

INF1711

Computação Gráfica

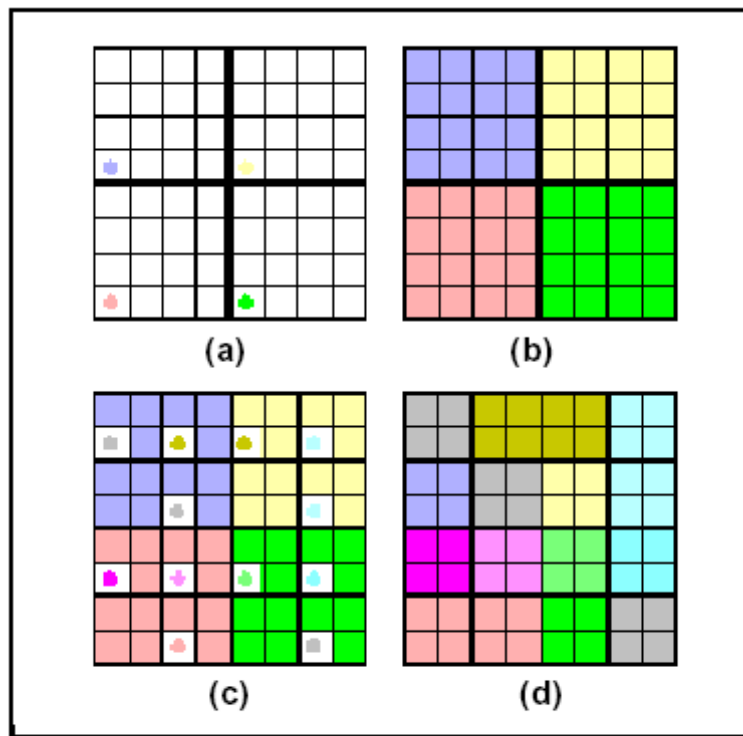
Trabalho 2

“RayTracing”

Alunos:

- Mauricio A. Lage Ferreira
- Giovani Mancuso Tadei

(1) Refinamento Progressivo:



[Hollasch (1992)]

Método para a figura ser visualizada ao mesmo tempo que é montada (a princípio, por pixels bem grandes) e tem noções cada vez mais detalhadas de sua aparência à medida que vai sendo calculada, até o nível final de refinamento. O efeito obtido passa a impressão de que os pixels diminuem de tamanho, o que pode ser comparado a um mosaico no sentido inverso.

Problemas / Soluções:

- **PROBLEMA:** A ordem de escolha sequencial causa um efeito desagradável à medida que o refinamento avança, parece um efeito de inclusão de linha.
- **SOLUÇÃO:** [Giacomantonio(1976)] Nova sequência baseada no hábito de leitura.

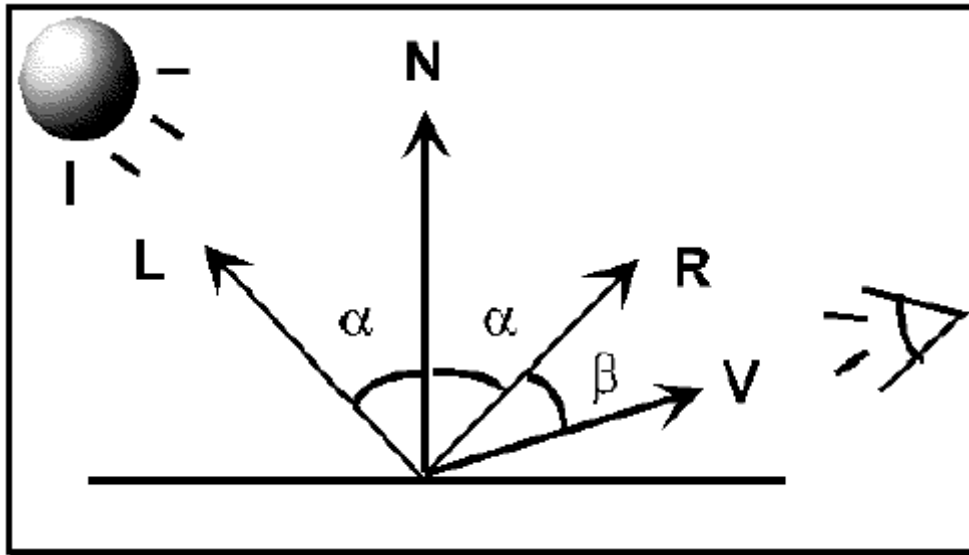
1	3
4	2

[Giacomantonio (1976)]

Problemas / Soluções:

- **PROBLEMA:** Algoritmo não pode ser recursivo. Sempre o primeiro quadrante subdividido será novamente subdividido, antes dos cálculos dos demais. Assim, não haveria refinamento progressivo em toda a imagem, mas somente dentro de cada quadrante.
- **SOLUÇÃO:** A solução encontrada foi utilizar uma fila seqüencial dos quadrantes subdivididos. Cada novo quadrante dividido tem a sua cor calculada e é colocado no final da fila. Para se calcular as cores do próximo quadrante, utiliza-se o primeiro da fila. Novamente, divide-se o novo quadrante em quatro subquadrantes e calcula-se as cores dos três que não contêm o *pixel* previamente calculado. A seguir, os quatro são colocados no final da fila, e assim por diante até a finalização de todos os pixels.

(2) Modelo de Iluminação:



Lambert / Phong

$$I = I_a \cdot k_a + \sum_{i=1}^{i=m} I_{fonte_i} \cdot \left[k_d \cdot \cos \alpha + k_{esp} \cdot \cos^n \beta \right]$$

[a] **Luz Ambiente:**

A componente de luz ambiente do modelo de Phong foi adicionada para levar em consideração luz gerada por interações inter-objetos. Em ambientes reais, superfícies que não são iluminadas diretamente geralmente não se encontram completamente escuras.

[b] **Luz Difusa:**

A componente difusa segue a lei de Lambert, que relaciona a intensidade de luz refletida pela superfície ao cosseno do ângulo α , formado pela normal à superfície (N) e pelo vetor (L), que liga o ponto a ser iluminado com a fonte luminosa.

$$I_d = I_{fonte} \cdot k_d \cdot \cos \alpha$$

[c] **Luz Especular:**

A componente especular é dada pela “contribuição de Phong”, uma fórmula empírica que relaciona a intensidade especular a uma potência do cosseno do ângulo β , formado pela direção de reflexão (R) e pelo vetor (V), que liga o ponto da superfície ao observador

$$I_{esp} = I_{fonte} \cdot k_{esp} \cdot \cos^n \beta$$

(3) Sombra:

[a] **Luz Ambiente:**

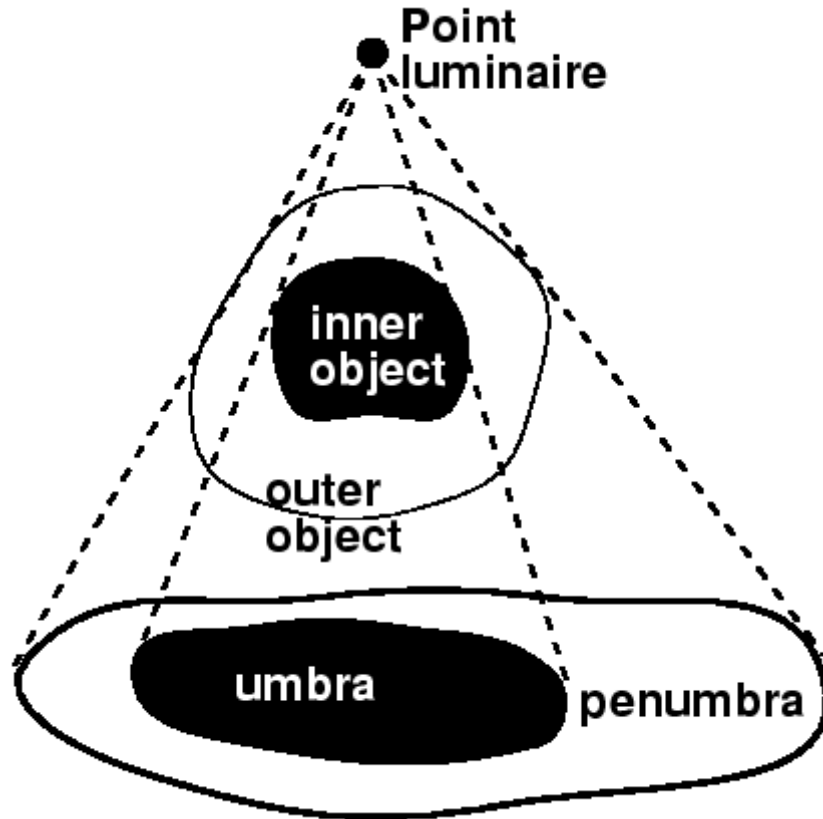
A idéia é lançar um outro raio, chamado “raio de sombra”, que une o ponto do objeto que foi atingido ao ponto de luz. Se entre o ponto e a luz existir um outro objeto, este ponto estará na sombra, ou seja, será pintado com a intensidade de luz ambiente.

[b] **Sombra de Objetos Transparentes:**

Para gerar esse tipo de sombra foi usado o seguinte cálculo:

$$\begin{aligned}\text{Opacidade} &= 1 - \text{Transparência} \\ \text{Fator de Sombra} &= 0.25^{\text{Opacidade}} \\ \text{Iluminação} &= \text{Fator de Sombra} * \text{Iluminação Difusa}\end{aligned}$$

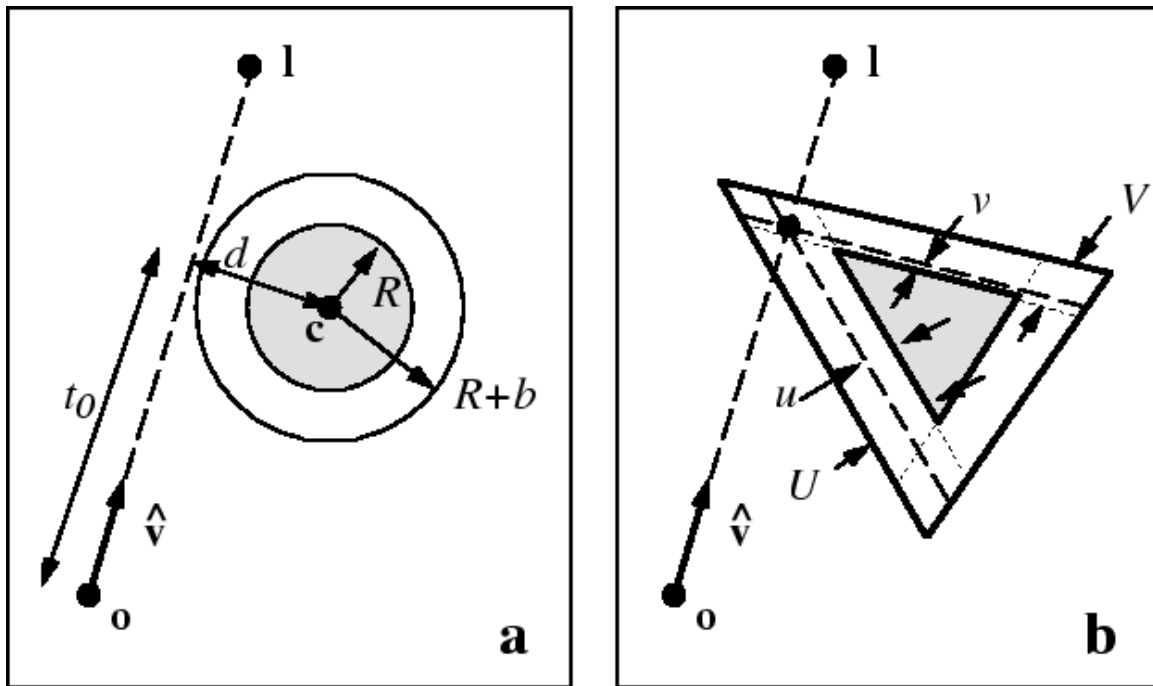
[c] **Soft Shadow:**



O algoritmo usado foi bem simples: Criar 8 fontes de luz em volta da fonte principal dividindo o valor de cor de cada uma por 8 dando o efeito de penumbra a sombra.

Um algoritmo mais interessante pesquisado foi o "Single Sample Soft Shadows"

(<http://www.cs.utah.edu/%7Ebes/papers/coneShadow/shadow.html>)

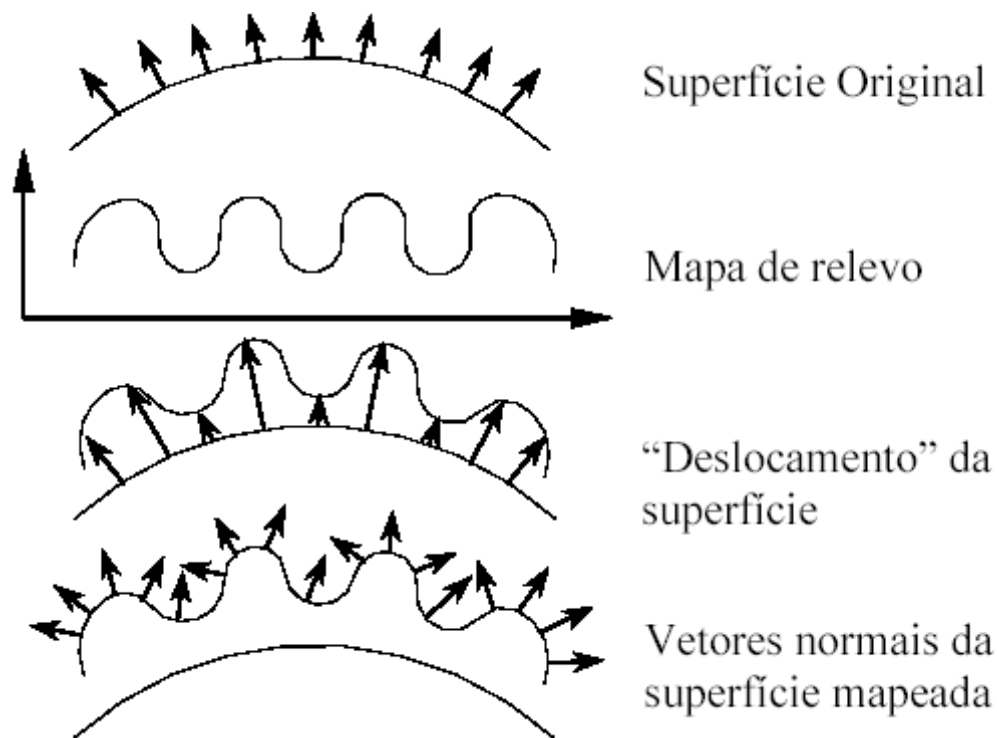


Nele através de uma álgebra é possível encontrar a Umbra e a Penumbra da figura e de posse dessas informações na região da umbra variar um “t” iniciando de $t=0$ até $t=1$ utilizando a interpolação Bernstein abaixo para ter o valor da sombra “s”.

$$s = 3\tau^2 - 2\tau^3$$

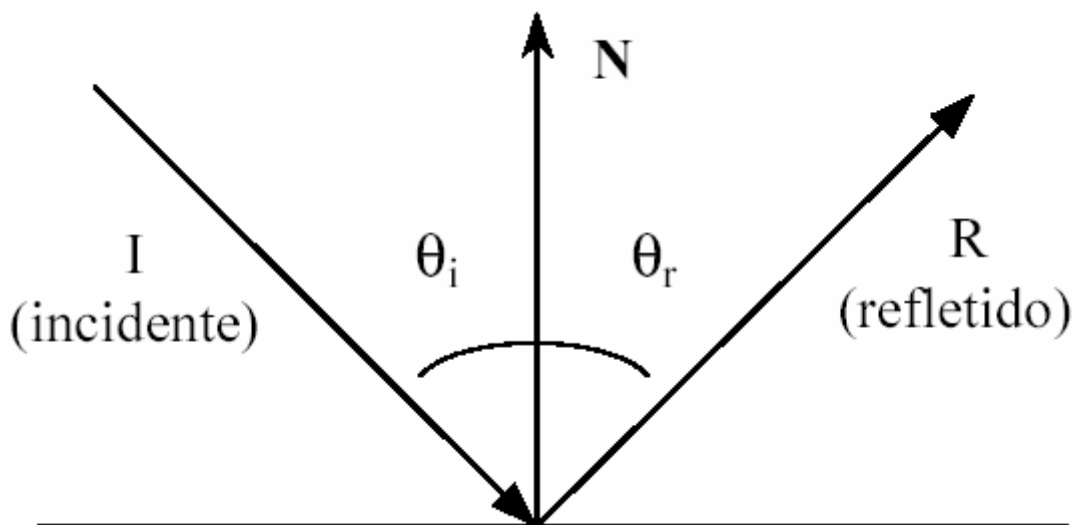
(5) Bump Mapping:

O algoritmo utilizado pega a normal do objeto naquele ponto, de norma um, e a partir dela encontra um vetor no plano definido pela normal, também de norma 1. Esse vetor é multiplicado por $1 - \text{luminância do bumpmap}$, e é adicionado à normal, e essa passa a ser a nova normal.

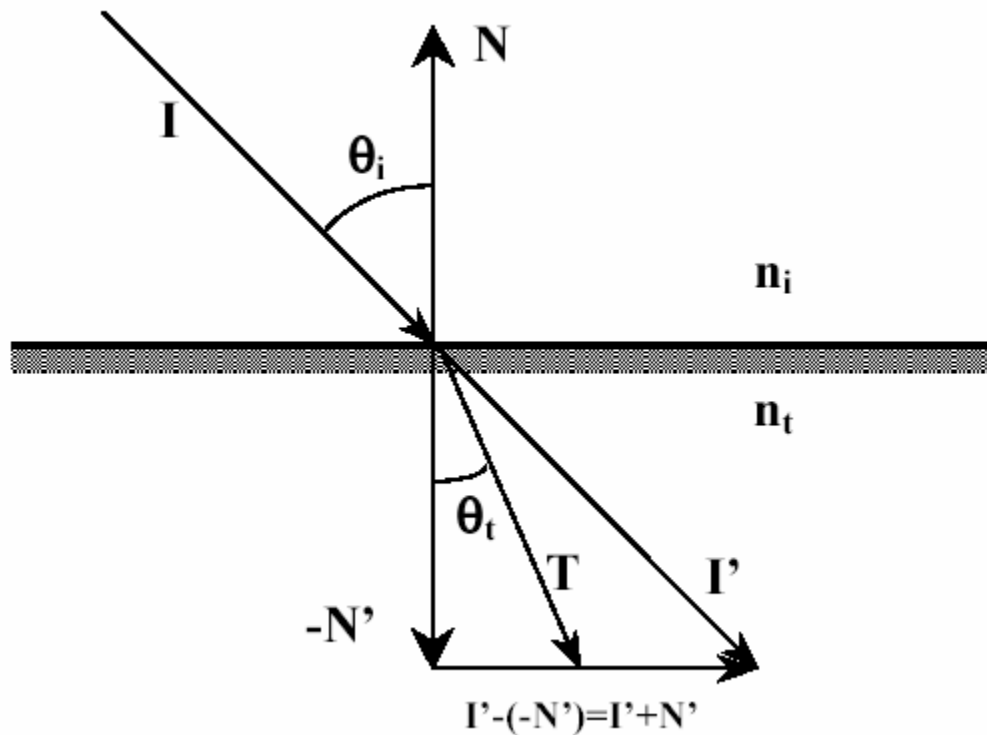


(6) Reflexão:

Quando executamos Ray Tracing em uma superfície refletora (um espelho, por exemplo), o raio que foi lançado bate nesta superfície e é refletido segundo a lei da ótica, que diz que o raio refletido tem a mesma direção do raio que incidiu sobre a superfície. Assim, em termos de algoritmo, é como se fosse lançado um novo raio visual a partir deste ponto, só que na direção de reflexão. Este ponto terá a cor calculada a partir do raio refletido.



(7) Transparência:



Ao atingir um objeto transparente com índice de refração N_1 , a luz atravessa a superfície e é desviada num fenômeno chamado refração, que é regulado pela lei de Descartes-Snell. A cor do ponto é calculada a partir da cor obtida por um novo raio lançado com a direção refratada (T). O cálculo do vetor T é mostrado na seguinte fórmula:

$$T = \left(N_{21} \cdot (N \cdot I) - \sqrt{1 - (N_{21})^2 \cdot (1 - (N \cdot I)^2)} \cdot N - N_{21} \cdot I \right)$$

Referências Bibliográficas

-Refinamento Progressivo da Cena com Traçado de Raios Distribuído

RENATO GONÇALVES DE ARAUJO

-Introdução ao Ray Tracing

Fernando Wagner Serpa Vieira da Silva

Laboratório de Computação Gráfica

LCG COPPE / UFRJ

- [Phong73] - Bui-Tuong, Phong, “Illumination for Computer Generated Surfaces”,

Doctoral Thesis, University of Utah, 1973.

-[Kay79] - Kay, Douglas S., “Transparency, Refraction and Ray Tracing for Computer

Synthesized Images”, Master Thesis, Program of Computer Graphics, Cornell

University, Jan 1979.

-Single Sample Soft Shadows

Steven Parker - Peter Shirley - Brian Smits

University of Utah

-Extensões do RayTracing Parametrizado

Eduardo Toledo