

ACELERAÇÃO DO MÉTODO "RAY-TRACING"

Rudnei Dias da Cunha

Carla Maria Dal Sasso Freitas

Centro de Processamento de Dados

Universidade Federal do Rio Grande do Sul

Rua Ramiro Barcelos 2574

Porto Alegre - RS - BRASIL

Resumo - A aceleração do método "ray-tracing" é discutida, apresentando-se algoritmos para a determinação mais rápida das intersecções e considerações sobre as primitivas geométricas utilizadas.

Palavras-chave : realismo, "ray-tracing"

1. Introdução.

O método "ray-tracing", proposto, em sua forma revisada, por Whitted [1], permite a obtenção de imagens geradas por computador com alto grau de realismo, simulando as interações da luz com os objetos. No entanto, este método apresenta tempos elevados de execução. A aceleração do método tem sido proposta por diferentes autores, seja pela implementação de novos algoritmos [2][3] ou pelo uso de outras arquiteturas [4][5].

Neste trabalho, discutiremos alguns algoritmos para acelerar o método [6], bem como mostraremos que o uso adequado de primitivas geométricas pode contribuir decisivamente para a redução do tempo de execução do método.

2. Algoritmos para acelerar o método.

O método "ray-tracing" é um método de exibição de imagens orientado ao dispositivo. O cálculo da cor de um ponto no dispositivo de exibição é feita determinando-se as intersecções de um raio (dirigido do observador a este ponto) com os objetos, criando uma árvore de raios refletidos e transmitidos [1]. Dessa forma, o método apresenta um tempo de execução proporcional à resolução do dispositivo, ao número de objetos e à profundidade da referida árvore.

Quando da determinação das intersecções, muitas destas serão calculadas sem que efetivamente se fizessem necessárias, pelo fato dos objetos não se encontrarem na direção do raio. Este desperdício de operações é devida à falta de recursos, no método, que nos permitam determinar o caminho percorrido por um raio dentro da cena.

É possível determinar-se tal caminho utilizando uma combinação de estruturas de armazenamento dos objetos, orientadas à distribuição espacial dos mesmos, e de um algoritmo que opera sobre estas estruturas, em função de um dado raio.

Em nossa implementação [6], foram desenvolvidas duas maneiras diferentes de se determinar o caminho percorrido por um raio:

- Subdivisão da área ocupada pelos objetos em quadrantes;
- Subdivisão do volume ocupado pelos objetos em octantes.

2.1 Subdivisão da área ocupada pelos objetos em quadrantes.

Esta técnica consiste em subdividir em quatro quadrantes a área ocupada pelos objetos no plano de projeção, segundo a fig. 1. As retas NS e WE determinam a localização de qualquer ponto no plano, limitado pelos pontos extremos (X_{min}, Y_{min}) e (X_{max}, Y_{max}) . Estas retas passam pelo centro C da área ocupada pelos objetos.

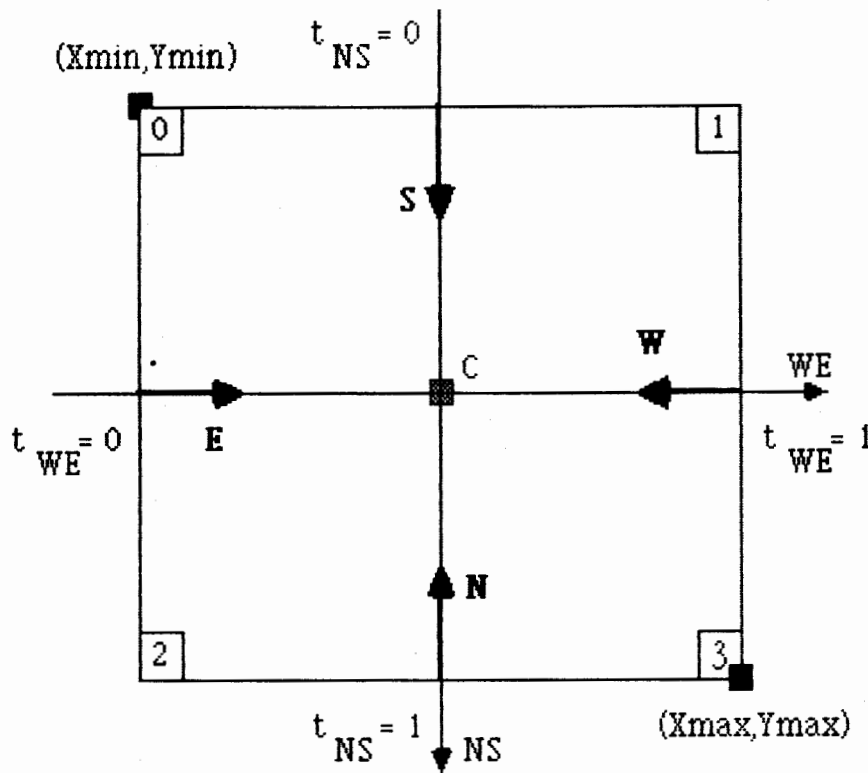


Figura 1 - Subdivisão da área ocupada pelos objetos em quadrantes.

As retas NS e WE são expressas na forma paramétrica,

$$\begin{aligned} \text{NS} : \quad x &= (X_{\min} + X_{\max})/2 \\ y &= Y_{\min} + t (Y_{\max} - Y_{\min}) \end{aligned}$$

$$\begin{aligned} \text{WE} : \quad x &= X_{\min} + q (X_{\max} - X_{\min}) \\ y &= (Y_{\min} + Y_{\max})/2 \end{aligned}$$

Os vetores **N**, **S**, **W** e **E** são chamados de vetores interiores, pois apontam para o interior da área ocupada pelos objetos. Pela disposição convencional destes vetores, cada quadrante terá associados a si dois vetores interiores.

A determinação de quais quadrantes são percorridos por um raio é feita em função do quadrante no qual este raio se originou. Inicialmente, calculam-se os produtos internos entre o vetor direção **v** do raio e cada um dos dois vetores interiores associados ao quadrante de origem.

Se ambos os produtos forem iguais a zero, então necessariamente o raio dirige-se para fora da área ocupada pelos objetos e, portanto, somente poderá interceptar objetos que se encontram em seu quadrante de origem. Se algum destes produtos for maior ou igual a zero (ou ambos), calculam-se as intersecções do raio com as retas NS e WE, em função dos parâmetros "t" e "q".

De acordo com o quadrante de origem, comparam-se os valores destes parâmetros com os intervalos $(-\infty, 0]$, $[0, 0.5)$, $(0.5, 1]$, $[1, +\infty)$ e determinam-se quais os quadrantes percorridos pelo raio [6]. Na seção 2.1.1 apresentamos o algoritmo para a determinação do caminho percorrido por uma raio originado no quadrante 0. Os quadrantes são armazenados em uma lista conforme a ordem de proximidade que guardam em relação ao quadrante de origem. Esta ordem permite efetuar mais rapidamente a determinação das intersecções.

2.1.1 Caminho percorrido por um raio com origem no quadrante 0.

| | |
|-------------------------------|---------------------------------|
| se ($v \cdot S < 0$) então | |
| se ($t > 0$) então | Quadrantes = { 0, 1 } |
| senão | Quadrantes = { 0 } |
| senão | |
| se ($v \cdot E < 0$) então | |
| se ($q > 0$) então | Quadrantes = { 0, 2 } |
| senão | Quadrantes = { 0 } |
| senão | |
| se ($t = q = 0.5$) então | Quadrantes = { 0, 3 } |
| senão | |
| se ($0 < t < 0.5$) então | |
| se ($0.5 < q \leq 1$) então | Quadrantes = { 0, 1, 3 } |
| senão | |
| se ($q > 1$) então | Quadrantes = { 0, 1 } |
| senão | |
| senão | |
| se ($0 \leq q < 0.5$) então | |
| se ($0.5 < t \leq 1$) então | Quadrantes = { 0, 2, 3 } |
| senão | |
| se ($t > 1$) então | Quadrantes = { 0, 2 } |

2.2 Subdivisão do volume ocupado pelos objetos em octantes.

Nesta técnica, o volume ocupado pelos objetos é subdividido em octantes, dispostos segundo a fig. 2a. Cada octante é um paralelepípedo, cujas faces são identificadas de acordo com a fig. 2b. Os octantes podem armazenar um número máximo, pré-definido, de objetos.

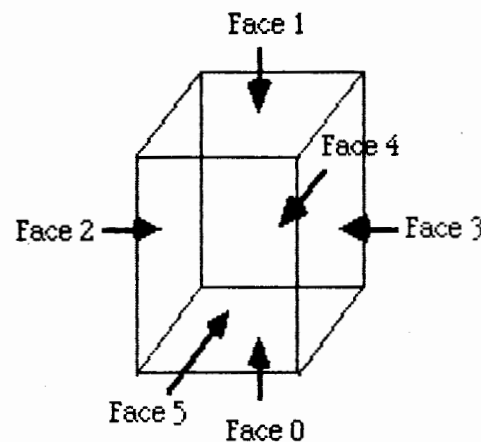
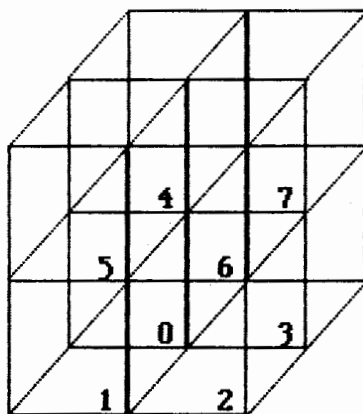


Figura 2a - Subdivisão em octantes.

Figura 2b - Faces de um octante.

A subdivisão em octantes é dinâmica, em contraste com a técnica anteriormente descrita. Sempre que o número máximo de objetos por octante é excedido, o octante é subdividido em oito novos octantes ("filhos") e os objetos residentes naquele octante são redistribuídos entre seus octantes filhos. Ao se estabelecer um número máximo de objetos residentes por octante bem menor do que o número de objetos na cena, esta subdivisão permite obter um número reduzido de objetos por octante [6].

A determinação dos octantes percorridos por um raio é feita de acordo com o grafo da fig. 3. Este grafo mostra as relações de justaposição entre as faces de cada octante. Cada nodo do grafo contém a identificação do octante (**O**) e da face de interesse (**I**), na forma **O/I**.

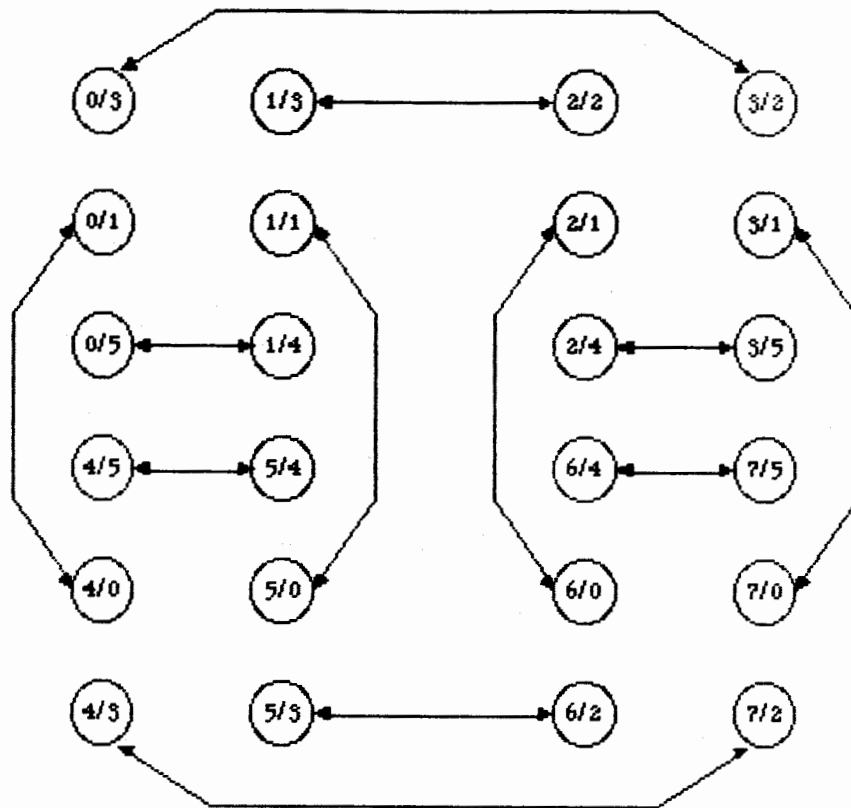


Figura 3 - Grafo de relacionamento octante/face.

É interessante notar que nem todas as faces de todos os octantes estão presentes no grafo. As faces que não se encontram representadas no grafo são aquelas que, se um raio interceptar, não vão corresponder à passagem do raio por um outro octante. Coincidentemente, estas faces são aquelas que definem a fronteira do espaço ocupado pelos objetos. Por exemplo, se um raio intercepta a face 1 do octante 7, isto significa que o raio está se dirigindo para fora do espaço ocupado pelos objetos.

O grafo de relacionamento entre octantes e faces determina também que no máximo três octantes serão percorridos por um raio. Se um raio intercepta uma das faces fronteiriças do espaço ocupado pelos objetos, apenas um octante é percorrido pelo raio.

A determinação de quais octantes são percorridos por um raio é feita localizando-se qual o octante de origem do raio. Determina-se então quais faces do octante são interceptadas pelo raio, armazenando-as em uma lista. Para cada face desta lista, verifica-se qual octante está justaposto ao de origem e qual face daquele octante justapõe-se à face interceptada, de acordo com o grafo. Este processo repete-se até que não hajam mais faces interceptadas (ou seja, a lista de faces interceptadas seja esgotada).

É importante ressaltar que a face justaposta à interceptada deve ser desconsiderada quando da determinação das faces do octante justaposto interceptadas pelo raio, de forma a evitar a ocorrência de um ciclo no processo.

A determinação do caminho percorrido pelo raio cria uma lista dos octantes, na ordem em que eles são efetivamente percorridos pelo raio. Portanto, da mesma forma que na técnica de subdivisão em quadrantes, sempre que se determinar a existência de uma intersecção entre o raio e um objeto de um octante desta lista, os demais octantes não mais necessitam ser testados, pois a lista já impõe uma ordem de proximidade dos objetos em relação à origem do raio.

A subdivisão espacial empregada, bem como o critério de utilização de cada octante (número máximo de objetos por octante), asseguram-nos que cada octante filho tem, necessariamente, um número menor de objetos nele contidos, do que no octante pai.

Como a estrutura utilizada neste algoritmo é efetivamente uma árvore, o processo de determinação dos octantes percorridos pelo raio deve ser repetido para os octantes filhos (se houverem) de um octante que esteja contido na lista de octantes percorridos pelo raio. Desta forma, estamos procurando testar sempre o menor número possível de objetos que potencialmente encontram-se no caminho do raio.

2.3 Comparação entre as técnicas.

As duas técnicas de determinação do caminho percorrido por um raio, descritas anteriormente, contribuem para que algoritmos de "ray-tracing" que as utilizem sejam aproximadamente 1.5 e 4.5 vezes mais rápidos do que um algoritmo implementando o método canônico. O gráfico 1 apresenta os tempos de execução para estes três algoritmos, obtidos com a nossa implementação [6].

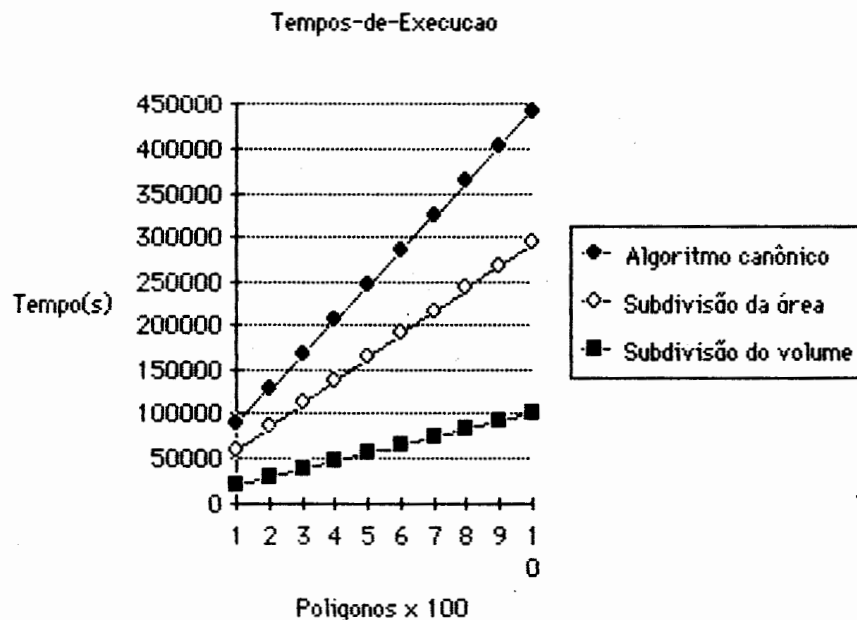


Gráfico 1 - Tempos de execução dos algoritmos.

3. Uso de primitivas geométricas.

A escolha de quais primitivas geométricas serão utilizadas para a descrição dos objetos é decisiva para a melhor "performance" do método "ray-tracing". Como este método faz uso intensivo do cálculo das intersecções existentes entre raios e objetos, é razoável utilizar primitivas que tenham um reduzido gasto computacional no cálculo da intersecção.

Em nossa implementação, as primitivas geométricas utilizadas eram polígonos, os quais apresentam um cálculo simples de intersecção. Tais polígonos eram obtidos pela subdivisão poligonal da superfície de superelipsóides [7], conóides e cilindróides [6].

Para se obter uma descrição razoável da superfície de uma esfera, por exemplo, utiliza-se aproximadamente 400 polígonos. Como o método "ray-tracing" apresenta um tempo proporcional ao número de objetos [6], torna-se mais vantajosa a utilização da própria primitiva, por envolver apenas um objeto na cena.

Supondo uma cena contendo seis esferas, necessitar-se-iam aproximadamente 2400 polígonos. Utilizando-se o algoritmo com subdivisão do volume ocupado pelos objetos, com polígonos como primitivas geométricas, o tempo estimado de execução é de 3800 min. Utilizando-se um algoritmo que implementa o método "ray-tracing" em sua forma canônica, com o uso das próprias esferas como primitivas geométricas [8], o tempo estimado de execução é de 70 min. O uso das esferas como primitivas geométricas implicou numa redução do tempo de execução da ordem de 54 vezes; tal redução indica não ser razoável proceder à subdivisão poligonal da superfície de objetos, se for possível obter uma expressão matemática que os determine.

A qualidade da imagem gerada por algoritmos que utilizam as próprias primitivas é bem superior à obtida com aqueles que utilizam polígonos obtidos da subdivisão poligonal da superfície de objetos, devido principalmente ao problema da diferença de intensidade de cor nos vértices e arestas dos polígonos.

4. Conclusão.

A determinação do caminho percorrido por um raio contribui enormemente para a aceleração do método "ray-tracing", por evitar o desperdício de cálculo de intersecções com objetos que não se encontram na direção do raio. O uso das próprias primitivas geométricas, sem recorrer à subdivisão poligonal da superfície de tais primitivas, apresenta ganhos bastante expressivos, indicando ser bastante apropriado tal uso para acelerar o método.

A combinação de alguma das técnicas de determinação do caminho percorrido por um raio com o uso de primitivas geométricas afigura-se capaz de acelerar o método "ray-tracing" de maneira mais significativa do que as já obtidas. Encontra-se em desenvolvimento um novo algoritmo no qual implementaremos tal combinação, de forma a verificar resultados ainda melhores.

5. Referências bibliográficas.

- [1] WHITTED, T., "An improved illumination model for shaded display", CACM 23:343-9, 1980.
- [2] FUJIMOTO, A., TAKAYUKI, T., IWATA, K., "ARTS : accelerated ray-tracing system", IEEE Computer Graphics and Applications, 6(4):16-26, 1986.
- [3] HAINES, E. A., GREENBERG, D. P., "The light buffer : a shadow-testing accelerator", IEEE Computer Graphics and Applications, 6(9):6-16, 1986.
- [4] CLEARY, J.G., WYVILL, B.M., BIRTWISTLE, G.M., VATTI, R., "Multiprocessor ray tracing", Computer Graphics Forum, 5:3-12, North-Holland.
- [5] GAUDET, S., HOBSON, R., CHILKA, P., CALVERT, T., "Multiprocessor experiments for high speed ray-tracing", ACM Transactions on Graphics, 7(3):151-79, 1988.
- [6] da CUNHA, R.D., "Algoritmos para a síntese de imagens através do método ray-tracing". Trabalho de Diplomação, DI/UFRGS, Dezembro de 1988.
- [7] BARR, A., "Superquadrics and angle-preserving transformations", IEEE Computer Graphics and Applications, 1(1):11-23, 1981.
- [8] da CUNHA, R.D., "Um sintetizador de imagens através do método ray-tracing", 1989 (em preparação).