

UNIVERSIDADE DE SÃO PAULO

Instituto de Ciências Matemáticas e de Computação

**β -Conexão: Uma Família de Objetos Tridimensionais
Reconstruídos a Partir de Seções Planares**

Alex Jesús Cuadros Vargas
Luis Gustavo Nonato

Nº 62

NOTAS



São Carlos - SP

UNIVERSIDADE DE SÃO PAULO
Instituto de Ciências Matemáticas e de Computação
ISSN 0103-2577

**β -Conexão: Uma Família de Objetos Tridimensionais
Reconstruídos a Partir de Seções Planares**

Alex Jesús Cuandros Vargas
Luis Gustavo Nonato

Nº 62

NOTAS

Série Computação



São Carlos – SP
Out./2001

β -CONEXÃO: UMA FAMÍLIA DE OBJETOS TRIDIMENSIONAIS RECONSTRUÍDOS A PARTIR DE SEÇÕES PLANARES

Alex Jesús Cuadros Vargas

Luis Gustavo Nonato

Departamento de Ciências de Computação e Estatística

Instituto de Ciências Matemáticas e de Computação

ICMC/USP – São Carlos

Caixa Postal 668

13560-970, São Carlos – SP, Brasil

{acvargas, gnonato}@icmc.sc.usp.br

Resumo. Neste trabalho, uma nova técnica de reconstrução tridimensional a partir de seções planares é apresentada. Esta técnica oferece uma família de modelos, governada por um parâmetro chamado β , que pode ser obtida a partir de um mesmo conjunto de seções planares. Este fato torna possível múltiplas opções para o problema da correspondência, tarefa difícil de ser realizada por outros métodos de reconstrução. Além da flexibilidade na conexão entre regiões de níveis consecutivos, o algoritmo produz LP-variedades e respeita a condição de re-exemplificação.

Palavras chave: Reconstrução tridimensional, Triangulação de Delaunay, Visualização Científica.

Abstract. In this work a new three-dimensional reconstruction technique from planar cross sections is presented. This technique offers a family of tridimensional models, governed by a parameter denoted β , that can be obtained from a same set of cross sections. This fact makes possible multiples choices for the correspondence problem, a difficult task to be performed by other reconstruction methods. Besides the flexibility in the connection among regions of the adjacent slices, the algorithm produces PL-manifolds and respects the re-sampling condition.

Keywords: 3D Reconstruction, Delaunay Triangulation, Scientific Visualization.

1 Introdução

A visualização científica tem sido utilizada em muitas áreas da ciência, especialmente na área médica, onde desempenha um papel importante para diagnósticos, cirurgias e terapias. Dentre as técnicas de visualização, a reconstrução tridimensional ganhou ampla aceitação já que permite visualizar objetos tridimensionais a partir de dados bidimensionais ou seções planares paralelas (imagens 2D). Em medicina, dados bidimensionais podem ser obtidos por dispositivos de aquisição de imagens como Tomografia por Ressonância Magnética (*MRI - Magnetic Resonance Imager*), Tomografia Computarizada (*CT - Computerized Tomography*) ou Microscopia Eletrônica por Laser. Estes

resultados são apresentados como uma lista de imagens 2D com resolução e clareza variáveis. Todos os algoritmos desenvolvidos para reconstruir modelos tridimensionais têm que lidar com três problemas intrínsecos à reconstrução, a saber: O problema da **Correspondência** (*correspondence*), aparece quando as regiões contidas em dois níveis consecutivos se dividem ou se juntam, o problema de **Geração de malha** (*tiling*), refere-se ao aspecto da triangulação na faixa entre dois níveis consecutivos, e o problema de **Ramificação** (*branching*), relacionado com a identificação de pontos de sela (singularidades) que aparecem quando o número de contornos nas seções adjacentes é diferente.

Embora os algoritmos de reconstrução descritos na literatura apresentem estratégias diferentes para resolver os problemas de correspondência, geração de malha e ramificação, eles têm uma característica em comum: a inflexibilidade na escolha da correspondência. A maioria das técnicas de reconstrução supõe a correspondência previamente definida ou assume um critério de proximidade para conectar as regiões contidas em seções planares adjacentes. A importância da flexibilidade na escolha da correspondência é evidente, já que somente a informação contida nas seções planares não é suficiente para garantir uma correta conexão. Este fato pode ser observado melhor na figura 1.



Figura 1 – Problema da correspondência.

(a) Conjunto de regiões em planos paralelos. (b, c, d) Conexões possíveis a partir de um mesmo conjunto de regiões.

Este trabalho apresenta uma nova técnica de reconstrução a partir de seções planares. Esta técnica permite múltiplas escolhas na definição da correspondência entre regiões contidas em seções planares adjacentes. Baseado na triangulação de Delaunay, o algoritmo oferece uma família de modelos que podem ser obtidos a partir de um mesmo conjunto de seções planares, sendo assim, uma ferramenta interessante na reconstrução de objetos com alto grau de ramificação ou união. A família de modelos está governada por um valor inteiro positivo denominado: parâmetro β . Com $\beta = 0$, não existe correspondência entre as regiões e quando β cresce a correspondência cresce também. Além da flexibilidade oferecida na correspondência, o algoritmo garante que o modelo final é uma variedade tridimensional linear por partes (*PL-manifold*), o que é desejado em simulações numéricas. Outra importante característica deste algoritmo é que ele respeita o critério de re-exemplificação (*re-sampling*), este critério significa que a interseção do modelo com os planos originais de corte deve resultar nas regiões originais.

Este trabalho está organizado como segue: na seção 2 são apresentados alguns trabalhos prévios de reconstrução tridimensional. Os conceitos fundamentais do Diagrama de Voronoi e Triangulação de Delaunay são explicados na

seção 3. A seção 4 apresenta o pipeline de reconstrução tridimensional a partir de seções planares, utilizando o novo algoritmo β -Conexão proposto por este trabalho. Na seção 5 são apresentados alguns resultados obtidos. Finalmente na seção 6 são apresentados algumas conclusões e trabalhos futuros.

2 Trabalhos relacionados

Desde a aparição dos dispositivos de medição não invasivos (*MRI, CT, etc*) capazes de representar um objeto real a partir de uma seqüência de imagens, muitas técnicas de reconstrução tridimensional têm sido desenvolvidas. Algumas destas técnicas constroem uma superfície que representa o bordo do objeto e outras geram modelos com representações volumétricas. Entre as principais técnicas de reconstrução podemos citar: as técnicas implícitas, os modelos baseados em *voxels*, os métodos de otimização, os modelos deformáveis, os modelos baseados em nuvens de pontos e as técnicas heurísticas. A seguir algumas destas estratégias de reconstrução são analisadas dando ênfase na forma em que elas resolvem o problema da correspondência.

As técnicas implícitas geram uma função implícita suave de forma que o bordo do objeto que se quer representar está no conjunto zero desta função. No trabalho feito por Jones *et al.* [Jon94], uma função é definida a partir de uma distância com sinal para cada um dos contornos. Esta função, juntamente com a técnica *Marching Cubes* [Lor87], é utilizada para extrair a superfície do modelo. Utilizando técnicas implícitas é possível garantir que a função criada contenha os contornos originais no seu conjunto de valores zero, mas é difícil garantir que os contornos nas superfícies geradas sejam os mesmos que os contornos originais. A correspondência é automaticamente definida pela função implícita, fazendo difícil outra opção de conexões.

Os modelos baseados em *voxels* [Kau98] visualizam o volume do objeto diretamente a partir das propriedades das unidades do volume de dados, chamados *voxels*. Esta visualização é feita sem construir uma superfície ou malha volumétrica. Embora este tipo de estratégia evite problemas como segmentação, correspondência, geração de malha e ramificação, é praticamente impossível especificar a correspondência entre as regiões, porque as interpolações usadas para gerar a imagem estipulam as conexões. No entanto, este tipo de modelo pode ser uma alternativa interessante para a visualização de informações volumétricas.

Tentativas de reconstrução mediante métodos de otimização utilizam usualmente teoria de grafos. Um dos primeiros trabalhos em reconstrução a partir de seções planares foi desenvolvido por Keppel [Kep75]. Neste trabalho a superfície que maximiza o volume do objeto que esta sendo reconstruído é obtida a partir de uma busca ótima em um grafo. Shinagawa *et al.* [Shi91] expandem um grafo toroidal discreto para formar um grafo contínuo, e uma homotopia é usada para reconstruir superfícies paramétricas e definir as correspondências. Meyers *et al.* [Mey92] aproximam os contornos dentro de cilindros e uma árvore geradora mínima determina a correspondência entre eles. Os métodos de otimização podem resolver o problema de geração de malha, mas alguma heurística é necessária para encontrar uma solução ao problema de ramificação.

Os modelos deformáveis combinam geometria, física e teoria da aproximação para gerar o modelo. A partir de um modelo inicial, deformações são executadas para aproximar o objeto original. Embora conexões diferentes possam ser

obtidas usando distintos modelos iniciais, realizar tais inicializações não é uma tarefa trivial. Uma boa revisão de modelos deformáveis pode ser encontrada no trabalho feito por Singh *et al.* [Sin98]. A vantagem de utilizar modelos deformáveis é que a segmentação pode ser incorporada ao processo de reconstrução, embora objetos com topologias complexas sejam difíceis de serem tratados. Delingette [Del94], utilizando *simplex meshes*, apresenta uma alternativa para o problema da topologia.

Existem também técnicas que reconstróem o objeto original a partir de uma nuvem de pontos. A maioria destas técnicas assumem que a amostra de pontos de entrada satisfaz alguns critérios de densidade. Sendo assim, é possível garantir que a saída seja topologicamente correta e convergente à superfície original. Nina Amenta *et al.* [Ame97], baseados no diagrama de Voronoi, e Dey *et al.* [Dey00], baseados na triangulação de Delaunay, criaram técnicas deste tipo. A correspondência em este caso é calculada baseada na proximidade dos pontos fazendo como que seja única.

Existe uma quantidade enorme de algoritmos que fazem uso de heurísticas para decidir a respeito das conexões entre contornos em planos consecutivos. Ekoule *et al.* [Eko91] mapeam contornos planares dentro de seus fechos convexos e um critério baseado na aresta mais curta é usado para conectar os fechos convexos que estão em níveis adjacentes. O problema de ramificação é resolvido criando um contorno intermediário entre dois níveis consecutivos. Bajaj *et al.* [Baj96] definem alguns critérios que a superfície reconstruída deve satisfazer, e a partir destes critérios, um modelo é construído. Nas regiões onde as condições não podem ser satisfeitas, a superfície é completada através de uma triangulação, levando em consideração o esqueleto de Voronoi. A correspondência entre regiões é implicitamente definida pelo critério de reconstrução, e um modelo volumétrico pode ser gerado usando um processo de tetradralização [Baj99]. Boissonnat [Boi88] utiliza um grafo obtido a partir da projeção ortogonal dos diagramas de Voronoi 2D para gerar a triangulação de Delaunay 3D. Depois, utilizando este grafo e eliminação de tetraedros para cada dois níveis consecutivos, Boissonnat gera o modelo volumétrico do objeto. A correspondência é dada, neste caso, pela triangulação de Delaunay juntamente com a eliminação de tetraedros. Geiger [Gei93] utiliza a idéia de Boissonnat projetando os esqueletos de Voronoi bidimensionais em níveis consecutivos para manipular ramificações complexas e contornos desiguais. Nesta alternativa, não é tratado o problema de singularidades geradas no modelo durante a eliminação de tetraedros.

Nonato [Non98],[Non01], motivado pelas idéias de Boissonnat e de Geiger, utiliza a triangulação de Delaunay tridimensional para construir um modelo volumétrico do objeto. Para estabelecer a correspondência entre contornos, Nonato baseia-se num critério de proximidade que pode ser obtido verificando a existência de um tipo especial de tetraedro na triangulação de Delaunay, que ele chama de tetraedro reverso. Ele considera que quando este tipo de tetraedro aparece na triangulação de Delaunay gerada a partir dos pontos dos contornos, estes contornos estão geometricamente bem posicionados, e então são conectados. Este algoritmo pode aceitar outras opções de conexão se previamente definidas. Uma importante característica deste algoritmo é que o modelo gerado, além de tratar os problemas da reconstrução de forma bastante satisfatória, satisfaz também os critérios de re-amostragem e 3-variedade LP.

O algoritmo apresentado neste trabalho, β -Conexão, está baseado na triangulação de Delaunay e nas idéias do algoritmo de Nonato. Com o β -Conexão é possível ter múltiplas opções para o problema da correspondência, oferecendo uma família de modelos tridimensionais a partir de um mesmo conjunto de seções planares. Além da flexibilidade oferecida na escolha da correspondência, fato dificilmente de ser obtido por outros algoritmos, β -Conexão produz 3-variedades LP e respeita o critério de re-amostragem.

3 Diagramas Voronoi e Triangulação de Delaunay

Um conjunto finito não vazio $A = \{x_1, \dots, x_z\}$ em R^m está em **posição geral** se nenhum subespaço afim de R^m contém A , e se não há uma esfera A^{m-1} passando através de um subconjunto de A com $m+k$, $k>1$ pontos. No texto a seguir sempre vamos considerar conjuntos de pontos em posição geral.

O **Diagrama de Voronoi** V de A é uma decomposição de R^m em células m -dimensionais $\{V_1, \dots, V_n\}$ com as seguintes propriedades (Figura 2a):

- Cada V_i contém apenas um ponto $x_i \in A$
- Dado $x \in R^m$, $x \in V_i$ se e somente se $d(x, x_i) \leq d(x, x_j)$ para todo $i \neq j$, onde d é a distância euclidiana.

Cada V_i será chamado **poliedro de Voronoi** associado a x_i . Note que $R^m = \cup V_i$ e que cada poliedro de Voronoi é definido pela interseção de $n-1$ semi-espacos fechados. Como semi-espacos são convexos e a interseção de conjuntos convexos é convexa tem-se que cada poliedro de Voronoi é convexo. Destas propriedades pode-se mostrar que a interseção de k poliedros do Diagrama de Voronoi, onde $2 \leq k \leq m+1$, é vazia ou é uma célula $(m-k+1)$ -dimensional contida no diagrama.

A **Triangulação de Delaunay** DT (Figura 2b) é o dual geométrico do Diagrama de Voronoi (VD), isto é, para cada célula p -dimensional do diagrama pode-se associar de forma natural um $(m-p)$ -simplexo da triangulação. Por exemplo, no caso de R^2 (Figura 2c):

- Um triângulo (2-simplexo) de DT está associado a um vértice do Diagrama de Voronoi,
- Uma aresta (1-simplexo) de DT está associada a uma aresta do Diagrama de Voronoi,
- Um vértice (0-simplexo) de DT está associado a uma célula de dimensão 2 do Diagrama de Voronoi.

A consequência da dualidade e da posição geral dos pontos é que cada célula 0-dimensional do Diagrama de Voronoi é o centro da esfera que circunscreve um m -simplexo da Triangulação de Delaunay. Uma importante propriedade que caracteriza a triangulação de Delaunay é que a esfera que circunscreve um m -simplexo não pode conter qualquer ponto de A em seu interior (Figura 2d). Uma descrição detalhada de alguns algoritmos de Triangulação de Delaunay e Diagramas de Voronoi pode ser encontrada em [For94].

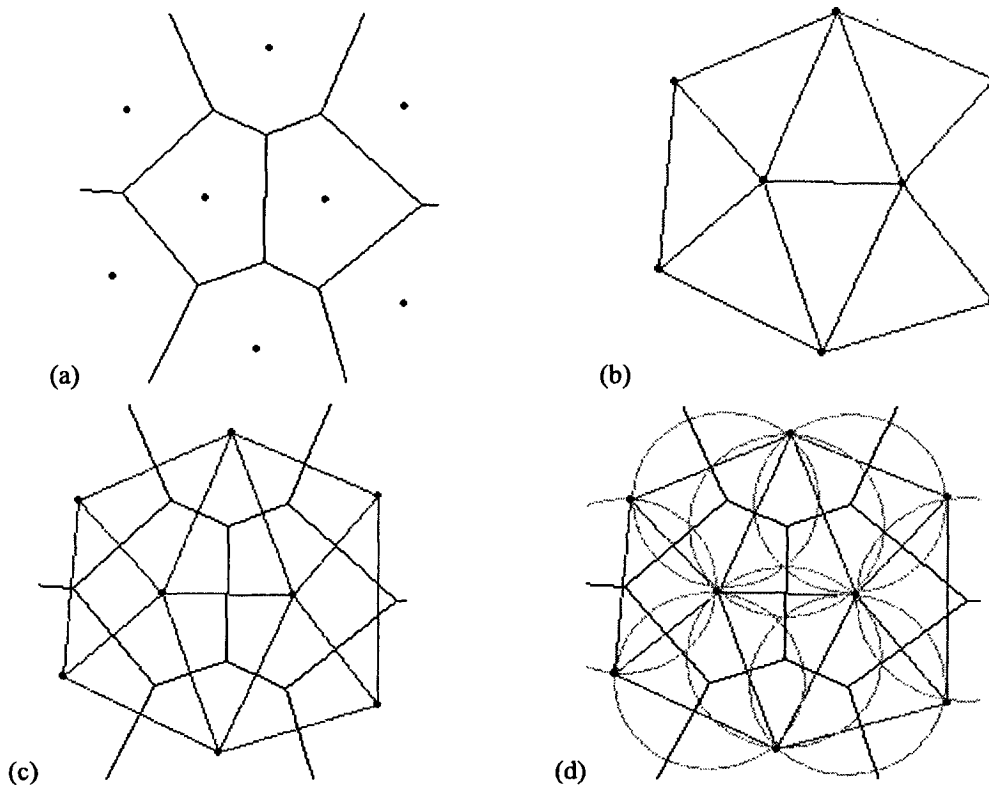


Figura 2 – Diagrama de Voronoi e triangulação de Delaunay.

(a) Diagrama de Voronoi, (b) Triangulação de Delaunay, (c) Triangulação de Delaunay como dual geométrico do diagrama de Voronoi e (d) Conseqüência da dualidade.

4 Pipeline de Reconstrução Tridimensional

O esquema geral do *pipeline* de reconstrução implementado neste trabalho pode ser apreciado na figura 3.

As entradas para o *pipeline* de reconstrução são:

- Um conjunto de arquivos arq_1, \dots, arq_n , $n \geq 2$, contendo o conjunto de seções planares $\pi_1 \dots \pi_n$, a partir das quais se quer reconstruir o objeto original. Cada seção planar π_i contém um conjunto de contornos C_1, \dots, C_k , $k \geq 2$, representando o bordo de regiões.
- O parâmetro β , que especifica a "densidade" com que as regiões nos planos paralelos serão conectadas.

A saída do *pipeline* é um objeto tridimensional formado por um conjunto de tetraedros $T_1 \dots T_t$, $t \geq 3$.

Na implementação deste trabalho foi escolhido o formato da biblioteca VTK (*Visualization ToolKit*) [Sch98] para representar os arquivos de entrada e de saída.

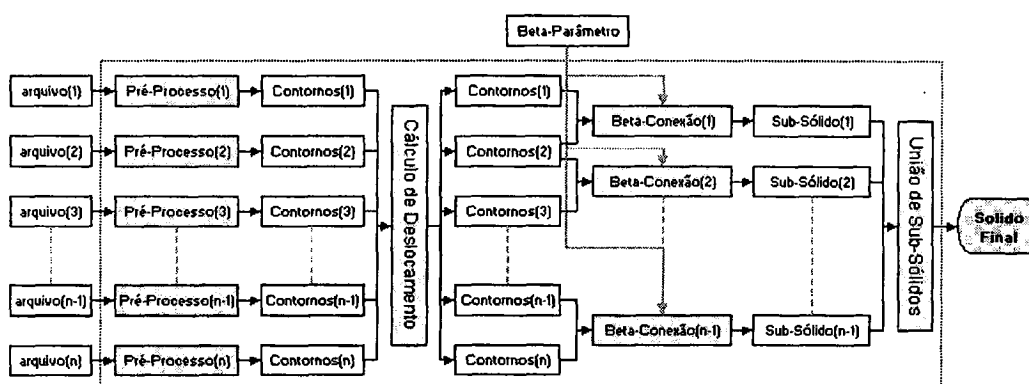


Figura 3 – Sub-processos principais do *pipeline* de reconstrução.

O *pipeline* possui quatro processos principais: pré-processamento, cálculo de deslocamento, β -Conexão e União de sub-sólidos. A seguir, cada um destes processos são explicados.

4.1 Pré-Processo de contornos

Nesta etapa cada nível de dados de entrada é processado individualmente, com a finalidade de obter contornos que facilitem a etapa de reconstrução tridimensional. O pré-processamento está dividido em três sub-processos: Leitura de contornos, Orientação de contornos e Verificação de arestas Delaunay.

Na **leitura de contornos** um arquivo arq_i contendo os pontos dos contornos $C_1 \dots C_k$, $k \geq 1$, de uma seção planar π_i é lido. Seguidamente, os contornos lidos são orientados de forma coerente na etapa de **orientação de contornos**.

Camihando por um contorno que possui orientação coerente, tudo o que está à esquerda é denominado **região interna**, e tudo o que permanece à direita será denominado **região externa**. Os contornos coerentes são de muita utilidade para o algoritmo de reconstrução β -Conexão. A última etapa do pré-processo é a **Verificação de arestas Delaunay**. Na etapa de reconstrução tridimensional será necessário calcular a triangulação de Delaunay 3D a partir dos vértices dos contornos contidos em dois níveis consecutivos. Nesta triangulação, todas as arestas dos contornos devem estar presentes. O objetivo desta etapa é garantir este fato.

4.2 Cálculo de deslocamento

Os sub-sólidos gerados pelo algoritmo de reconstrução da etapa de reconstrução deverão ser unidos na última etapa para formar o sólido final. O objetivo do cálculo de deslocamento é facilitar esta união. Para esta finalidade, em cada

plano π_i devem ser calculados dois dados: O número de pontos no plano e o somatório do número de pontos dos níveis anteriores.

4.3 Reconstrução tridimensional - β -Conexão

Nesta seção, apresentamos o β -Conexão, o novo algoritmo de reconstrução tridimensional a partir de seções planares proposto neste trabalho. Baseado na triangulação de Delaunay, e nas idéias do algoritmo de Nonato [Non98, Non01], criou-se uma nova técnica para resolver o problema da correspondência. Esta técnica oferece múltiplas escolhas de conexão entre regiões contidas em seções planares adjacentes.

Este algoritmo recebe duas entradas:

- A primeira entrada são dois níveis ou seções planares consecutivas π_i e π_{i+1} , contendo um conjunto de contornos já pré-processados.
- A segunda entrada é o β -Parâmetro. Este parâmetro expressa a densidade com que os contornos serão conectados.

A saída deste algoritmo é o sub-sólido reconstruído.

Este algoritmo está dividido em 6 etapas: Triangulação de Delaunay 3D, Classificação de arestas, Classificação de tetraedros, Cálculo de componentes conexas, Desconexão de componentes e Subdivisão de tetraedros.

O β -Conexão começa executando a triangulação de Delaunay 3D com pontos contidos em duas seções planares adjacentes. Em seguida, as arestas desta triangulação, que estão contidas nas duas seções planares são classificadas. A partir desta classificação de arestas, os tetraedros da triangulação são classificados também. Baseado nesta última classificação, o β -Conexão determina as componentes conexas do modelo, ou seja, quais regiões ficarão conectadas entre si. Para isso, calcula-se o caminho mais curto em tetraedros entre cada par de regiões nos níveis consecutivos. Depois, utiliza-se um valor inteiro positivo chamado β (estabelecido pelo usuário) para expressar uma mínima distância em tetraedros. Esta distância mínima serve como condição para que as regiões sejam conectadas. Isto é, para um $\beta = n$, todas as regiões que estejam a uma distância em tetraedros menor o igual a n ficarão conectadas. Desta forma, a partir de um conjunto de contornos em dois níveis consecutivos, é possível gerar uma família de objetos reconstruídos atribuindo diferentes valores a β . Após ter determinado as componentes conexas, os grupos de regiões que ficaram conectadas no modelo devem ser isoladas no processo de desconexão de componentes. Finalmente, para refinar o modelo, um processo de subdivisão de tetraedros é realizado.

O algoritmo β -Conexão, da mesma forma que o algoritmo de Nonato, satisfaz os critérios de Re-amostragem e 3-variedade LP.

- O critério de re-amostragem exige que a interseção do modelo gerado com os planos de corte resulte nos contornos originais.
- O critério de 3-variedade LP impõe que o modelo gerado seja sempre uma 3-variedade linear por partes.

Em seguida, cada um dos processos deste algoritmo será detalhado.

4.3.1 Triangulação

A triangulação de Delaunay 3D, denotada DT , é o primeiro passo deste algoritmo de reconstrução. Esta triangulação é construída com pontos de duas seções planares consecutivas de entrada, como é mostrado na figura 4.

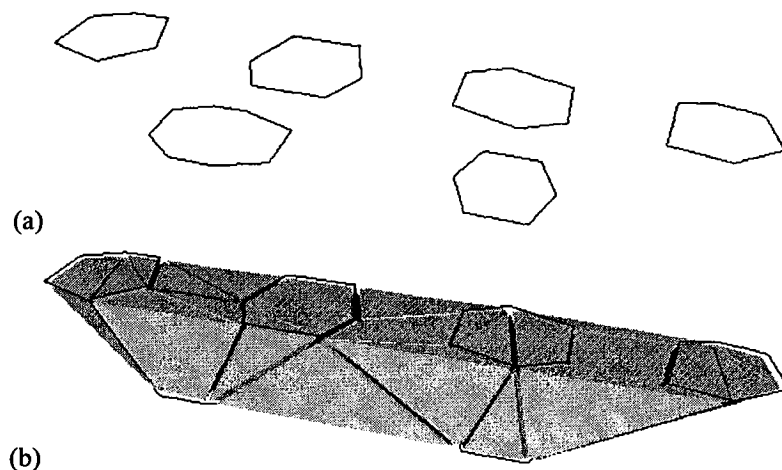


Figura 4 – Triangulação de Delaunay 3D.

(a) Regiões em seções planares paralelas, (b) triangulação de Delaunay 3D com pontos destas regiões.

4.3.2 Classificação de arestas

A partir da triangulação de Delaunay 3D DT obtida no passo anterior e dos contornos orientados, as arestas que estão nos planos π_i e π_{i+1} são classificadas. Isto é, caminhando por um contorno com orientação coerente, as arestas da triangulação DT contidas nos planos π_i e π_{i+1} (figura 5a) podem ser classificadas da seguinte forma.:

- As arestas que estão à esquerda do contorno serão chamadas **arestas internas**.
- As arestas que permanecem à direita do contorno serão chamadas **arestas externas**.
- As arestas dos planos π_i e π_{i+1} , que não são arestas internas nem externas serão chamadas **arestas de contorno**.

A figura 5b apresenta o resultado da classificação de arestas.

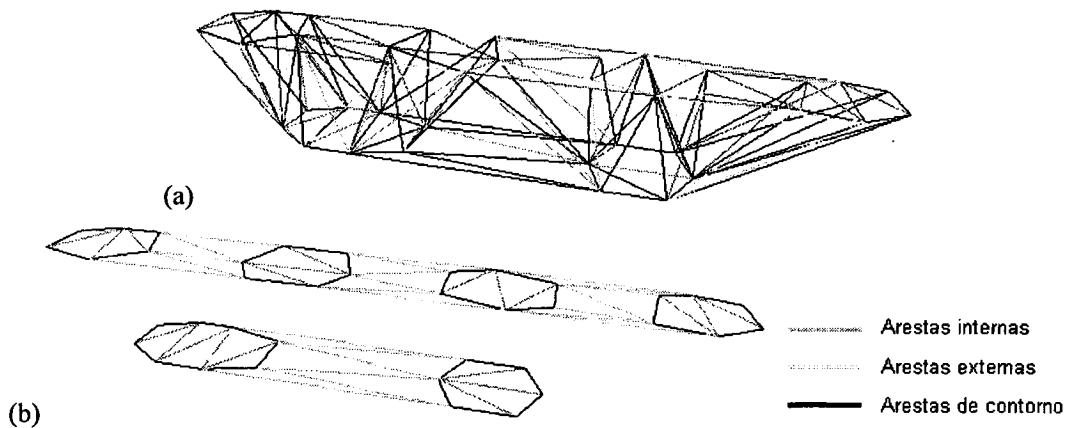


Figura 5 – Classificação de arestas.

(a) Exemplo de uma triangulação de Delaunay em *Wirre frame*,

(b) Arestas internas, Externas e de contorno.

Note que somente as arestas de *DT* que estão nos planos π_i e π_{i+1} são classificadas. As arestas que estão entre π_i e π_{i+1} não são classificadas.

Antes de definir o que são as componentes conexas do modelo, é preciso classificar os tetraedros de *DT*.

4.3.3 Classificação de tetraedros

Tendo classificadas as arestas nos planos π_i e π_{i+1} pelo passo anterior, os tetraedros da triangulação *DT* podem ser classificados também. Pelo algoritmo de *Boissonat* [Boi88], que assume como entrada um conjunto de pontos em \mathbb{R}^3 distribuídos em planos paralelos, nós sabemos que em *DT* existem somente dois tipos de tetraedros:

- O primeiro tipo de tetraedro tem 3 arestas em um dos dois planos paralelos, este tipo de tetraedros será chamado **tetraedro do tipo 1**.
- O segundo tipo de tetraedro possui uma aresta em cada um dos dois planos, este segundo tipo de tetraedro será chamado **tetraedro do tipo 2**.

A partir destes dois tipos de tetraedros, tipo 1 e 2, e da classificação de arestas, em *DT* (Figura 6a), pode-se diferenciar cinco tipos de tetraedros, da seguinte maneira:

- Caso um tetraedro t seja do tipo 1 então:
 - Se t possui uma ou mais arestas internas e não possui arestas externas, ou possui três arestas de contorno de um contorno com orientação anti-horária, t será chamado **tetraedro interno** (Figura 6b).
 - Se t possui uma ou mais externas e não possui arestas internas, ou possui três arestas de contorno de um contorno com orientação horária, t será chamado **tetraedro externo** (Figura 6c).
- Caso um tetraedro t seja do tipo 2 então:

- Se t possui duas arestas internas, t será chamado **tetraedro reverso interno** (Figura 6d).
- Se t possui uma aresta interna e uma externa, t será chamado **tetraedro reverso externo** (Figura 6e).
- Se t possui duas arestas de contorno e não tem arestas internas nem externas, t será chamado **tetraedro redundante** (Figura 6f).
- Se t possui duas arestas externas ou, uma aresta de contorno e uma aresta externa, t será chamado **tetraedro externo** (Figura 6c).
- Se t possui uma aresta de contorno e uma aresta interna, t será chamado **tetraedro interno** (Figura 6b).

A figura 6g apresenta um exemplo do resultado da classificação de tetraedros.

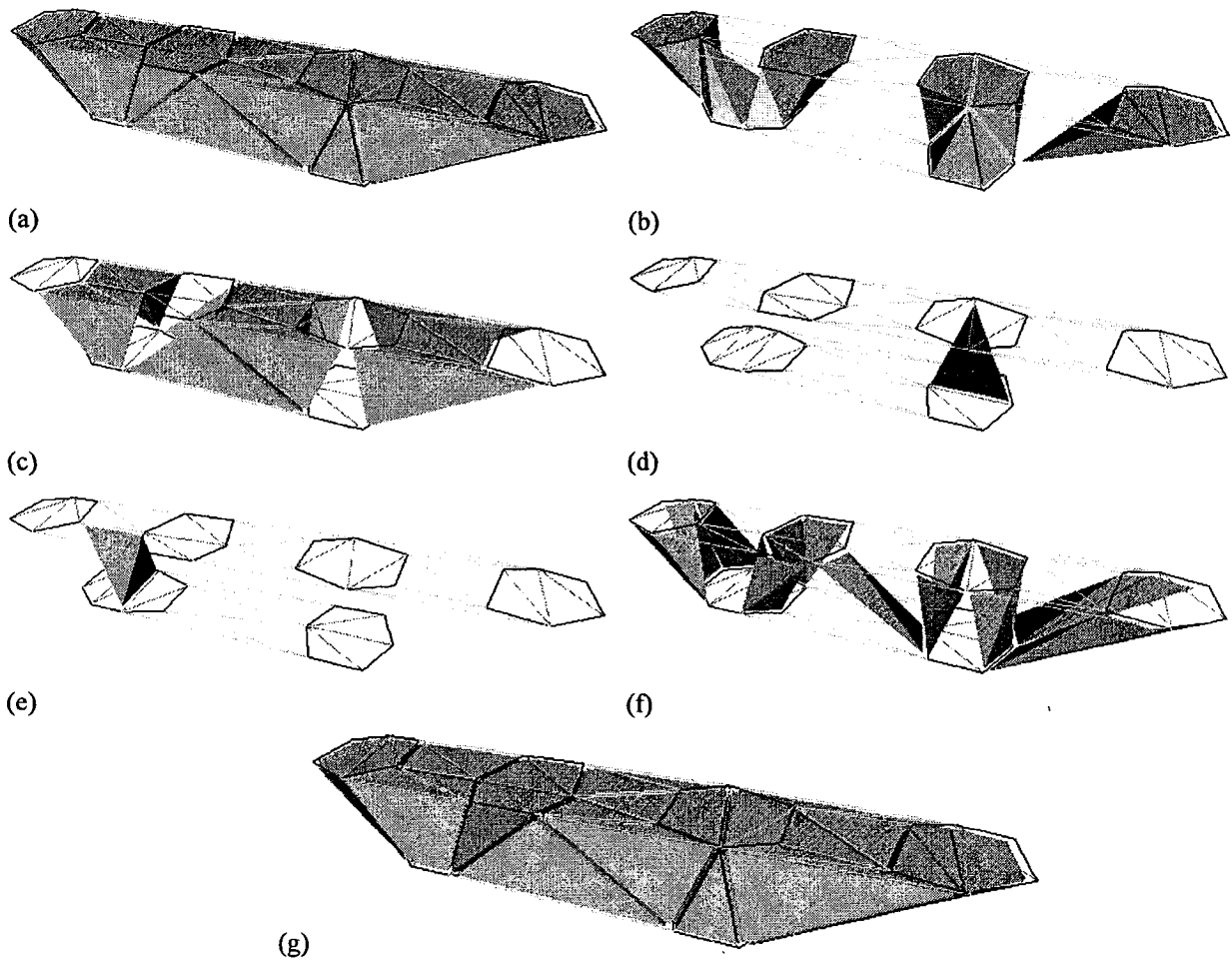


Figura 6 – Classificação de tetraedros.

(a) Exemplo de uma Triangulação de Delaunay com arestas classificadas, (b) Tetraedros internos, (c) Tetraedros externos, (d) Tetraedros reversos internos, (e) Tetraedros reversos externos, (f) Tetraedros redundantes, (g) Resultado da classificação de tetraedros.

No algoritmo de Nonato, considera-se que dois contornos $c_1 \in \pi_i$ e $c_2 \in \pi_{i+1}$ estão geometricamente bem posicionados se e somente se na triangulação de Delaunay tridimensional com pontos de π_i e π_{i+1} existe um tetraedro reverso com vértices em c_1 e c_2 . A heurística para resolver o problema da correspondência do algoritmo β -Conexão, considera também, entre outras coisas, a existência destes tetraedros reversos.

A etapa explicada a seguir é o cálculo de componentes conexas.

4.3.4 Cálculo de componentes conexas

Definir quais regiões devem ser conectadas é uma tarefa difícil, já que a informação destas não é suficiente para garantir uma correta correspondência. Nesta seção, será apresentada uma nova heurística baseada em uma distância mínima em tetraedros, representada por um valor inteiro positivo chamado β . Esta distância é utilizada como uma condição para que as regiões sejam conectadas. Isto é, para um $\beta = n$, todas as regiões que estejam a uma distância em tetraedros menor ou igual a n ficarão conectadas. Desta forma, a partir de um conjunto de regiões contidas em níveis consecutivos π_i e π_{i+1} , é possível gerar uma família de objetos reconstruídos atribuindo diferentes valores a β .

Sejam C_1 e C_2 dois conjuntos de contornos representando o bordo de dois conjuntos de regiões R_1 e R_2 contidos nas duas seções planares de entrada π_i e π_{i+1} , e seja DT a triangulação de Delaunay 3D gerada com pontos em C_1 e C_2 .

A triangulação DT (figura 7a) pode ser considerada, de forma lógica, como um grafo não dirigido G (Figura 7b) contendo dois tipos de nós. Tal grafo pode ser obtido da seguinte forma:

- Os nós de G
 - Cada região $r \in R_1 \cup R_2$, junto com todos seus tetraedros internos, é considerada como um nó chamado **nó região**.
 - Cada tetraedro $t \in DT$, do tipo redundante ou externo, é considerado como um nó chamado **nó externo**.
- As arestas de G
 - Se duas regiões r_1 e r_2 , possuem tetraedros internos com uma face em comum, ou se existe um tetraedro reverso interno com arestas em r_1 e r_2 , então, considera-se uma aresta em G conectando os nós região que representam r_1 e r_2 .
 - Se um tetraedro externo ou redundante, representado por um nó externo w , possui uma face em comum com um tetraedro interno de uma região r , então, considera-se uma aresta em G entre os nós w e r .
 - Considera-se também uma aresta entre dois nós do tipo externo, se os tetraedros que representam estes nós possuem uma face em comum.

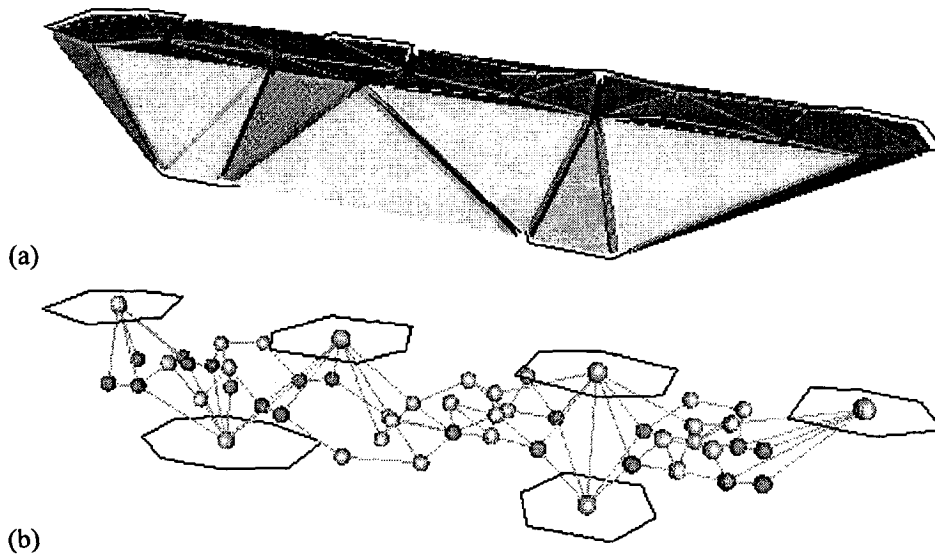


Figura 7 – Grafo lógico associado a uma triangulação de Delaunay 3D com pontos distribuídos em seções planares.

(a) Exemplo de uma triangulação de Delaunay 3D, (b) Grafo gerado a partir da triangulação da figura a.

Por definição, as componentes conexas de um grafo não dirigido são os subconjuntos de nós sob uma relação "é alcançável desde" [Cor90](pg.88). No contexto do algoritmo β -Conexão, chamaremos **componentes conexas** aos subconjuntos de regiões em R_i e R_{i+1} que estão unidos sob a relação "distância mínima" especificada pelo parâmetro β .

A partir do grafo lógico G obtido de DT , as componentes conexas do modelo podem ser calculadas em dois passos:

- **Cálculo de distâncias em tetraedros entre regiões**

Para cada nó região r em G aplica-se o algoritmo de busca em largura (*Breadth-first search*) [Cor90](pg.469-477), tomando como nó inicial o próprio nó região r . Este algoritmo, a partir de um nó inicial, expande a fronteira entre os nós conhecidos e os desconhecidos de forma sistemática. Isto é, o algoritmo alcança todos os nós a uma distância k antes de descobrir qualquer outro nó a uma distância $k+1$. Este algoritmo produz também uma árvore (*Breadth-first tree*) contendo todos os nós alcançáveis desde o nó inicial. A execução deste algoritmo é feita em tempo linear, $O(V+E)$, onde V é o número de nós e E é número de arestas em G [Cor90](pg. 471).

Para qualquer nó região $s \in G$, $s \neq r$, o caminho na árvore desde s até r corresponde ao caminho mais curto entre s e r .

Depois de ter sido feito este processo para cada nó região r em G , origina-se um novo grafo J de distâncias entre regiões. A figura 8 apresenta o grafo de distâncias obtido a partir do grafo da figura 7.

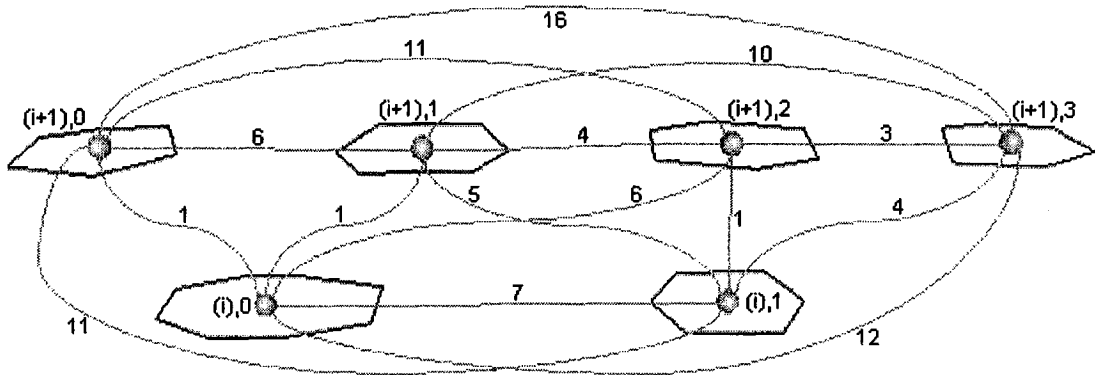


Figura 8 –Grafo de distâncias entre regiões.

- Determinação de componentes conexas

A partir do parâmetro β e do grafo de distâncias entre regiões J , é fácil determinar as componentes conexas do modelo. Isto é, para um $\beta = n$, todas as regiões que no grafo J estejam a uma distância menor ou igual a n , ficarão em uma mesma componente conexa.

Por último, cada tetraedro $t \in DT$, é atribuído à componente conexa da região que este está associado. Os tetraedros externos e redundantes que possuem vértices em duas componentes conexas distintas não são atribuídos a nenhuma componente conexa e serão eliminados na seguinte etapa.

Desta forma, segundo o grafo J , se tem:

- Com $\beta = 0$, cada região forma uma componente conexa (Figura 9a).
- Com $\beta = 1$, todas as regiões que possuam um tetraedro reverso interno entre elas ou tetraedros internos com faces em comum ficarão conectadas (Figura 9b).
- Com $\beta = d, d \geq 2$, todas as regiões que estejam a uma distância menor o igual a d ficarão conectadas (Figura 9c).

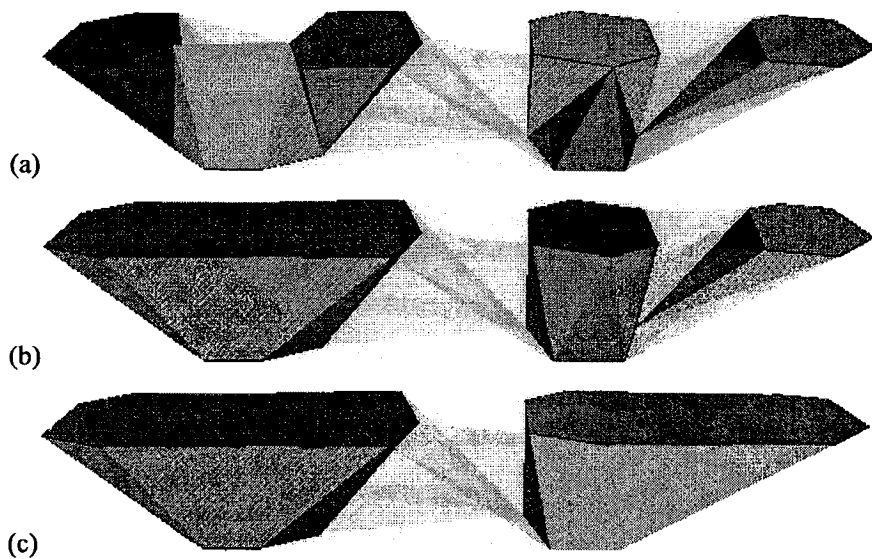


Figura 9 – Componentes conexas de um modelo

(a) Componentes conexas com $\beta = 0$, (b) Componentes conexas com $\beta = 1$,

(c) Componentes conexas com $\beta = 3$.

4.3.5 Desconexão de componentes conexas

Após ter calculado as componentes conexas do modelo, estas devem ser isoladas. Com esta finalidade, os tetraedros externos e redundantes com vértices em duas componentes conexas distintas têm que ser eliminados. Após a eliminação de tetraedros, algumas componentes conexas poderiam não estar totalmente separadas, gerando vértices e arestas singulares. A figura 10 apresenta um modelo que, com $\beta = 2$, gera uma aresta singular e um vértice singular. Para eliminar estas singularidades, os vértices singulares devem ser duplicados e depois deslocados até uma posição apropriada.

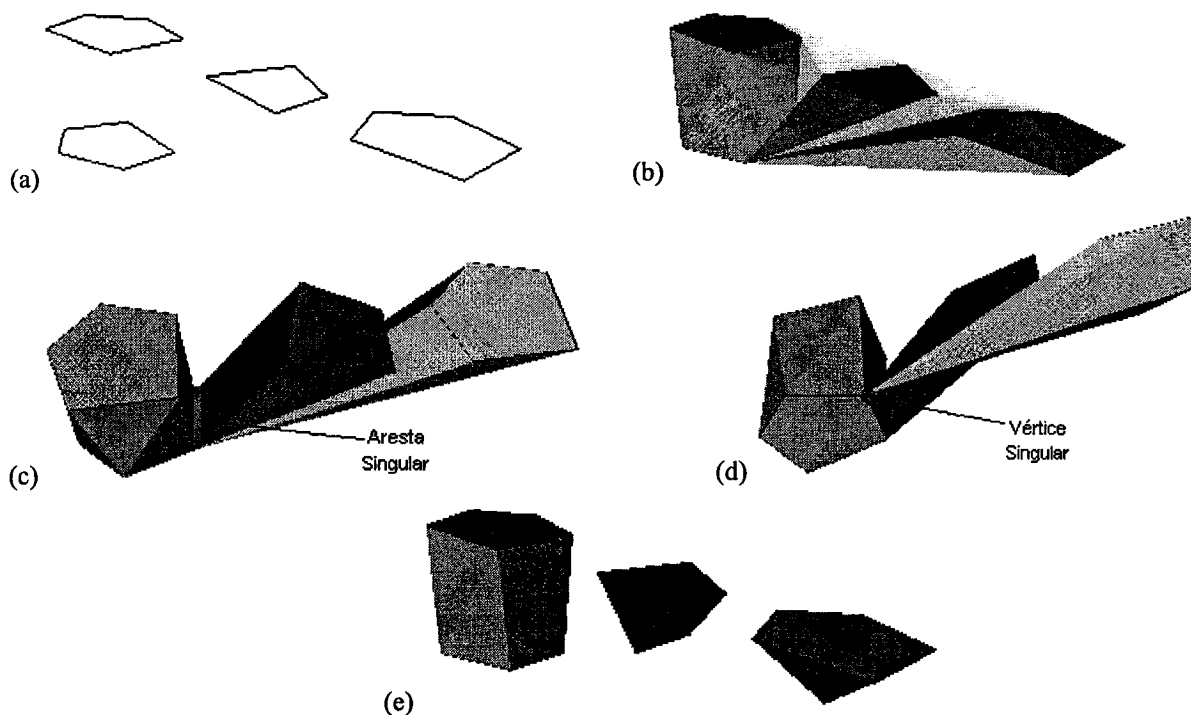


Figura 10 – Exemplo de desconexão de componentes com singularidades
 (a) Contornos do modelo exemplo, (b) Componente conexas,
 (c) Aresta Singular, (d) Vértice singular, (e) Componente Conexas separadas

Existem também casos especiais com $\beta = 0$ junto com tetraedros reversos internos no modelo. A existência de um tetraedro reverso (Figura 11a) indica uma forma sobreposição entre duas regiões. Um tetraedro deste tipo não pode ser separado somente com translação de vértices porque geraria singularidades (Figura 11b). Para evitar estas singularidades, estes tetraedros têm que ser duplicados (Figura 11c).

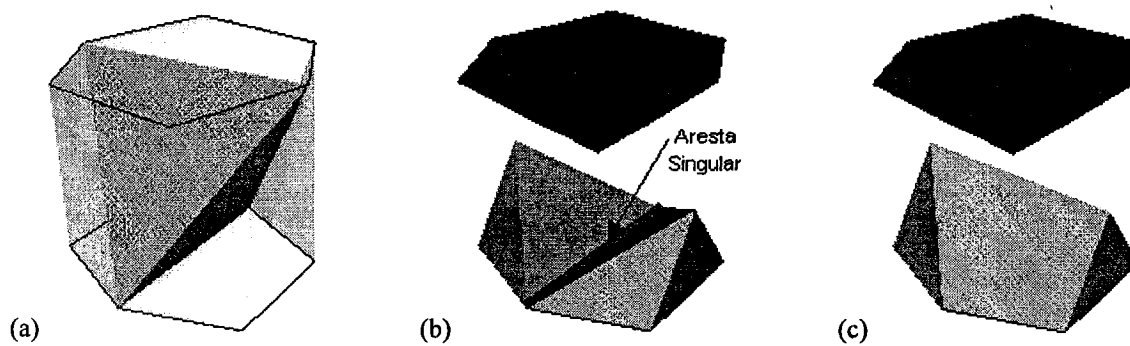


Figura 11 – Caso especial com $\beta = 0$, e tetraedros reversos.
 (a) Singularidade gerada por arrancar um tetraedro reverso,
 (b) Tetraedro Reverso duplicado para tampar as singularidades.

A seguir será explicado como um processo de subdivisão de tetraedros pode ser usado para eliminar estas singularidades.

4.3.6 Subdivisão de tetraedros

Para satisfazer o critério de re-amostragem, isto é, que os contornos originais possam ser recuperados interceptando o modelo com os planos de corte originais, devemos eliminar os tetraedros externos (tetraedros dos tipos externo e reverso externo), o problema é que esta eliminação pode gerar singularidades. Por exemplo, a eliminação de um tetraedro reverso externo (com uma aresta externa) (Figura 12a), gera uma aresta singular no interior do contorno que contém este tetraedro (Figura 12b).

Por outro lado, a fim de garantir o segundo critério de reconstrução, ou seja, garantir que o modelo reconstruído seja sempre uma 3-variedade LP, os tetraedros cuja eliminação dá origem a singulares devem ser mantidos. Uma maneira de fazer com que a presença destes tetraedros não viole o primeiro critério de reconstrução, é introduzir um ponto na aresta externa, subdividir os tetraedros que incidem nela (Figura 12c) e transladar este ponto para uma posição entre os contornos (Figura 12d).

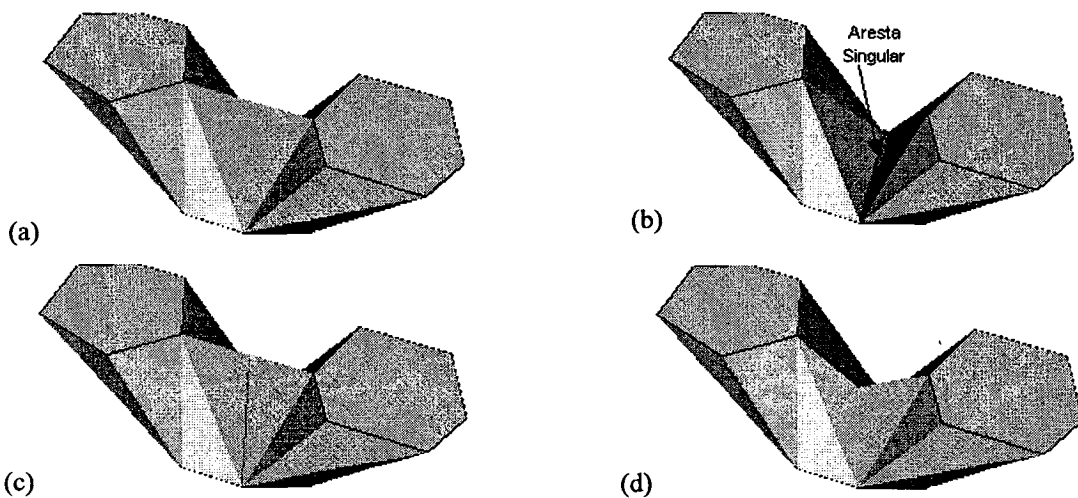


Figura 12 – Aparição de singularidades e sua eliminação com subdivisão de tetraedros.

Exemplo de tetraedro reverso externo,

Eliminação do tetraedro causando uma singularidade em aresta,

(c) Subdivisão de tetraedro, (d) Traslado dos novos pontos da subdivisão.

A idéia principal do processo de subdivisão de tetraedros é:

1. Eliminar os tetraedros externos que não geram singularidades.
2. Subdividir aqueles tetraedros externos que não devem ser eliminados no caso 1.
3. Transladar os novos pontos inseridos na subdivisão até uma posição intermediária.

Estas subdivisões de tetraedros podem ser feitas da seguinte maneira:

Seja t é um tetraedro com uma face contida em um dos planos adjacentes (tetraedro do tipo 1):

- Se t possui apenas uma aresta externa, acrescenta-se um vértice no ponto médio desta aresta e divide-se t em dois novos tetraedros (Figura 13a).
- Se t possui duas arestas externas, acrescenta-se um vértice no ponto médio de cada uma delas e divide-se t em três novos tetraedros (Figura 13b).
- Se t possui três arestas externas, acrescenta-se um vértice no ponto médio de cada uma delas e divide-se t em quatro novos tetraedros (Figura 13c).

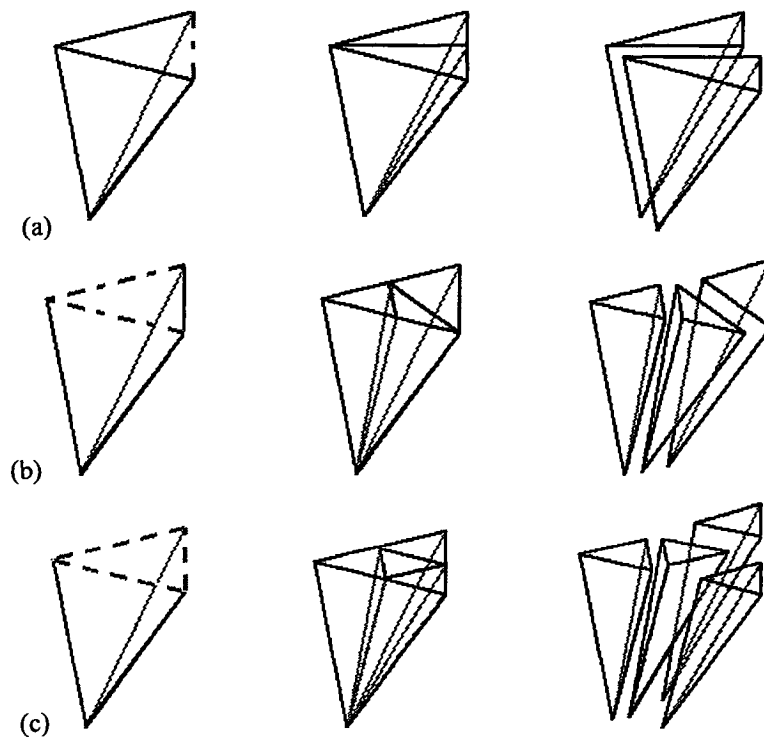


Figura 13 – Subdivisão de tetraedros do tipo 1.

(a) Com uma aresta externa, (b) Com duas arestas externas, (c) Com três arestas externas.

Seja t é um tetraedro com uma aresta em cada um dos planos adjacentes (tetraedro do tipo 2):

- Se t possui apenas uma aresta externa, acrescente um vértice no ponto médio desta aresta e divida t em dois novos tetraedros (Figura 14a).
- Se t possui duas arestas externas, acrescente um vértice no ponto médio de cada uma delas e divida t em quatro novos tetraedros (Figura 14b).

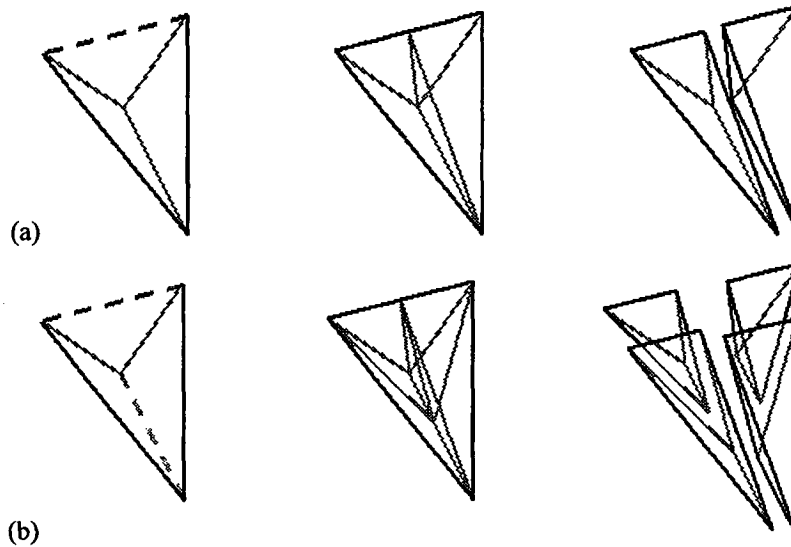


Figura 14 – Subdivisão de tetraedros do tipo 2.

(a) Com uma aresta externa , (b) Com duas arestas externas.

Note que a subdivisão de tetraedros respeita os contornos. Por exemplo, se um tetraedro a ser subdividido tem uma aresta de contorno, a face que contém esta aresta é mantida intacta, ou seja, somente as faces com arestas externas são subdivididas.

Desta forma, o modelo gerado pela aplicação do processo de subdivisão de tetraedros satisfaz os dois critérios de reconstrução: re-amostragem e 3-variedade LP. A figura 15a apresenta um conjunto exemplo de contornos em seções planares adjacentes e as figuras 15b, 15c e 15d apresentam os sub-sólidos, com β igual 0, 1, e 3 respectivamente, depois do processo de subdivisão de tetraedros.

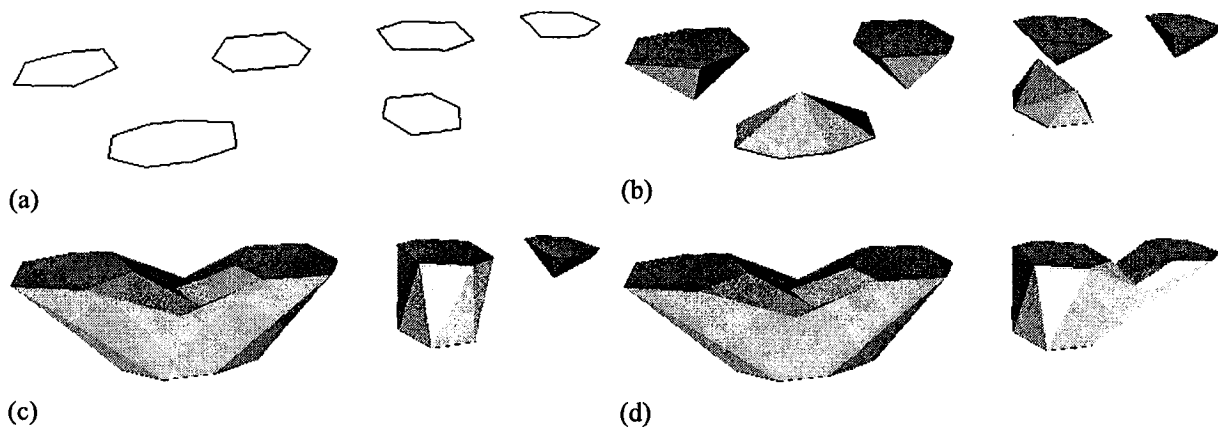


Figura 15 – Exemplos de sub-sólidos.

- (a) Conjunto de contornos de exemplo , (b) Sub-sólido com $\beta = 0$,
 (c) Sub-sólido com $\beta = 1$, (d) Sub-sólido com $\beta = 3$.

Com a subdivisão de tetraedros, é finalizada a etapa do processo de reconstrução pelo algoritmo β -Conexão, dando origem a um sub-sólido.

Todos os sub-sólidos criados nesta etapa serão unidos pelo último processo, união dos sub-sólidos, do *pipeline* de reconstrução.

4.4 União de Sub-Sólidos

A união dos sub-sólidos é o último processo do *pipeline* de reconstrução. Neste processo, todos os sub-sólidos criados pelo algoritmo de reconstrução β -Conexão devem ser unidos para formar um sólido final. Já que o algoritmo de reconstrução é executado entre cada dois níveis consecutivos, dois sub-sólidos consecutivos possuem um conjunto de regiões em uma seção planar em comum. Por outro lado, tem-se conhecimento, pelo algoritmo de Boissonat [Boi88] para pontos distribuídos em seções planares, que as triangulações de Delaunay 2D dos planos estão contidas na triangulação de Delaunay 3D também destes planos. A partir destas regiões em comum com triangulações idênticas, os sub-sólidos criados pelo algoritmo β -Conexão podem ser unidos com facilidade.

A figura 16a apresenta 3 níveis consecutivos de seções planares. As figuras 16b, 16c e 16d, apresentam os sólidos finais com valores de β iguais a 0, 1 e 3 respectivamente, depois do processo de união de sub-sólidos.

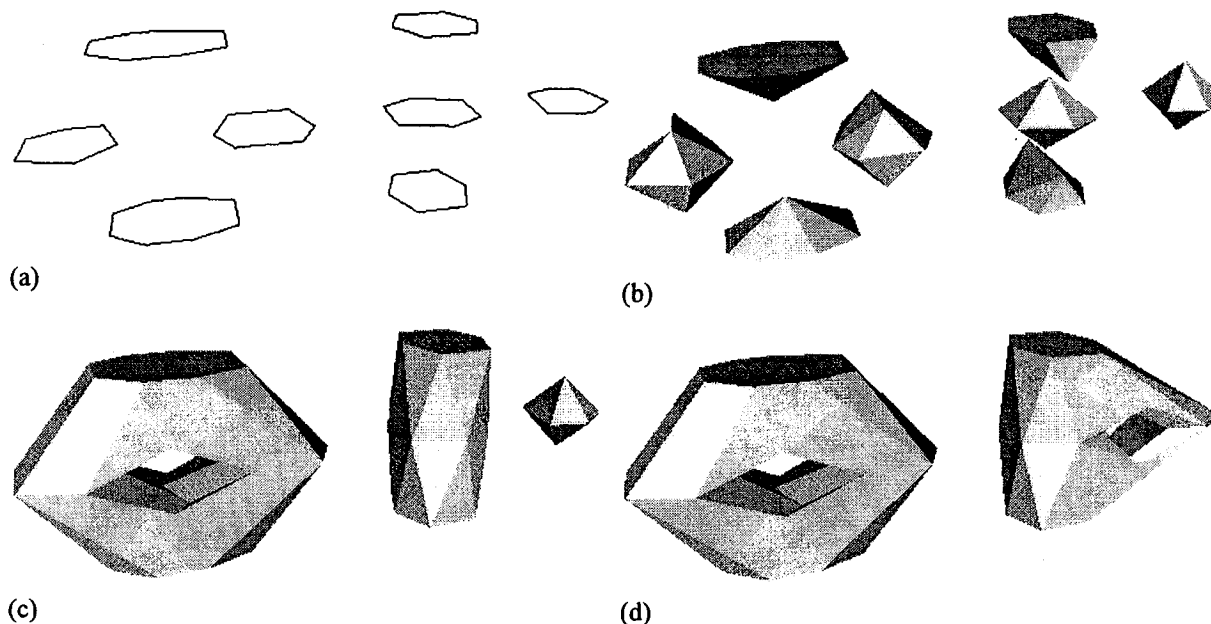


Figura 16 – Exemplos de sólidos finais.

- (a) Conjunto de seções planares em três seções planares,
 (b) Sólido final com $\beta = 0$, (c) Sólido final com $\beta = 1$, (d) Sólido final com $\beta = 3$.

5 Resultados

Durante os testes de reconstrução observou-se um fato interessante. Quando duas regiões em planos adjacentes estão bem posicionadas (figura 17a), aparecem na triangulação 3D um ou mais tetraedros reversos (figura 17b e 17c). Como consequência, a distância segundo o grafo associado à triangulação 3D é de 1 (figura 17d). Se nós afastamos ligeiramente estas duas regiões, aparece uma barreira de tetraedros do tipo redundante (figura 17e), aumentando a distância entre elas, como pode ser observado a partir do grafo associado na figura 17f. Se nós afastamos ainda mais estas duas regiões, a barreira de tetraedros redundantes é maior (figura 17g), e como consequência a distância entre estas duas regiões aumenta ainda mais (figura 17h). Desta forma quanto maior a "distância" entre as regiões, maior deverá ser o valor do parâmetro β para que estas possam ser conectadas.

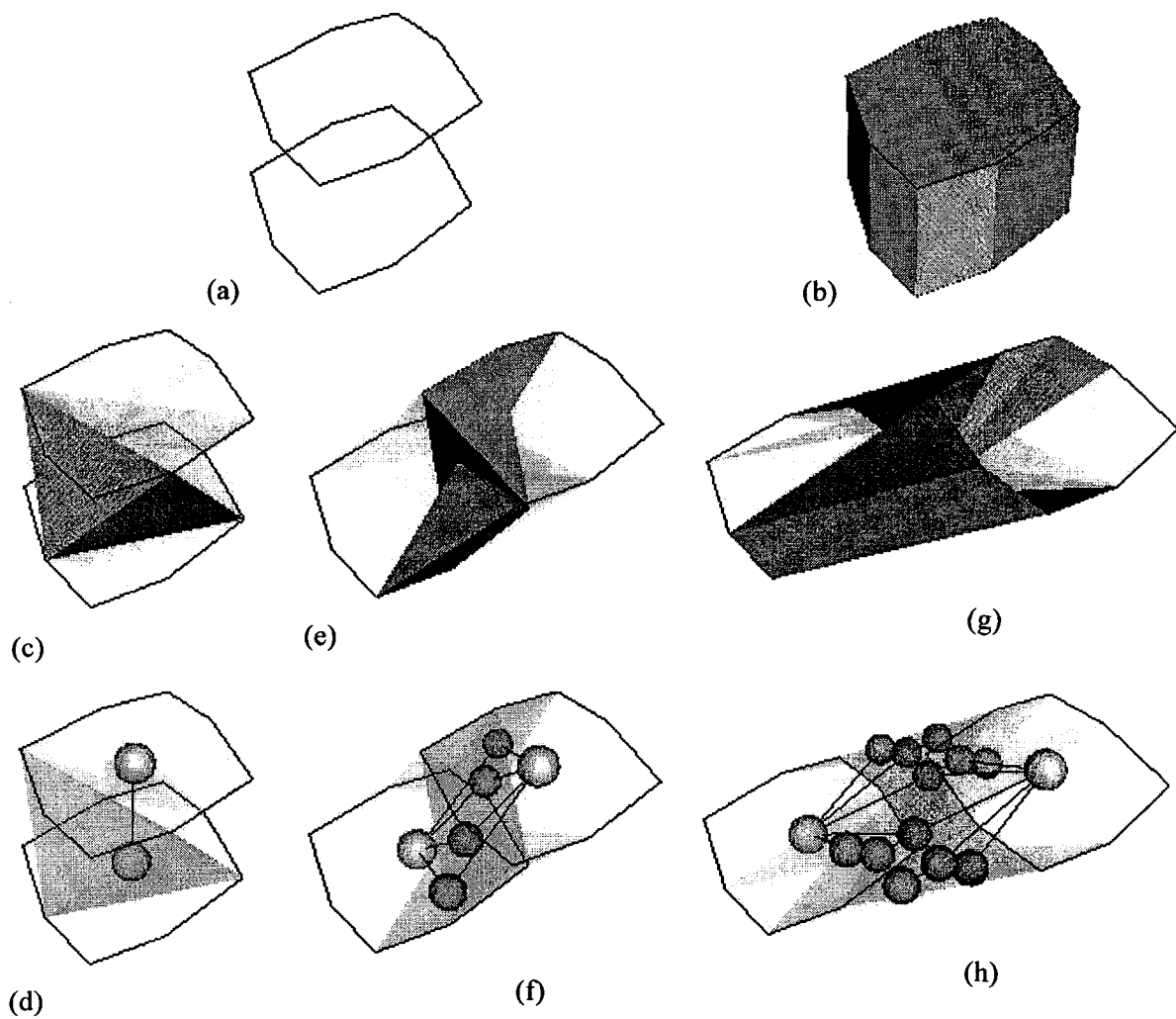


Figura 17 – Fato interessante acontecido durante o cálculo de componentes conexas.

(a) Contornos bem posicionados, (b) Triangulação 3D,

(c) Aparição de tetraedro reverso em contornos bem posicionados, (d) Grafo associado à figura a,

(e) Aparição de uma barreira de tetraedros redundantes em contornos ligeiramente afastados,

(f) Grafo associado à figura e, (g) Barreira maior de tetraedros reversos em contornos mais afastados, (h) Grafo associado à figura g.

O exemplo da figura 5.3 apresenta o comportamento do pipeline de reconstrução com um mesmo conjunto de seções planares e diferentes valores para o parâmetro β . A figura 18a apresenta um conjunto de 6 seções planares. As figuras 18b, 18c, 18d, 18e e 18f apresentam sólidos finais com valores de β iguais a 0, 1, 2, 3, e 4 respectivamente.

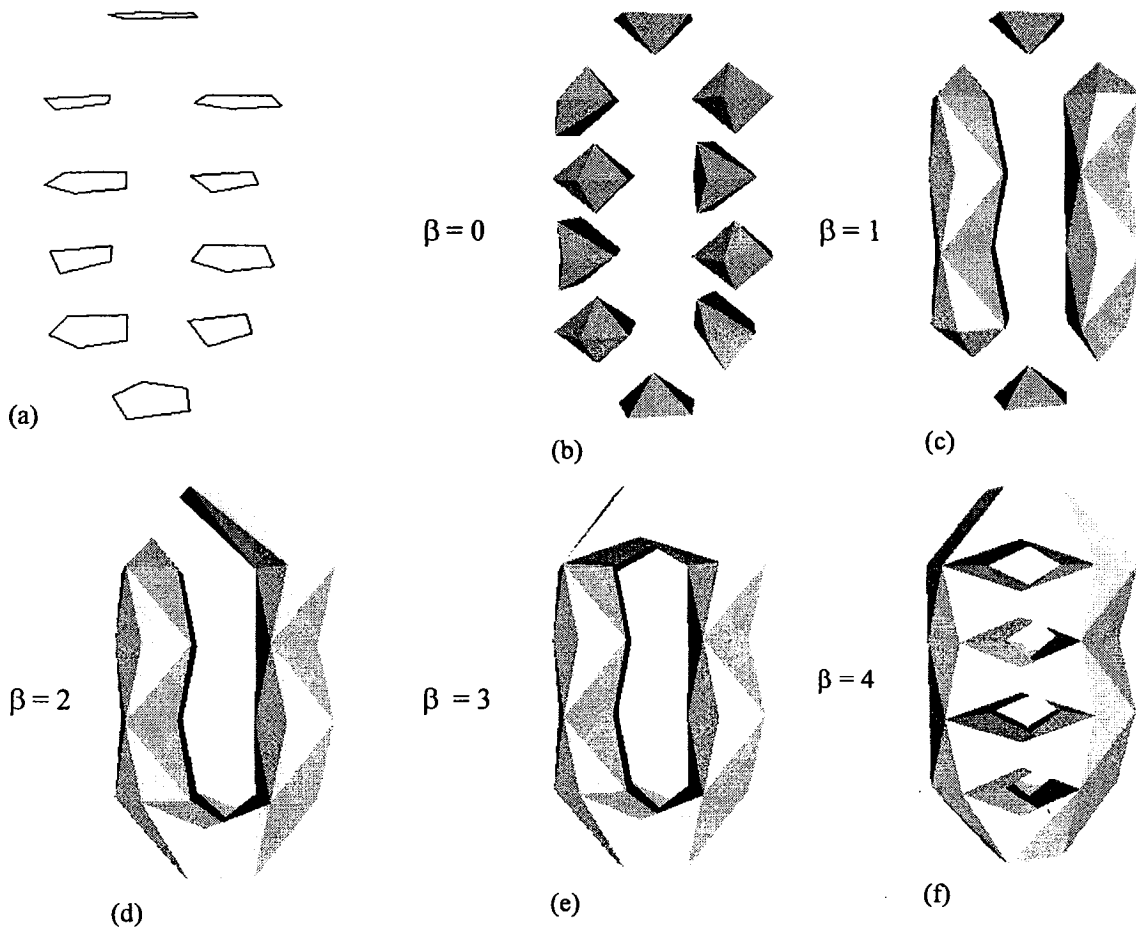


Figura 18 – Comportamento do algoritmo β -Conexão com o mesmo conjunto de seções planares e diferentes valores para o parâmetro β .

(a) Conjunto de seções planares de exemplo, (b) Sólido final com $\beta = 0$, (c) Sólido final com $\beta = 1$,
 (d) Sólido final com $\beta = 2$, (e) Sólido final com $\beta = 3$, (f) Sólido final com $\beta = 4$,

A figura 19 apresenta a importância da flexibilidade na escolha das componentes conexas. A figura 19a apresenta um conjunto de regiões contidas em uma seqüência de seções paralelas. Pelo fato de que o processo de reconstrução é restrito a duas seções planares adjacentes, é possível especificar diferentes valores para β entre cada duas seções planares adjacentes. As figuras 19b, 19c e 19d apresentam diferentes modelos que podem ser obtidos escolhendo distintos valores para β entre cada par de seções planares. Note que a topologia dos modelos finais é completamente diferente em cada caso. Este fato faz o algoritmo β -Conexão uma poderosa ferramenta para modelar sólidos.

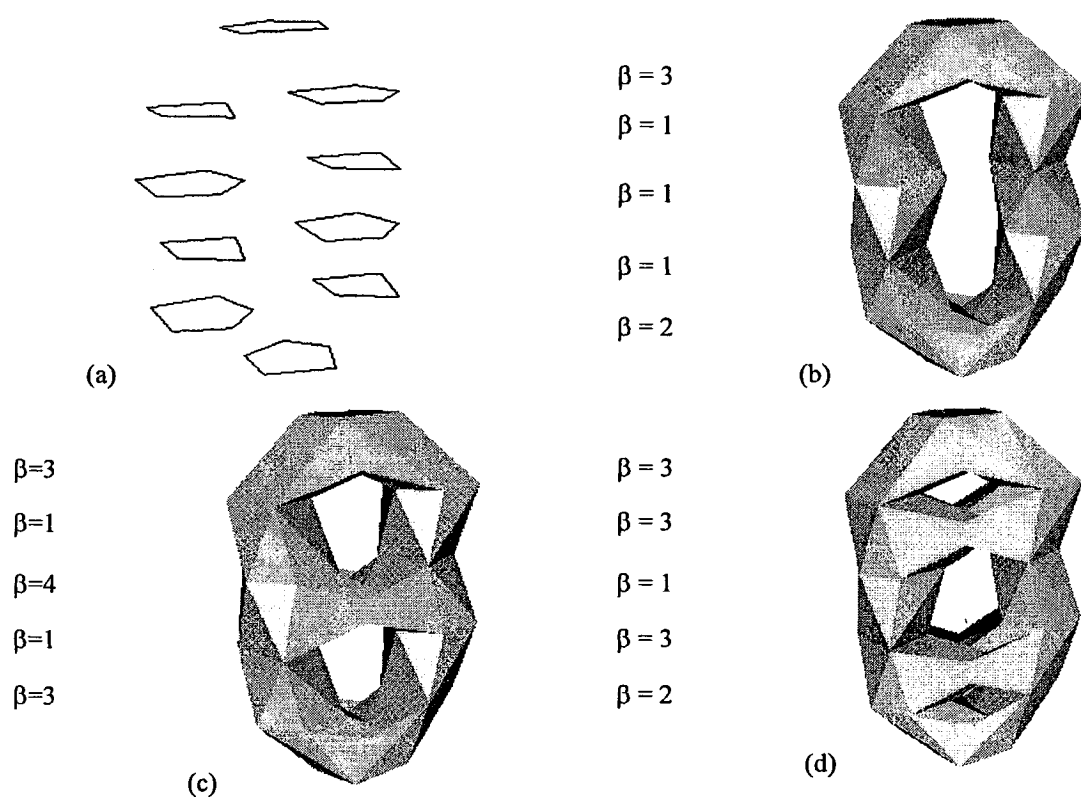


Figura 19 – Modelos finais gerados a partir de um mesmo conjunto de seções planares paralelas e diferentes valores de β entre seções planares.

(a) Seções planares de exemplo, (b, c, d) Figuras geradas com diferentes valores de β entre cada par de seções planares.

6 Conclusões e trabalhos futuros

Este trabalho apresentou todo um processo de reconstrução tridimensional a partir de seções planares. O algoritmo de reconstrução utilizado neste processo é um algoritmo novo denominado β -Conexão. A nova heurística utilizada pelo algoritmo β -Conexão está baseada em um valor inteiro positivo β , que especifica uma distância mínima em tetraedros como condição para definir as conexões entre um conjunto de seções planares. Esta técnica oferece mais flexibilidade na escolha das componentes conexas, gerando uma família de objetos a partir de um mesmo conjunto de seções planares, fato dificilmente obtido por outros algoritmos da literatura. Esta técnica também resolve os problemas de ramificação e geração de malha de forma satisfatória.

O algoritmo foi implementado na linguagem C++ fazendo uso do VTK (Visualization Toolkit) [Sch98], [Kit01] para a visualização. Utilizou-se também a estrutura de dados volumétrica *singular Half-Edge* de dados implementada por Biscaro [Bis00].

Este processo de reconstrução tridimensional começa com a entrada de um conjunto de contornos distribuídos em uma seqüência de seções planares. Em geral, muitos algoritmos de reconstrução, citados na literatura, estão limitados por esta etapa prévia de segmentação para obter os contornos. Seria muito interessante, em um trabalho futuro, incluir o processo de segmentação dentro do processo da reconstrução. Desta forma a reconstrução poderia ser realizada diretamente a partir de uma seqüência de imagens obtidas, por exemplo, por um aparelho de ressonância magnética.

É muito importante para processos de simulações numéricas de fenômenos físicos que o sólido reconstruído por esta nova técnica seja uma malha livre de singularidades. No entanto, em um trabalho futuro, seria válido acrescentar um processo para melhorar a qualidade da malha pelo nosso algoritmo.

O casamento da visualização volumétrica com a reconstrução 3D tem sido fonte de interesse em muitos grupos de pesquisa, sendo essa junção, mais um trabalho futuro apontado por nós.

Referências Bibliográficas

- [Ame97] Amneta, N.; Bern, M.; Kamvysseis, M. – “A new Voronoi-Based Surface Reconstruction Algorithm”, artigo disponível em: <http://web.mit.edu/manoli/crust/www/crust.html>, visitado em agosto de 2001.
- [Baj96] Bajaj, C.L.; Coyle, E.J.; Lin, K. – “Arbitrary topology shape reconstruction from planar cross sections”, *Graphical Models and Image Processing*, 58 (6), pp. 524-543 (1996).
- [Baj99] Bajaj, C.L.; Coyle, E.J.; Lin, K. – “Tetrahedral meshes from planar cross-sections”, *Comput. Methods Appl. Mech Engrg.*, 179, 31-52 (1999).
- [Bis00] Biscaro, H.H. – “Estrutura Half-Face”, Dissertação de mestrado, Instituto de Ciências Matemáticas e de Computação - Universidade de São Paulo, (2000).
- [Boi88] Boissonnat, J-D. – “Shape Reconstruction from Planar Cross Sections”, *Computer Vision, Graphics and Image Processing*, (44), 1-29 (1988).
- [Cor90] Cormen, T.H.; Leiserson, C.E.; Rivest, R.L. – “Introduction to Algorithms”, The MIT Electrical Engineering and computer Science Series, McGraw-Hill Edition (1990).
- [Del94] Delingette, H. – “Simplex Meshes: A General Representation for 3D Shape Reconstruction”, Technical Report 2214, INRIA (1994).
- [Dey00] Dey, T. K.; Giesen, J.; Hudson, J.; – “Delaunay based shape reconstruction”, Ohio State University, Columbus USA, (2000).
- [Eko91] Ekoule, A.B.; Peyrin, F.C.; Odet, C. L. – “A Triangulation Algorithm from Arbitrary Shaped Multiple Planar Contours”, *ACM Trans. On Graph.*, 10(2), pp. 182-199 (1991).
- [For94] Fortune, S. – “Voronoi Diagrams and Delaunay Triangulation”, in D.-Z. Du, F K. Hwang (eds.), *Computing in Euclidean Geometry*, World Scientific Pub. Co., 193-233 (1994).
- [Gei93] Geiger, B. – “Three Dimensional Modeling of Human Organs and its Application to Diagnosis and Surgical Planning”, Technical Report 2105, INRIA Sophia-Antipolis France (1993).
- [Jon94] Jones, M. W.; Chen, M. – “A New Approach to the Construction of Surfaces from Contour Data”, *Proc. EUROGRAPHICS'94*, 13(3), pp. 75-84 (1994).
- [Kau98] Kaufman, A. – “Advances in Volume Visualization”, SIGGRAPH'98, Course Notes no. 24, Orlando (1998).
- [Kep75] Keppel, E. – “Approximating complex surface by triangulation of contour lines”, *IBM. J. Res. Dev.* 19, pp. 2-11 (1975).
- [Kit01] Kitware, “The Visualization Toolkit”, Object-Oriented Approach to 3D Graphics, disponível em: <http://public.kitware.com>, visitado em agosto de 2001.
- [Lor87] W.E. Lorensen; H.E. Cline – “Marching cubes: A High Resolution 3D Surface Construction Algorithms”, *ACM SIG. Comp.Graph.*, 21,163-169 (1987).
- [Mey92] Meyers, D.; Skinner, S.; Sloan, K. – “Surface from Contours”, *ACM Trans. On Graphics*, 11(3), pp. 228-258 (1992).

- [Non01] Nonato, L.G.; Minghim, R.; Oliveira, M.C.F.; Tavares, G. – "A Novel Approach for Delaunay 3D Reconstruction with a Comparative Analysis in the Light of Applications", *Computer Graphics Forum*, 20(2), 161-174, (2001).
- [Non98] Nonato, L.G. – "Reconstrução Volumétrica de Objetos definidos por seções Bidimensionais", Tese de Doutorado, Departamento de Matemática, Pontifícia Universidade Católica, Rio de Janeiro, Brasil, (1998).
- [Sch98] Schröder, W.J.; Martin, K.; Lorensen, B. – "The Visualization Toolkit", *Object-Oriented Approach to 3D Graphics*, 2nd Edition; Prentice-Hall, (1998).
- [Shi91] Shinagawa, Y.; Kunii, T.L. – "The homotopy model: a generalized model for smooth surface generation from cross sectional data", *The Visual Computer*, 7, pp. 72-86 (1991).
- [Sin98] Singh, A.; Goldof, D.; Terzopoulos, D. (ed.) – "Deformable Models in Medical Image Analysis", IEEE CS Press (1998).

NOTAS DO ICMC

SÉRIE COMPUTAÇÃO

- 061/2001 OLIVEIRA JR., O N; MARTINS, R T; RINO, L H M ; NUNES, M G V – O uso de interlínguas para comunicação via internet: o projeto UNL/Brasil.
- 060/2001 SILVA, E Q ; MOREIRA, D A – Use of software agents to the management of distance education courses over the internet.
- 059/2001 OLIVEIRA, M.C.F.; LEVKOWITZ, H. – Visual data exploration and mining: a survey.
- 058/2001 SOARES, M D.; FORTES, R P M; MOREIRA, D A – Version-web : a tool for helping web pages version control.
- 057/2001 LIANG, Z; MACAU, E E N; OMAR, N - Scene Segmentation of the Chaotic Oscillator Network.
- 056/2000 BATISTA, G E A P A; CARVALHO, A C P L F; MONARD, M C – Applying one-sided selection to unbalanced datasets.
- 055/2000 NONATO, L G; MINGHIM, R.; OLIVEIRA, M C F; TAVARES, G. – A novel approach for delaunay 3D reconstruction with a comparative analysis in the light of applications.
- 054/2000 MORSELLI JR., J C M; SANTANA, R H C; SANTANA, M J; ULSON, R S – An approach for dynamic swapping of distributed simulation synchronisation protocols.
- 053/2000 SPOLON, R.; SANTANA, M J; SANTANA, R H C – A methodology for performance evaluation of optimistic distributed simulation synchronisation mechanisms.
- 052/2000 BRANCO, K R L J C; SANTANA, M J S; SANTANA, R H C; CALÔNEGO JR, N – A parallel programming supporting tool.