

DETECÇÃO DE COLISÕES AUXILIADAS POR BOUNDING BOXES

Aluno: Matheus Felipe Ferreira Maciel

Orientador: Thomas Lewiner

Introdução

Existem diversos métodos de detecção de colisões em sistemas gráficos, sendo um deles as *bounding boxes*. Elas são ferramentas utilizadas para facilitar tal detecção por determinar da melhor maneira possível o espaço interior/exterior da malha em questão. Sendo assim, foi realizado um estudo para obter maneiras de calcular uma *bounding box* de uma estrutura e, finalmente, montado um programa para cálculo das mesmas.

Objetivos

Estudar métodos para geração de *bounding boxes* para malhas de triângulos gerais, de acordo com a orientação de eixos cartesianos padrões (malhas simples) ou eixos orientados (malhas complexas) e desenvolver um programa de computador utilizando C/C++ e OpenGL que aplique tais métodos.

Metodologia

Dado que uma malha de triângulos em um sistema gráfico comum é gerada por pontos em R^3 e que temos todas as informações referentes às coordenadas desses pontos, pode-se determinar uma maneira para a construção da *bounding box* associada a malha conforme a complexidade desejada. Nesse trabalho foram desenvolvidos dois tipos de *bounding boxes* para malhas: *AABB* (*Axis Aligned Bounding Box*) e *BB* (*bounding box*).

Objetos com simetria e pouco irregulares podem ser contidos dentro de *AABBs*, tendo em vista que essas são construídas de acordo com pontos máximos e mínimos definidos pelas coordenadas da malha dada como entrada. Para tanto, basta a análise de tais pontos e a definição dos limites da *bounding box* de acordo com eles. No programa existem três estruturas utilizadas para determinar uma *bounding box* sendo eles:

- Utilizando um ponto identificador (inferior esquerdo conforme os eixos XYZ padrões) e calculando a diferença entre os máximos e mínimos da malha;
- Utilizando as coordenadas do ponto inferior esquerdo frontal e superior direito de trás da malha;
- Utilizando o centro da malha.

A detecção de colisões entre *AABBs* é feita por base de testes dentro da estrutura do programa que analisa a interseção de *bounding boxes* de malhas dadas como valores de entrada. Esses testes são feitos verificando se os pontos guia de uma *bounding box*, determinados pelos métodos definidos anteriormente, estão contidos dentro da outra *bounding box* em questão.

Objetos irregulares precisam de uma maneira mais eficaz de determinação para uma *bounding box*, pois o método mais simples não comporta toda a malha de uma maneira eficiente, tendo em vista que utiliza somente os eixos XYZ para construir os limites do objeto.

A estrutura de dados utilizada para a montagem das *BBs* é mais complexa, tendo em vista que necessitam mais que coordenadas de pontos em XYZ: é necessário definir uma matriz que contenha informações sobre as transformações básicas da matriz, sendo elas a rotação, escalonamento e translação [1]. Tal estrutura é definida dentro do programa obedecendo a regras de construção de classes [2].

Para calcular a bounding box para uma malha, é necessário considerar X como a SVD [3] do espaço de dados. Resolvendo o sistema, temos que os vetores singulares da esquerda são os eixos principais do sistema. Sendo assim, aplicando mínimos quadrados temos a variância do espaço de dados, sendo que podemos então definir a bounding box que melhor contém a malha escolhida.

Conclusões

O estudo teórico permitiu maior conhecimento e envolvimento da álgebra linear com a computação gráfica, principalmente no que diz respeito à construção de estruturas de auxílio como sistemas de colisão.

O desenvolvimento do programa teve um aspecto importante na duração, tendo em vista que a utilização de uma biblioteca matemática é relativamente fácil, pois muitas funções estão inclusas nela, sendo necessário ter o conhecimento teórico para utilizá-las.

Criar e aplicar os conceitos e a mecânica é algo extremamente válido tendo em vista que esse método de colisão é utilizado em diversas plataformas modernas de desenvolvimento, principalmente aquelas ligadas a indústria de jogos.

No que diz respeito a complexidade, o módulo pode ser aprimorado para detectar colisões em tempo real de forma eficiente, com um certo número de modificações. Modelos mais refinados poderiam ser decompostos em partes e ter *bounding boxes* associadas a essas partes, para refinar ainda mais a detecção de colisões.

Referências

- 1 - Maia, J. G. R.; Vidal, C. A. & Cavalcante-Neto, J. B. **Transformation Semantics: An Efficient Approach for Collision Detection**. Disponível em <<http://sibgrapi.sid.inpe.br/col/sid.inpe.br/sibgrapi%4080/2006/08.24.13.51/doc/>>. Acesso em: 18 de agosto 2007.
- 2 - H. M. DEITEL & P. J. DEITEL. **C++ Como Programar**. 3º Ed. Porto Alegre: Bookman, 2001. 1098 p.
- 3 - LEE, Joon Lae. **SVD (Singular Value Decomposition) and Its Applications**, Disponível em www.mathnet.or.kr/real/2006/1/1101_lee.ppt> Acesso em: 20 de outubro 2007.