

UM AMBIENTE PARA SÍNTESE DE IMAGENS REALÍSTICAS BASEADO NA  
METODOLOGIA RAY-TRACING

ARI DE MOURA VILLAÇA

MESTRANDO EM ENGENHARIA ELÉTRICA

FACULDADE DE ENGENHARIA ELÉTRICA DA UNICAMP (FEE/UNICAMP)

DCA/FEE CAIXA POSTAL 6101

13100 - CAMPINAS - S. P.

SUMÁRIO

ESTE TRABALHO APRESENTA O AMBIENTE PARA SÍNTESE DE IMAGENS REALÍSTICAS DESENVOLVIDO NO LABORATÓRIO DE COMPUTAÇÃO GRÁFICA DA FEE/UNICAMP UTILIZANDO-SE A METODOLOGIA RAY-TRACING.

1- INTRODUÇÃO

Nos últimos anos o método ray-tracing tem sido intensamente pesquisado com o objetivo de sintetizar imagens realísticas em computador.

Este método caracteriza-se pela sua simplicidade conceitual, flexibilidade quanto aos tipos de modelos que podem ser utilizados para descrever uma cena e sua capacidade de simulação de transparências, reflexões especulares e sombras.

O objetivo deste trabalho é apresentar o ambiente para síntese de imagens criado no Laboratório de Computação Gráfica da FEE/UNICAMP com propósitos de pesquisa.

A metodologia ray-tracing é conceituada na seção 2. Na seção 3, o ambiente para síntese de imagens realísticas é discutido, e suas ferramentas de apoio são apresentadas. Finalmente, na seção 4, apresentam-se as conclusões e comentam-se algumas técnicas que permitirão aumentar o número de efeitos visuais alcançados com este ambiente e melhorar o seu desempenho.

2- O MÉTODO RAY-TRACING

Goldstein e Nagel [1] foram os primeiros a utilizar o

método ray-tracing para sintetizar imagens em computador, gerando objetos opacos em tons de cinza. Whitted [2] estendeu com sucesso o método permitindo a visualização de objetos transparentes, especulares e o cálculo de sombras.

O conceito de ray-tracing é bastante simples. Um raio de luz parte do olho do observador na direção da cena. A partir do ponto onde o raio intersecta a superfície mais próxima do observador novos raios de luz são produzidos devido a reflexão especular e transmissão de luz. Estes raios também intersectam outras superfícies da cena gerando uma árvore binária com interseções entre raios e superfícies. A intensidade de luz que chega ao olho do observador é computada percorrendo-se a árvore criada, traçando-se em cada nó um raio entre o ponto de interseção e a fonte de luz para o cálculo da intensidade de luz que ilumina o ponto e fazendo-se uma combinação das intensidades de luz calculadas em cada nó da árvore segundo o modelo de iluminação proposto por Whitted em [2]. A figura 1 ilustra o conceito de ray-tracing mostrando um raio percorrendo uma cena e a respectiva árvore de interseções criada.

### 3- O AMBIENTE PARA SÍNTESE DE IMAGENS REALÍSTICAS

Roth [3] estabeleceu um algoritmo ray-tracing para cenas modeladas com Constructive Solid Geometry (CSG) de objetos opacos que tornou-se padrão devido a sua simplicidade, flexibilidade e elegância, podendo ser estendido para simular reflexões especulares, transparências e sombras.

O ambiente básico para síntese de imagens realísticas foi construído segundo a metodologia ray-tracing utilizando o algoritmo de Roth como base e é composto por um conjunto de programas com o objetivo de modelar cenas com CSG, sintetizar imagens realísticas, criar lookup tables para cada imagem e mostrar a imagem produzida no monitor colorido.

A arquitetura do ambiente de produção de imagens é mostrada na figura 2.

O programa responsável pelo modelamento de cenas com CSG foi implementado em linguagem Pascal em ambiente PC sendo similar ao de Roth [3]. Suas funções são criar, transformar, deletar, copiar, combinar e desfazer combinações entre sólidos, salvar e carregar a descrição de um sólido armazenada em disco e

mostrar silhuetas de sólidos na tela. O usuário trabalha interativamente criando instâncias de sólidos primitivos, transformando os sólidos no sistema de coordenadas da cena, combinando e visualizando sólidos, formando, dessa forma, objetos mais complexos. A possibilidade de visualização do sólido sendo modelado permite ao usuário do programa controlar cada etapa de projeto do sólido prevenindo-se de erros.

O programa sintetizador de imagens realísticas (SIR) utiliza as cenas criadas pelo modelador de sólidos mais os arquivos com características de reflexão de luz de cada superfície da cena, posição do observador e das fontes de luz para criar uma imagem colorida com transparências, reflexões e sombras. O SIR simula uma câmera fotográfica onde o observador é o foco e a tela do monitor é o filme. A partir do observador, são lançados raios de luz que passam através de cada pixel da tela e seguem na direção da cena, onde intersectam as superfícies visíveis pelo observador em cada pixel. A imagem criada é armazenada em três arquivos, cada um com uma das três componentes RGB que definem a cor de cada pixel da tela.

SIR foi implementado em linguagem Pascal no ambiente VAX do Centro de Computação da Unicamp contando com aproximadamente 2500 linhas de programa na versão atual. As principais características deste programa são :

a- cenas modeladas em CSG, utilizando os sólidos primitivos cubo, esfera, cone e cilindro e permitindo todas as operações de conjunto definidas (união, diferença e interseção);

b- raios devido a transmissão e reflexão de luz são traçados, permitindo a simulação de transparências e espelhos;

c- raios na direção das fontes de luz são traçados permitindo a simulação de sombras, penumbras e várias fontes de luz do tipo pontual;

d- possibilidade do uso de texturas para a definição de características de iluminação das superfícies da cena;

e- utilização de envoltórios cúbicos [3] para diminuir o esforço computacional do método ray-tracing e diminuir o tempo de execução do programa.

O Laboratório de Computação Gráfica da FEE/UNICAMP conta com o monitor ELTEK-MA120 com 1024 por 780 pixels e lookup table de 256 cores entre 256k cores possíveis.

A escolha da lookup table mais adequada para cada imagem é vital para se conseguir uma imagem de boa qualidade. A escolha das cores relaciona-se com as características das superfícies da cena. Por exemplo, se uma cena contém superfícies amarelas e vermelhas, a lookup table deve ser montada dividindo-se as 256 cores possíveis entre amarelo e vermelho, não utilizando outras cores.

Para criar lookup tables foi desenvolvido um programa que utiliza o modelo HSV de cores [4] e a frequência da cor na imagem. A versão atual do programa trabalha com uma matriz HSV, para contagem da frequência de cores, que representa 36 cores possíveis (dimensão H), 30 saturações (dimensão S) e 30 valores diferentes (dimensão V) por cor podendo produzir 900 tons diferentes de cada cor. O total de 32400 tons diferentes é bem menor que as 256k cores possíveis de serem representadas no monitor mas, partindo da suposição que, num monitor RGB, várias cores com triplas RGB diferentes são indistinguíveis ao olho humano parecendo ser iguais, o mapeamento de 8 cores em 1 (256k em 32k) parece ser suficiente para representar todas as cores possíveis sem perda de qualidade notável nas imagens.

Dada a matriz de frequência HSV, a lookup table é criada da seguinte forma :

- a) o valor de cada pixel da imagem é lido dos arquivos R, G, B que representam a imagem e transformado em valores h, s, v;
- b) os valores h, s, v são utilizados como índices para incrementar a frequência de uma cor na matriz HSV e indicar o número de vezes que a cor aparece na imagem;
- c) calcula-se o número de cores diferentes da imagem verificando na matriz HSV as posições com frequência diferente de zero e o número de tons diferentes em cada cor (dimensão H) da matriz HSV;
- d) calcula-se a porcentagem de utilização de cada cor na matriz HSV dividindo-se o número de tons diferentes da cor pelo número total de cores diferentes;
- e) multiplica-se a porcentagem de utilização de cada cor por 256 (número de posições da lookup table) calculando-se o número de posições da lookup table que devem ser inicializadas com tons dessa cor;
- f) percorre-se a matriz SV de cada cor escolhendo os tons

que apareçam com maior frequência para preencher a lookup table.

Este algoritmo gera dois arquivos relativos a lookup table: um com triplas RGB e outro com triplas HSV que identificam a mesma cor em dois sistemas de coordenadas diferentes.

Uma imagem criada por SIR é mostrada no monitor colorido da seguinte forma :

a) a lookup table do monitor é inicializada com a lookup table RGB para a imagem;

b) as componentes RGB de cada pixel da imagem são lidas e transformadas em valores HSV; pesquisa-se o valor HSV mais próximo na lookup table HSV; o índice desse valor é armazenado na memória de vídeo do monitor e identifica a cor mostrada pelo pixel na tela.

#### 4- CONCLUSÕES E COMENTÁRIOS FINAIS

Este trabalho apresenta o ambiente de produção de imagens desenvolvido no Laboratório de Computação Gráfica da FEE/UNICAMP cujo objetivo é criar condições de pesquisa fornecendo um sistema básico de produção de imagens realísticas que possa ser modificado com relativa facilidade.

As imagens produzidas neste ambiente revelam um grande grau de realismo. Porém, a modelagem com CSG restringe bastante o universo de cenas que podem ser descritas tornando praticamente impossível modelar certos tipos de objetos como carros, aviões, árvores, etc. Portanto, espera-se que novos métodos de modelagem de cenas sejam acrescentados ao ambiente.

A simulação de superfícies com rugosidades será o próximo passo no desenvolvimento do SIR com a utilização da técnica de perturbação da normal [5] o que permitirá a modelagem de pequenos detalhes nas superfícies da cena.

Atualmente não é feito nenhum tratamento anti-aliasing o qual também deve ser implementado em breve permitindo eliminar os pequenos defeitos notados principalmente nas bordas das superfícies da cena.

O algoritmo ray-tracing é muito lento devido ao grande gasto computacional que ocorre no cálculo de interseções entre raio e superfícies. Apesar de utilizarmos envoltórios cúbicos no programa SIR parece ser interessante utilizar a técnica de subdivisão 3D [6] para diminuir o tempo de execução do programa.

## AGRADECIMENTOS

A Celso Hiroshi Yamaguchi, aluno de graduação da FEE/UNICAMP e a Hélio Azevedo, do CTI, pela implementação do software básico que controla o monitor colorido ELTEK-MA120,

A Helga Hartelt, doutoranda da FEE, pelas discussões sobre como trabalhar com cores e

ao meu orientador Dr. Clesio Luiz Tozzi pelo apoio no desenvolvimento deste projeto que é o esforço principal de minha teste de mestrado.

Sem a ajuda destas pessoas jamais conseguiria concluir este trabalho.

## REFERÊNCIAS BIBLIOGRÁFICAS

- [1] Goldstein, R.A. & Nagel R. 3-D visual simulation. Simulation, 16(1):25-31, jan., 1971.
- [2] Whitted, T. An improved model for shaded display. Communications ACM, 23(6):343-49, jun., 1980.
- [3] Roth, S.D. Ray casting for modeling solids. Computer Graphics and Image Processing, 18(2):79-144, fev., 1982.
- [4] Foley, J.D., Van Dam, A. Fundamentals of interactive computer graphics. Reading, Mass., Addison-Wesley, 1984.
- [5] Blinn, J.F. Simulation of wrinkled surfaces. Computer Graphics, 12(3):282-92, ago., 1978.
- [6] Bouatouch, K. et al A new algorithm of space tracing using a CSG model. Eurographics'87, Amsterdam, ago. 24-28, 1978, p.65-77.

Este trabalho foi financiado pelo convênio UNICAMP/CCPG/CAPG/CAPES/DS-192/82 Proc. 10039/87.

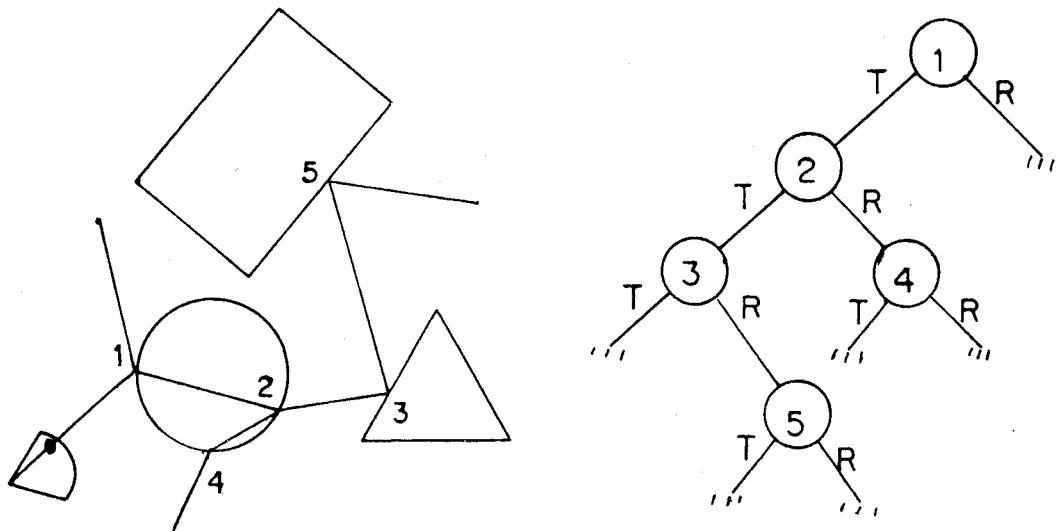


FIG. 1 - Caminho de um raio e sua árvore de interseções.

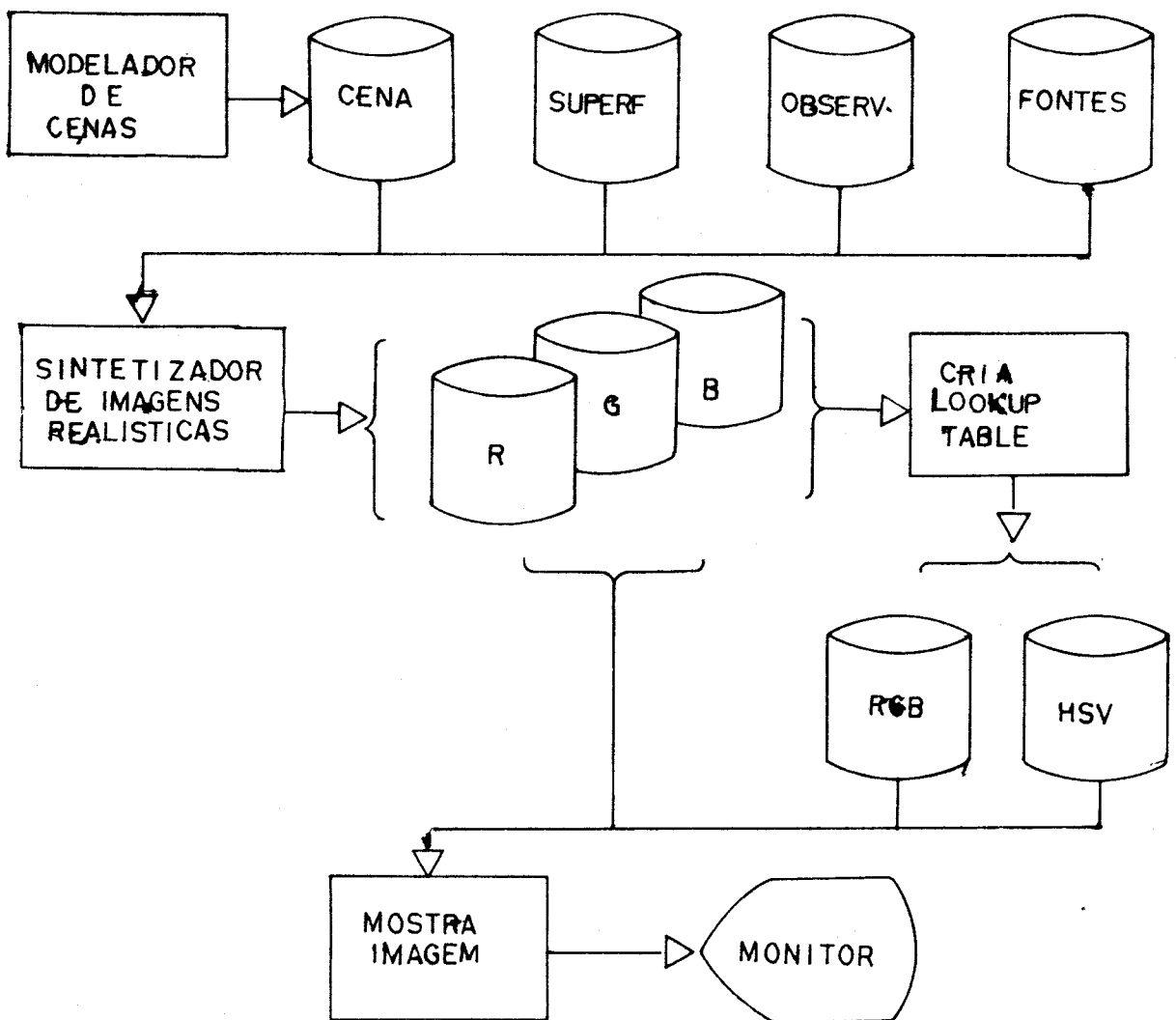


FIG. 2 - Arquitetura do ambiente.