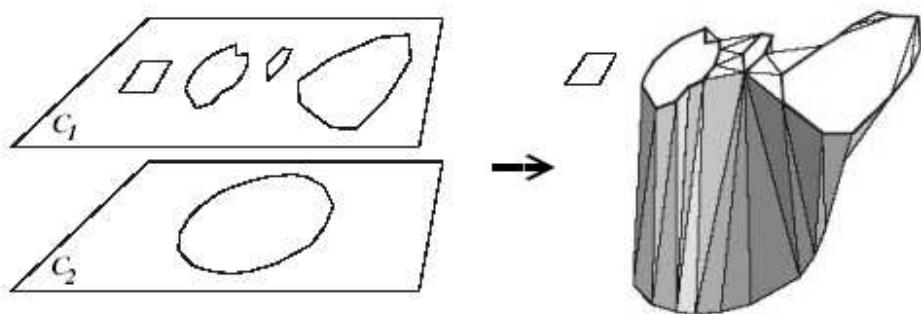


Figura 3.10: Objeto reconstruído sem bifurcação. Fonte: Gatass e Peixoto (2000)

### 3.2.2 Algoritmo de Treece, Prager, Gee e Berman (1999)

Outro trabalho que também trata de reconstrução a partir seções planares considerando heurísticas foi desenvolvido por Treece et al (1999) e é baseado no cálculo da distância entre regiões das curvas. Para cada plano contendo as curvas são relacionados discos, que podem ser internos ou externos as curvas, os discos externos são utilizados para representar regiões entre as curvas do mesmo plano, enquanto os discos internos são utilizados para representar cada região interna de cada curva. A Figura 3.11 apresenta em (a) o conjunto inicial de curvas, e em (b) os discos internos e externos que são atribuídos a cada plano aos quais as curvas pertencem.

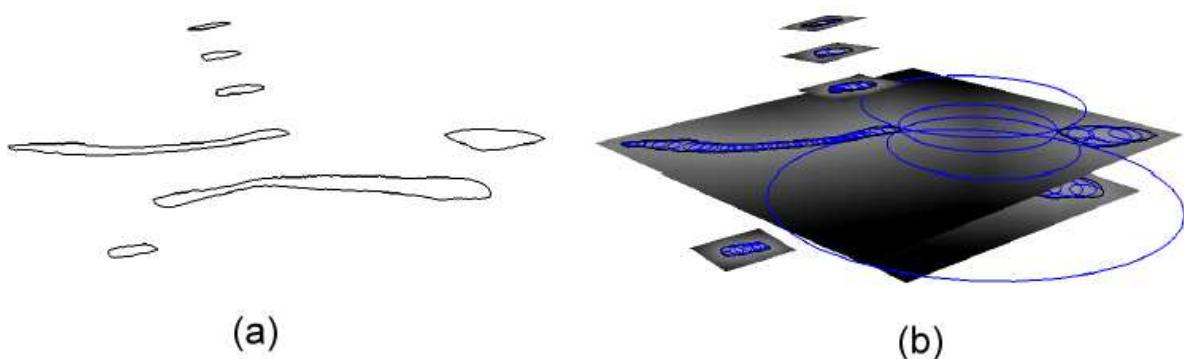


Figura 3.11: Definição dos discos para as curvas. Fonte: Treece et al (1999)

Para cada disco é calculado seu centro, denominado centróide, que será utilizado para calcular a distância entre cada par de discos de dois planos consecutivos. O cálculo da probabilidade de correspondência baseado na distância de cada par de discos é a etapa mais importante neste

processo. A heurística definida neste algoritmo é a seguinte: os discos de dois planos consecutivos com menor distâncias entre si serão correspondentes. É feita então, para cada dois planos consecutivos, uma comparação de distâncias entre os centróides de cada par de discos. A correspondência, neste caso, não leva em consideração toda a área da curva e sim cada região representada por um disco.

A Figura 3.12 apresenta esse processo ilustrando uma projeção de dois planos consecutivos A e B e três curvas, uma composta pelos discos  $a_1$ ,  $a_2$  e  $a_3$  e outra pelo disco  $a_4$ , ambas localizadas no plano A, e a relação delas com uma outra curva representada pelo disco  $b_1$  localizada no plano B, em (a) é comparada a distância do disco  $b_1$  com o disco  $a_4$ , em (b) a distância do disco  $b_1$  com o disco  $a_1$ , em (c) a distância do disco  $b_1$  com o disco  $a_2$ , e finalmente em (d) a distância do disco  $b_1$  com o disco  $a_3$ , resultando na conexão das regiões destes discos, pelo fato de terem a menor distância.

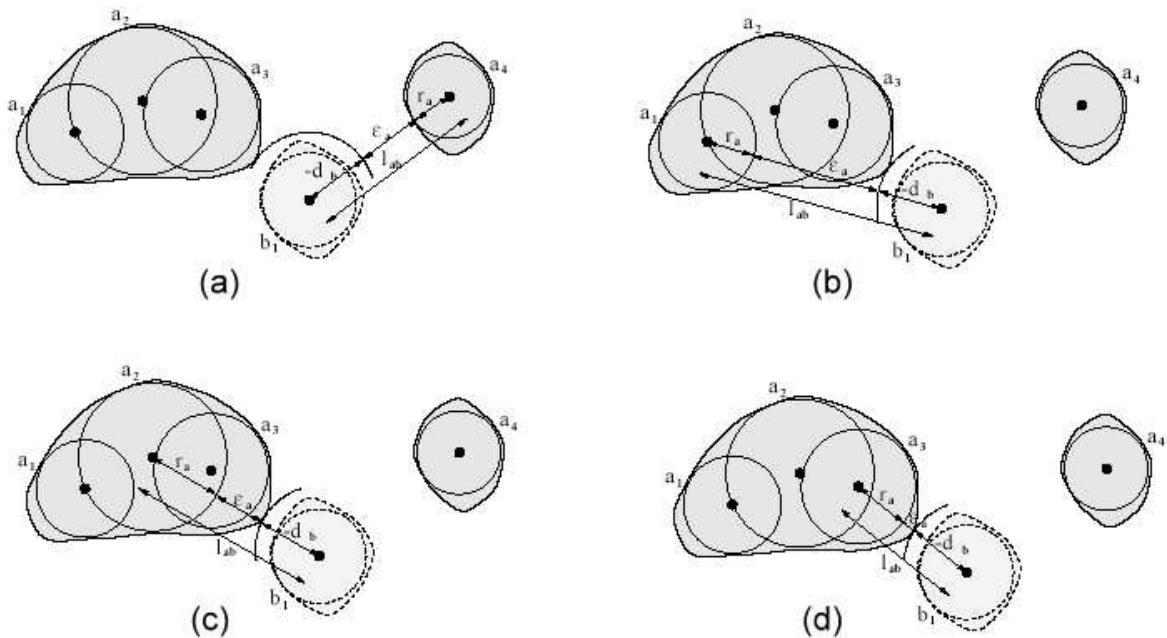


Figura 3.12: Comparação de distâncias entre centróides. Fonte: Treece et al (1999)

Feita a correspondência de todos os discos, as curvas que não tiverem nem um par de discos conectados não serão conectadas. Então, depois de decidida a correspondência de todas as curvas agora, além da heurística utilizada para esta etapa, é utilizada uma função implícita para gerar a superfície do objeto resultante. A Figura 3.13 representa em (a) o resultado da

correspondência dos discos da Figura 3.11, e em (b) a superfície do objeto final gerada a partir de uma função implícita, método discutido no capítulo 2 deste trabalho.

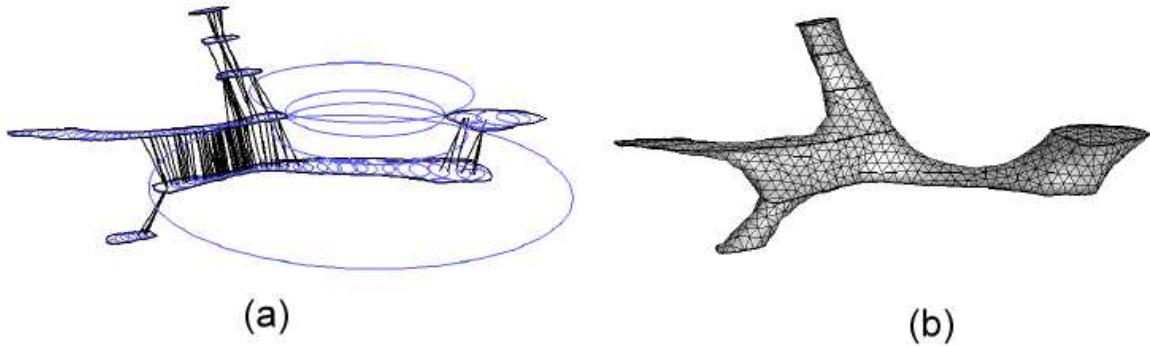


Figura 3.13: Resultado da correspondência e objeto final. Fonte: Treece et al (1999)

Algumas características podem ser analisadas do trabalho de Treece et al (1999). Uma delas é que as regiões representadas pelos discos só vão conectar com as outras regiões mais próximas de cada plano consecutivo se essa relação for recíproca, isso permite deixar regiões sem conexão se for preciso. Uma característica em relação a proposta de Barequet e Sharir (1996) discutida anteriormente neste capítulo é que não é necessário duas curvas se sobreponem para de conectarem. E apesar de utilizar heurística para a definição da correspondência, o trabalho de Treece et al (1999) também utiliza funções implícitas para a geração de malha do objeto final.

### 3.2.3 Algoritmo $\beta$ -Connection (2001)

Uma outra técnica que utiliza heurística é proposta por Vargas (2001). Trata-se de uma estratégia de reconstrução volumétrica chamada de *Beta-Connection* que tem a flexibilidade para produzir uma família de objetos construídos a partir de um mesmo conjunto de seções planares, tornando possível múltiplas opções de um objeto final. Para resolver o problema de correspondência (*correspondence*) o algoritmo efetua um cálculo de distância e recebe um parâmetro  $\beta$  do usuário pra decidir quais curvas vão se conectar. Este parâmetro é um inteiro positivo que expressa uma distância entre as curvas para a conexão. A heurística definida é a seguinte: Se a distância entre duas curvas quaisquer for menor que o valor do parâmetro

$\beta$  definido pelo usuário, então estas curvas são conectadas. Isto é, para um  $\beta$  definido pelo usuário, todas as curvas que estejam a uma distância menor ou igual a  $\beta$  ficarão conectadas.

A Figura 3.14 apresenta diferentes soluções de conexão resultantes da estratégia proposta por Vargas (2001). A partir das curvas situadas em seções paralelas (a), na primeira solução (b) o parâmetro  $\beta$  definido é menor que todas as distâncias entre as curvas de dois planos consecutivos, não acontecendo nenhuma conexão, na segunda solução (c) o valor do parâmetro  $\beta$  vale 3, então todas as curvas de dois planos consecutivos com distância entre elas menor ou igual a 3 são conectadas, e na ultima ilustração (d) é atribuído um valor grande para  $\beta$  sendo maior que todas as distâncias entre as curvas de dois planos consecutivos, neste caso todas elas se conectam.

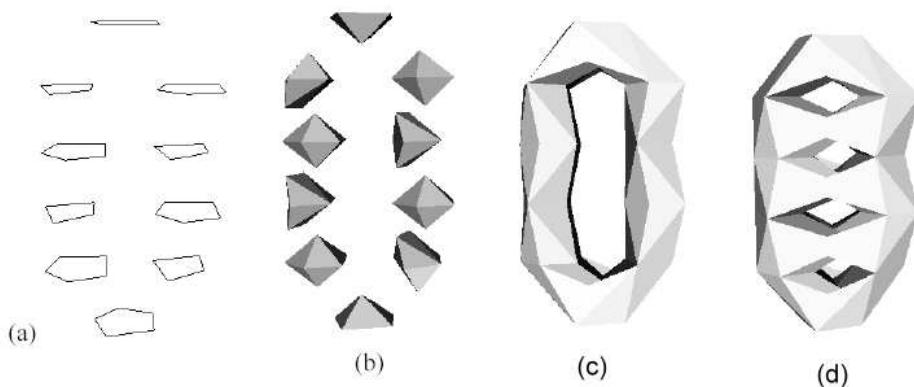


Figura 3.14: Diferentes valores para  $\beta$ . Fonte: Vargas (2001)

O trabalho de Vargas (2001) não constrói somente a superfície do objeto final, é feita uma triangulação entre todos os vértices das curvas iniciais e então são gerados tetraedros para representar todo o volume do objeto final. A etapa de bifurcação (*branching*) é tratada detalhadamente por duas etapas de seu algoritmo: a classificação de todos os tetraedros do objeto e a eliminação dos tetraedros que não pertencerão ao objeto final. A classificação dos tetraedros é feita analisando se as arestas dos mesmos têm alguma relação com as arestas das curvas iniciais, assim são identificados os tetraedros internos e externos ao modelo final. A Figura 3.15 ilustra este processo, onde partindo das curvas (a) são gerados tetraedros a partir de todos os vértices das mesmas, gerando em (b) o volume total formado por estes tetraedros, é feita então a classificação dos mesmos (c) com a finalidade de eliminar os necessários para representar

a bifurcação, e gerar finalmente o modelo final do objeto reconstruído (d), com os tetraedros externos eliminados.

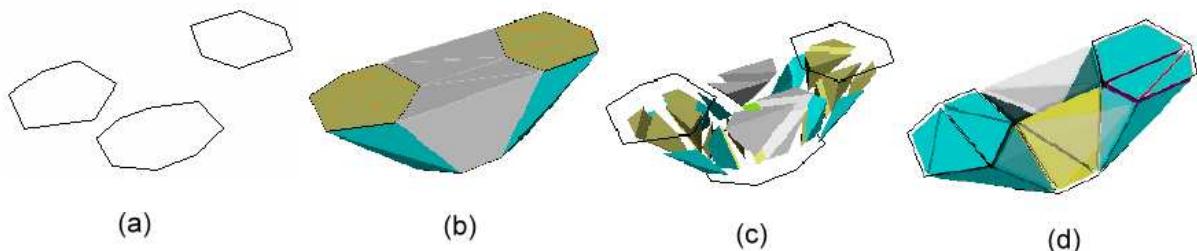


Figura 3.15: Classificação e eliminação de tetraedros. Fonte: Vargas (2001)

Segundo Vargas, (2001) a técnica de reconstrução *Beta-Connection* oferece mais flexibilidade na escolha das componentes conexas, pois partindo de um mesmo conjunto de seções planares pode-se chegar a diferentes formas de objetos, o que é difícil através de outros algoritmos da literatura, resolvendo os problemas de ramificação e geração de malha de forma satisfatória.

O algoritmo proposto neste trabalho é construído com base nas contribuições descritas neste capítulo, partindo da idéia do  $\beta$ -*Connection* [Vargas, 2001] onde se tem flexibilidade para a etapa de decisão de correspondência através de um parâmetro. Como  $\beta$ -*Connection* é uma técnica de reconstrução volumétrica, e neste trabalho está se tratando da reconstrução de superfícies, não serão utilizados tetraedros para o cálculo de distância entre as curvas e sim através da projeção dos planos [Barequet e Sharir, 1996], serão calculadas as distâncias entre os centróides das curvas [Treece et al, 1999] afim de compará-las com o parâmetro definido pelo usuário para decidir a correspondência. Então, a proposta apresentada no capítulo a seguir incorpora características das três abordagens anteriores.

## 4 *Algoritmo $\Delta$ -connection*

Com base nos trabalhos relatados nos capítulos anteriores, é possível perceber que se faz reconstrução 3D para obter diversas informações sobre o modelo original, basicamente com dois motivos:

- Estudar a estrutura do modelo, a relação entre as partes, o entendimento do todo. Existência ou não de certas formações (bolsões, conexões, etc.) e;
- Obter características mensuráveis do modelo (volume, área, comprimento, etc.).

Percebe-se que esta divisão leva a duas possíveis abordagens quanto às aplicações das técnicas de reconstrução 3D, uma voltada à visualização do objeto como um todo, outra voltada para a precisão das partes do objeto. A Figura 4.1 apresenta estas duas abordagens, onde em (a) a reconstrução pode ser feita de uma forma mais simples, sem levar em consideração todos os detalhes do objeto final, por exemplo, o local que irá ocorrer cada bifurcação, realizando uma reconstrução preocupada principalmente com a visualização do modelo como um todo, já em (b) se a reconstrução não tratar com precisão a bifurcação que vai ser gerada, vai ser visível qualquer má formação na superfície do modelo final.

A reconstrução voltada para visualização da estrutura sem necessariamente ter-se precisão já encontra grande utilidade para várias aplicações como a identificação de adensamento de capilaridades (várias ramificações agrupadas) ou uma má formação congênita (aparecimento de uma geometria estranha). Estas aplicações se valem da vantagem que estas técnicas trazem de poder tornar visível externamente uma formação cuja visualização, de outra forma, seria invasiva. Para este tipo de abordagem a flexibilidade e as heurísticas de identificação de cor-

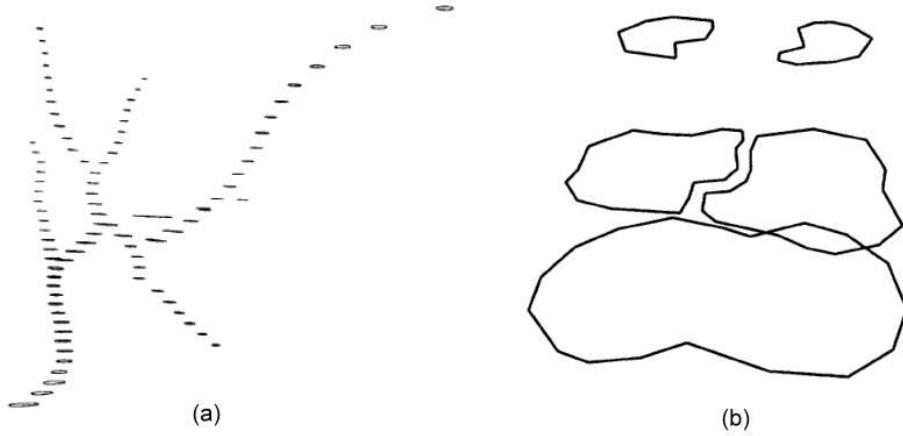


Figura 4.1: Diferentes abordagens para reconstrução. Fonte: Meyers et al (1992)

respondências são de grande importância aliadas ao fato de que se busca uma resposta rápida à esta abordagem

A proposta deste trabalho é voltada para esta abordagem, ou seja, para a visualização de estruturas como um todo, ilustrada em (a) na Figura 4.1. A etapa de definição de correspondência (*correspondence*) terá ênfase, e não será tratado com precisão cada local de bifurcação (*branching*), sendo resultado direto e impreciso, obtido da etapa de geração de malha (*tiling*).

Na abordagem adotada aqui, basicamente centrada na definição de correspondências e geração direta do *tiling* resulta numa sobreposição de superfícies que proporciona um efeito visual do *branching* sem cálculos adicionais e sem processamento diferenciado para o *tiling*. Assim, economizam-se tempo e processamento para estes dois detalhes. Vale ressaltar que efetuar o cálculo da intersecção das correspondências, em média não altera a contagem de faces do módulo, apenas sua precisão. Sendo assim não influencia no tempo de *rendering* do modelo com esta abordagem. A seguir será descrito como cada uma das etapas de reconstrução será tratada neste trabalho.

## 4.1 Correspondência no $\Delta$ -connection

A etapa de definição de correspondências (*correspondence*) será realizada em três passos. Primeiramente será calculado um centróide para cada curva. O centróide será calculado através

da média aritmética dos pontos  $x$  e  $y$  extremos da curva.

É então calculada a distância entre cada centróide das curvas pertencentes aos planos consecutivos. As distâncias serão calculadas com as curvas projetadas no mesmo plano, ou seja, desprezando a altura entre os planos, o que simplifica o cálculo da distância. Os valores das distâncias calculadas serão armazenados em uma matriz tendo como linhas as curvas de um plano e como colunas as curvas do plano consecutivo. A Figura 4.2 ilustra, a partir de uma projeção dos dois planos, as distâncias entre os centróides das curvas e a matriz onde elas são armazenadas. As curvas desenhadas em pontilhado pertencem ao plano projetado para o cálculo as distâncias.

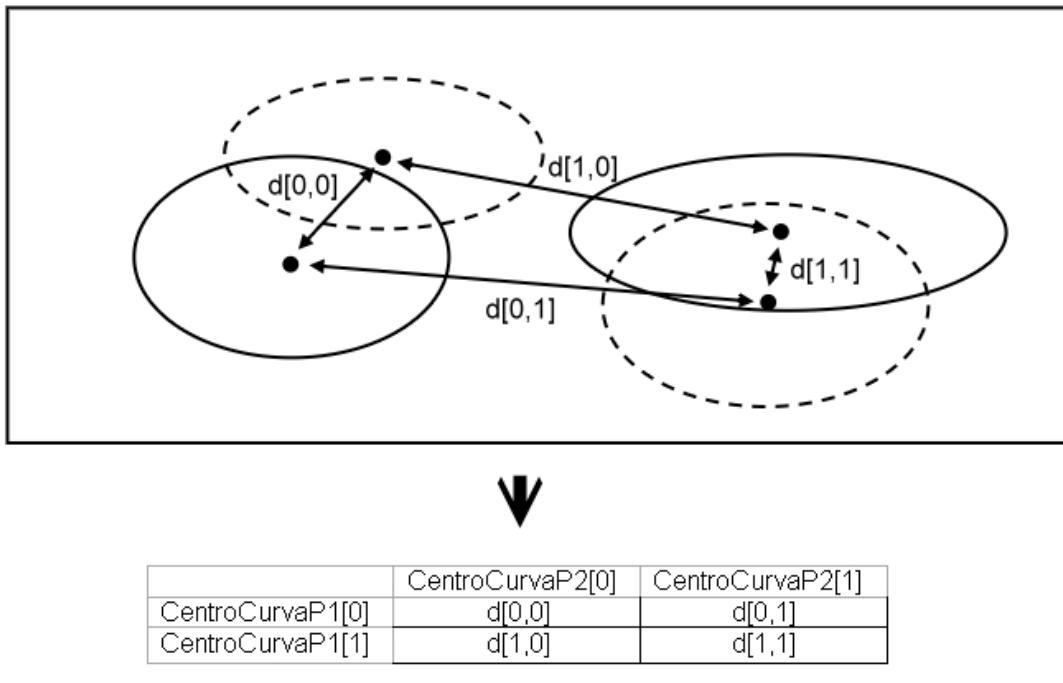


Figura 4.2: Matriz de distâncias

Definida a matriz de distâncias, são calculadas também as distâncias mínimas e máximas na matriz com o objetivo de informar ao usuário qual o intervalo de distâncias entre todas as curvas. O algoritmo recebe um parâmetro  $\Delta$ , definido pelo usuário, que é comparado com cada distância calculada, ou seja, cada elemento da matriz de distâncias, afim de decidir se duas determinadas curvas serão conectadas ou não. A heurística definida é a seguinte:

- Se  $\Delta$  for menor que a distância mínima da matriz, nenhuma curva é interpolada;

- Se  $\Delta$  for maior ou igual que a distância máxima da matriz, todas as curvas são interpoladas;
- Se  $\Delta$  estiver no intervalo entre a distância mínima e máxima, é feita a seguinte regra para cada uma das distâncias:
  - Se a distância entre duas curvas for menor ou igual a  $\Delta$ , então essas curvas são conectadas;
  - Se a distância for maior que  $\Delta$ , então não ocorrerá conexão dessas duas curvas.

Para cada duas curvas que forem conectadas a partir da heurística definida, essas curvas são marcadas no algoritmo como “conectadas” e é realizada a próxima etapa do algoritmo que trata da geração de malha (*tiling*). Para as curvas que não tiverem nenhuma conexão (curvas não marcadas), será apenas gerada uma face (para funcionar como topo ou base) a partir de seus vértices para a visualização, como ilustrada pela seta na Figura 4.3 (b).

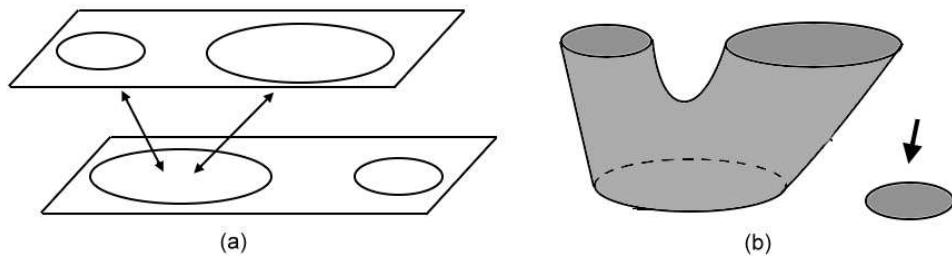


Figura 4.3: Curva sem nenhuma correspondência

Como a heurística apresentada aqui depende da distância “d” (em grego delta) o algoritmo será denominado de  $\Delta$ -connection, a exemplo do  $\beta$ -connection. A seguir será descrita a etapa de geração de malha para as curvas correspondentes.

## 4.2 Geração de Malha no $\Delta$ -connection

Uma vez identificadas as correspondências, a geração de malha (tiling) será feita para cada par de curvas correspondentes. Nesta etapa será adotada a técnica de *B-rep* poliédrico para representar o objeto a partir da sua geometria e topologia.

O VRML conta com um recurso para modelar objetos, que parte do princípio da técnica *B-rep*. Trata-se do comando *IndexedFaceSet* que permite descrever as coordenadas e a topologia dos vértices, definindo cada face do objeto final através dos nós *Coordinate* e *coordIndex*. O nó *Coordinate* é onde estão listadas as coordenadas de todos os vértices no espaço 3D a partir do nó *point* que guarda uma lista de pontos, aqui é definida a geometria das faces que serão definidas. O nó *coordIndex* descreve uma lista de faces a partir dos índices do nó *Coordinate*, ou seja, a partir da geometria é agora definida a topologia das faces, sempre em um mesmo sentido (horário). A Figura 4.4 apresenta como é feita a descrição das faces de um objeto em VRML a partir do comando *IndexedFaceSet*.

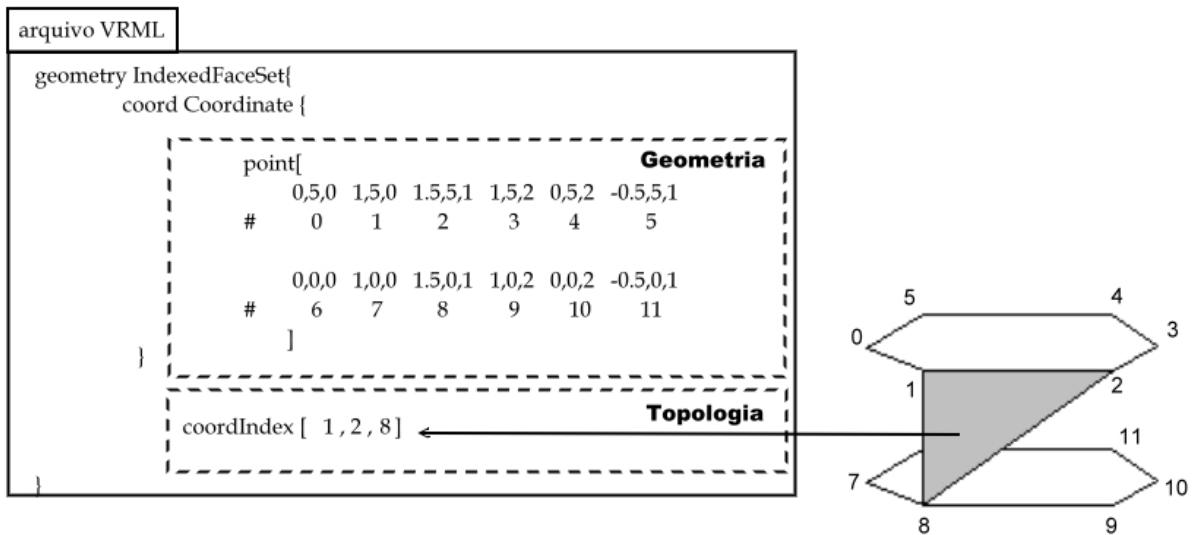


Figura 4.4: Representação *B-rep* em VRML

Então para cada par de curvas correspondentes, a geração de malha será feita da seguinte forma: primeiro serão descritos os vértices das duas curvas (geometria) através do nó *Coordinate* e depois será feita a conexão destes vértices (topologia) através do nó *coordIndex*, gerando todas as faces do objeto final. A geração de malha não pode ser feita por uma simples triangulação entre os vértices das duas curvas devido a possível diferença do número de vértices entre as mesmas, é preciso então ser feito um mapeamento destes vértices a fim de decidir quantos vértices de uma curva vão se conectar com quantos da outra.

A Figura 4.5 representa esta situação de forma bem simples onde é realizada a geração de

malha entre uma curva com quatro vértices, representada pelos pontos P ( $P_1, P_2, P_3$  e  $P_4$ ) e uma curva com seis, representados pelos pontos Q ( $Q_1, Q_2, Q_3, Q_4, Q_5$  e  $Q_6$ ). Em (a) é realizada uma simples triangulação, ou seja, cada ponto de uma curva é conectado com apenas os dois mais próximos da curva seguinte e como o número de pontos não é igual, a geração de malha não conecta todos eles ( $Q_6$  fica desconectado). Em (b) é feito uma reordenação dos pontos das duas curvas para que todos eles se conectem e gerem a superfície desejada, neste caso alguns vértices se conectaram com mais de dois pontos da curva correspondente, como é o caso de  $P_3$  e  $P_4$ .

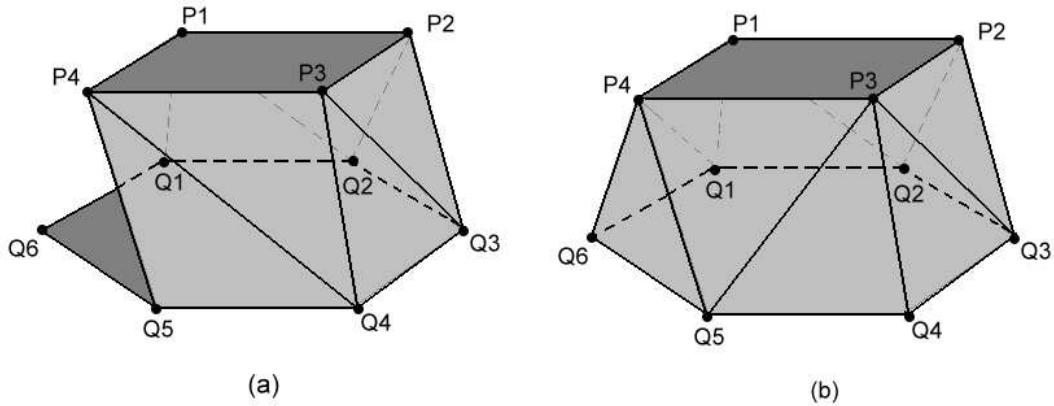


Figura 4.5: Diferentes geração de malha

Para resolver tal problema, considera-se uma situação genérica como, por exemplo, a interpolação de uma curva com 4 vértices com outra de 11 vértices. O problema inicial é decidir quantos vértices da curva maior vão se conectar com cada um da curva menor. Para isso é feita uma divisão do número de pontos da curva maior (11) pelo número de pontos da curva menor (4) como representado na Figura 4.6.

Conforme a Figura 4.6, ao valor inteiro da divisão é chamado de  $N$ , e para ser feita uma distribuição uniforme dos pontos entre as duas curvas fica decidido que alguns pontos da curva menor vão conectar-se com  $N$  pontos da curva maior e alguns com  $N+1$ . O problema agora é quantos pontos da curva menor serão conectados com  $N$  pontos da curva maior e quantos com  $N+1$ . Para isso utiliza-se do valor decimal da divisão realizada na Figura 4.6, que será chamado de  $R$ . A regra é a seguinte: será multiplicado este valor decimal  $R$  pelo número de pontos da

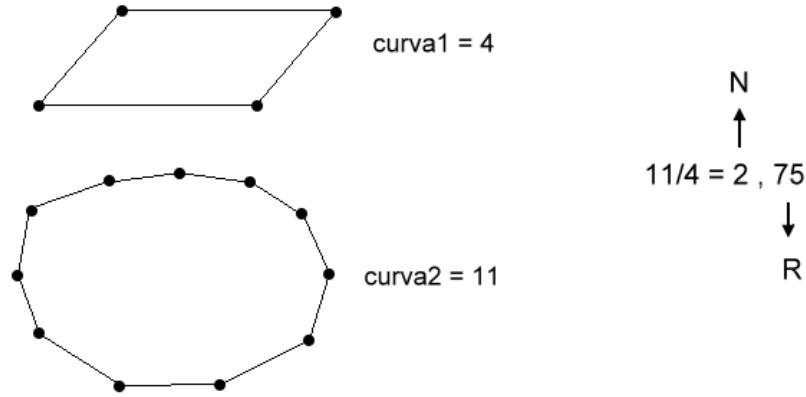


Figura 4.6: Solução para a geração de malha

curva menor ( $R*4 = 0.75*4 = 3$ ) para decidir quantos pontos dela serão conectados com  $N+1$  pontos da curva maior, e o restante será conectados com  $N$  pontos. Assim resulta que três dos quatro pontos da curva menor, serão conectados com três pontos ( $N+1$ ) da curva maior, e o ponto restante será conectado com dois ( $N$ ), como representado na Figura 4.7.

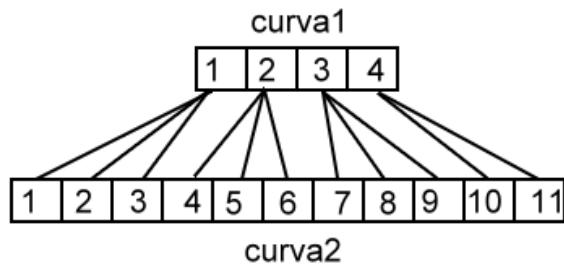


Figura 4.7: Mapeamento dos pontos

### 4.3 Bifurcação no $\Delta$ -connection

A etapa de geração de bifurcações (*branching*) será resultado direto da geração de malha. Não será decidido o local em que ocorrerá a bifurcação, sendo ele resultado da malha gerada entre as curvas, sem nenhum cálculo de intersecção de superfícies. A Figura 4.8 demonstra como o algoritmo tratará das bifurcações quando existir mudança de topologias nas curvas, ou seja, onde uma curva se divide em duas ou vice-versa.

É possível notar na Figura 4.8 que a bifurcação é resultado direto das duas superfícies

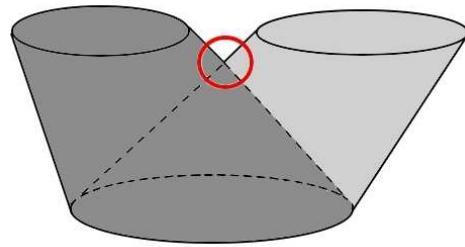


Figura 4.8: Local da bifurcação - *Branching*

geradas, sendo uma o resultado da interpolação da primeira curva do plano superior com a curva do plano inferior e outra da segunda. Esta solução não trata com precisão do local onde irá ocorrer a bifurcação, mas é uma solução automática e não exige um custo computacional alto, o que se torna interessante para o método de reconstrução adotado, que é voltado para a visualização externa do sólido como um todo e não para uma reconstrução precisa de cada parte da superfície.

## 4.4 Arquitetura do $\Delta$ -connection

O processo de reconstrução do  $\Delta$ -connection é dividido em etapas, desde a geração das curvas até a visualização do objeto final. A Figura 4.9 representa arquitetura da proposta e as etapas envolvidas.

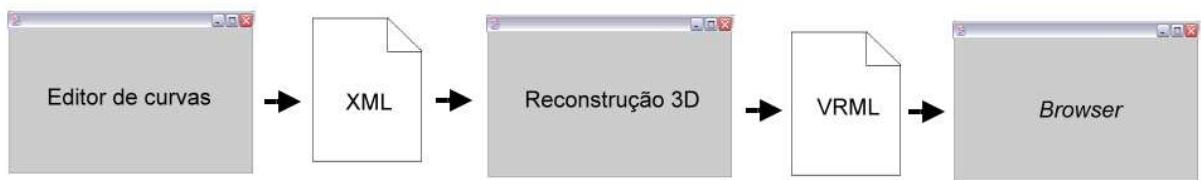


Figura 4.9: Arquitetura do  $\Delta$ -connection

Para a implementação proposta, a primeira etapa é uma aplicação GUI-2D em JAVA que será desenvolvida para a edição das curvas. Depois de editadas as curvas, as informações relativas a elas (coordenadas dos pontos e plano pertencente) serão armazenadas em um arquivo XML. O algoritmo de reconstrução  $\Delta$ -connection receberá a entrada desses dados e realizará a interpolação das curvas, criando um arquivo VRML com as informações geométricas do ob-

jeto final, que pode ser visualizado em qualquer *browser*<sup>1</sup> VRML. Abaixo são listadas algumas características/requisitos do algoritmo:

- Curvas representadas pela forma poligonal e orientadas no sentido horário;
- Curvas côncavas e convexas, não auto-interceptantes;
- Curvas fechadas não intersectantes;
- Centróides calculados pela média dos pontos extremos;
- Arquivo de representação das curvas em XML;
- Algoritmo com parâmetro de controle de flexibilidade;
- Algoritmo simétrico (reconstrução *Top-down* = reconstrução *Bottom-up*);
- Objeto representado em *B-rep*, usando VRML com faces orientadas no sentido horário;
- A decisão do local de bifurcação (*branching*) não será tratada;
- Aplicação preferencial: visualização de estruturas (canais).

A seguir será mostrada a aplicação utilizada para a edição das curvas, a primeira etapa deste processo.

## 4.5 Editor de curvas

A aplicação em JAVA que será desenvolvida para a edição dos planos vai permitir que o usuário desenhe as curvas de um determinado plano para posteriormente ser realizada a interpolação. A Figura 4.10 apresenta a proposta da interface GUI-2D desta aplicação onde são desenhadas duas curvas no plano que está sendo editado e é visualizada uma terceira curva em pontilhado que representa as curvas que compõe o plano anterior.

---

<sup>1</sup>Navegador de internet com suporte (*plugin*) VRML

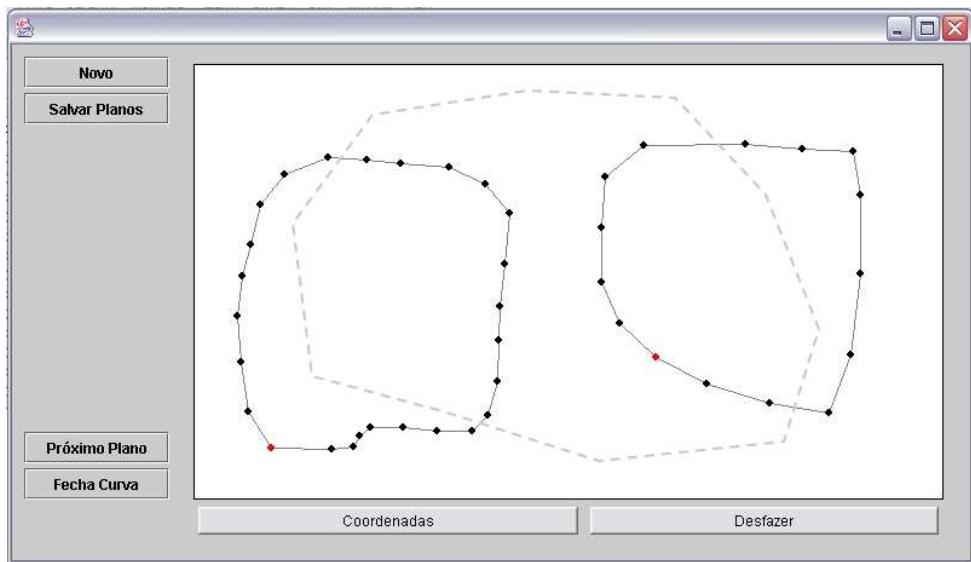


Figura 4.10: GUI-2D de edição das curvas

Quanto aos componentes da interface apresentada para o editor de curvas, a função de cada botão é a seguinte:

- Novo: inicia um novo projeto para a edição das curvas em uma nova seqüência de planos;
- Salvar Planos: salva toda a seqüência de planos que contém as curvas editadas até o momento;
- Próximo Plano: salva o plano editado no momento e habilita o plano seguinte para continuar a edição das curvas;
- Fecha Curva: fecha a curva que está sendo editada (se esta estiver aberta) e habilita o usuário para desenhar a próxima curva do plano;
- Coordenadas: mostra as coordenadas de cada ponto da curva que está sendo editada;
- Desfazer: apaga o ultimo ponto da curva que está sendo editada.

O usuário, então, tem a opção de desenhar quantas curvas quiser em cada um dos planos e também visualizar suas coordenadas a medida que forem sendo editadas. Uma vez escolhida a opção “salvar planos”, todos os dados relativos a todas as curvas que foram editadas serão armazenados em um arquivo XML, como será descrito a seguir.

## 4.6 Representação da curva

O arquivo XML utilizado para o armazenamento das informações das curvas editadas será organizado de maneira que fique clara a representação de cada plano e de cada curva pertencente ao mesmo. A Figura 4.11 representa a estrutura do arquivo XML que será criado após a edição dos planos.

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
- <Planos>
  <n_planos>2</n_planos>
  <dist_planos>0.5</dist_planos>
  - <plano1>
    <n_curvas>2</n_curvas>
    <curva1>0.0,0.0;0.3,0.0;0.0,0.5;0.3,0.8;1.4,0.8 ...</curva1>
    <curva2>0.0,0.0;0.3,0.0;0.0,0.5;0.3,0.8;1.4,0.8 ...</curva2>
  </plano1>
  - <plano2>
    <n_curvas>1</n_curvas>
    <curva1>0.0,0.0;0.3,0.0;0.0,0.5;0.3,0.8;1.4,0.8 ...</curva1>
  </plano2>
</Planos>
```

Figura 4.11: Armazenamento em XML

A estrutura do arquivo é formada por um *tag* raiz denominado “Planos”. O primeiro elemento dentro do *tag* raiz, denominado “n\_planos”, informa o número de planos que foram editados. O segundo elemento do nó raiz, chamado de “dist\_planos”, informa a distância entre os planos que serão interpolados, e depois para cada um dos planos é criado um elemento contendo as informações dos mesmos. Para cada plano, é informado no elemento ”n\_curvas” o número de curvas que o mesmo contém, e para cada curva é são descritas as coordenadas x e y de cada um de seus vértices separadas pelo caractere ”;”. Depois de escrito o arquivo XML, pode ser executado então o algoritmo de reconstrução que é apresentado a seguir.

## 4.7 Reconstrução 3D: $\Delta$ -connection

O algoritmo de reconstrução carrega o arquivo XML e a partir de um parâmetro, definido pelo usuário, realiza a reconstrução (interpolação das curvas). Um pseudo-código do algoritmo foi desenvolvido para representar como será implementada a solução proposta nesse capítulo. As Figuras 4.12 e 4.13 apresentam como ele foi detalhado.

```

Início
  Para todos os índices de [ p ]      //para todos os planos

    Entrada( planos[ p ] )
      Para cada CurvaP1[i]
        CentroCurvaP1[i] = calculaCentro( CurvaP1[i] );

    Entrada( planos[ p+1 ] )
      Para cada CurvaP2[i]
        CentroCurvaP2[i] = calculaCentro( CurvaP2[i] );

    D** = matrizDeDistâncias( CentroCurvasP1 , CentroCurvasP2 )

    Entrada( $\Delta$ )
      Se  $\Delta < \text{distanciaMin}(D)$       // não interpola          heurística
        Para todos os índices [ i ] e [ j ]
          fechaCurva( CurvaP1(i) ) e fechaCurva( CurvaP2(j) )

      Senao
        Se  $\Delta \geq \text{distanciaMax}(D)$  // interpola todas as curvas
          Para todos os índices [ i ] e [ j ]
            Tiling ( CurvaP1(i) , CurvaP2(j) )

        Senao
          Para todos os índices [ i ] e [ j ]
            Se  $D[i][j] \leq \Delta$ 
              Tiling ( CurvaP1(i) , CurvaP2(j) )
            Senão
              fechaCurva( CurvaP1(i) ) e fechaCurva( CurvaP2(j) )

      p++
  Fim

```

Figura 4.12: Pseudo-código do  $\Delta$ -connection

```

Tiling( CurvaP1 , CurvaP2 )
{
  geometria = pontos( CurvaP1 , CurvaP2 );
  topologia = mapeamento ( CurvaP1 , CurvaP2 );
  escreveVRML(geometria ,topologia);
}

```

Figura 4.13: Função *Tiling*

O algoritmo interpola as curvas a cada dois planos consecutivos, então ele é executado para

cada par de planos, que estão descritos no arquivo XML, para gerar o objeto resultante de todos os planos armazenados. Conforme a Figura 4.12 o algoritmo funcionará da seguinte maneira: para cada dois planos consecutivos é calculado o centróide de todas as curvas. A função do cálculo do centróide será colocada em módulo separado para permitir posteriormente que este cálculo possa ser experimentado com várias técnicas.

Depois de calculados os centróides, são calculadas as distâncias entre eles. Uma pequena otimização será feita no cálculo das distâncias no sentido de que estas serão calculadas no plano XY, ou seja, desconsiderando a distância Z entre os planos (que será sempre a mesma), e ainda será evitada a extração da raiz quadrada, que é um cálculo custoso. Então o cálculo representado pela equação 4.1 será resumido pelo apresentado na equação 4.2.

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2} \quad (4.1)$$

$$d^2 = (x_1 - x_2)^2 + (y_1 - y_2)^2 \quad (4.2)$$

Todas as distâncias são armazenadas em uma matriz ( $D[i][j]$ ) onde serão calculadas as distâncias mínima e máxima entre as curvas e informadas ao usuário para o mesmo definir o parâmetro  $\Delta$  que irá expressar uma distância máxima e decidirá na correspondência das curvas através da heurística já detalhada.

A interpolação consiste na geração de malha entre duas determinadas curvas e é realizada pela função *Tiling*. Esta função tem como entrada duas curvas correspondentes, e são armazenados em dois vetores a geometria e topologia dos pontos das curvas recebidas. O vetor *geometria* recebe as coordenadas dos pontos das duas curvas e o vetor *topologia* recebe a lista de mapeamento dos pontos realizadas pela função *mapeamento* que define quais pontos de uma curva vão se conectar com quais da curva correspondente através da técnica apresentada na seção 4.2. O último passo da função *Tiling* é escrever as faces da superfície que será gerada em VRML através do comando *IndexedFaceSet* descrito também na seção 4.2 utilizando a geometria e topologia dos pontos como parâmetro.

A cada execução da função *Tiling*, é construída a superfície entre as curvas recebidas como

parâmetro. Ao final da execução do algoritmo, o arquivo VRML contém a superfície final do objeto, que é formada por todas as superfícies geradas entre cada par de curvas correspondentes. O arquivo, então, pode ser exportado para a visualização em qualquer *browser* VRML.

## 5 Considerações Finais

A busca por um método de interpolação de curvas com mudança topológica que seja visualmente interessante e rápido exigiu um maior entendimento de conceitos relacionados à reconstrução 3D. Posteriormente, a identificação da abordagem heurística propiciou o caminho que parece se adequar melhor aos objetivos traçados para este trabalho.

Até a presente etapa deste trabalho de conclusão de curso, uma solução para o problema de interpolação de curvas com mudança topológica baseado em abordagens heurísticas de reconstrução tridimensional foi apresentada. Esta solução tem como critérios de eficiência principalmente a rapidez e simplicidade no cálculo das etapas de reconstrução.

O algoritmo que realiza a interpolação das curvas foi desenvolvido com ênfase na etapa de definição de correspondências, citada na literatura como sendo a principal etapa de reconstrução tridimensional. Para realizar esta etapa o algoritmo conta com um parâmetro de entrada, que o torna flexível para decidir no resultado do objeto reconstruído.

O local de bifurcações, que ocorrem quando existe mudança de topologia entre as curvas, é decidido automaticamente, sendo assim, esta etapa de reconstrução não foi detalhada no algoritmo. Então, a reconstrução realizada será voltada para a visualização de toda estrutura do objeto e não para a precisão, sendo que a área de aplicação recomendada é para a visualização de canais.

Para a edição das curvas, foi adotado desenvolver uma aplicação por não se dispor de dados originais de modelos. Assim apesar de terem sido solicitados os dados à alguns autores, não há dependência deste trabalho de respostas deles.

O presente trabalho alcançou seus objetivos traçados para o período, o que permite prosseguir imediatamente para os próximos passos: a implementação e validação da proposta.

## *Referências*

- AZEVEDO, E.; CONCI, A. *Computação Gráfica: Teoria e Prática*. [S.l.]: Editora Campus, 2003. 353 p.
- BAJAJ, C. L.; COYLE, E. J.; LIN, K.-N. Arbitrary topology shape reconstruction from planar cross sections. *Graphical Models and Image Processing*, 1996.
- BAJAJ, C. L.; COYLE, E. J.; LIN, K.-N. Tetrahedral meshes from planar cross sections. *Computer Methods in Applied Mechanics and Engineering*, 1999.
- BAREQUET, G.; SHARIR, M. Piecewise-linear interpolation between polygonal slices. *Computer Methods in Applied Mechanics and Engineering*, 1996.
- CARRARD, M. *Tutorial VRML 2.0*. [S.l.]: Universidade Regional do Noroeste do Estado Rio Grande do Sul- UNIJUÍ / Departamento de Tecnologia. Ijuí (RS), 2001. 88 p.
- DÉCIO, O. C. *XML: Guia de Consulta Rápida*. [S.l.]: São Paulo. Novatec Editora Ltda, 2000. 96 p.
- DIEGUEZ, J. P. P. *Modelagem Geométrica para Computação Gráfica*. [S.l.]: Achiamé, 1989. 165 p.
- FIGUEIREDO, L. H.; CARVALHO, P. C. P. *Introdução à Geometria Computacional*. [S.l.]: Rio de Janeiro: Instituto de Matemática Pura e Aplicada, 18º Colóquio Brasileiro de Matemática, 1991. 108 p.
- FILHO, A. C. *Modelagem Geométrica: Representação e Manipulação de Objetos Geométricos Utilizando o Computador*. [S.l.]: São José Do Rio Preto: Sociedade Brasileira de Matemática Aplicada e Computacional, 1998. 30 p.
- FOLEY, J. D. et al. *Computer Graphics: Principles and Practice*. 2. ed. [S.l.]: Addison-Wesley Publishing Company, 1996. 1174 p.
- GATTASS, M.; PEIXOTO, A. Reconstrução de superfícies a partir de seções bidimensionais. *Qualificação MCC28/00*, Julho 2000.
- GRAVES, M. *Projeto de Banco de Dados em XML*. [S.l.]: São Paulo. Makron Books, 2003. 518 p.
- MCINERNEY, T.; TERZOPOULOS, D. Deformable models in medical image analysis: A survey. *Medical Image Analysis*, 1996.
- MEYERS, D.; SKINNER, S.; SLOAN, K. Surface from contours. *ACM Trans. On Graphics*, 1992.

NONATO, L. G. et al. Beta-connection: Generating a family of models from planar cross sections. *ACM Transactions on Graphics*, Outubro 2005.

ROMEU, R. K. et al. Impacto dos sistemas de visualização 3d e de realidade virtual nos estudos integrados de reservatórios. *Rio Oil and Gas Expo and Conference, Rio de Janeiro*, 2000.

SPECK, J. H. *Avaliação Comparativa das Metodologias Utilizadas em Programas de Modelagem Sólida*. 185 p. Dissertação (Mestrado) — PPGEP/UFSC, 2001.

TRAINA, A. J. M.; OLIVEIRA, M. C. F. *Apostila de Computação Gráfica*. [S.l.], Setembro 2004.

TREECE, G. M. et al. Surface interpolation from sparse cross-sections using region correspondence. *Technical Report CUED/F-INFENG/TR 342*, Março 1999.

VARGAS, A. J. C. *Beta-Conexão: Uma Família de Objetos Tridimensionais Reconstruídos a partir de Seções Planares*. 73 p. Dissertação (Mestrado) — USP - São Carlos, 2001.

# ***APÊNDICE A – *Plano de TCC****

## **Plano de Trabalho de Conclusão de Curso**

### **Interpolação de Curvas com Mudança Topológica**

UDESC - Centro de Ciências Tecnológicas

Departamento de Ciências da Computação

Bacharelado em Ciência da Computação - Integral

Turma 2006/1 - Joinville - Santa Catarina

**Guilherme Rossetti Anzolin - [dcc6gra@joinville.udesc.br](mailto:dcc6gra@joinville.udesc.br)**

**Orientador: Marcelo da Silva Hounsell - [marcelo@joinville.udesc.br](mailto:marcelo@joinville.udesc.br)**

**Resumo** - Através deste trabalho de conclusão de curso serão estudados métodos de interpolação entre curvas em casos que ocorrem mudança de topologia das mesmas. Para tratar este problema, que tem aplicações desde a área médica até a representação tridimensional de terrenos, serão estudados conceitos e técnicas de modelagem geométrica, bem como a criação e representação de modelos tridimensionais através da interpolação de curvas. Será implementado um sistema para demonstrar a solução proposta para este problema de maneira eficiente.

**Palavras-chave:** interpolação de curvas, modelagem geométrica, computação gráfica.

## A.1 Introdução e Justificativa

### A.1.1 Introdução e Justificativa

A base do que seria a computação gráfica foi lançada na década de 60, e pela primeira vez passou-se a interagir com o computador através da linguagem usual de desenho, ou seja, solicitando-se o desenho de linhas e círculos, que eram mostrados nas telas de terminais.

A Modelagem Geométrica é uma área da Computação Gráfica que estuda, através de métodos matemáticos, mecanismos de criação de modelos de objetos, tratando da síntese, manipulação e topologia dos objetos gráficos no computador [Dieguez, 1989][Foley et al, 1996]. Na Modelagem Geométrica, a representação de curvas é a base para a descrição de modelos, desde formas simples até objetos complexos. Representar curvas através de uma simples seqüência de retas pode ser suficiente para algumas aplicações, porém quando se trata de curvas mais complexas existem inúmeras formas de representação, por exemplo, através de um conjunto de pontos; uma representação analítica; formas paramétricas e não-paramétricas [Azevedo e Conci, 2003].

De uma maneira geral, as superfícies são uma generalização das curvas. Uma superfície também pode ser gerada por uma seqüência de pontos, e assim como uma curva pode ser representada de forma analítica; explícita ou implícita; paramétrica ou não [Azevedo e Conci, 2003]. É possível interpolar superfícies a partir de pontos, essa é uma forma de gerar objetos através de seus contornos.

Através de superfícies, podem ser representados objetos 3D, esta é uma forma de representação denominada *Boundary Representation (B-rep)*, e consiste na descrição de um objeto em termos de seus limites de superfícies: vértices, bordas e faces [Foley et al, 1996]. A grande vantagem na utilização do método de representação de superfícies, em relação a métodos de representação volumétrica (que leva em consideração todo o volume do objeto) é a quantidade de informação armazenada, pois para gerar as fronteiras (superfícies) de um objeto são necessárias bem menos informações do que para gerar seu volume [Gattass e Peixoto, 2000].

Neste trabalho a descrição dos modelos geométricos será feita pela representação de superfícies através de uma interpolação entre curvas que estarão localizadas em seções paralelas. Uma definição formal para este problema é:

Dado um conjunto de curvas  $c_{ij}$ , onde  $i=1, 2, \dots, n$  e  $n$  é o numero de curvas diferentes no plano  $x_j$ , criar um objeto  $O$  sendo que as curvas  $c_{ij}$  sejam a interseção de  $O$  com os planos  $x_j$ . Cada curva  $c_{ij}$  é representada por uma seqüência de  $p_i$  pontos orientados em sentido horário.

A Figura A.1 ajuda a ilustrar o problema, mostrando no primeiro espaço as curvas fechadas paralelas em seus respectivos planos, e no segundo a superfície gerada a partir da interpolação.

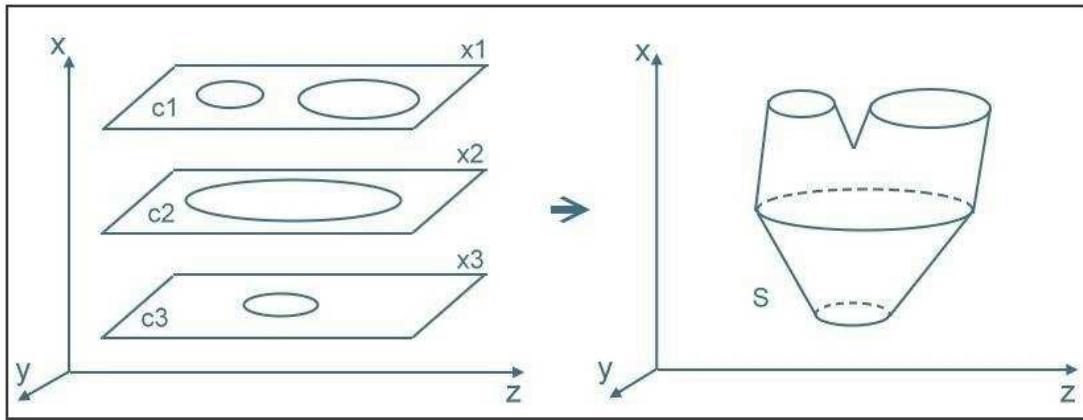


Figura A.1: Interpolação de curvas

Como a Figura A.1 pode mostrar, uma fatia  $c_i$  pode conter uma ou mais curvas fechadas independentes no mesmo plano, gerando assim uma mudança topológica na interpolação entre as fatias  $c_i$  e  $c_{i+1}$ . Como nada é definido para o comportamento das curvas no entre-planos, percebe-se que podem existir vários objetos que satisfaçam ao enunciado, ou seja, através de seções com mudança topológica é possível realizar diferentes interpolações, gerando diferentes objetos. Para o caso de  $c_i$  e  $c_{i+1}$  representarem cada, uma única curva fechada e uma interpolação linear, então, pode-se garantir unicidade na solução. Este trabalho visa obter uma solução algorítmica para o caso de interpolações lineares, a fim de estabelecer critérios para a interpolação quando ocorrer mudança topológica entre as seções.

Uma das possíveis aplicações em reconstrução de superfícies é na área da medicina, onde

através de tomografias computadorizadas e ressonâncias magnéticas as informações sobre o corpo humano são armazenadas em fatias bidimensionais que podem ser interpoladas para gerar a superfície 3D da imagem em questão [Gattass e Peixoto, 2000]. A Figura A.2 ilustra a reconstrução de um fêmur a partir imagens bidimensionais.

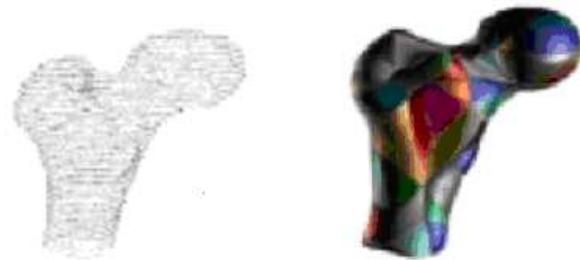


Figura A.2: Reconstrução de um fêmur. Fonte: Gattass e Peixoto (2000)

Outra aplicação interessante está na reconstrução de terrenos, principalmente para a identificação de canais em reservatórios petrolíferos, onde através de imagens de fatias do solo pode-se interpolar os canais identificados representando toda a estrutura do canal. Modelar os terrenos onde se encontram reservatórios permite a visualização e entendimento de sua estrutura e comportamento de forma mais clara, permitindo ao profissional um auxílio na tomada de decisões para a alocação de novos poços de petróleo [Romeu et al, 2000].

## A.2 Objetivos

### A.2.1 Objetivo Geral

Desenvolver uma solução eficiente para a interpolação entre curvas com mudança topológica.

### A.2.2 Objetivos Específicos

Os resultados parciais a serem alcançados no decorrer do trabalho são os seguintes:

- Entender conceitos e técnicas de modelagem geométrica para a criação e representação de modelos;
- Identificar técnicas existentes na literatura que sejam apropriadas à Interpolação;

- Estabelecer uma solução para a interpolação de curvas considerando critérios de eficiência (à serem levantadas e estabelecidas);
- Projetar e implementar a solução proposta.

### A.2.3 Metodologia

Para o desenvolvimento deste trabalho será adotado o método de pesquisa aplicada. Será feito um levantamento bibliográfico sobre o assunto proposto e a partir de uma base teórica será proposta uma solução algorítmica para o problema e implementado um sistema para demonstrar tal solução.

O trabalho será desenvolvido com as seguintes etapas:

- 1.Fundamentação Teórica sobre Modelagem Geométrica (representação de curvas e superfícies, modelagem de sólidos, características);
- 2.Levantamento de técnicas possíveis consolidadas e artigos correlatos ao tema (interpolação de curvas);
- 3.Definição do escopo do projeto, opções de solução e solução adequada;
- 4.Seleção, aprofundamento e desenvolvimento da interpolação;
- 5.Projeto do sistema, levantamento das tecnologias, funções do software e interface;
- 6.Escrita de Monografia I;
- 7.Implementação;
- 8.Avaliação, otimização e manutenção do sistema;
- 9.Site do projeto (o programa, um visualizador de objetos em VRML e dados do projeto);
- 10.Escrita de Monografia II;

Ao final deste TCC será obtido um melhor entendimento do problema da interpolação de curvas e reconstrução de superfícies; será obtida a descrição detalhada de uma solução; estará disponível um *software* com interface gráfica que demonstre a solução adotada gerando objetos que poderão ser exportados e haverá um *site* com a descrição dos principais aspectos do projeto, o software com vários exemplos de objetos e referências.

#### **A.2.4 Cronograma**

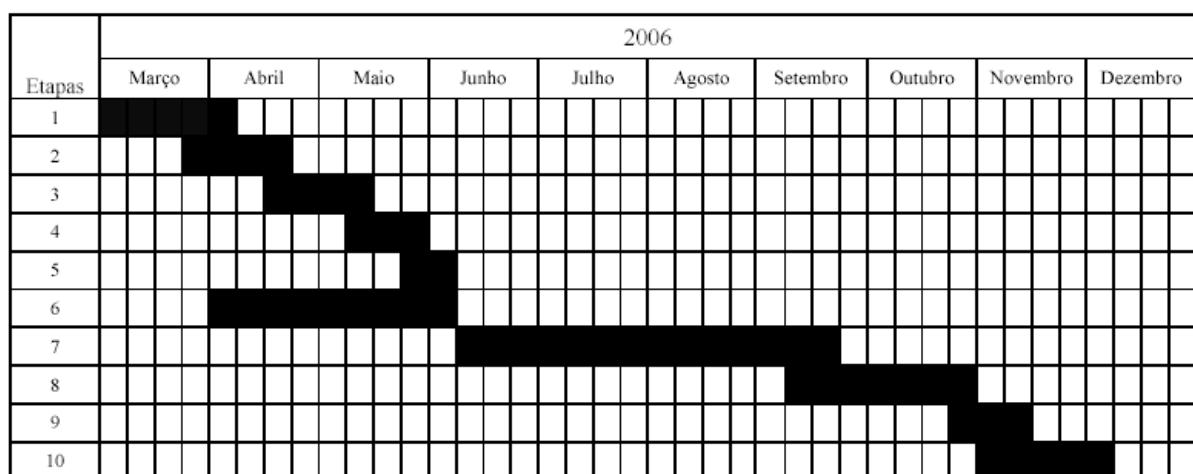


Figura A.3: Cronograma

#### **A.2.5 Linha e Grupo de pesquisa**

A linha de pesquisa deste trabalho de conclusão de curso é o estudo de técnicas de modelagem geométrica, especificamente a interpolação de curvas. O trabalho se insere nas linhas de pesquisa do grupo de pesquisa LARVA (LABoratório de Realidade Virtual Aplicada).

#### **A.2.6 Forma de Acompanhamento/Orientação**

O acompanhamento com o orientador será através de uma reunião presencial semanal com horário a ser definido. Se houver a necessidade, outros horários poderão ser definidos. Para um registro mais preciso do comparecimento aos horários combinados será utilizada uma ficha de controle de reuniões e atividades.