

Técnicas de Aceleração para Visualização de Modelos Massivos em Tempo Real

Paulo Ivson
PUC-Rio
psantos at tecgraf.puc-rio.br

Waldemar Celes Filho (orientador)
PUC-Rio
celes at inf.puc-rio.br

Resumo

Para a visualização em tempo real de modelos massivos, é necessária a utilização de técnicas de aceleração. Este trabalho investiga a eficácia da técnica de descarte por oclusão aliada ao descarte contra o volume de visão. Foram analisadas diferentes metodologias para construção de hierarquias e seu impacto nestes algoritmos. Além disso, foi avaliada a qualidade do ajuste de diferentes volumes envolventes e sua importância no desempenho da aplicação. Os resultados evidenciam as qualidades de uma boa hierarquia para o descarte por oclusão, além de comprovar o uso desta técnica para acelerar a visualização de modelos massivos.

1. Introdução

Avanços nas tecnologias de aquisição, modelagem e simulação de dados têm resultado em modelos geométricos cada vez maiores. Estes modelos extremamente complexos são comumente referidos por “modelos massivos”. Exemplos de destaque incluem: aplicações de modelagem em CAD, visualização de simulações de fluidos e reconstrução de objetos reais com o uso de scanners a laser.

Apesar do rápido avanço no desempenho das placas gráficas, a visualização interativa destes modelos tem se tornado cada vez mais difícil. É necessário reduzir o volume de informação a ser processado de modo a melhorar o desempenho das aplicações, sem prejudicar a qualidade da imagem final.

2. Trabalhos Relacionados

Uma técnica de aceleração já consolidada é o descarte contra o volume de visão [1]. Utiliza-se uma hierarquia de volumes envolventes para descartar geometrias que se encontram fora do campo de visão

do observador – *View Frustum Culling* ou (VFC)¹. Em geral, apenas este algoritmo não é capaz de garantir um bom desempenho durante a visualização.

Placas gráficas modernas tornaram possível a identificação de objetos ocultos para o observador através de testes de oclusão. Evita-se o envio para a placa gráfica destes objetos que não contribuem para a imagem final. A proposta de descarte por oclusão hierárquico [2] elimina eficientemente grupos destes objetos – *Coherent Hierarchical Culling* ou (CHC).

Os resultados mais recentes na visualização de modelos massivos provêm de duas técnicas conhecidas por Quick-VDR [3] e Far Voxels [4]. A primeira agrupa espacialmente objetos para utilizar algoritmos de simplificação de malhas. A segunda procura substituir geometrias que teriam uma contribuição pequena na imagem final por representações simplificadas chamadas *voxels*. Ambas utilizam hierarquias para descartar grupos de objetos ocultos utilizando o algoritmo de CHC.

Neste trabalho, a principal técnica estudada é o descarte por oclusão hierárquico. Seu objetivo é determinar se grupos de objetos estão visíveis ou não, evitando o processamento desnecessário de geometrias que não são visíveis para o observador. Foi implementado como referência o descarte contra o volume de visão.

3. Descarte por Oclusão Hierárquico

O teste de oclusão para uma dada geometria envolve seu processamento pela placa gráfica. Por isso, utiliza-se o volume envolvente dos objetos a serem testados. Volumes de baixo custo de visualização reduzem também o custo do teste de visibilidade. Porém, quanto melhor seu ajuste, maior será a fidelidade do resultado obtido pelo teste.

¹ Para facilitar a referência às técnicas estudadas, serão utilizadas siglas entre parênteses que irão identificar os algoritmos conforme são descritos ao longo deste documento.

Dependendo da hierarquia construída, objetos de diferentes complexidades geométricas serão testados. Encontrar um equilíbrio entre o particionamento espacial e o uso eficiente de testes de oclusão é um desafio investigado neste trabalho.

É importante observar que existe um tempo de espera para o resultado de um teste de oclusão. Portanto, é necessário manter a CPU enviando dados para a placa gráfica. Caso contrário, a CPU pode travar esperando o resultado de um teste ou então a GPU pode desperdiçar tempo sem ter dados para processar.

Estas situações prejudicam o desempenho final da aplicação. Para evitá-las, o algoritmo de CHC intercala o envio de novos testes de visibilidade com o caminhamento na hierarquia e a obtenção de resultados de testes já efetuados. Caso o resultado de um teste esteja disponível e a visibilidade seja positiva, as geometrias correspondentes são imediatamente enviadas para a placa gráfica.

O algoritmo de CHC também explora a coerência temporal entre um quadro e o anterior. Somente efetuam-se testes de oclusão para folhas previamente visíveis e nós internos previamente invisíveis. Além disso, assume-se que uma folha previamente visível permanece visível no quadro atual. Dessa forma, sua geometria pode ser enviada para a placa gráfica sem que se espere pelo resultado de seu teste de visibilidade. Neste caso, efetua-se o teste de oclusão não sobre seu volume envolvente mas diretamente sobre a geometria, a fim de atualizar sua visibilidade para o próximo quadro.

4. Volumes Envolventes

Foram analisados dois dos volumes mais comuns na literatura: a caixa alinhada com os eixos -AABB- e a caixa orientada -OBB-. Enquanto o algoritmo de construção de uma AABB possui complexidade $O(n)$ e é simples, existem inúmeras propostas para construção de uma caixa orientada.

O algoritmo clássico de construção de uma OBB [5] utiliza a média dos vértices a serem envolvidos e os autovetores de sua matriz de covariância como as três direções principais da caixa orientada. Este algoritmo será referido por (OC). Sua complexidade é $O(n)$.

Como esta abordagem é influenciada negativamente por concentrações de vértices no espaço, utiliza-se o fecho convexo dos vértices para o cálculo da média e covariância [6]. Obtemos o fecho através do algoritmo chamado QuickHull [7], de complexidade $O(n \lg n)$ no caso médio e $O(n^2)$ no pior caso.

Foram implementados quatro algoritmos nesta linha. O primeiro utiliza o método da covariância de

[6] sobre o fecho (OFC). Os três demais propõem: uma amostragem uniforme sobre sua superfície (OFS) – conforme [6]; ponderar a covariância pelos centróides dos triângulos do fecho (OFT) e ponderá-la pelas áreas dos triângulos (OFA) – segundo a proposta em [8].

Finalmente, um quinto algoritmo encontra uma aproximação satisfatória da caixa envolvente mínima [9]. Neste caso, a implementação fornecida pelos autores possui três heurísticas: um processo iterativo de aproximação do menor diâmetro dos vértices (CMD), uma amostragem sobre uma grade regular (CAG) e o uso da grade para todos os vértices envolvidos (CMG). Esta abordagem possui complexidade $O(n \lg n + n/\epsilon^3)$, onde ϵ é o erro tolerado.

5. Hierarquia Espacial

As hierarquias analisadas neste trabalho incluem BSP-Trees, Octrees, kd-Trees [10] e OBB-Trees [7]. O sucesso da kd-Tree e da OBB-Tree na literatura recente indicou um estudo mais aprofundado de ambas.

Na construção destas hierarquias, primeiro obtém-se o volume envolvente de cada nó a ser subdividido. No caso da OBB, esta etapa é influenciada diretamente pelo algoritmo de construção utilizado. Em seguida, escolhe-se um plano de corte perpendicular ao eixo mais longo da caixa envolvente. Este posicionamento determina a qualidade final da hierarquia.

Por conta disso, foram implementadas cinco heurísticas diferentes para o posicionamento do plano de corte: mediana espacial (ME), mediana do número de objetos (MO), mediana dos centros (MC), média dos centros (MeC) e equilibrar a complexidade geométrica (EC). Destaca-se o uso de uma hierarquia para garantir a escalabilidade dos algoritmos.

6. Resultados

O desempenho da visualização foi medido através de um passeio automático pelas cenas de teste, variando a quantidade de geometrias visíveis e o grau de oclusão, vide a Figura 1.

As cenas *Teapots*(n,s), onde n é o número de objetos e s o fator individual de escala, são constituídas de vários *teapots* (1.364 vértices) distribuídos aleatoriamente em um volume alinhado com os eixos. O segundo tipo de cena corresponde ao modelo CAD da plataforma de petróleo P-38, composto por 90.867 objetos totalizando 4.717.749 vértices.

Os testes foram conduzidos em um PC desktop AMD Athlon XP 3800+ 2.4GHz com 2GB de RAM e uma placa gráfica Nvidia GeForce 7800 GT 256MB.

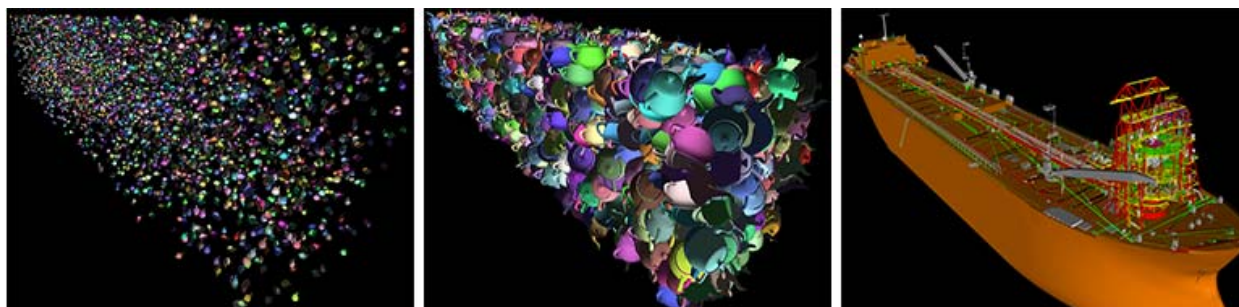


Figura 1. Cenas de teste para o algoritmo de descarte por oclusão. Teapots(5000,3) à esquerda, Teapots(5000,20) ao centro e modelo CAD P-38 à direita.

6.1. Ajuste de Volumes Envolventes

Cada volume envolvente foi avaliado segundo seu tempo de construção (em segundos) e sua área de superfície. O primeiro identifica a escalabilidade dos algoritmos e o segundo o ajuste das caixas construídas. Os resultados encontram-se na Tabela 1.

Observa-se que na cena Teapots(5000,10) todas as caixas que utilizam covariância apresentaram os piores resultados (OC, OFC, OFS, OFT, OFA). Isto se explica pelo fato das geometrias estarem distribuídas em um volume alinhado com os eixos, favorecendo a AABB. Mesmo assim, as aproximações CMD, CAG e CMG da caixa mínima obtiveram as caixas mais bem ajustadas.

Destaca-se que o fecho convexo não foi capaz de garantir um bom ajuste, além de seu tempo de construção maior. Neste caso é preferível a abordagem CMD, ciente de que seu tempo de construção pode torná-la proibitiva para modelos massivos.

Tabela 1. Volumes envolventes

| Caixa | Teapots(500,10) | | Teapots(5000,10) | |
|-------|-----------------|-----------|------------------|-----------|
| | tempo | área | tempo | área |
| AABB | 0,15 | 10.623,51 | 1,59 | 10.835,00 |
| OC | 0,17 | 13.117,38 | 1,78 | 13.928,76 |
| OFC | 17,92 | 13.413,87 | 164,91 | 15.149,93 |
| OFS | 17,99 | 11.927,99 | 164,78 | 13.253,46 |
| OFT | 17,88 | 13.298,16 | 166,89 | 12.786,44 |
| OFA | 17,85 | 31.878,56 | 164,98 | 36.239,65 |
| CMD | 14,68 | 10.581,23 | 206,01 | 10.802,03 |
| CAG | 102,51 | 10.581,15 | 1.351,43 | 10.802,03 |
| CMG | 6.886,68 | 10.581,23 | N/D | N/D |

6.2. Custos dos Testes de Oclusão

Foi analisado o impacto dos testes de oclusão em uma aplicação de visualização simples, sem uso de hierarquia. Foi constatado que não há custo adicional significativo nas seguintes etapas: envio de um teste de oclusão, verificação da disponibilidade de um resultado e obtenção de um resultado da placa gráfica.

O tempo para que um resultado se torne disponível é linearmente proporcional à complexidade da geometria sendo testada. Além disso, foi identificado um limite no hardware em torno de 500 testes por quadro. A partir daí, cada teste torna-se cada vez mais caro, prejudicando o desempenho da aplicação.

6.3. Desempenho de Visualização

Para analisar o desempenho dos algoritmos, foram utilizadas três cenas de Teapots além da cena P-38. A cena Teapots(500,30) avalia o algoritmo de CHC para casos onde já existe um bom desempenho, devido à baixa complexidade da cena. Já o teste Teapots(5000,3) avalia o pior caso, quando há muitas geometrias e pouca oclusão. Inversamente, o melhor caso é avaliado na cena Teapots(5000,20), quando há muita oclusão. Finalmente, a cena do modelo P-38 reflete um caso real de uso. Nota-se na Figura 1 diferentes graus de oclusão nos testes.

No caso da cena Teapots(500,30), a técnica de descarte contra o volume de visão obtém um ganho de 5% no desempenho, enquanto a técnica de CHC 40%. Já na cena Teapots(5000,3), o algoritmo de VFC obtém uma melhoria de 2% e o descarte por oclusão uma melhoria de 12%. Mesmo no pior caso, não há perda de desempenho utilizando-se a técnica de CHC.

No teste Teapots(5000,20), o algoritmo de VFC obtém uma melhoria de 6% no desempenho enquanto o algoritmo de CHC obtém 81%. No melhor caso, acelera-se em quase 5x a taxa de quadros por segundo.

Até a cena P-38, não se observa nenhuma diferença no uso de diferentes volumes envolventes e hierarquias. A Tabela 2 exhibe os tempos em segundos do passeio pela cena de CAD. Percebe-se que as piores hierarquias são baseadas em ME e MC, enquanto a melhor é baseada na heurística EC. A Tabela 3 mostra que neste caso as piores hierarquias são também as mais desbalanceadas. Nesta cena, o algoritmo de VFC obteve um ganho de 5% e o de CHC um ganho de 57%, atestando sua eficiência em um caso real.

Tabela 2. Cena P-38

| Heuríst. | Hierarquia | Visualização | | |
|----------|------------|--------------|-------|-------|
| | tempo | simples | VFC | CHC |
| MO | 19,37 | 78,72 | 74,83 | 34,97 |
| ME | 27,33 | 76,09 | 74,60 | 45,86 |
| MC | 35,08 | 77,71 | 92,53 | 44,77 |
| MeC | 20,91 | 76,12 | 74,71 | 33,96 |
| EC | 16,98 | 78,81 | 75,02 | 33,38 |

Tabela 3. Descrição das hierarquias

| | MO | ME | MC | MeC | EC |
|------------|------|------|------|------|------|
| Altura | 14 | 21 | 17 | 17 | 13 |
| No. Nós | 4299 | 1345 | 3789 | 4365 | 5779 |
| No. Folhas | 2150 | 673 | 1895 | 2183 | 2890 |

7. Conclusões e Trabalhos Futuros

Os resultados obtidos neste trabalho comprovam a eficácia do algoritmo de descarte por oclusão hierárquico. Como já esperado, o algoritmo de descarte contra o volume de visão é adequado apenas como método complementar para aceleração da visualização.

Além dos testes efetuados, foram implementadas três tentativas de melhoria do algoritmo CHC: evitar testes consecutivos para um mesmo nó, usar uma probabilidade de um nó previamente visível permanecer visível e evitar testes de oclusão para objetos mais próximos do observador. Nenhuma delas trouxe uma melhoria significativa no desempenho.

No pior caso, o algoritmo de CHC não causou nenhuma perda de desempenho, enquanto foram observadas melhorias de 2x a 5x na taxa de quadros por segundo da aplicação. Além disso, o desempenho consistente nas diferentes cenas comprova a escalabilidade do algoritmo através do uso de uma hierarquia de divisão espacial.

A análise de diferentes heurísticas para construção desta hierarquia nos leva à seguinte conclusão: heurísticas baseadas em uma divisão do espaço (ME e MO) possuem pior desempenho do que heurísticas que procuram equilibrar a complexidade geométrica ao longo da hierarquia (EC). Para reforçar este argumento, o uso de volumes envolventes de melhor ajuste não alterou significativamente o desempenho da aplicação. Segundo os resultados obtidos, o uso de uma caixa alinhada com os eixos (AABB) já é suficiente.

Pode-se concluir que o custo principal no processamento dos testes de oclusão não se encontra na rasterização dos volumes envolventes dos nós internos invisíveis, e sim na complexidade das geometrias sendo testadas nas folhas.

Identificamos como trabalhos futuros uma implementação mais eficiente e robusta da técnica de aproximação da caixa mínima. Além disso, existem melhorias propostas no algoritmo de CHC que não foram investigadas, como especular o resultado de um teste de oclusão que não esteja disponível. Finalmente, pode-se modelar a construção da hierarquia através de uma função de custo, similar ao SAH utilizado em algoritmos de traçado de raios [11], porém de forma apropriada para os testes de oclusão.

Referências

- [1] Assarsson U., Moller T.. "Optimized view frustum culling algorithms for bounding boxes". Journal of Graphics Tools, 5(1):9--22, 2000.
- [2] Bittner J. et al. "Coherent Hierarchical Culling: Hardware Occlusion Queries Made Useful". Institute of Computer Graphics and Algorithms, Vienna University of Technology, EUROGRAPHICS 2004.
- [3] Yoon. S., Salomon, B., Gayle, R. and Manocha, D. "Quick-VDR: Out-of-Core View-Dependent Rendering of Gigantic Models". IEEE Transactions on Visualization and Computer Graphics, July/August 2005, p369-382.
- [4] Gobbetti, E. and Marton, F. 2005. "Far voxels: a multiresolution framework for interactive rendering of huge complex 3D models on commodity graphics platforms". ACM Trans. Graph. 24, 3 (Jul. 2005), p878-885.
- [5] Duda R. O., Hart P. E., "Pattern Classification and Scene Analysis". John Wiley and Sons, 1973.
- [6] Gottschalk S., Lin M., Manocha D. "OBB-Tree: A Hierarchical Structure for Rapid Interference Detection", Computer Graphics (SIGGRAPH 96 Proceedings), p171-180, August 1996.
- [7] Barber C., Dobkin D., Huhdanpaa H., "The Quickhull Algorithm for Convex Hull", Geometry Center Technical Report GCG53, Univ. of Minnesota, MN, 1993.
- [8] Eberly, D. "3D Game Engine Design: A Practical Approach to Real-Time Computer Graphics", 2nd edition. Morgan Kaufmann Publishers Inc, 2006.
- [9] Barequet, G., Har-Peled, S. "Efficiently approximating the minimum-volume bounding box of a point set in three dimensions". Proceedings of the 10th ACM-SIAM Symposium on Discrete Algorithms (SODA) (1999), p82-91.
- [10] Möller, Thomas & Haines, Eric. "Real-Time Rendering", A K Peters, 2002.
- [11] Havran, V. "Heuristic Ray Shooting Algorithms". Czech Technical University, Ph.D. thesis, 2001.