3
Método da oclusão implícita

Dados volumétricos de grandes dimensões estão presentes em vários campos como medicina, sísmica, etc. A geração e visualização de isosuperfícies para esses dados pode ser impraticável em algumas situações, pois a isosuperfície gerada pode conter milhões de polígonos sobrecarregando o processamento gráfico. Além disso, a busca de todas as células do dado que contém a isosuperfície e a consequente geração da triangulação resultam em um alto consumo de tempo.

Uma proposta para viabilizar esse processo procura explorar o fato de que, para alguns dados volumétricos grandes, porções da isosuperfície desejada são internas, estando portanto, ocultas pelas porções visíveis da isosuperfície. Assim, se gerarmos apenas porções visíveis da isosuperfície, reduziremos o número de faces geradas e visualizadas resultando em um processo mais rápido.

Este é o princípio básico de uma classe de algoritmos denomidado view-dependent (dependência do observador) (8), no qual baseia-se o método da oclusão implícita que mencionaremos a seguir.

Inicialmente descreveremos alguns trabalhos anteriores relacionados.

3.1
Trabalhos anteriores

Livnat (8) propõe uma clasificação das principais estratégias utilizadas para acelerar a extração de isosuperfícies em três categorias: decomposição geométrica do espaço, decomposição do espaço valor e decomposição do espaço imagem.

3.1.1
Decomposição geométrica do espaço

Especificamente para grids volumétricos estruturados obtemos uma organização espacial dos dados, imposta pela própria estrutura ortogonal do grid. Sendo assim, os métodos baseados na geometria do espaço procuram explorar a coerência entre células adjacentes.

Um primeiro exemplo é o algoritmo Marching Cubes (10), criado em 1987 com o objetivo de renderizar imagens médicas em tomografias computadorizadas e ressonâncias magnéticas. A idéia básica do algoritmo é aproximar uma isosuperfície em um dado volumétrico através de triângulos. O algoritmo calcula os vértices do triângulo utilizando interpolação linear dos vértices de cada cubo do dado volumétrico, daí o nome "Marching Cubes", pois o algoritmo "marcha"em todos os cubos do volume de dados. O Marching Cubes não se preocupa em localizar de uma forma eficiente porções do grid que não contenham a isosuperfícies. Visto que os dados volumétricos são na maioria dos casos de grandes dimensões, muito se têm pesquisado na direção de algoritmos que melhorem a eficiência do Marching Cubes.

Nessa direção, Wilhelms e Van Gelder (14) realizaram um trabalho que utiliza uma octree para otimizar a renderização de um dado volumétrico. Inicialmente o dado volumétrico regular é ajustado a octree de forma que cada nó da octree delimita um subconjunto do grid. Em seguida, dado um isovalor (ou isolevel) w, classifica-se cada nó da octree como: positivo, negativo ou zero, segundo o critério: se o valor da função em todo o subconjunto do grid contido nesse nó for maior do que w, o nó é "positivo". Caso o valor da função em todos os vértices for menor que w, o nó é classificado como "negativo", e se houver mudança de sinal, dizemos que o nó é "zero"(ver figura 3.1). O objetivo de classificar os nós da octree, é aplicar o algoritmo Marching Cubes apenas nos nós "zero"da octree, ou seja, nós que contém a isosuperfície, evitando assim, percorrer regiões do grid que não contenham a isosuperfície. Este algoritmo não é view-dependent, pois a quantidade de nós "zero"da octree independe da posição do observador.

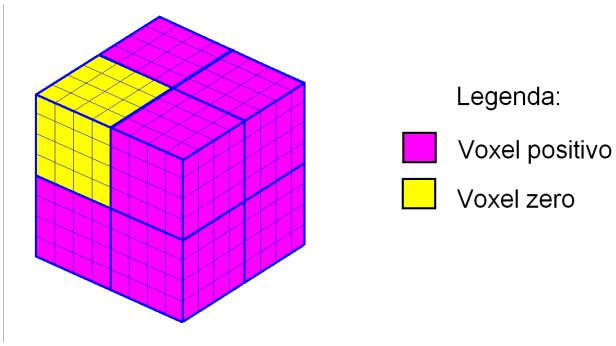

Figura 3.1: Octree de nível 1 onde há 1 nó "zero"e 7 nós "positivos", Logo, aplica-se o algoritmo de Marching Cubes em apenas 1/8 de todo o volume de dados.

3.1.2
Decomposição do espaço valor

A decomposição do espaço valor considera a decomposição dos valores do campo escalar f definido sobre o grid, em geral, através de intervalos cujos extremos correspondem respectivamente ao mínimo e máximo que f atinge em um dado nó.

Assim, através de estratégias que organizem e ordenem os intervalos é possível acessar eficientemente os nós que contenham regiões da isosuperfície. Usando esta técnica, Livnat et al. (8) definem a noção do span space e utilizam uma kd-tree para organizar os intervalos. Cignoni et al. (4) utilizam o span-space para reduzir a complexidade na fase de busca dos nós. Gao e Shen (5) aplicam o conceito de span space para desenvolver um processamento em paralelo.

Para a extração de isosuperfícies em dados de dimensões arbitrárias, Chiang et. al. (3) e (2) propõe a utilização de memória externa (out-of-core).

3.1.3
Decomposição do espaço imagem

Nesta classificação inclui-se os métodos que atuam principalmente na geração da imagem final, buscando evitar a extração da isosuperfície em regiões que não são visíveis.

Parker et al. (11) propõe um ray tracing em tempo real cujo princípio é gerar imagens da isosuperfície sem uma representação explícita dos polígonos da isosuperfície.

Em (8) Livnat et al. introduziram uma classe de algoritmos deno- minados view-dependent (dependência do observador), baseado em um trabalho anterior de Greene (6). O objetivo é reduzir os tempos de busca, geração e renderização selecionando, para isso, somente células que contenham porções visíveis da isosuperfície para uma dada posição do observador. O princípio básico do trabalho proposto por Livnat et al. (9) é construir regiões de oclusão (via espaço imagem) por intermédio da própria isosuperfície extraída incrementalmente. Para isso utiliza uma estrutura hierárquica com percorrimento de frente para trás.

3.1.4
Apresentação do trabalho

Nessa dissertação prosseguimos com o trabalho proposto em Pesco et al. (12) que, seguindo a classificação proposta, corresponde a um algoritmo view-dependent. A principal diferença entre Pesco at al. (12) e Livnat (7) é que a

construção da região de oclusão é feita sem calcular a isosuperfície. A geração da região de oclusão é feita apenas com a informação do campo escalar.

3.2
Método da oclusão implícita

Considere um campo escalar contínuo $f : D \rightarrow \Re$ definido sobre um domínio convexo $D \subset \Re^3$ e um isovalor w. O objetivo é determinar ocluders para a isosuperfície $f^{-1}(w)$, ou seja, determinar regiões que delimitam porções visíveis e não visíveis da isosuperfície.

Explorando a continuidade de f podemos gerar ocluders sem a necessidade de calcular a isosuperfície $f^{-1}(w)$. Para isso, estudamos as trocas de sinal de $f^*(x) = f(x) - w$ de positivo para negativo (ou vice versa) numa vizinhança da isosuperfície. Sem perda de generalidade, vamos considerar $w = 0$. Caso $w \neq 0$, basta considerar a função f^*.

3.2.1
Descrição do método

Considere $A \subset D$ uma região onde f é sempre positivo e $B \subset D$ outra região em que f é sempre negativo conforme a figura 3.2.

Desde que D seja convexo, qualquer reta r conectando um ponto em A a um ponto em B está totalmente contida em D. Então, se o valor de f sobre r muda continuamente de positivo para negativo, segundo o teorema do valor intermediário, existe um zero de f entre os 2 valores, que é exatamente a interseção da reta r com a isosuperfície.

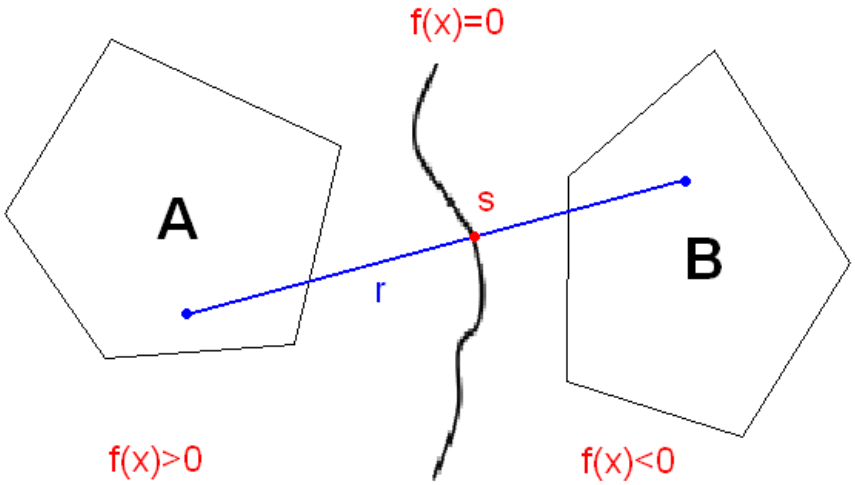

Figura 3.2: A reta r conectando o conjunto A ao conjunto B intercepta a isosuperfície $f^{-1}(0)$.

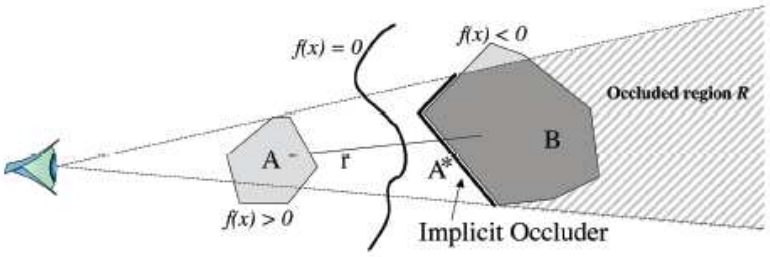

Figura 3.3: A idéia fundamental na oclusão implícita é explorar a continuidade do campo escalar f para definir uma região de oclusão, sem a necessidade de computar a isosuperfície.

Em seguida, considerando uma direção particular do observador projetamos a região A sobre o bordo da região B (ou vice versa) obtendo A^* conforme indicado na figura 3.3.

Assim qualquer raio partindo do obsevador e alcançando A^* deve obrigatoriamente ter interseção com a isosuperfície. Podemos, portanto, concluir que a região atrás de A^* é não visível e eleger A^* como um ocluder.

3.2.2 Observações sobre o método

Algumas observações sobre o método:

- A determinação de A^* depende de uma posição particular do observador. Isso justifica a classificação desse método como view-dependent.
- A determinação de A^* é feita através do estudo das trocas de sinal de f, não sendo necessário calcular a isosuperfície. Por esta razão a denominação oclusão "implícita".
- A região entre a isosuperfíce $f^{-1}(0)$ e A^* é considerada visível pelo método, embora não seja pois está atrás da isosuperfície. Dessa forma o método define um conjunto visível *conservativo* (ver figura 3.4), *from point*, pois calcula a visibilidade a partir da posição da câmera e *image space*, pois verificamos a visibilidade (discreta) de porções do objeto.
- Para ser efetiva, qualquer implementação do método precisa detectar a primeira mudança de sinal na direção do raio para um melhor aproveitamento do ocluder (veja figura 3.5)

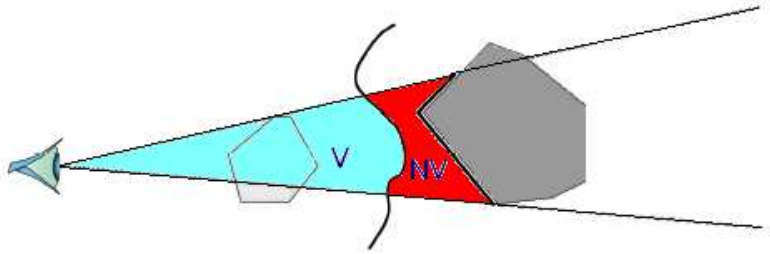

Figura 3.4: O conjunto conservativo de visibilidade é a união dos dois conjuntos indicados V e NV, pois inclui toda a porção visível e possivelmente uma porção não visível.

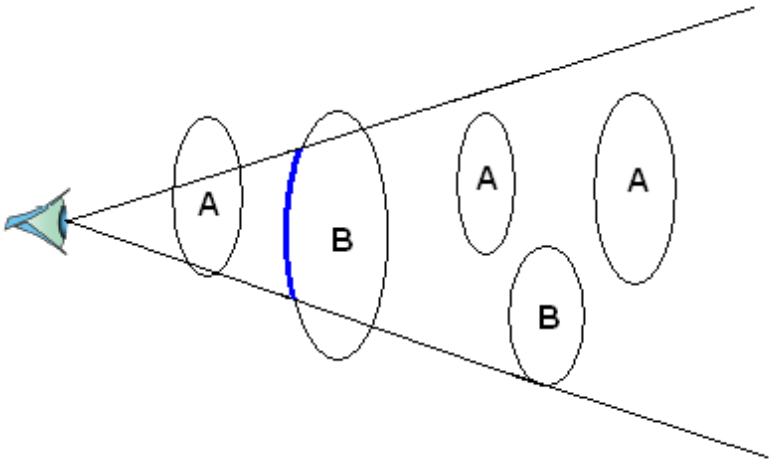

Figura 3.5: A idéia do occluder é detectar a primeira mudança de sinal.