

Algoritmos para a Síntese de Cenas através do Método "Ray-Tracing"

Rudnei Dias da Cunha
Centro de
Processamento de Dados
Rua Ramiro Barcellos, 2574
Porto Alegre - RS - CEP 90049

Carla Maria Dal Sasso Freitas
Pós-Graduação em
Ciências de Computação
Av. Osvaldo Aranha, 99
Porto Alegre - RS - CEP 90001

Universidade Federal do Rio Grande do Sul

Sumário - O presente trabalho apresenta dois algoritmos para a síntese de cenas através do método "ray-tracing". A partir dos resultados obtidos são comparados os desempenhos dos algoritmos.

1. Introdução.

A síntese de cenas através do método "ray-tracing" foi primeiramente proposta por Whitted [WHI80]. O realismo atingido por algoritmos que o implementam é considerável, por levar em conta os efeitos da reflexão e transmissão da luz, bem como permitir simular o efeito das sombras. No entanto, o tempo de execução destes algoritmos é elevado, e a redução deste tempo é objeto de trabalhos recentes, seja melhorando os métodos de intersecção [COQ85] ou utilizando estruturas de dados hierárquicas [GLA84] [FUJ86], passando pela utilização de computadores baseados em outros tipos de arquiteturas [PLU85] [CLE86].

O trabalho aqui apresentado é resultado parcial de um projeto que visa comparar diferentes algoritmos de síntese de cenas através do método "ray-tracing", ora em desenvolvimento na UFRGS [CUN87]. O objetivo deste projeto é permitir sintetizar cenas em um menor espaço de tempo, mantendo-se um razoável grau de realismo.

2. Conceitos.

Os algoritmos implementados sintetizam uma cena cujos objetos são polígonos. Cada polígono é descrito pelos seus vértices (vetores no espaço vetorial R^3), pela equação do plano que o contém, pelo seu envelope retangular e pela esfera limitante (aquela de menor raio na qual o polígono esteja contido), além das características de reação aos efeitos da luz [CUN86].

A síntese da cena é realizada simulando o efeito dos raios de luz que atingem os objetos. Um raio é definido como uma reta em R^3 , tendo um vetor direção e uma origem. Quando da intersecção de um raio com um polígono, são gerados um ou dois raios (refletido e transmitido). A definição das direções destes raios pode ser encontrada em [HEC84] e [YAY72].

O modelo de iluminação utilizado baseia-se no proposto por Phong [PHO82], considerando-se os efeitos da luz refletida e transmitida, conforme proposto por Whitted [WHI80] e revisto em [ROG85].

3. Descrição dos Algoritmos.

3.1 Algoritmo A.

O algoritmo A é o mais simples dos dois implementados. Basicamente, para cada ponto do vídeo que pertença a um polígono, um raio orientado do observador ao ponto é disparado.

Este raio, ao incidir sobre o polígono, gera um raio refletido e, dependendo das características do polígono, gera também um raio transmitido [YAV72]. Cada raio, por sua vez, é testado contra todos os polígonos da cena à procura de alguma intersecção. Se esta ocorrer, são gerados novamente um ou dois raios, e este processo é repetido, obtendo-se uma árvore de raios refletidos e transmitidos (ART), até que não ocorram mais intersecções ou que o nível máximo de profundidade da ART seja excedido. Este nível máximo é especificado quando da modelagem da cena [CUN 86].

Cada nodo da ART corresponde a um raio gerado pela intersecção de um raio incidente a um polígono. Sempre que ocorre uma intersecção, a cor do ponto de intersecção é calculada, segundo o modelo de iluminação citado anteriormente, e esta cor é armazenada no nodo. Este cálculo leva em consideração a atenuação da luz que incide sobre o ponto devido às sombras causadas por outros polígonos presentes na cena.

Ao término da verificação das intersecções dos raios com os polígonos, existe uma ART que será percorrida visitando-se primeiro os nodos filhos, de forma a se obter a contribuição de luz que um ponto recebe das direções dos raios refletido e transmitido (se houver). Ao final, como o ponto para o qual foi disparado o raio está na raiz da ART, a cor deste terá recebido as contribuições devidas à todas as intersecções ocorridas. Esta cor é aquela a ser exibida para o referido ponto. O processo é repetido para todos os pontos de todos os polígonos presentes na cena.

As seções seguintes detalharão os aspectos referentes à criação dos raios incidentes, intersecção dos raios com os polígonos, cálculo da cor do ponto e obtenção da contribuição da luz nas direções refletida e transmitida.

3.1.1 A criação dos raios incidentes.

A maneira mais simples de se criar os raios incidentes seria, para cada ponto do vídeo, disparar um raio em sua direção, a partir do observador. Isso, no entanto, leva a uma taxa elevada de raios disparados porém não utilizáveis (por não interceptarem qualquer objeto), aumentando consideravelmente o tempo de execução do algoritmo, pois implica em se testar este raio contra todos os objetos da cena (ou alguns, se se empregar uma descrição posicional dos mesmos, como "quadrees", "octrees" e outras) à procura de intersecções [GLA 84].

Devido ao problema acima, são disparados unicamente raios sobre pontos do vídeo que pertençam a algum polígono. Para cada linha Y do vídeo, localizam-se os polígonos cujos envelopes a contenham. Considerando-se cada um destes polígonos, para cada coordenada X de seus envelopes, verifica-se se o ponto (X,Y) não teve algum raio disparado anteriormente para ele e se o ponto pertence ao polígono. Se ambas as condições forem satisfeitas, então um raio será disparado sobre este ponto.

Os polígonos estão organizados em ordem crescente da sua distância ao observador. Por esta razão, não é necessário preocupar-se com a situação de mais de um polígono conter o ponto. O primeiro polígono encontrado, por estar menos distante do observador, oculta os demais polígonos que porventura contenham o ponto, e portanto este é que deve ser considerado para o disparo do raio sobre o ponto.

3.1.2 A intersecção dos polígonos com os raios.

A medida que ocorrem as intersecções dos raios com os polígonos, a árvore de raios refletidos e transmitidos (ART) vai sendo formada. Cada intersecção de um raio com um polígono gera um ou dois raios. O raio refletido (R) será colocado à esquerda do nodo do raio interceptante, e o raio transmitido (T), se houver, será colocado à direita. A fig. 1 ilustra um raio interceptando um objeto, gerando outros raios que por sua vez interceptam outros objetos, e a ART correspondente.

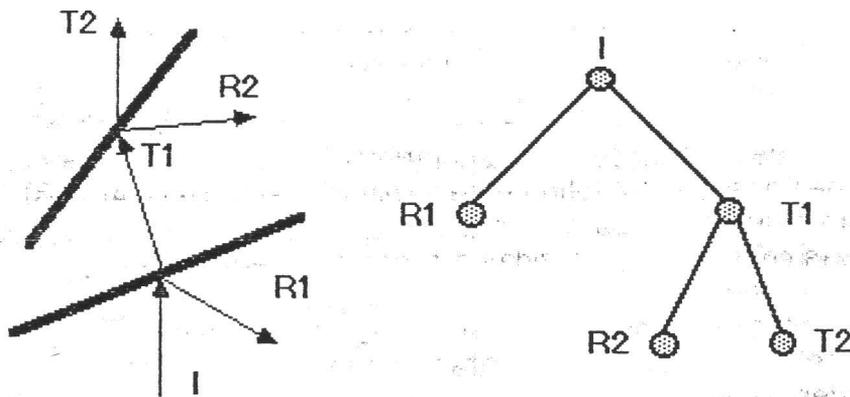


Fig. 1

A cada novo raio gerado, deve-se procurar o polígono interceptado por ele. Como ao se disparar o raio sobre o ponto já se conhece a qual polígono ele pertence, esta pesquisa não é efetuada para este raio, mas apenas para os demais raios que forem gerados quando das sucessivas intersecções.

Uma das características do processo de construção da ART é que, para um determinado raio, somente aqueles que estiverem em sua direção tem a possibilidade de serem interceptados. Todos os demais polígonos que não satisfizerem esta condição podem ser desconsiderados do processo de localização das intersecções.

O estabelecimento de critérios de selecção de alguns dos polígonos presentes na cena para serem testados na localização das intersecções reduz o tempo de execução do algoritmo. Os seguintes critérios de selecção foram adotados:

a. Não-parallelismo entre o raio e o polígono: este teste verifica se o ângulo formado entre o vetor normal do plano que contém o polígono e o vetor direcção do raio é diferente de zero. Se for, então o raio possivelmente intercepta o polígono;

b. Posição relativa entre o raio e o polígono: este teste verifica se o polígono encontra-se à frente do raio. Para isso, considera-se um plano que contém o ponto de origem do raio e cujo vetor normal é o vetor direcção do raio. Se a equação do plano, aplicada sobre o centro da esfera limitante do polígono, gerar um valor positivo, então o polígono encontra-se à frente do raio;

c. Intersecção do raio com a esfera limitante: este teste permite-nos verificar se, dentre aqueles polígonos que se encontram à frente do raio, qual (ou quais) tem mais probabilidade de ser interceptado pelo raio.

d. Contenção do ponto interceptado: neste teste, é verificado se o ponto interceptado pertence ao polígono.

Quando um polígono satisfaz a todos estes testes, verifica-se se a distância entre o ponto de origem do raio e o ponto resultado da intersecção do raio com o polígono é menor do que a encontrada para um polígono que anteriormente tenha satisfeito aos testes. Se aquela for menor, então aquele é o polígono que deve ser considerado para a construção da ART. Quando todos os polígonos tiverem sido testados, se tiver ocorrido alguma intersecção, o polígono mais próximo do ponto de origem do raio é aquele que efetivamente deve ser considerado como interceptado.

A ordem na qual os testes são aplicados é aquela na qual foram apresentados. Esta ordem foi estabelecida de forma a seleccionar progressivamente aqueles polígonos que satisfaçam aos mesmos. Além disso, não deve-se gastar tempo de execução com testes de maior grau de abrangência, de vez que muitos polígonos poderão satisfazer ao teste, sem com isto assegurar efetivamente qual destes é aquele interceptado pelo raio. O teste **d** é aquele que efetua maior número de operações de ponto flutuante, sendo este número proporcional ao dobro do número de vértices do polígono.

3.1.3 Cálculo da cor do ponto

O cálculo da cor do ponto utiliza o modelo de iluminação mencionado na secção 2. Podem existir no máximo dez fontes de luz na cena, e cada uma destas pode ser do tipo pontual, flap ou spot [CUN86]. Para cada ponto interceptado, deve-se verificar quais fontes de luz contribuem para a iluminação do mesmo. Fontes de luz do tipo pontual podem difundir sua luz em todas as direcções, contribuindo para a iluminação de todos os pontos interceptados. As fontes de luz do tipo spot e flap somente irão iluminar aqueles pontos que estejam dentro da área iluminada pelo seu fecho. As fontes de luz que contribuem para a iluminação de um ponto formam o conjunto efetivo de fontes de luz (CEFL) para aquele ponto.

Para cada uma das fontes de luz neste conjunto, é calculado o vetor Z que une o ponto de origem do raio à posição da fonte de luz. A determinação das sombras em relação a este ponto é efetuada verificando-se se o vetor Z de cada fonte de luz do CEFL intercepta algum polígono. Se interceptar, então a contribuição desta fonte de luz é atenuada pelo fator de transmissividade do polígono interceptado. A localização dos polígonos interceptados é efetuada de maneira semelhante à exposta na seção 3.1.2. O fator de atenuação de uma fonte de luz é o produto dos fatores de transmissividade de todos os polígonos interceptados pelo vetor Z .

3.1.4 Contribuição da luz nas direções refletida e transmitida

Após a ART estar construída, em cada um de seus nodos está calculada a cor do ponto de intersecção. A ART é então percorrida em pós-ordem de forma a se completar o modelo de iluminação com as contribuições da luz nas direções dos raios refletidos e transmitidos - os termos $k_S S$ e $k_T T$ do modelo de iluminação [ROG85].

Ao se visitar um nodo, verifica-se a existência dos raios refletido e transmitido. A cor associada aos nodos dos referidos raios será atenuada pela distância existente entre o ponto de origem destes raios e os pontos de intersecção dos mesmos raios com outros polígonos da cena. A fig. 2 ilustra este processo.

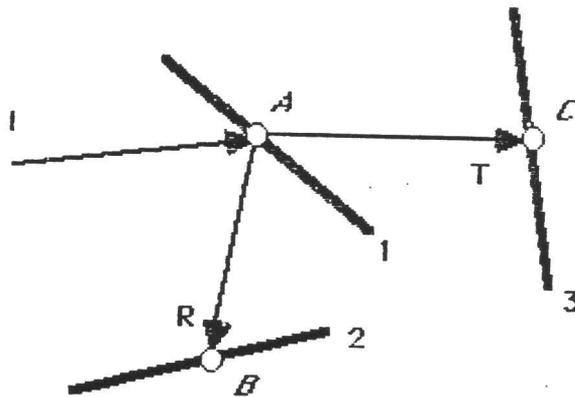


Fig. 2

O raio I , incidindo sobre o polígono 1 no ponto A , gera os raios R e T , os quais interceptam os polígonos 2 e 3 nos pontos B e C . Chamemos de C_A , C_B e C_C as cores dos pontos A , B e C . Chamemos D_{AB} e D_{AC} as distâncias entre os pontos A e B e entre A e C . A cor C_A será acrescida das contribuições das cores nas direções refletida e transmitida como segue :

$$C_A = C_A + k_{SA} C_B / D_{AB} + k_{TA} C_C / D_{AC} ,$$

onde k_{SA} e k_{TA} são os coeficientes de especularidade e transmissividade do polígono 1.

4.2 Algoritmo B.

O algoritmo B é uma evolução do algoritmo A, diferenciando-se deste quanto à maneira de criação dos raios incidentes. Este algoritmo vale-se da propriedade de coerência de imagem.

A propriedade de coerência de imagem nos diz que, se um ponto tem uma determinada cor, muito provavelmente os pontos situados na sua vizinhança terão a mesma cor ou tons desta. No algoritmo A, aproximadamente 80% do tempo de execução, em média, é gasto no processo de cálculo das intersecções, o qual é iniciado pela incidência de um raio sobre um ponto. Nos testes realizados, detectou-se que, muitas vezes, as cores de dois pontos contíguos (para os quais foram disparados dois raios), apresentam diferenças da ordem de 10^{-3} a 10^{-5} . Esta diferença, além de ratificar a validade desta propriedade, permite-nos reduzir o tempo de execução do algoritmo A disparando um menor número de raios.

O algoritmo B inicia subdividindo as linhas do vídeo em n conjuntos. Para a primeira linha de cada conjunto, localizam-se os polígonos cujos envelopes a contêm. Considerando-se cada um destes polígonos, o intervalo ao longo do eixo X dos seus envelopes é igualmente subdividido em n conjuntos. Com isso, forma-se uma malha de $n \times n$ pontos, e somente serão disparados raios sobre os pontos localizados nos quatro vértices da malha (fig. 3).

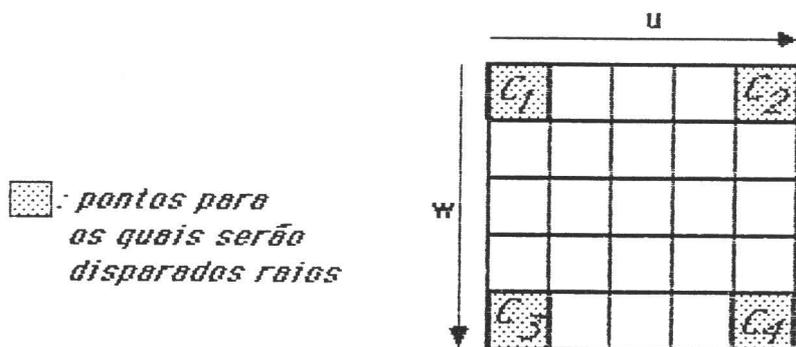


Fig. 3

As cores (C_1 , C_2 , C_3 e C_4) dos pontos sobre os quais os raios são disparados são calculadas da mesma forma que no algoritmo A. Nos restantes pontos da malha, as cores são calculadas através de uma aproximação linear das quatro cores dos vértices da malha. Foi escolhido este tipo de aproximação de forma a não onerar o tempo de execução.

Esta aproximação é feita utilizando-se dois parâmetros, u e w . Os valores destes estão limitados no intervalo $[0,1]$. A expressão que descreve a distribuição de cor na malha em função dos parâmetros u e w é

$$(1-u)[(1-w)C_1 + wC_3] + u[(1-w)C_2 + wC_4], \quad 0 \leq u \leq 1, \quad 0 \leq w \leq 1.$$

A tabela 1 permite comparar as diferenças existentes nas cores obtidas através dos algoritmos A e B, para uma mesma cena. As diferenças percentuais são das cores obtidas pelo algoritmo A em relação àquelas calculadas pelo algoritmo B.

Cores Geradas		Diferenças (%)
Algoritmo A	Algoritmo B	
0.29674	0.29674	0.0
0.30375	0.30375	0.0
0.29830	0.29830	0.0
0.30172	0.30172	0.0
0.29899	0.29911	0.04014
0.30063	0.30075	0.03992
0.30232	0.30240	0.02646
0.29739	0.29746	0.02354

Tabela 1

5. Comparação do desempenho dos algoritmos.

Para a comparação do desempenho entre os dois algoritmos, foram sintetizadas algumas cenas. As imagens correspondentes foram preparadas para exibição em um monitor gráfico Mitsubishi, conectado a uma placa ARTIST, com resolução de 1024x768 pontos.

Os algoritmos foram implementados na linguagem Pascal e executados no computador Burroughs A10F da UFRGS.

De forma a se verificar o ganho de tempo que o algoritmo B apresenta sobre o algoritmo A, foi sintetizada uma cena com 648 polígonos e com duas fontes de luz, uma do tipo pontual e a outra do tipo spot. Os tempos de execução de cada uma das fases do algoritmo são apresentados na tabela 2.

A tabela 3 apresenta os tempos de execução obtidos para diversas cenas, onde o número de polígonos e o número de fontes de luz presentes aumentam a complexidade das mesmas. As cenas 4, 7 e 8 ilustram as diferenças obtidas no tempo de execução de ambos os algoritmos para um mesmo número de polígonos, porém com diferentes fontes de luz e um nível de profundidade a mais para a ART.

Fases do Algoritmo	Tempos de Execução (s)	
	Algoritmo A	Algoritmo B
Disparo dos Raios	1.998,55	835,78
Intersecções , Cálculo parcial da cor e Sombras	1.608,15	388,56
Contribuição da cor nas direções refletida e transmitida	16,42	3,22
Total dos tempos de execução	3.623,12	1.227,56

Tabela 2

Cena	Nº de Polígonos	Nível da ART	Fontes de Luz (tipo/nº)	Algoritmo A		Algoritmo B	
				Nº de Raios	T(s)	Nº de Raios	T(s)
1	91	2	pontual - 1 spot - 1	292.483	2.557,10	73.121	858,09
2	102	2	pontual - 1 spot - 1	247.570	2.372,56	61.893	765,34
3	88	2	pontual - 1 spot - 1	61.094	637,77	15.274	219,92
4	648	2	pontual - 1 spot - 1	38.506	3.623,54	9.627	1.228,32
5	5	2	pontual - 1 spot - 1	189.727	1.289,40	47.432	415,94
6	10	2	pontual - 1 spot - 1	237.923	1.609,03	59.481	519,04
7	648	2	pontual - 3	38.506	3.230,45	9.627	1.113,95
8	648	3	pontual - 3	38.506	6.752,31	9.627	2.328,38

Tabela 3

6. Conclusão.

A análise dos dois algoritmos mostra os seguintes resultados :

- a. O algoritmo B apresenta uma economia de $n^2 - 4$ raios disparados, onde n é o tamanho da malha ;
- b. O tempo de execução do algoritmo B é $\sim(n-1)$ vezes menor que o do algoritmo A.

Apesar das vantagens que o algoritmo B apresenta sobre o algoritmo A, a malha utilizada não pode ser muito grande, de forma a impedir que a cena gerada por este algoritmo seja muito diferente daquela gerada pelo algoritmo A. O tamanho da malha depende da resolução do vídeo no qual será exibida a imagem sintetizada da cena e da complexidade desta última. Tamanhos usuais da malha estão entre 4 e 8.

Além dos algoritmos apresentados, já se encontram em fase de testes mais dois novos algoritmos ("C" e "D"), os quais terão os polígonos presentes na cena armazenados em uma "octree". Acredita-se que, para determinados tipos de cenas, estes algoritmos demandarão um menor tempo.

7. Bibliografia.

[CLE86] Cleary, J.G., Wywill, B.M., Birtwistle, G.M., Vatti, R., "Multiprocessor Ray Tracing", Computer Graphics Forum 5:3-12, North-Holland

[COQ85] Coquillart, S., "An Improvement of the Ray-Tracing Algorithm". In: Vandoni, C.E. (Ed.). EuroGraphics '85, North-Holland, 1985

[CUN86] da Cunha, R.D., "Um Modelador para a Síntese de Cenas através do Método Ray-Tracing", Porto Alegre, DI/CPD-UFRGS, 1986. (Pelatório parcial)

[CUN87] da Cunha, R.D., Freitas, C.M.D.S., "Síntese de Cenas através do Método Ray-Tracing". In: Simpósio Brasileiro de Banco de Dados, II^o, Porto Alegre, 7-8 Maio 1987, Anais UFRGS/SBC, Porto Alegre, 1987. pp.154-5

[FUJ86] Fujimoto, A., Takayuki, T., Iwata, K., "ARTS : Accelerated Ray-Tracing System", IEEE Computer Graphics and Applications, 6(4):16-26, 1986

[GLA84] Glassner, A. S., "Space Subdivision for Fast Ray Tracing", IEEE Computer Graphics and Applications, 4(10):15-22, 1984

[HEC84] Heckberth, P.S., Hanrahan, T., "Beam Tracing Polygonal Objects", ACM Computer Graphics 18: 137-45

[PHO82] Bui-Tuong, Phong, "Illumination for Computer Generated Images", CACM **18**:311-7, 1975

[PLU85] Plunkett, D.J., Bailey, M.J., "The Vectorization of a Ray-Tracing Algorithm for Improved Execution Speed", IEEE Computer Graphics and Applications, **5**(8):52-60, 1985

[ROG85] Rogers, D.F., Procedural Elements for Computer Graphics, McGraw-Hill, New York, 1985

[WHI80] Whitted, T., "An Improved Illumination Model for Shaded Display", CACM **23**:343-9, 1980

[YAV72], Yavorsky, B., Detlaf, I., Handbook of Physics, Mir, Moskva, 1972