

Geração da Curva Interseção de Superfícies Trianguladas

André Luiz Firmino Alves

Rosana Marques da Silva (Orientadora)

*Universidade Federal de Campina Grande – UFCG
Departamento de Matemática e Estatística - DME
andre.alfa@gmail.com, rosana@dme.ufcg.edu.br*

Resumo

O presente trabalho apresenta uma implementação do algoritmo para a obtenção da curva de interseção de duas superfícies trianguladas, apresentado por Lo e Wang [3]. O algoritmo determina a curva de interseção entre duas malhas trianguladas criando novos vértices em cada uma das superfícies, gerando uma poligonal pertencente a ambas as superfícies, as quais devem ser reconstruídas localmente a partir desses novos vértices, recortando as faces excedentes. O problema de interseção entre duas superfícies trianguladas é tratado como um problema de interseção entre dois triângulos e dividido nas seguintes etapas: Determinação do ponto de interseção entre o plano que contém uma das faces e uma aresta da outra face e a verificação da pertinência do ponto de interseção determinado a ambos os triângulos. A eficiência do algoritmo é melhorada pelo uso de caixas limitantes.

1. Introdução

A operação de interseção entre superfícies modeladas por malhas poligonais, para a representação de estruturas arbitrárias e complexas, requer algoritmos de alta performance, visto que as interseções, geralmente, ocorrem em pequenas regiões das superfícies. Assim, é desejável a definição de um processo preciso e eficiente para determinar as curvas de interseções.

Através da curva de interseção de duas superfícies poliédricas pode-se construir uma única malha a partir da junção das malhas que modelam cada uma das superfícies envolvidas, onde a malha resultante deve ser formada pelas malhas iniciais modificadas localmente nas regiões onde ocorrem as interseções.

As operações de interseção e união permitem a construção de modelos geométricos de superfícies complexas, a partir de modelos de superfícies mais simples, construídos individualmente [2,3].

Este trabalho apresenta uma implementação do algoritmo de interseção de superfícies, representadas por malhas trianguladas, apresentado por Lo e Wang [3]. O algoritmo determina a curva de interseção entre duas malhas trianguladas criando novos vértices, em cada uma das superfícies, gerando uma poligonal pertencente a ambas as superfícies, as quais devem ser reconstruídas, localmente, a partir desses novos vértices, recortando as faces excedentes.

O cálculo da curva de interseção em superfícies trianguladas, apresentado neste trabalho, foi utilizado no desenvolvimento de um protótipo de modelador de cenários geológicos tridimensionais de reservatórios petrolíferos [1]. Os volumes conexos que formam um reservatório têm formas arbitrárias e complexas, e podem ser obtidos seguindo o método da superposição das unidades geológicas que compõem o reservatório, através das operações de interseção entre essas unidades modeladas individualmente como superfícies trianguladas [5].

2. Superfícies Trianguladas

A utilização de faces triangulares na poligonização de superfícies apresenta algumas vantagens, entre elas: Planaridade, que facilita cálculos entre superfícies, tais como as interseções, que podem ser vistas como interseções entre retas (arestas) e planos (triângulos); e Sistema de coordenadas, visto que um triângulo tem associado a ele, naturalmente, um sistema de coordenadas lineares (coordenadas baricênticas) [4].

3. Interseção entre Faces Triangulares

A interseção entre duas superfícies trianguladas pode ser tratada como um problema de interseção entre triângulos (Figura 1), podendo ser dividido nas seguintes etapas:

- 1- Determinação do ponto de interseção entre o plano que contém uma das faces e uma aresta da outra face;
- 2- Verificação se o ponto de interseção determinado pertence a ambos os triângulos.

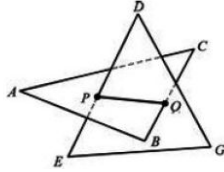


Figura 1. Interseção entre dois triângulos [3]

3.1. Interseção entre os triângulos ABC e DEG.

O problema de interseção entre uma face triangular ABC com outra face triangular DEG, como já mencionado, pode ser tratado como um problema de interseção entre planos e retas. Mais precisamente deve-se verificar a ocorrência de interseção das arestas DE, EG e GD com a face triangular ABC (Figura 1). Para verificar se a aresta DE intercepta o triângulo ABC as seguintes condições devem ser satisfeitas [3]:

1. Os pontos D e E pertencem a lados opostos do plano que contém o triângulo ABC, ou seja,

$$(\overrightarrow{AD} \cdot \vec{N})(\overrightarrow{AE} \cdot \vec{N}) < 0, \quad (1)$$

onde $\vec{N} = \overrightarrow{AB} \times \overrightarrow{AC}$ é o vetor normal ao plano que contém a face ABC.

2. O ponto P de interseção está contido no triângulo ABC.

Dado que a condição anterior é satisfeita, tem-se que a aresta DE corta o plano que contém o triângulo ABC. Então basta verificar se o ponto P de interseção da aresta DE com o plano está contido no triângulo ABC. Observe que além de P pertencer à aresta (reta) DE, P também pertence ao plano que contém o triângulo ABC, logo P satisfaz o sistema de equações:

$$\begin{cases} P = tE + (1-t)D \\ (\overrightarrow{AP} \cdot \vec{N}) = 0. \end{cases} \quad (2)$$

Assim o parâmetro t é dado por

$$t = \frac{\overrightarrow{AD} \cdot \vec{N}}{\overrightarrow{AD} \cdot \vec{N} - \overrightarrow{AE} \cdot \vec{N}}. \quad (3)$$

3.2. Verificando se o ponto P pertence ao triângulo ABC

O ponto P estará dentro do triângulo ABC se

$$(\overrightarrow{AB} \times \overrightarrow{AP}) \cdot \vec{N} > 0, (\overrightarrow{BC} \times \overrightarrow{BP}) \cdot \vec{N} > 0, (\overrightarrow{CA} \times \overrightarrow{CP}) \cdot \vec{N} > 0. \quad (4)$$

De forma análoga, pode-se verificar se as arestas EG e GD interceptam o triângulo ABC (Figura 1), para o cálculo do segundo ponto da interseção. Se as outras arestas do triângulo DEG não interceptam o triângulo ABC, então o segundo ponto de interseção pode ser determinado calculando a interseção das arestas do triângulo ABC (AB, BC e CA) com o triângulo DEG. Seja Q o segundo ponto da interseção, logo o segmento PQ compõe a curva de interseção entre as duas faces triangulares ABC e DEG.

A Figura 2 mostra os resultados da implementação realizada para os cálculos da verificação de interseção entre faces triangulares (Figura 2.a) a fim de determinar a curva de interseção entre os dois triângulos: determinação dos pontos de interseção (Figura 2.b) e do segmento da curva de interseção (Figura 2.c).

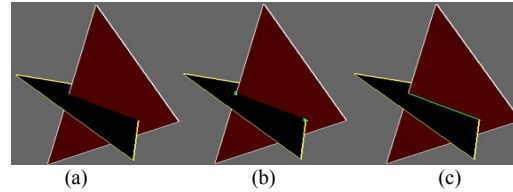


Figura 2. Interseção entre dois triângulos [1].

3.3. Casos previstos nas interseções dos triângulos

Na interseção entre dois triângulos, existem pelo menos quatro casos possíveis (Figura 3) [3]:

1. O ponto de interseção P ocorre dentro do triângulo ABC e pertence a uma das arestas do triângulo DEG (Figura 2 e 3.a);
2. Os pontos de interseção P e Q pertencem às arestas do triângulo ABC (Figura 3.b);
3. O ponto de interseção P coincide com um vértice do triângulo ABC (Figura 3.c);
4. A interseção não é mais um ponto, e sim um domínio planar, ou seja, o produto vetorial entre os vetores normais dos triângulos ABC e DEG é nulo, pois os triângulos estão no mesmo plano (Figura 3.d).

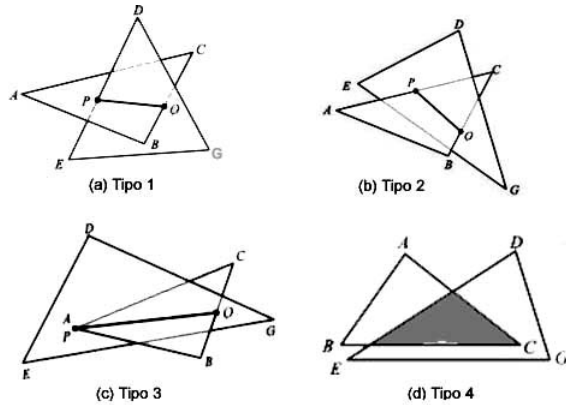


Figura 3. Possibilidades de interseção entre dois triângulos

4. Determinação da Curva de Interseção

A interseção entre duas superfícies normalmente ocorre apenas em algumas de suas regiões, necessitando-se, portanto, em termos de eficiência computacional, de um mecanismo para filtrar as faces onde não há possibilidade de ocorrência de interseção (Figura 4). Lo e Wang [3] utilizam a técnica da Caixa Limitante para reduzir o tempo de pesquisa das faces trianguladas, filtrando as faces onde não ocorrem interseções nas superfícies. A utilização de Caixas Limitantes reduz drasticamente o tempo de checagem das interseções entre as faces trianguladas, tornando o tempo do processo de interseção quase linear em relação ao número de elementos (faces) envolvidos.

O algoritmo utiliza uma Caixa Limitante, chamada neste trabalho de Caixa Limitante Global, para encapsular os objetos (malhas), ou parte dos objetos envolvidos no processo, e caixas limitantes para cada face triangulada de cada superfície, chamadas de *bounding box*. A Caixa Limitante Global trata-se de uma partição espacial (Figura 5.a) em células de dimensões menores, doravante chamadas de *voxels* (Figura 5.b). As dimensões dos *voxels* podem ser arbitrárias, como também ajustadas dinamicamente de acordo com o tamanho dos objetos. Lo e Wang [3] utilizam *voxels* de mesma dimensão e cada *voxel* poderá conter mais de um objeto (face triangular).

5. Implementação do algoritmo

Esta seção apresenta uma implementação do algoritmo apresentado por Lo e Wang [3], com algum detalhamento.

Dadas duas superfícies trianguladas S_1 e S_2 (Figura 5), segue os passos do algoritmo implementado:

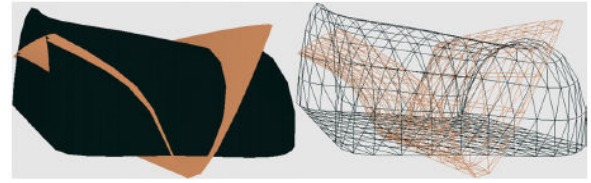


Figura 4. Interseção entre superfícies poliédricas

1. Definir a Caixa Limitante Global (CLG) (Figura 5.a), os *Bounding Box's* das faces triangulares (Figura 6) e determinar as partições da CLG, ou seja, os *voxels* V_i (Figura 5.b);

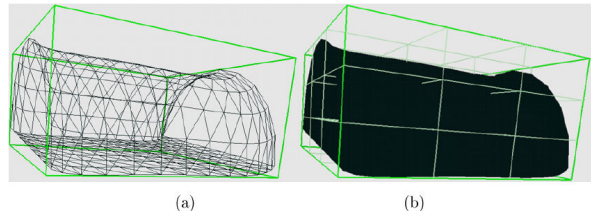


Figura 5. (a) CLG de uma superfície (b) Os Voxels.

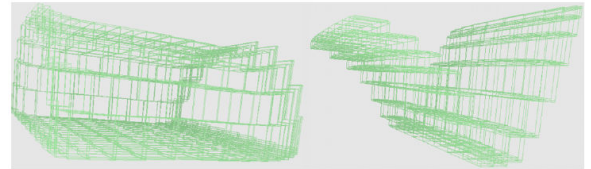


Figura 6. *Bounding box's* das faces trianguladas das superfícies S_1 e S_2 .

2. Identificar as interseções entre os *voxels* (Figura 5.b) e os *bounding box's* (Figura 6) das faces triangulares, ou seja, identificar as faces T_k em S_1 que interceptam os *voxels* V_i , e as faces F_j em S_2 que interceptam os *voxels* V_i (Figura 7.a):

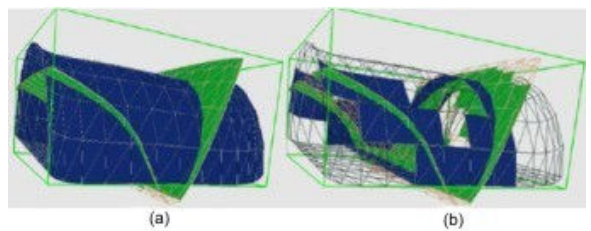


Figura 7. As faces candidatas a interseção.

3. Selecionar faces candidatas (Figura 7.b):
 - (a) Cada *voxels* V_i (Figura 5b) é examinado por vez. V_i é ignorado se o *bounding box* (Figura 6) da face T_k em S_1 ou o *bounding box* da face F_j em S_2 não está em seu interior.
 - (b) Se V_i contém o *bounding box* de uma face T_k em S_1 e o *bounding box* de uma face F_j em S_2 , e os *bounding box* dessas faces se interceptam,

então essas faces são consideradas candidatas a interseção (Figura 7.b e Figura 8).

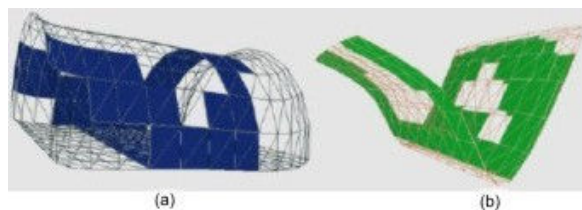


Figura 8. As faces candidatas a interseção em cada uma das superfícies.

4. Determinar a curva de interseção: Para todos os voxels V_i que contém faces candidatas a interseção:
 - (a) Calcular o ponto de interseção entre as faces candidatas de ambas as superfícies
 - (b) Se o ponto de interseção entre as faces candidatas de ambas as superfícies pertencerem as faces, então registra os pontos nas respectivas faces.
 - (c) Criar a curva poligonal a partir dos pontos de interseção (Figura 9).

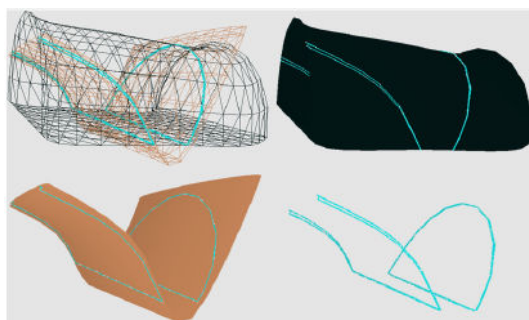


Figura 9. Obtenção da curva de interseção.

A implementação do algoritmo que determina a curva de interseção em malhas trianguladas foi realizada em linguagem C, na plataforma Linux, utilizando funções da biblioteca TDM (Módulo de Dados Topológicos), desenvolvida por Wu [6] para representar e gerenciar o modelo das superfícies, e funções da biblioteca Mesa/OpenGL para auxiliar a visualização do modelo gerado.

5. Considerações Finais

Este trabalho apresentou uma implementação do algoritmo, apresentado por Lo e Wang [3], para obtenção de uma poligonal que representa a curva interseção entre duas superfícies trianguladas. O algoritmo usa caixas limitantes, reduzindo, com isso, significativamente o tempo para identificar as regiões

em que há possibilidade de ocorrência de interseções. Geralmente as interseções entre superfícies ocorrem em pequenas regiões, portanto o procedimento que utiliza as caixas limitantes é bastante eficiente.

Um próximo passo para a obtenção de uma única malha através da operação de interseção entre duas superfícies seria uma reconstrução local na região que contém a curva de interseção das superfícies. Além do mais o algoritmo implementado deve ser adaptado aos casos de interseção de n superfícies ($n > 2$) e superfícies cuja curva interseção contém múltiplos laços (auto-interseções).

6. Agradecimentos

À ANP/PRH-25 pelo apoio financeiro e ao Departamento de Matemática e Estatística/UFCG pelo suporte para a realização deste trabalho.

7. Referências

- [1] ALVES, A. L. F. Geração de Cenários Tridimensionais de Reservatórios Petrolíferos Canalizados. Monografia de Graduação. UFCG/CCT/DME/PRH-25, 2007.
- [2] COELHO, L. C., GATTASS, M. Intersecting and trimming parametric meshes on finite element shells. International Journal for Numerical Methods in Engineering, vol.0(0),1-10, 1999.
- [3] LO, S. H., WANG, W. X. A Fast Robust Algorithm for the Intersection of Triangulated Surfaces. Engineering with Computers, 2004.
- [4] MIRANDA, A. C. O., MARTHA, L. F.. Geração de malhas estruturadas a partir de um espaço paramétrico de triangulações não estruturadas. Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio), Rio de Janeiro, 2004.
- [5] SANTOS, J. A. ARAÚJO, C. E. S. e SILVA, R. M.. Geração de um Modelo Geométrico de Canais Encontrados em Reservatórios Petrolíferos. XVIII Brazilian Symposium on Computer Graphics and Image Processing, SIBGRAPI - WORKSHOP of Undergraduate Work, Natal, RN, 2005.
- [6] WU, S.-T. Non-manifold Data Models: Implementational Issues. In Proceedings of MICAD'92, pg 38-56, Paris, 1992.