

Construção de Octrees de Voxels Esparsas

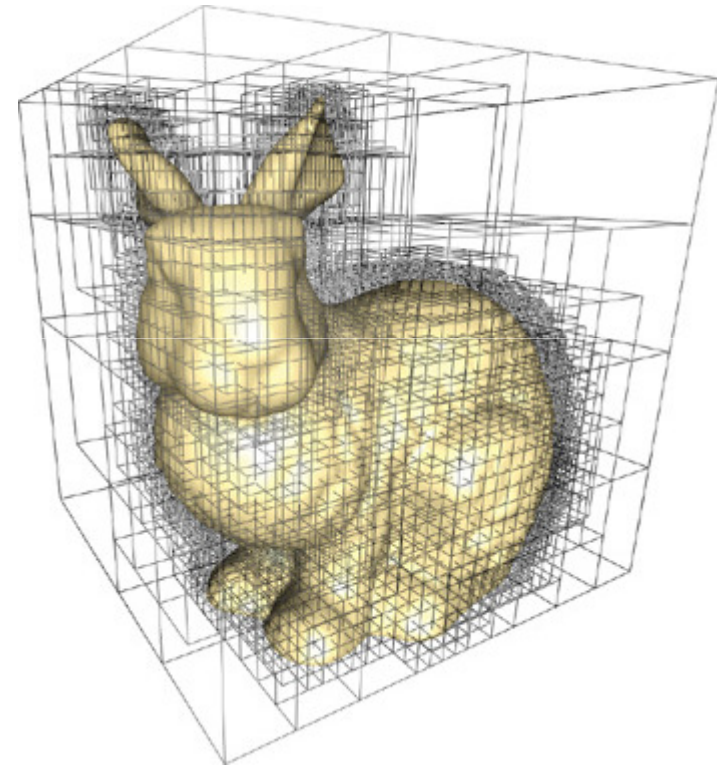
Fabrício Cardoso da Silva

Voxels

- Utilização em visualização científica e efeitos especiais
- Estrutura simples, porém custosa
 - Tempo de computação
 - Consumo de memória
- Geração de imagens com alto nível de detalhes
 - Controle da resolução espacial da cena

Octrees de voxels

- Estruturas hierárquicas em que cada voxel corresponde a um nó folha
- Acesso aos nós folha em $\log_8(n)$ em uma estrutura balanceada
- Constitui uma estrutura regular que favorece o desempenho do lançamento de raios
- Streaming de dados possibilita visualização de modelos *out-of-core*

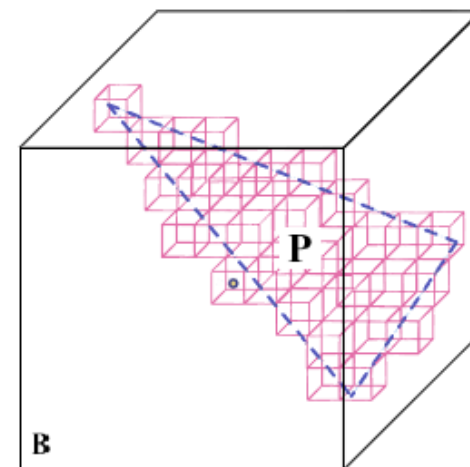
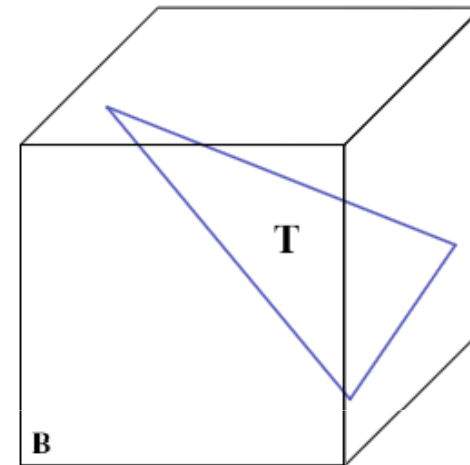


Objetivo

Descrever a implementação na CPU de octrees de voxels esparsas, bem como da etapa de voxelização de malhas de triângulos para construção do volume de dados.

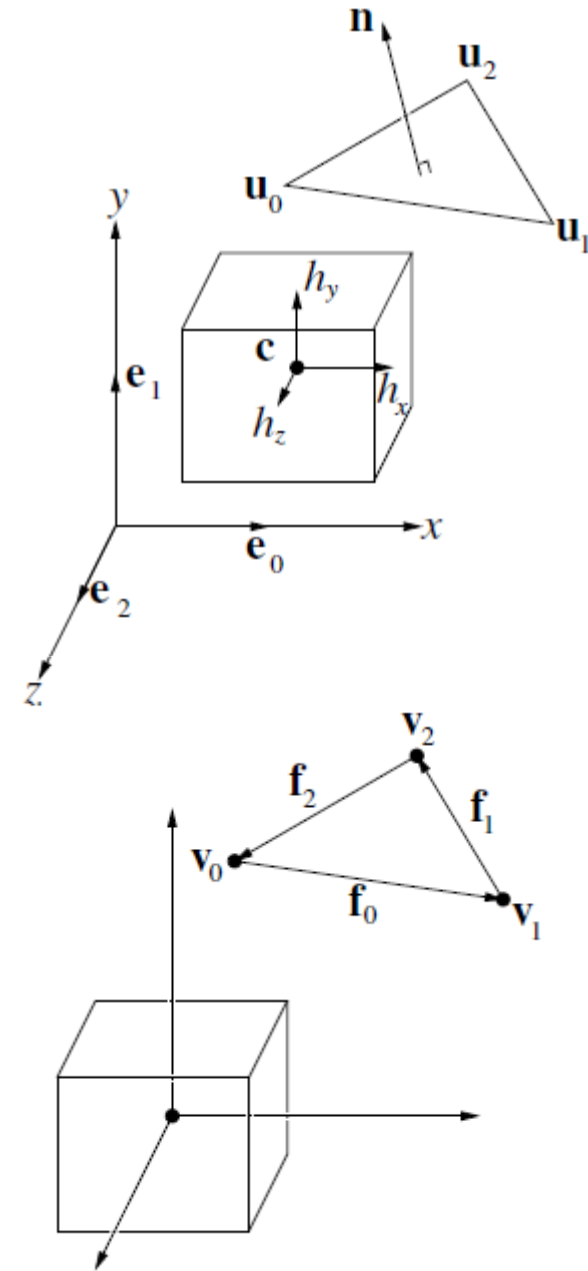
Etapa de voxelização

- Extração de voxels a partir de superfícies de malhas de triângulos
- Informações são armazenadas em uma grade regular
 - Interseção entre célula e geometria determina geração de voxel



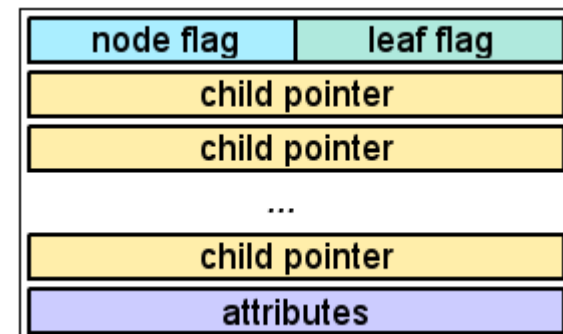
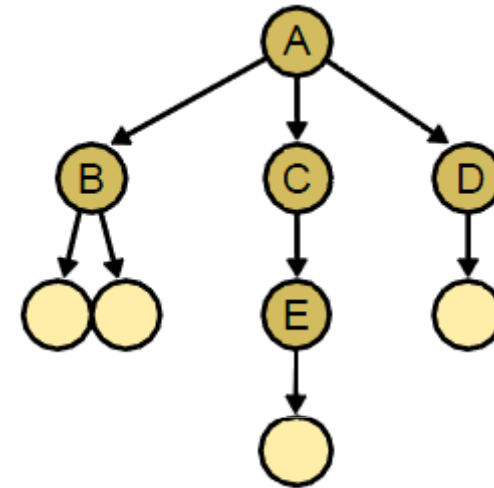
Procedimento

- Criação de uma grade regular
 - Definição da resolução espacial (nível de detalhes)
 - Definição das dimensões da estrutura (ajustada ao tamanho do modelo)
- Determinação dos voxels sólidos
 - Para cada triângulo, é definida sua bounding box e avaliadas quais células do grid a intersectam
 - Definidas as células, são realizados os testes de interseção com os triângulos individualmente



Especificação da octree

- Representação compacta dos nós da árvore
 - Dois conjuntos de flags determinam quais filhos contêm geometria e quais constituem folhas
- Arranjos de ponteiros definem as conexões da estrutura
 - Estruturas diferentes para nós internos e nós folha
- Nós folha constituem a maioria dos nós na octree
 - Redução do armazenamento de ponteiros para nós internos



Método de construção

- Construção bottom-up utilizando algoritmo recursivo
 - Recebe como entrada o arranjo de voxels
- Divisão em blocos de índices
 - Caso bloco seja maior que um voxel, é subdividido em oito sub-blocos
 - Novas chamadas recursivas para cada sub-bloco
 - Geração de nós internos a cada chamada
- Nós folha são gerados quando o bloco atinge o tamanho de um voxel

Método de construção (cont.)

- Verificações são realizadas nos nós internos após retorno das recursões
- Eliminação de redundância
 - Nós que possuem somente filhos vazios
 - Nós cujos filhos possuem a mesma cor
- Ajuste dos níveis de detalhe
 - Atribuição da cor média dos filhos ao nó atual

Resultados

- Trabalho parcialmente implementado...
 - Etapa de voxelização completa
 - Construção da octree em andamento, devido a problemas com resultados na etapa anterior

Referências

Laine, S., Karras, T. 2010. Efficient sparse voxel octrees. In *Proc. of ACM SIGGRAPH 2010 Symposium on Interactive 3D Graphics and Games*, 55-63.

Crassin, C., Neyret F., Lefebvre, S., Eisemann, E. 2009. Gigavoxels: Ray-guided Streaming for Efficient and Detailed Voxel Rendering. In *Proc. I3D 2009*, 15-22.

Akenine-Möller, T.. 2001. Fast 3D Triangle-Box Overlap Testing. *Journal of Graphics Tools*, vol. 6, no. 1, 29-33.

Rosen, D. 2009. Triangle Mesh Voxelization.

http://www.gamasutra.com/blogs/DavidRosen/20091202/3702/Triangle_Mesh_Voxelization.php