

Um rastreador de raios para micros "PC".

**Luiz Antonio M. Pereira
Rubens Nascimento Melo**

Departamento de Informática - PUC/RJ

Sumário

Esse trabalho aborda os principais aspectos da implementação de um rastreador de raios (ray tracer) em um microcomputador do tipo "PC" visando a produção de imagens realistas de objetos tridimensionais.

Abstract

This work covers the principal aspects of the implementation of a ray tracer on a "PC" microcomputer regarding the production of realistic images of tridimensional objects.

Introdução.

A produção, por computador, de imagens de formas tridimensionais tem sido objeto de estudos desde os primórdios dos computadores digitais. O aperfeiçoamento das técnicas de produção das imagens e o estágio atingido pela eletrônica nos anos 70, mais especificamente com o advento de dispositivos de exibição do tipo raster de maior resolução e elementos de memória (chips) cada vez mais capazes e baratos, tornou possível a produção de imagens com altos graus de realismo e custos de produção reduzidos. O interesse despertado pelo assunto, o esforço em pesquisas, tanto de novas técnicas quanto de hardware, provocou, como efeito secundário, a criação de uma nova especialidade da Computação Gráfica: a Fotografia Sintética ou Imagem Digital, como preferem alguns. As imagens assim criadas têm sido empregadas em cinema, publicidade e em simuladores com fins científicos e militares, substituindo com grande vantagem os métodos tradicionais.

Uma das áreas de pesquisa no Departamento de Informática da PUC/RJ é a Computação Gráfica Interativa e suas aplicações. Em particular um dos tópicos de estudo recente é a síntese de imagens realistas (ou fotografia sintética). Em um trabalho anterior, o método de mapeamento de textura para a produção de imagens foi estudado ([ALVAR84] e [ALVAR85]). Mais recentemente, o método de rastreamento de raios foi estudado e um protótipo implementado ([PERE187]).

O método de rastreamento de raios (ray tracing, em inglês) é conhecido por sua flexibilidade, precisão e adequação à modelagem de efeitos de reflexo e refração.

Nesse artigo apresentaremos as características do método de rastreamento de raios, abordaremos os principais tópicos da implementação do método, descrevendo o ambiente (hardware e

software) utilizado nas pesquisas e, ao final, apresentaremos alguns resultados práticos de nosso trabalho. Maiores detalhes sobre o método podem ser encontrados em [PEREI87], [FOLEY83] e [NEUMA79].

Características do Método.

O método de rastreamento de raios baseia-se no acompanhamento, através de equações matemáticas que modelam leis da Física, das trajetórias dos raios visuais em suas interações com os objetos que compõem a cena. A figura 1, adiante, ilustra a aplicação do método para cálculo das cores de dois pontos (pixels) do dispositivo de exibição. O método foi escolhido por sua capacidade de modelar bastante precisamente os efeitos dos raios visuais nos seus caminhos desde a(s) fonte(s) de luz até o olho do observador, ou seja, a reflexão nos objetos refletores, a refração quando atravessam objetos transparentes, o brilho e a sombra suave (degradê) produzida ao longo das superfícies dos objetos.

Ambiente Usado.

Para a criação dos arquivos contendo as informações das imagens utilizamos um microcomputador PC modelo AT, de 8 MHz, com um processador numérico 80287. Os arquivos foram transportados para outro PC AT equipado com a placa gráfica TARGA 16 (512 X 400 pontos a 32768 cores), da AT&T, onde as imagens foram produzidas e fotografadas. Os programas foram desenvolvidos em Turbo Pascal sob o MS-DOS.

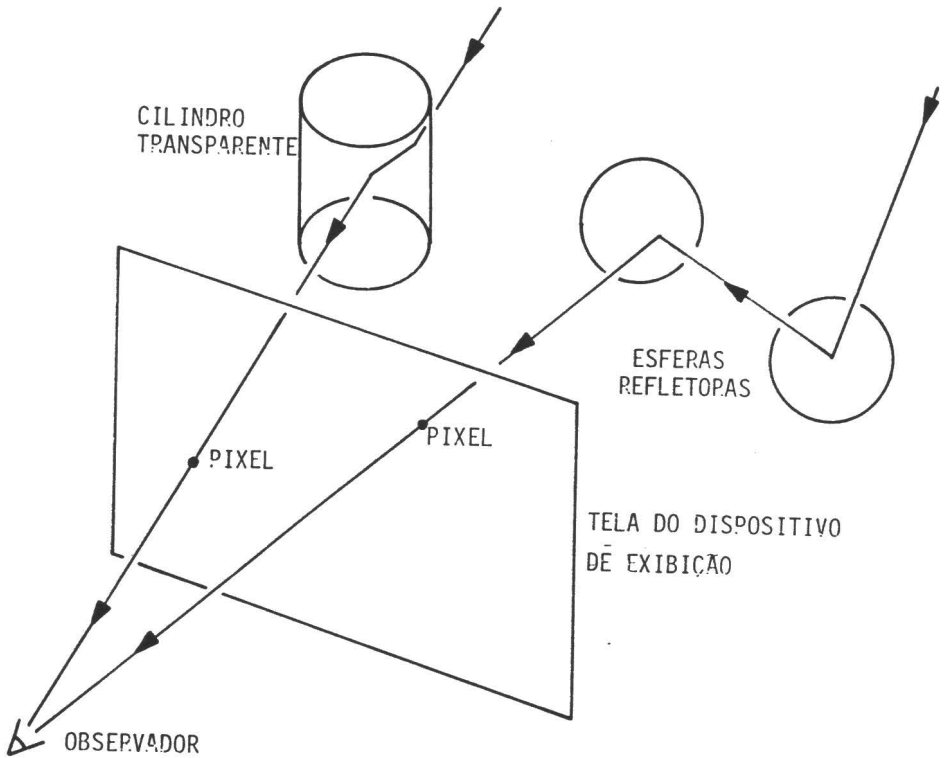


Fig. 1 - Esquema ilustrativo do rastreamento de dois raios visuais.

Aspectos da Implementação.

O diagrama da figura 2 ilustra a arquitetura do sistema. O único programa ainda não disponível é o modelador geométrico (MG). O arquivo que contém os dados da geometria da cena deve ser, portanto, criado e modificado via editor de textos.

O esquema de representação usado para modelar os objetos da cena é um esquema híbrido GSC (geometria sólida construtiva) e por fronteiras (b-rep) e as primitivas disponíveis são: paralelogramos, triângulos, círculos, anéis de círculo, esferas e cones/cilindros. Os objetos da cena são construídos, partindo-se deste conjunto de primitivas, via

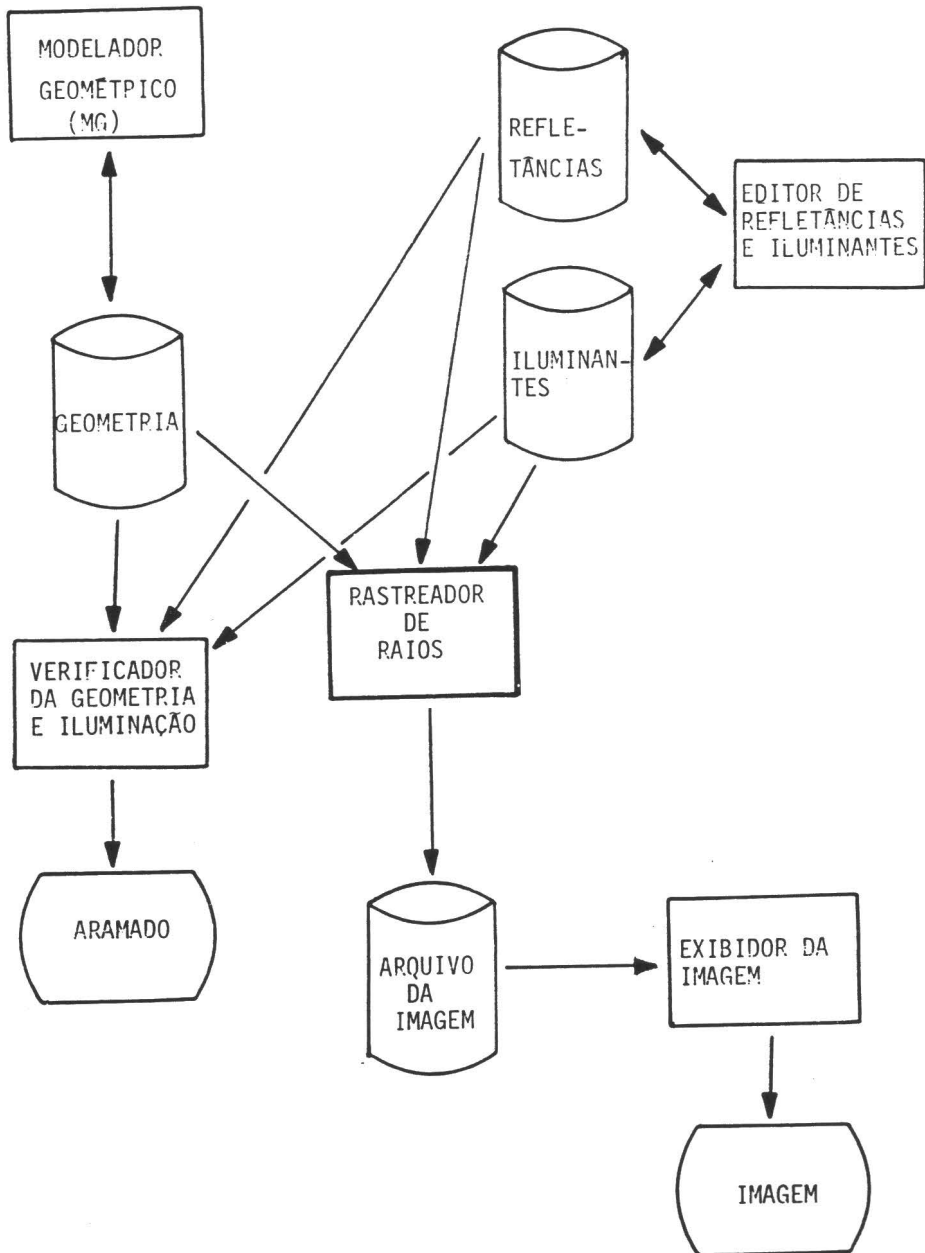


Fig. 2 - Arquitetura do sistema.

operações de adição, exclusivamente.

A figura 3 apresenta o pseudo-código simplificado do programa de rastreamento de raios.

Foram desenvolvidas expressões matemáticas que calculam as coordenadas dos pontos de interseção das visadas (retas) com cada uma das primitivas disponíveis. Essas expressões foram operadas ao máximo e os valores das sub-expressões invariantes são pré-calculados e armazenados na estrutura de dados.

Afim de diminuir o número de cálculos de interseções, foi adotado um esquema de coerência de objetos: cada primitiva é inscrita em uma esfera de centro e raio pré-calculados, sendo esses valores também armazenados na estrutura de dados. Ao se rastrear um determinado raio visual, as distâncias de sua reta suporte aos centros dessas esferas são comparadas com os respectivos raios. Os cálculos das interseções devem, então, ser executados apenas para as primitivas cujas distâncias à reta suporte do raio visual forem menores que os raios das esferas envolventes. Verificamos mais tarde que esse processo passou a responder por parte significativa do tempo total gasto nos cálculos das interseções. Isso acontece devido ao fato de que, apesar de simples, o cálculo é feito para todas as primitivas e para todos os pixels da tela, enquanto que as interseções com primitivas (processos muito mais complexos e caros) só são determinadas em alguns casos. Procedendo-se dessa maneira o tempo total de cálculo das interseções varia bastante com a complexidade da imagem. Embora não se tenha tomado a iniciativa de comparar os tempos de processamento, acredita-se que, dessa forma, o tempo seja bem inferior ao tempo necessário caso não se adotasse esquema algum de coerência.

Início (Ray tracing)

Solicitar as dimensões da janela;
 Solicitar as coordenadas do S0 em relação ao SG;
 Solicitar a distância do ponto observador à janela;
 Solicitar ângulos de visada;
 Calcular a matriz de transf. de coord. de SG para S0;
 Calcular a matriz de S0 para SG por inversão simples, ;
 Calcular as coordenadas globais do ponto observador;
 Solicitar as coordenadas globais da fonte de iluminação;
 Solicitar os atributos dos elementos da cena;

Para cada pixel da janela fazer

Calcular as coordenadas, segundo o S0, do pixel;
 Determinar as coord., segundo o SG, do pixel;
 Interceptar a visada com todos os elementos da cena;
 Eliminar os pontos que estão aquém do plano projetante;
Se ainda existe algum ponto de interseção
 Determinar o ponto mais próximo do observador;
 Se o pto. é iluminado calcular iluminação da luz direta;
 Adicionar a iluminação ambiente

Fim-se

Senão fazer a cor do ponto := Cor_do_Fundo;

Armazenar a cor do ponto

Fim-paraFim. (Ray tracing)

Fig. 3 - Pseudo-código do programa de rastreamento de raios.

O anti-aliasing, processo que visa suavizar a transição entre cores diferentes dando, inclusive, a sensação de que o dispositivo de exibição possui maior resolução, é executado baseado na coerência da imagem. Para o cálculo da cor de um pixel da tela (representado pelo ponto central da figura 4) devemos já ter calculados os 16 pontos (não pixels) circunvizinhos. Caso os pontos 6, 7, 10 e 11 não correspondam ao mesmo objeto, um décimo sétimo ponto (coincidente com o pixel em estudo) também deverá ser calculado. A cor do pixel é o resultado de médias ponderadas entre as cores de todos os pontos, calculadas para cada cor primária R, G e B independentemente. Nossa experiência mostrou que se considerarmos, apenas, os quatro pontos imediatamente circunvizinhos (pontos 6, 7, 10 e 11 da fig. 4) não chegamos a resultados satisfatórios enquanto que se considerarmos mais do que dois níveis de pontos não obtemos diferenças apreciáveis (ao ponto de começarmos a degradar a imagem).

O método de cálculo da iluminação adotado foi desenvolvido por Robert Cook e Kenneth Torrance e descrito em [COOK81]. Esse modelo, baseado no modelo de Bui Tuong Phong ([PHONG75]), considera que a luz que chega aos olhos do observador é formada por três componentes resultantes da reflexão da luz ambiente, da luz difusa e da reflexão especular. Cada uma dessas componentes é, por sua vez, a luz incidente (ambiente ou direta) atenuada pelos valores das refletâncias dos materiais dos objetos. Essa forma de abordar o problema exige que se descreva as superfícies dos objetos da cena através da informação de suas propriedades físicas (que podem já estar armazenadas em arquivo). As fontes de luz têm suas características (intensidades e cores) especificadas por suas distribuições espectrais de energia. Como os cálculos da iluminação são feitos no sistema XYZ (espaço perceptual de

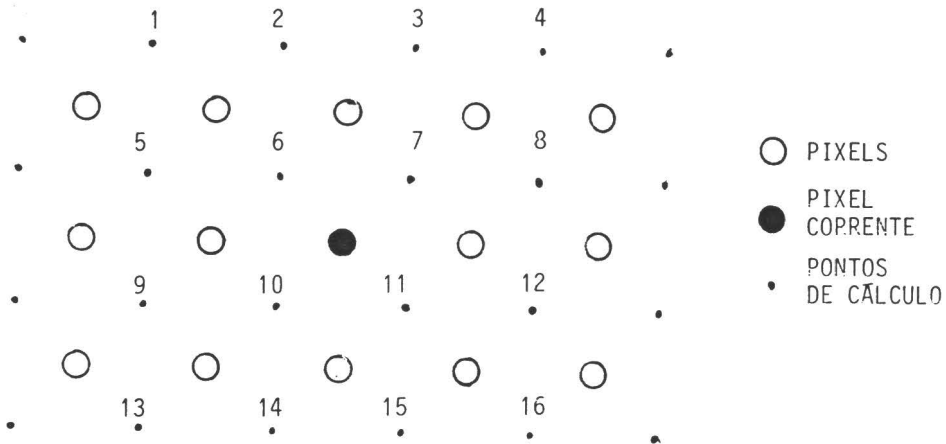


Fig. 4 - Disposição dos raios a serem rastreados para o cálculo da cor de um ponto visando a obtenção da imagem com anti-aliasing.

cores baseado nos tri-estímulos - vide [PEREIR87]), a conversão XYZ→RGB deve ser efetuada ao final, através da multiplicação do vetor XYZ por uma matriz 3x3, cujos coeficientes foram determinados experimentalmente e que variam de monitor para monitor. A fase de ajuste desse modelo (determinação dos coeficientes da matriz XYZ→RGB e de outras constantes) foi, sem dúvida, a que demandou mais tempo relativamente ao número de linhas do código fonte correspondente. O cálculo da iluminação através desse modelo responde, em média, por 40% do tempo total de processamento.

Ferramentas Para o Projeto da Cena.

A produção de uma imagem realista é, em geral, um processo muito demorado (várias horas) e muitas vezes requer ajustes de parâmetros para que se chegue ao efeito final desejado. É fundamental, portanto, que o sistema forneça algumas ferramentas que auxiliem o projetista da cena nesta tarefa.

Devido à dificuldade em se determinar as refletâncias dos objetos de modo a obtermos uma determinada cor, desenvolveu-se um programa (fig. 5) que permite a manipulação dos valores das refletâncias, calculando prontamente os valores R, G e B resultantes. Esse processo se repete iterativamente até que esses valores se igualem às primárias R, G e B da cor desejada.

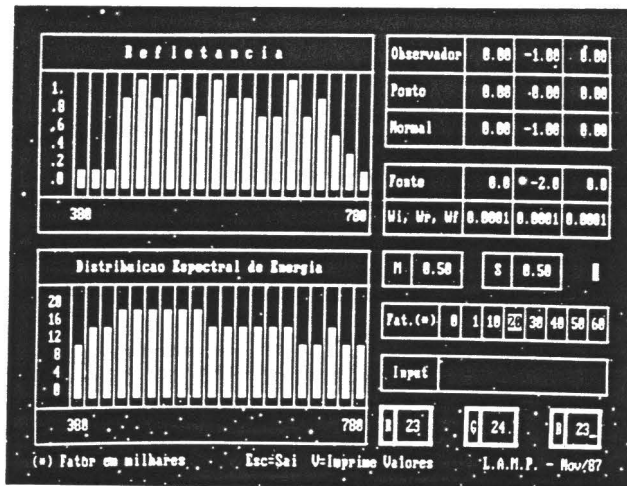


Fig. 5 - Tela do utilitário que determina as componentes RGB resultantes da variação das refletâncias dos objetos.

Dois outros utilitários foram desenvolvidos visando a verificação da geometria e da iluminação da cena. O primeiro deles (fig. 6) produz uma imagem do modelo aramado (wire frame) dos objetos, permitindo que se verifique o correto posicionamento, tanto das primitivas, quanto do observador. Permite, também, que se determine os valores das componentes R, G e B nos pontos indicados pelo cursor do mouse. Essa verificação é necessária pois em nenhum ponto da imagem pode-se ter qualquer componente com valor superior a 31, o que

resultaria em anomalias na coloração dos objetos. Essa limitação é devida à placa gráfica usada (TARGA 16) que permite que cada uma das componentes R, G e B assumam valores inteiros de 0 a 31 (daí as $32768 = 32 \times 32 \times 32$ cores possíveis). Caso isso aconteça, é necessário que se diminua a intensidade da luz ambiente e/ou direta ou que se altere os valores da refletância.

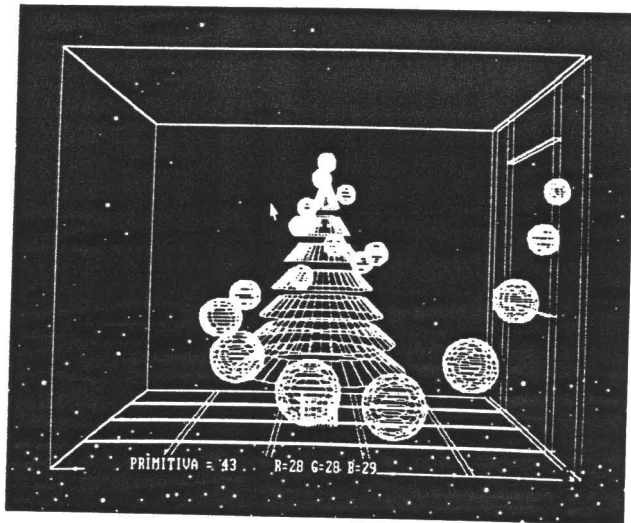


Fig. 6 - Resultado do utilitário de verificação da geometria e iluminação, através da exibição do aramado (wire frame) e da possibilidade de se determinar os valores das componentes R, G e B nos pontos indicados pelo cursor do mouse.

O segundo utilitário (fig. 7) produz a imagem da cena em baixa resolução, ou seja, um raio visual rastreado para cada grupo de 8×8 pixels da tela. Isso reduz em pouco mais de 64 vezes o tempo necessário para a produção da imagem e serve para se dar uma "última verificação", tanto na geometria quanto na iluminação. A imagem em baixa resolução pode servir,

também, para a detecção dos pontos com brilho (valores R, G e B mais sujeitos a "overflow"), facilitando a determinação dos máximos através do programa descrito anteriormente.

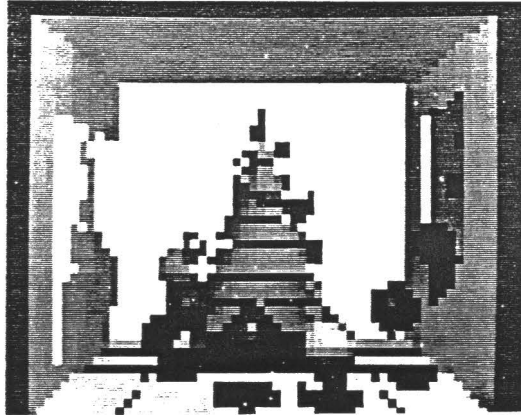


Fig. 7 - Resultado do rastreador de baixa resolução. Objetiva verificar a geometria, a iluminação e a identificação dos pontos de brilho.

Resultados.

Apresenta-se, a seguir, alguns resultados do rastreador de raios.

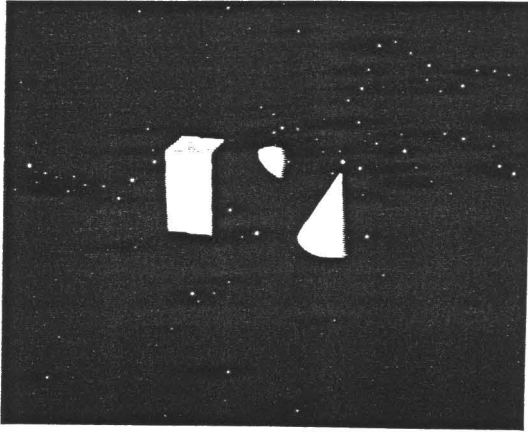


Fig. 8 - Objetos com refletâncias características do cobre; 8 primitivas, 1 fonte, 4:50 h de processamento.

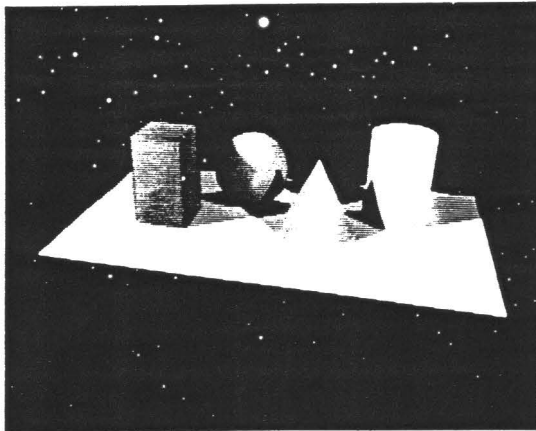


Fig. 9 - Experiência com cores diversas; 10 primitivas, 2 fontes, 6:13 h de processamento.



Fig. 10 - Tabuleiro de Xadrez; 43 primitivas, 1 fonte de luz, 18:20 horas de processamento.

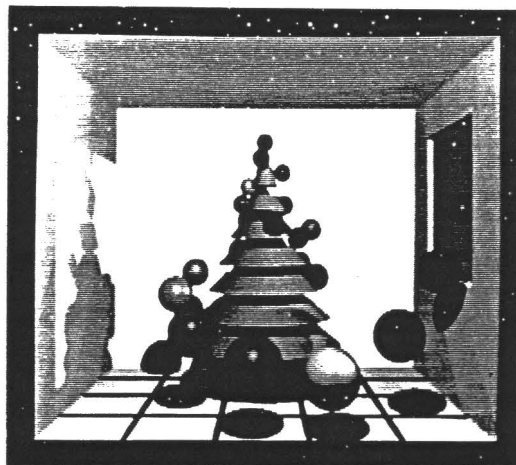


Fig. 11 - Árvore de Natal; 61 primitivas, 2 fontes, 48 h de processamento.

Comentários Finais.

O tempo necessário para a geração de imagens pelo método de rastreamento de raios é, comparado com outros métodos, demasiadamente grande (aliás, essa é a principal crítica que se faz do método). Muitos diriam que, com os tempos de processamento na ordem de grandeza dos tempos gastos na produção dos exemplos não seria viável se utilizar comercialmente o método implementado em microcomputadores. Duas observações, no entanto, se contrapõem a esse raciocínio:

- A presente implementação do método não teve por objetivo a criação de um programa altamente otimizado. No entanto, algumas providências nesse sentido foram tomadas, sempre que fossem simples de serem implementadas.

- O computador usado nos testes não é o mais rápido do mercado. Existem versões do mesmo microprocessador rodando a 10, 12, 14 e até 16 MHz. Além disso estão vindo aí os micros baseados nos processadores 80386/80387s da Intel e 68030s da Motorola e, certamente, em outros mais rápidos ainda (sem contar os que utilizarão arrays de processadores, já que o método se presta bastante ao processamento em paralelo)!

Pretende-se dar continuidade ao trabalho implementando-se novas primitivas (formas livres, formas geradas por arrasto de esferas, etc), outras operações de construção (interseção, complemento e diferença) e os efeitos de reflexo, transparência e refração que, afinal, são os que permitem

qualificar o método de rastreamento de raios como o mais flexível de todos.

Neste trabalho apresentamos uma experiência de síntese de imagens realistas por computador. Essa área ainda requer mais pesquisas, não só no método de rastreamento de raios aqui apresentado, como em outros métodos de fotografia sintética. Outros estudos e experiências a partir dos trabalhos pioneiros como este e o estudo de mapeamento de texturas deverão continuar no Departamento de Informática da PUC/RJ.

Agradecimentos.

Agradecemos a Ricardo Richers, professor de Desenho Industrial e Programação Visual da PUC do Rio de Janeiro, pela colaboração na parte artística das imagens produzidas.

Referências Bibliográficas.

- [ALVAR84] Alvarenga, Claudia de Campos
Um Estudo Comparativo de Técnicas de Mapeamento de Textura.
Dissertação de Mestrado - PUC/RJ - 1984
- [ALVAR85] Alvarenga, Claudia de Campos
Mapeamento de Textura na Síntese de Imagens Realistas.
V Congresso da SBC - Julho/1985
- [COOK81] Cook, Robert L & Torrance, Kenneth E.
A Reflectance Model for Computer Graphics.
Computer Graphics, vol 15 #3 PP 307-316
ACM/SigGraph

- [FOLEY83] Foley, J. D. & Van Dam, A.
Fundamentals of Interactive Computer Graphics.
Addison-Wesley Publishing Company - 1983.
- [NEWMA79] Newman, William M. & Sproull, Robert F.
Principles of Interactive Computer Graphics.
McGraw-Hill International Book Company - 1979.
- [PERE187] Pereira, Luiz Antonio M.
Síntese de Imagens Realistas por Rastreamento de Raios.
Dissertação de Mestrado - PUC/RJ - 1987
- [PHONG75] Phong, Bui Tuong
Illumination for Computer Generated Pictures.
Communications of the ACM, vol 18 #6 PP 311-317