### Douglas Messias Uba

## Síntese de Imagens Realistas Utilizando Traçado de Raios

Juiz de Fora05/02/2007

### Douglas Messias Uba

## Síntese de Imagens Realistas Utilizando Traçado de Raios

Orientador: Marcelo Bernardes Vieira

Universidade Federal de Juiz de Fora Instituto de Ciências Exatas Departamento de Ciência da Computação

> Juiz de Fora 05/02/2007

## Sum'ario

### Lista de Figuras

1	Intr	Introdução				
	1.1	Algoritmos de Iluminação	p. 9			
	1.2	Objetivos	p. 10			
	1.3	Organização do Trabalho	p. 10			
2	Física da Luz					
	2.1	Natureza da Luz	p. 12			
	2.2	Radiometria	p. 13			
		2.2.1 O Fóton	p. 13			
		2.2.2 Quantidade Radiométricas	p. 13			
	2.3	Espalhamento da Luz	p. 17			
	2.4	BRDF	p. 18			
3	3 Modelos de Iluminação Local					
	3.1	Fundamentos Básicos	p. 20			
		3.1.1 Reflectância	p. 21			
		3.1.2 Reflexão Difusa	p. 21			
		3.1.3 Reflexão Especular	p. 22			
	3.2	Modelo Modificado de Phong				
	3 3	R. Modolo do Schlick				

4	Traçado de Raios						
	4.1	Raio	o. 27				
	4.2	Traçado de Raios Clássico	o. 28				
		4.2.1 Algoritmo	o. 31				
	4.3	Cálculo das Interseções	o. 32				
		4.3.1 Esfera	o. 32				
		4.3.2 Plano	o. 34				
		4.3.3 Triângulo	o. 34				
		4.3.4 Superfícies Quádricas	o. 35				
	4.4	Limitações do Traçado de Raios Clássico	o. 37				
5	Tra	çado de Raios Monte Carlo	o. 39				
	5.1	Método de Monte Carlo	o. 39				
		5.1.1 Integração Monte Carlo	o. 39				
	5.2	Equação de Rendering	o. 40				
	5.3       Traçado de Raios Distribuído						
		5.4.1 Algoritmo	o. 45				
6	Imp	olementação e Resultados	o. 47				
	6.1	Implementação	o. 47				
		6.1.1 Estruturas básicas	o. 47				
		6.1.2 Câmera	o. 48				
		6.1.3 Primitivas e Materiais	o. 48				
		6.1.4 BRDF	o. 49				
		6.1.5 Fontes de Luz	o. 49				
		6.1.6 Cena	o. 50				

	6.2 Resultados		p. 50
	6.3	Dificuldades e Limitações	p. 53
7	Con	nclusão e Trabalhos Futuros	p. 54
Referências			p. 56

# Lista de Figuras

1	Iluminação Local: apenas a iluminação direta é considerada	p. 9
2	Iluminação Global: iluminação direta e indireta são consideradas	p. 9
3	Irradiância (fluxo incidindo na superfície); Radiosidade (fluxo abandonando a superfície)	p. 14
4	Intensidade radiante: fluxo radiante $(d\Phi)$ por ângulo sólido $(d\vec{\omega})$	p. 15
5	Área projetada é definida como a área da superfície projetada em um plano perpendicular à direção de incidência ou saída do fluxo	p. 16
6	Radiância é a relação entre fluxo radiante $(d\Phi)$ e a área projetada $(dA^{\perp})$ da superfície pelo ângulo sólido $(d\vec{\omega})$	p. 16
7	Espalhamento de um fóton em uma superfície. Podemos descrever esse fenômeno através de uma BSSDRF	p. 17
8	Na BDRF, assumimos que o fóton é refletido instantaneamente no mesmo ponto que incidiu na superfície.	p. 17
9	Reflexão difusa ideal ou Lambertiana	p. 21
10	(a) Reflexão especular ideal. (b) Glossy	p. 22
11	Exemplo de um raio com origem $o$ , direção $d$ e diferentes valores para o parâmetro $t$	p. 28
12	Image sintetizada através do algoritmo clássico de traçado de raios, possuindo inter-reflexões múltiplas, sombras e transparência. Fonte: (JENSEN, 2001)	p. 29
13	Algoritmo de traçado de raios. Raios com origem no observador passam pelos respectivos <i>pixels</i> no plano de projeção. A cor de cada <i>pixel</i> é a cor do objeto interceptado pelo raio ou a cor de fundo caso não exista	1
	interseção	p. 29

14	Superfícies Quádricas: (a) Elipsóide (b) Cilindro elíptico (c) Cone elíptico (d) Parabolóide elíptico	p. 36
15	Reflexão borrada para diferentes números de raios distribuídos em torno da direção de reflexão ideal. Fonte: (MARTIN, 2007)	p. 41
16	Translucência para diferentes números de raios distribuídos em torno na direção de refração. Fonte: (MARTIN, 2007)	p. 41
17	Penumbra para diferentes números de raios de sombra. Fonte: (MARTIN, 2007)	p. 42
18	Profundidade de campo: lente colocada em frente ao plano da imagem.  O raio primário tem origem em um ponto no círculo definido pela lente e passa pelo ponto focal	p. 43
19	Profundidade de campo para diferentes valores de distância focal. Fonte: (BUSS, 2003)	p. 43
20	Imagen com borrão de movimento. Os raios são distribuídos em instantes de tempo diferentes. Fonte: (BUSS, 2003)	p. 43
21	Algoritmo de traçado de raios distribuído. Vários raios são lançados por pixel. Os raios de sombra são distribuídos sobre a área da fonte de luz. Para superfícies especulares, são distribuídos raios de reflexão em torno da direção de reflexão ideal	p. 44
22	Presença de ruído em imagens geradas através do traçado de caminhos. A utilização de 100 raios/pixel ainda assim não é suficiente para eliminar por completo o ruído na imagem. Fonte: (JENSEN, 2001)	p. 45
23	Esfera gerada com diferentes valores para os parâmetros do modelo de Phong Modificado. A última é perfeitamente especular e tem a iluminação indireta calculada.	p. 50
24	Superfície quádrica: Cilindro gerado com diferentes valores para os parâmet do modelo de Phong Modificado.	ros p. 51
25	Esferas e plano: iluminação indireta calculada para uma esfera perfeitamente especular. Utilização de 50 raios/pixel	p. 51
26	Refração: o índice de refração da cena é igual a 1,003 (ar), enquanto o da esfera é 1,59 (vidro).	p. 51

27	Reflexão borrada ( $glossy$ ): utilizando 20 raios em torno da direção de	
	reflexão ideal	p. 52
28	Sombras suaves ( $soft\ shadows$ ) para diferentes números de raios de sombra.	p. 52
29	Superfície quádrica especular (elipsóide). Devido a utilização de apenas	
	1 raio por <i>pixel</i> , temos o problema de <i>jittering</i>	p. 52
30	Utilização de 10 raios por pixel com distribuição uniforme	p. 53
31	Utilização de 10 raios por pixel com distribuição normal	p. 53

## 1 Introdução

Atualmente a Computação Gráfica já faz parte do dia-a-dia da sociedade. Com a disponibilidade de equipamentos de alto desempenho e custo mais baixo, rapidamente surgiram aplicações em campos como publicidade, desenho industrial, estudo de iluminação, visualização de modelos tridimensionais complexos, efeitos especiais, CAD, visualização médica, jogos eletrônicos, entre tantos outros. Em muitas destas aplicações, cada vez mais é necessário gerar imagens mais sofisticadas e mais realistas.

A síntese de imagens por computador consiste no processo de, a partir da descrição geométrica tri-dimensional de um mundo, produzir uma imagem bi-dimensional correspondente a um determinado ponto de observação. Essa descrição geométrica do mundo, comumente chamada de modelo do mundo, pode incluir, além da geometria, outras características dependendo da aplicação, como por exemplo: propriedades dos materiais que formam os objetos da cena, posição das fontes de luz, atributos do meio onde a luz se propaga etc.

A **Síntese de Imagens Realistas** é um segmento da Computação Gráfica que procura gerar imagens sintéticas que são indistinguíveis de imagens do mundo real.

Desde o início da Computação Gráfica, foi percebido o seu grande potencial para geraração dessas imagens. Obviamente, as primeiras técnicas para esse propósito foram naturalmente limitadas pela tecnologia disponível e permaneceram esquecidas até o fim da década de 70. Em 1980, com a introdução do traçado de raios, a síntese de imagens realistas passou a utilizar simulações baseadas em fenômenos físicos reais, o que chamamos de síntese por simulação. O outro paradigma de síntese de imagens, que se opõe a síntese por simulação, é o da síntese por emulação.

As técnicas de síntese por emulação procuram gerar imagens retratando a realidade, mas sem se preocupar com os fênomenos físicos. Em outras palavras, são técnicas que utilizam artifícios algorítmicos para "imitar" a realidade.

Porém, um pensamento natural ao querermos gerar imagens que se aproximem ao

máximo da realidade é primeiro, compreendermos quais os processos ocorrem de fato para a formação de uma imagem, para, em seguida, modelar esse conhecimento e simular o processo através do computador. Isso envolve principalmente teorias que descrevem a natureza física da luz e seu modelo matemático. Logo, os métodos utilizados na síntese de imagens realistas fazem uso extensivo desses conceitos. Mais especificamente, a síntese de imagens realistas implica a modelação e simulação dos processos físicos ligados ao transporte da luz e sua interação com os materiais que constituem as superfícies.

### 1.1 Algoritmos de Iluminação

Podemos classificar os algortimos de iluminação como sendo local ou global. Algoritmos de iluminação local consideram apenas a iluminação direta, ou seja, a interação entre cada objeto e a energia luminosa proveniente de cada fonte de luz. Interações entre os objetos são ignoradas, tais como sombras, reflexões e transmissões (Fig.1). Os algoritmos de iluminação global consideram, além da iluminação direta, a luz que incide nos objetos após ter interagido com outros objetos da cena (Fig.2). A isso damos o nome de iluminação indireta. Essa classe de algoritmos é essencial para a síntese de imagens realistas, já que grande parte do aspecto final de uma imagem se deve a iluminação indireta.

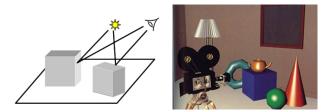


Figura 1: Iluminação Local: apenas a iluminação direta é considerada.

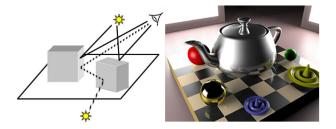


Figura 2: Iluminação Global: iluminação direta e indireta são consideradas.

### 1.2 Objetivos

O objetivo principal deste trabalho é estudar os conceitos e métodos utilizados na síntese de imagens realistas. Será necessário compreender a física natural da luz, a terminologia utilizada para descrevê-la, bem como a interação da luz com superfícies. Com isso, formamos o modelo matemático de iluminação.

Entendido estes conceitos, o **objetivo secundário** da monografia é o estudo e o desenvolvimento de um traçador de raios como o modelo computacional de iluminação. Além de simples e elegante, este é o método que permite a geração das imagens mais realistas atualmente.

### 1.3 Organização do Trabalho

Nesta monografia iremos percorrer todo o caminho que envolve a geração de imagens realistas, segundo a descrição física da luz.

No capítulo 2 apresentaremos a física da propagação da luz e um dos modelos utilizados para representá-la, a ótica geométrica. Faremos um estudo sobre a radiometria, parte da física que estuda a medição da radiação eletromagnética e fornece a terminologia usada para descrever a luz. Estudaremos em seguida o espalhamento da luz e como formular matematicamente este problema através de uma função de distribuição, a BRDF.

No capítulo 3 apresentaremos dois modelos de iluminação local: o modelo de Phong e o modelo de Schlick. Veremos os modelos de reflexão ideais, fundamentos básicos para compreensão de qualquer modelo de iluminação.

No capítulo 4 apresentaremos o método de traçado de raios. Iremos definir o que é um raio, como e de que forma ele é utilizado. Em seguida, apresentamos o primeiro algoritmo utilizado na síntese de imagens realistas, o traçado de raios recursivo.

No capítulo 5 apresentaremos o método de Monte Carlo e o conceito de integração Monte Carlo, fundamento básico para a compreensão das extensões do traçado de raios. Em seguida, formulamos matematicamente o problema que todo o algoritmo de iluminação global se propões a resolver, a equação de rendering e apresentamos dois algoritmos que utilizam método de Monte Carlo: o traçado de raios distribuído e o traçado de caminhos.

No capítulo 6 apresentaremos os resultados obtidos com a implementação de um sintetizador de imagens que utiliza técnicas de traçado de raios. Encerramos o trabalho

tratando as limitações e dificuldades encontradas durante o desenvolvimento.

### $\it 2$ $\it Fisica da Luz$

Estamos interessados em simular o processo que ocorre quando a luz é emitida por uma ou mais fontes e interage com os materiais que consitutem as superfícies. Para que essa simulação seja realizada de forma correta, devemos ser capazes de compreender a natureza da luz, a terminologia e o modelo matemático utilizado para descrevê-la, e também como ocorre a sua interação com as superfícies. O objetivo deste capítulo é apresentar esses conceitos.

Para isso, apresentaremos os modelos que foram propostos ao longo do tempo para representar a luz e como essa esta é parte do espectro eletromagnético, um estudo da radiometria se faz necessário, já que é a parte da física que estuda a medição da radiação eletromagnética. Serão definidas algumas grandezas radiométricas fundamentais, como fluxo radiante e radiância e, em seguida, apresentaremos a BRDF, função que descreve a interação da luz com um material.

### 2.1 Natureza da Luz

A natureza da luz é bastante complicada, existindo diferentes abordagens para capturar seu comportamento dual: corpuscular e ondulatório.

Atualmente temos três modelos para representá-la: a ótica quântica, a ótica eletromagnética e a ótica geométrica. Embora o modelo quântico seja o mais geral e através dele sejamos capazes de descrever todos os fenômenos capturados pelos outros modelos, é útil podermos escolher o modelo mais adequado para cada tipo de problema que queremos resolver. Como a ótica geométrica é o modelo mais utilizado na área da Computação Gráfica, este será adotado ao longo deste trabalho.

A ótica geométrica nos permite descrever fenômenos como emissão, reflexão, transmissão e absorção, importantes para o nosso propósito, e é baseado em algumas simplificações:

- A propagação da luz ocorre em linha reta e não é afetada por fatores externos, como campos magnéticos e a gravidade;
- A luz propaga-se instantaneamente entre dois pontos quaisquer;
- O comprimento da onda eletromagnética é muito menor do que os objetos com os quais esta interage.

As informações acima mostram o motivo pelo qual esse modelo é o mais utilizado na Computação Gráfica. Um modelo simplista, mas com capacidade de descrever fenômenos responsáveis em grande parte pela aparência do mundo que nos cerca com certeza é o melhor a ser utilizado comparado com os outros modelos, considerados demasiadamente detalhados para a síntese de imagens realistas.

### 2.2 Radiometria

A radiometria é a parte da física que estuda a medição da radiação eletromagnética e nos fornece a terminologia básica usada para descrever a luz.

Nesta seção será apresentado o fóton e as grandezas radiométricas necessárias para descrever e formular os algoritmos que calculam a distribuição da luz em um determinado modelo do mundo, isto é, os algoritmos de iluminação global.

#### 2.2.1 O Fóton

A quantidade básica de iluminação é o fóton. A energia de um fóton,  $e_{\lambda}$ , com comprimento de onda  $\lambda$  é dada por:

$$e_{\lambda} = \frac{hc}{\lambda} \tag{2.1}$$

onde h é a constante de Plack ( $h\approx 6,63\cdot 10^{-34}J\cdot s$ ) e c é a velocidade da luz (no vácuo, c=299.792.458 m/s).

### 2.2.2 Quantidade Radiométricas

Definido a energia de um fóton, podemos então definir a **energia radiante espectral**,  $Q_{\lambda}$ , que é a energia de  $n_{\lambda}$  fótons com comprimento de onda  $\lambda$ , dada por:

$$Q_{\lambda} = n_{\lambda} e_{\lambda} = n_{\lambda} \frac{hc}{\lambda} \tag{2.2}$$

medida em J/nm.

A energia radiante, Q, é a energia de um conjunto de fótons com diversos comprimentos de onda. É calculada integrando a energia radiante espectral,  $Q_{\lambda}$ , no domínio de todos os comprimentos de onda:

$$Q = \int_0^\infty Q_\lambda \ d\lambda \tag{2.3}$$

medida em J.

O fluxo radiante,  $\Phi$ , é a quantidade radiométrica fundamental. Expressa a quantidade de energia radiante, Q, que flui através de uma superfície por unidade de tempo. É dado por:

$$\Phi = \frac{dQ}{dt} \tag{2.4}$$

medida em Watts (J/s).

A potência das fontes de luz é normalmente expressa em fluxo radiante.

A densidade de fluxo radiante é dada pelo quociente do fluxo radiante pela área da superfície, expressando portanto a noção de densidade. É comum referir-se a densidade de fluxo radiante que abandona uma superfície em um ponto x por Radiosidade (B(x)) e a densidade de fluxo radiante que incide em uma superfície em um ponto x por Irradiância (E(x)) (Fig.3). Formalmente é definida por:

$$E(x) = B(x) = \frac{d\Phi}{dA} \tag{2.5}$$

medida em  $W/m^2$ .

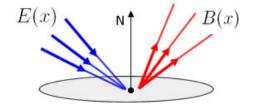


Figura 3: Irradiância (fluxo incidindo na superfície); Radiosidade (fluxo abandonando a superfície).

Antes de definirmos **intensidade radiante**, devemos primeiro definir o que é ângulo sólido, elemento bastante usado na descrição da luz.

Um ângulo sólido é a generalização para três dimensões de um ângulo normal em duas dimensões. O ângulo normal é a relação entre um trecho de circunferência, que chamamos de arco, e o raio dessa. Seguindo essa definição e estendendo para três dimensões, o ângulo sólido é a relação entre uma pequena área na superfície de uma esfera e seu raio ao quadrado. Medimos um ângulo normal em **radianos** (rad), enquanto um ângulo sólido é medido em **esterradiando** (sr).

Entendido esse conceito, podemos então definir a intensidade radiante,  $I(\vec{\omega})$ , como sendo o quociente do fluxo radiante por unidade de ângulo sólido:

$$I(\vec{\omega}) = \frac{d\Phi}{d\vec{\omega}} \tag{2.6}$$

Em outras palavras, a intensidade radiante nada mais é que o fluxo radiante em determinada direção (Fig.4). É medida em W/sr.

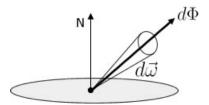


Figura 4: Intensidade radiante: fluxo radiante  $(d\Phi)$  por ângulo sólido  $(d\vec{\omega})$ .

Definidos densidade de fluxo radiante e intensidade radiante, estamos prontos para definir a grandeza mais importante envolvendo o transporte da luz, a **radiância**.

Formalmente, é definida como o fluxo radiante por unidade de área projetada por unidade de ângulo sólido:

$$L(x,\vec{\omega}) = \frac{d^2\Phi}{dA^{\perp} \ d\vec{\omega}} \tag{2.7}$$

medida em  $W/m^2sr$ .

A área projetada,  $dA^{\perp}$ , é definida como sendo a área da superfície, dA, projetada em um plano perpendicular à direção de incidência ou saída do fluxo (Fig.5). Isso nos dá a relação:

$$dA^{\perp} = dA \cdot \cos\theta \tag{2.8}$$

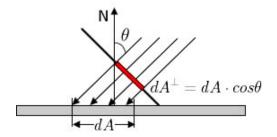


Figura 5: Área projetada é definida como a área da superfície projetada em um plano perpendicular à direção de incidência ou saída do fluxo.

Assim, podemos reescrever a radiância como:

$$L(x,\vec{\omega}) = \frac{d^2\Phi}{dA \cdot \cos\theta \ d\vec{\omega}}$$

Percebam que a radiância nada mais é que a combinação da densidade de fluxo radiante com a intensidade radiante e nos dá o número de fótons que incidem ou abandonam por unidade de tempo, uma determinada área com uma determinada direção (Fig.6).

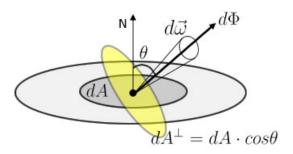


Figura 6: Radiância é a relação entre fluxo radiante  $(d\Phi)$  e a área projetada  $(dA^{\perp})$  da superfície pelo ângulo sólido  $(d\vec{\omega})$ .

Mais uma vez podemos reescrever a radiância, agora em termos da energia de  $n_{\lambda}$  fótons com diferentes comprimentos de onda, por unidade de área, por unidade de ângulo sólido, por unidade de tempo. Fazemos isso integrando em toda distribuição espectral, ou seja, no domínio de todos os comprimentos de onda:

$$L(x,\vec{\omega}) = \int_0^\infty \frac{d^4 n_\lambda}{dA \cdot \cos\theta \ d\vec{\omega} \ dt \ d\lambda} \frac{hc}{\lambda} \ d\lambda$$

Podemos utilizar a radiância para determinar a intensidade luminosa em um determinado ponto no espaço e em uma determinada direção. Isso explica sua notória importância nos algoritmos de iluminação global.

### 2.3 Espalhamento da Luz

Quando um fóton atinge uma superfície, ele pode ser absorvido ou geralmente interage com o material e é reemitido em outro ponto da superfície e em outra direção (Fig.7). Devemos modelar matematicamente esse problema, pois estamos interessados em calcular quanto de energia luminosa determinada superfície reflete em determinada direção. Isto pode ser feito através de uma Função de Distribuição Bidirectional de Reflectância em Superfície de Espalhamento, do inglês Bidirectional Scattering Surface Reflectance Distribution Function, BSSRDF.

Acontece que uma BSSRDF, para ser calculada, não depende apenas do ponto e da direção de incidência, já que a reflexão pode ocorrer em qualquer outro ponto da superfície. Isso traz alguns problemas, como um elevado custo computacional para o cálculo da BSSRDF.

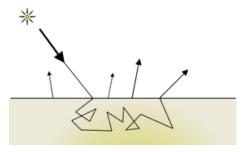


Figura 7: Espalhamento de um fóton em uma superfície. Podemos descrever esse fenômeno através de uma BSSDRF.

Assumimos então algumas simplificações (Fig.8):

- Um fóton, ao atingir uma superfície em um ponto, é reemitido nesse mesmo ponto;
- A reemissão ocorre instantaneamente.

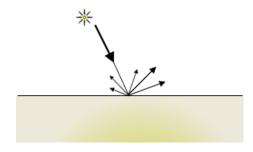


Figura 8: Na BDRF, assumimos que o fóton é refletido instantaneamente no mesmo ponto que incidiu na superfície.

Dessa forma, as propriedades de reflexão podem ser capturadas através de uma Função de Distribuição Bidirectional de Reflectância, do inglês *Bidirectional Reflectance Distribution Function*, BDRF, que é a aproximação de uma BSSRDF.

Ao assumirmos as simplificações acima, estamos sujeitos a perda de informações sobre as propriedades de reflexão dos materiais, mas isto é compensado pela redução da complexidade da função que as descreve. Outro ponto positivo é que através de uma BRDF nós podemos capturar com boa aproximação as propriedades de reflexão de vários materiais, com exceção dos chamados materiais translúcidos, como por exemplo:

- pele humana;
- mármore;
- pedras preciosas;
- leite,

onde é necessário o uso de uma BSSRDF para obtenção de resultados próximos da realidade.

### 2.4 BRDF

Nesta seção, apresentaremos com mais detalhes a BRDF, seu modelo matemático e duas importantes propriedades.

A BRDF foi introduzida por (NICODEMUS et al., 1977) e define a reflectância de um ponto em uma superfície, ou seja, quanto de energia luminosa é refletida. Formalmente é definida como a relação entre a radiância refletida e a irradiância. Denota-se por  $f_r$ :

$$f_r(x, \vec{\omega'}, \vec{\omega}) = \frac{dL_r(x, \vec{\omega})}{dE_i(x, \vec{\omega'})}$$
(2.9)

onde  $\vec{\omega'}$  é a direção de incidência e  $\vec{\omega}$  a direção de reflexão.

Aqui,  $dE_i(x, \vec{\omega'})$  significa a irradiância em x vinda da direção  $\vec{\omega'}$ . Desta forma, podemos reescrever esse termo como a radiância total que incide no ponto x, através da direção  $\vec{\omega'}$  da seguinte forma:

$$dE_i(x, \vec{\omega'}) = L_i(x, \vec{\omega'})(\vec{\omega'} \cdot \vec{n}) \ d\vec{\omega'}$$

onde  $\vec{n}$  é a normal no ponto x. Note que  $(\vec{\omega'} \cdot \vec{n}) = \cos\theta$ .

Reescrevendo  $f_r$ , temos:

$$f_r(x, \vec{\omega'}, \vec{\omega}) = \frac{dL_r(x, \vec{\omega})}{L_i(x, \vec{\omega'}) \cos\theta \ d\vec{\omega'}}$$

Estamos interessados em calcular a radiância refletida por um ponto em determinada direção. Para isso, basta agora, isolarmos o termo  $dL_r(x,\vec{\omega})$  na equação acima e integrarmos em todo o hemisfério de direções<sup>1</sup>,  $\Omega$ , a BRDF e a radiância incidente, obtendo assim a radiância refletida em todas as direções:

$$L_r(x,\vec{\omega}) = \int_{\Omega} f_r(x,\vec{\omega'},\vec{\omega}) \ L_i(x,\vec{\omega'}) \cos\theta \ d\vec{\omega'}$$
 (2.10)

Existem duas classes de BRDF: **isotrópicas** e **anisotrópicas**. Uma BRDF é chamanda de isotrópica quando as propriedades de reflexão não mudam com relação a rotação da superfície em torno de seu vetor normal. Se as propriedades de reflexão mudam com a rotação, dizemos que a BRDF é anisotrópica.

Duas importantes propriedades físicas também devem ser observadas em uma BRDF. A primeira diz que se mantermos as direções inalteradas e invertermos o sentido dos fluxos, a BRDF permanece igual. Essa propriedade é conhecida com Lei da Reciprocidade e matematicamente é expressa como:

$$f_r(x, \vec{\omega'}, \vec{\omega}) = f_r(x, \vec{\omega}, \vec{\omega'})$$
(2.11)

A segunda propriedade é dada pela Lei da Conservação da Energia. Uma superfície não pode refletir mais energia do que recebe. Assim, uma BRDF deve satisfazer a seguinte equação:

$$\int_{\Omega} f_r(x, \vec{\omega'}, \vec{\omega}) \ (\vec{\omega'} \cdot \vec{n}) \ d\vec{\omega'} \le 1, \forall \ \vec{\omega}$$
 (2.12)

Para expressarmos matematicamente essa lei, utilizamos um artifício matemático e assumimos que a radiância incidente,  $L_i$ , é constante em todas as direções. Isso nos permite escrever a equação acima.

Definido BRDF e apresentado duas importantes propriedades, devemos agora definir o que chamamos de **modelos de iluminação local**.

 $<sup>^{1}</sup>$ O Hemisfério de Direções,  $\Omega$ , são todas as direções visíveis por um ponto em uma superfície.

## 3 Modelos de Iluminação Local

Os modelos de iluminação local descrevem como determinado material reflete ou transmite a energia luminosa incidente, em determinada direção. Fazem isso definindo uma BRDF e são baseados em dois modelos de reflexão ideais: reflexão difusa e especular. Fazemos aqui um levantamento sobre os modelos existentes, para em seguida, apresentarmos os fundamentos básicos para compreendê-los.

Os primeiros modelos criados eram puramente fenomenológicos, com destaque para o modelo de (PHONG, 1975), que combina linearmente uma componente difusa com outra especular. Seguindo a idéia de Phong de combinar duas componentes, Blinn (BLINN, 1977) trouxe para a Computação Gráfica o primeiro modelo baseado na física (TORRANCE; SPARROW, 1967). Este modelo utiliza o conceito de microfacetas, ou seja, considera que a superfície é formada por inúmeras microfacetas com uma distribuição conhecida, e que cada uma dessas microfacetas possui reflexão especular ideal. Seguindo a mesma linha, um modelo aprimorado essas idéias foi desenvolvido em (COOK; TORRANCE, 1981). Outro modelo que utiliza o conceito de microfacetas é o modelo de (OREN; NAYAR, 1994), que propôs um modelo mais apropriado para reflexão difusa do que o modelo lambertiano. Outros modelos foram criados para capturar a reflectância de superfícies anisotrópicas, como o modelo (WARD, 1992) e o modelo de (POULIN; FOURNIER, 1990).

#### 3.1 Fundamentos Básicos

Apresentaremos nesta seção os fundamentos básicos necessários para compreender os modelos de iluminação local, que incluem um maior estudo sobre reflectância e os dois modelos de reflexão ideais.

#### 3.1.1 Reflectância

Reflectância,  $\rho(x)$ , é a relação entre o fluxo de radiação eletromagnética incidente em uma superfície e o fluxo que é refletido. Formalmente é definida como:

$$\rho(x) = \frac{d\Phi_r(x)}{d\Phi_i(x)} \tag{3.1}$$

 $\rho(x)$  é a fração de energia incidente que é refletida. Em outras palavras, a reflectância indica a capacidade de uma superfície refletir luz.

#### 3.1.2 Reflexão Difusa

Uma superfície com reflexão difusa tem como característica refletir a luz em todas as direções. Este tipo de reflexão ocorre geralmente em superfícies rugosas.

Chamamos de superfícies **lambertianas** aquelas que refletem a luz com a mesma intensidade, em todas as direções, ou seja, possuem reflexão difusa ideal (Fig.9).



Figura 9: Reflexão difusa ideal ou Lambertiana.

Podemos expressar uma reflexão Lambertiana através de uma BRDF. Como a radiância refletida é constante em todas as direções, independente da irradiância, temos:

$$L_r(x,\vec{\omega}) = f_{r,d}(x) \int_{\Omega} dE_i(x,\vec{\omega'}) = f_{r,d}(x)E_i(x)$$

Utilizando a relação de reflectância, podemos escrever:

$$\rho_d(x) = \frac{d\Phi_r(x)}{d\Phi_i(x)} = \frac{L_r(x)dA \int_{\Omega} (\vec{n} \cdot \vec{\omega})d\vec{\omega}}{E_i(x)dA} = \pi f_{r,d}$$

Logo, a BRDF de uma reflexão Lambertiana é uma constante, dada por:

$$f_{r,d} = \frac{\rho_d(x)}{\pi} \tag{3.2}$$

A direção de reflexão em uma superfície Lambertiana é perfeitamente randômica e podemos calculá-la através de uma distribuição uniforme. Dados dois números randômicos uniformes,  $\xi_1$  e  $\xi_2$ , ambos pertencentes ao intervalo [0,1], a direção  $\vec{\omega}_d$  é distribuída como:

$$\vec{\omega}_d = (\theta, \phi) = (\cos^{-1}(\sqrt{\xi_1}), 2\pi\xi_2) \tag{3.3}$$

onde  $(\theta, \phi)$  são coordenadas esféricas para a direção.

### 3.1.3 Reflexão Especular

Uma superfície com reflexão especular tem como principal característica refletir a luz em uma única direção. Quando o ângulo de incidência é igual ao ângulo de reflexão temos o que chamamos de reflexão especular ideal (Fig.10a). Este tipo de reflexão ocorre geralmente em superfícies lisas.



Figura 10: (a) Reflexão especular ideal. (b) Glossy.

Acontece que muitas superfícies possuem imperfeições, o que causa uma dispersão da luz refletida em torno da direção de reflexão ideal, formando um pequeno cone (Fig.10b). Quando isso acontece, a superfície é chamada de *glossy*.

A radiância refletida em uma superfície especular é calculada como:

$$L_r(x, \vec{\omega_s}) = \rho_s(x) L_i(x, \vec{\omega'})$$

Para uma reflexão especular ideal, a direção de reflexão  $\vec{\omega_s}$  é dada por:

$$\vec{\omega_s} = 2(\vec{\omega} \cdot \vec{n})\vec{n} - \vec{\omega'} \tag{3.4}$$

Assim, podemos expressar uma reflexão especular ideal através de uma BRDF, utilizando coordenadas esféricas para a direção:

$$f_{r,s}(x,\vec{\omega'},\vec{\omega}) = 2\rho_s \delta(sen^2\theta' - sen^2\theta)\delta(\phi' - \phi \pm \pi)$$
(3.5)

onde a distribuição delta de Dirac,  $\delta(x)$ , é usada para limitar a direção onde a BRDF não é zero ( $\delta(x)$  não é zero apenas quando x=0). Logo, devemos ter:

$$sen^2\theta' = sen^2\theta$$

$$\phi' = \phi \pm \pi$$

Isso nos garante que o ângulo de incidência é igual o ângulo de reflexão, característica da reflexão especular ideal.

Na realidade, a maioria das superfícies não pode ser descritas através dos modelos ideais de reflexão difusa ou especular. Por isso, vários modelos de iluminação local foram desenvolvidos com o intuito de descrever a aparência dos materiais reais.

Neste trabalho, vamos detalhar dois desses modelos. São eles: o modelo modificado de Phong e o modelo de Schlick.

### 3.2 Modelo Modificado de Phong

Sabemos que o modelo de Phong é totalmente **empírico** e combina uma componente difusa e outra especular. Por não ser um modelo com base física, não podemos utilizá-lo na forma de uma BRDF. Então, se faz necessário modificá-lo para que assim possa ser utilizado na síntese de imagens por simulação (LAFORTUNE; WILLEMS, 1994).

É possível escrevermos uma BRDF muito similar ao modelo original de Phong e que é comumente utilizada. Fazemos isso somando uma componente difusa e outra especular:

$$f_r(x, \vec{\omega'}, \vec{\omega}) = f_{r,d}(x, \vec{\omega'}, \vec{\omega}) + f_{r,e}(x, \vec{\omega'}, \vec{\omega})$$

Aqui,  $f_{r,d}$  é uma BRDF que descreve uma reflexão Lambertiana e  $f_{r,e}$  é uma BRDF que representa uma reflexão especular dispersa em torno da direção de reflexão especular ideal.

Reescrevendo a BRDF temos:

$$f_r(x, \vec{\omega'}, \vec{\omega}) = \rho_d \frac{1}{\pi} + \rho_e \frac{n+2}{2\pi} \cos^n \alpha$$
 (3.6)

onde:

 $\bullet$   $\alpha$  é o ângulo entre a direção de reflexão especular ideal e a direção de reflexão;

- $\rho_d$  é a reflectância difusa;
- $\rho_e$  é a reflectância especular;
- *n* é chamado **expoente especular** (grande valores para *n* resultam em uma maior reflexão especular).

A direção de reflexão é expressa diferentemente para as duas componentes, difusa e especular. Já apresentamos o cálculo da direção para uma reflexão Lambertiana, que é dada por uma distribuição uniforme. Agora, vamos definir a direção de reflexão da componente especular no modelo de Phong, que também é calculada através de uma distribuição uniforme e expressa em coordenadas esféricas  $(\theta, \phi)$ .

Dados dois números randômicos uniformes,  $\xi_1 \in [0,1]$  e  $\xi_2 \in [0,1]$ , a direção  $\vec{\omega'}$  é distribuída como:

$$\vec{\omega'} = (\alpha, \phi) = (\cos^{-1}(\xi_1^{\frac{1}{n+1}}), 2\pi\xi_2)$$

### 3.3 Modelo de Schlick

O modelo de Shilick (SCHLICK, 1994) pode ser considerado um modelo híbrido, ou seja, é baseado em conhecimentos empíricos e físicos. Na verdade, a base física vem de aproximações de alguns modelos de iluminação local, que são baseados na teoria de reflexão (modelo de Cook-Torrance, modelo de He e o modelo de Ward). Schlick identificou os pontos fracos presentes nesses modelos, propôs técnicas de otimização e somando a elas conhecimentos empíricos, desenvolveu um novo modelo de iluminação local.

O modelo possui três parâmetros:

- $F_0 \in [0,1]$ : coeficiente de reflexão especular
- $\sigma \in [0,1]$ : fator de rugosidade ( $\sigma = 0$ : reflexão especular ideal;  $\sigma = 1$ : reflexão Lambertiana)
- $\psi \in [0,1]$ : fator de isotropia ( $\psi = 0$ : anisotrópica;  $\psi = 1$ : isotrópica)

Segundo Schlick, a escolha por esses parâmetros se deu por dois motivos:

- a função que cada parâmetro desempenha pode ser compreendida intuitivamente, e por essa razão, podem ser definidos facilmente;
- os parâmetros podem ser determinados de acordo com dados experimentais (PALIK, 1985).

A BRDF do modelo de Schlick é uma combinação de um fator especular S(u) e um termo que controla a quantidade de reflexão difusa (d), especular (s) e glossy (g). Formalmente define-se como:

$$f_r(x, \vec{\omega'}, \vec{\omega}) = S(u) \left\{ \frac{d}{\pi} + gD(t, v, v', w) + sf_{r,s}(x, \vec{\omega'}, \vec{\omega}) \right\}$$
(3.7)

onde  $f_{r,s}$  é uma BRDF para uma reflexão especular ideal e D(t, v, v', w) é um termo de controle direcional da reflexão glossy. Os parâmetros u, t, v, v' e w são calculados através da orientação da superfície e das direções de incidência e reflexão:

$$\vec{H} = \frac{\vec{\omega} + \vec{\omega'}}{\left\| \vec{\omega} + \vec{\omega'} \right\|}$$

$$u = \vec{\omega} \cdot \vec{H}$$

$$t = \vec{n} \cdot \vec{H}$$

$$v = \vec{\omega} \cdot \vec{n}$$

$$v' = \vec{\omega'} \cdot \vec{n}$$

$$w = \vec{T} \cdot \frac{\vec{H} - (\vec{n} \cdot \vec{H})\vec{n}}{\left| \vec{H} - (\vec{n} \cdot \vec{H})\vec{n} \right|}$$

onde  $\vec{T}$  é um vetor tangente a superfície no ponto x.

O fator especular S(u) é calculado através da equação:

$$S(u) = F_0 + (1 - F_0)(1 - u^5)$$

que é uma aproximação derivada por Schlick do fator de Fresnel, conhecido também como coeficiente de reflexão de Fresnel.

Utilizando o conceito de microfacetas, o termo de controle direcional D(t, v, v', w) é calculado através de dois termos, Z(t) e A(w), e também através de um fator geométrico, G(v). Esses termos são definidos como:

$$Z(t) = \frac{\sigma}{(1 + \sigma t^2 - t^2)^2}$$
$$A(w) = \sqrt{\frac{\psi}{\psi^2 - \psi^2 w^2 + w^2}}$$
$$G(v) = \frac{v}{\sigma - \sigma v + v}$$

Definido esses termos, podemos escrever o fator direcional D(t, v, v', w):

$$D(t, v, v', w) = \frac{G(v)G(v)Z(t)A(w) + 1 - G(v)G(v')}{4\pi vv'}$$

Falta ainda definirmos os três fatores que controlam a quantidade de reflexão difusa, especular e glossy  $(d, s \in g)$ . Schlick sugeriu defini-los em função do fator de rugosidade  $\sigma$  da seguinte forma:

$$g = 4\sigma(1 - \sigma)$$

$$d = \begin{cases} 0 & se \ \sigma < 0.5 \\ 1 - g & sen\~ao \end{cases}$$

$$s = \begin{cases} 1 - g & se \ \sigma < 0.5 \\ 0 & sen\~ao \end{cases}$$

## 4 Traçado de Raios

Apresentados os conceitos da física da luz, sua interação com superfícies e os modelos de iluminação local, podemos agora apresentar o algoritmo de **traçado de raios** (ray tracing). Na verdade, a designação de traçado de raios aplica-se a um conjunto de algoritmos que foram desenvolvidos a partir do trabalho de Appel (APPEL, 1968). Em 1980, a técnica de traçado de raios foi estendida por Whitted (WHITTED, 1980) para o cálculo de iluminação. Whitted propôs o traçado de raios recursivo, que também é chamado de traçado de raios clássico.

O objetivo deste capítulo é apresentar o algoritmo de traçado de raios clássico e os fundamentos necessários para a sua compreensão.

#### 4.1 Raio

Antes de apresentarmos o algoritmo de traçado de raios, como o próprio nome nos indica, precisamos saber o que é um raio.

Um raio, no nosso contexto, nada mais é que uma semi-reta no espaço 3D descrita em sua forma paramétrica. Assim, um raio, r, tem a forma:

$$r(x,\vec{\omega}) = o + \vec{\omega} \ t \tag{4.1}$$

onde o é a origem do raio,  $\vec{\omega}$  é a direção do raio e t é o parâmetro da descrição paramétrica de uma reta, que aqui representa a distância percorrida pelo raio (Fig.11).

Sendo a origem  $o = (x_o, y_o, z_o)$  e o vetor direção  $\vec{\omega} = (x_d, y_d, z_d)$ , os pontos (x, y, z)

que estão sobre esta reta podem ser calculados via o parâmetro t, da seguinte forma:

$$x = x_o + t x_d$$

$$y = y_o + t y_d$$

$$z = z_o + t z_d$$

$$(4.2)$$

Logicamente, nos interessam apenas os valores positivos de t, pois assim teremos pontos à frente da origem do raio.

Raios são utilizados para simular o movimento da luz de um ponto a outro do espaço e como conseqüência também interagem com os objetos. Quando isso ocorre, a direção do raio é modificada em função da sua direção, da posição do objeto e das propriedades do material que esse é constituído. Lembramos que o modelo adotado neste trabalho para representar a luz é o da ótica geométrica, o que torna a utilização de raios bastante adequada.

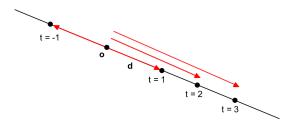


Figura 11: Exemplo de um raio com origem o, direção d e diferentes valores para o parâmetro t.

### 4.2 Traçado de Raios Clássico

Atualmente o método que permite a geração de imagens mais realistas é o traçado de raios. Utilizando-se apenas o algorimto clássico proposto por Whitted é possível a obtenção de cenas de alta qualidade com inter-reflexões múltiplas, transparências e sombras de forma bastante elegante e simples (Fig.12). Apresentamos nesta seção o algoritmo clássico de traçado de raios.

O algoritmo de traçado de raios considera raios com origem na posição do observador e cada um desses raios será traçado em direção a cena, passando pelo ponto correspondente a um *pixel* da imagem sobre o plano de projeção, para onde o observador está olhando (Fig.13). O objetivo então é definir a cor de cada um desses *pixels*, o que é feito calculando

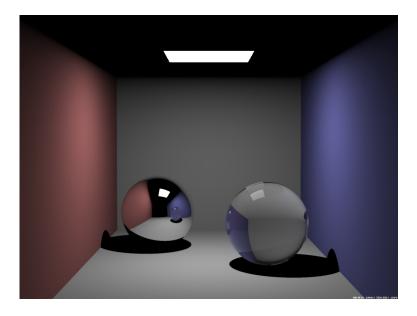


Figura 12: Image sintetizada através do algoritmo clássico de traçado de raios, possuindo inter-reflexões múltiplas, sombras e transparência. Fonte: (JENSEN, 2001).

a radiância refletida na direção do raio. Raios com origem no observador e que passam pelo pixel são chamados raios primários.

Para calcularmos a radiância de um raio primário, precisamos encontrar o primeiro objeto que o raio intercepta, ou seja, o menor valor positivo para o parâmetro t. A interseção do raio com os objetos da cena é uma questão importante e será tratada mais adiante. Se o raio não interceptar nenhum objeto, a cor atribuída ao *pixel* por onde o raio passa é a cor de fundo da cena. É importante notar que o algoritmo realiza o cálculo de superfícies visíveis implicitamente, pois somente o primeiro objeto interceptado terá sua cor atribuída ao *pixel*.

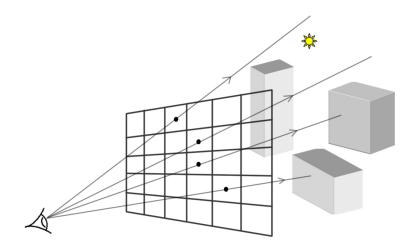


Figura 13: Algoritmo de traçado de raios. Raios com origem no observador passam pelos respectivos *pixels* no plano de projeção. A cor de cada *pixel* é a cor do objeto interceptado pelo raio ou a cor de fundo caso não exista interseção.

Encontrado a primeira interseção, devemos calcular o ponto onde esta ocorreu. Então, conhecendo esse ponto x, a normal  $\vec{n}$ , bem como a BRDF,  $f_r$ , podemos calcular a iluminação para cada fonte de luz e a radiância refletida na direção do raio. Para uma fonte de luz pontual com potência  $\Phi_l$  e posição p, a radiância refletida,  $L_r$ , é calculada como:

$$L_r(x,\vec{\omega}) = f_r(x,\vec{\omega},\vec{\omega'}) \frac{\vec{\omega'} \cdot \vec{n}}{\|p - x\|^2} V(x,p) \frac{\Phi_l}{4\pi}$$
(4.3)

onde  $\vec{\omega'}$  é um vetor unitário na direção da fonte de luz:

$$\vec{\omega'} = \frac{p - x}{\|p - x\|}$$

e V(x,p) é chamada **função de visibilidade**. Esta função define se o ponto de interseção é visível para uma dada fonte de luz, ou seja, se este ponto recebe ou não energia luminosa proveniente desta (iluminação direta). Ela é calculada traçando um raio do ponto de interseção x até a posição da fonte luminosa. Este raio recebe o nome de **raio de sombra**. Se o raio de sombra intercepta algum objeto entre x e a fonte de luz, V(x,p) = 0, caso contrário, V(x,p) = 1.

Para superfícies perfeitamente especulares, o algoritmo pode calcular a reflexão especular traçando um raio na direção que ocorre a reflexão,  $\vec{\omega_s}$  (Eq. 3.4). O cálculo da radiância refletida para esse raio é exatamente igual ao cálculo para os raios primários, caracterizando assim o traçado de raios recursivo.

É importante notar que o traçado de raios lança raios no sentido contrário ao que ocorre na realidade, ou seja, os raios são traçados do observador em direção às fontes de luz. Fisicamente, essas fontes emitem inúmeros fótons que interagem com os objetos e são refletidos na nossa direção. Acontece que apenas um pequena fração desses fótons atingem nossos olhos, contribuindo efetivamente para a formação da imagem. Simular esse processo não seria viável. Por isso é feita a inversão do sentido para o traçado dos raios, que é possível graças a duas propriedades:

- no vácuo, a radiância é constante ao longo de uma linha reta;
- o espalhamento da luz em superfícies é simétrico (Eq. 2.11).

Logo, essa inversão de sentido não causa nenhuma alteração na geometria envolvida, gerando a mesma imagem que o sentido real.

A seguir, apresentamos o traçado de raios clássico em termos algorítimicos.

### 4.2.1 Algoritmo

O algoritmo de traçado de raios é bastante simples e possui duas funções. A função trace() é responsável por traçar os raios em direção a cena e encontrar o primeiro objeto interceptado, bem como o ponto de interseção e a normal neste ponto. Encontrada uma interseção, a função trace() chama a função de colorização shade(), que tem a responsabilidade de calcular a radiância refletida (cor) na direção do raio. Isso é feito traçando raios de sombra para testar a visibilidade do ponto, em seguida, adicionando a contribuição de cada fonte de luz e traçando raios refletidos/refratados para as superfícies perfeitamente especulares. É importante dizer que a função shade() depende do modelo de iluminação local utilizado para calcular a iluminação no ponto.

```
início
   para cada pixel
        selecione um raio com origem no observador até o pixel;
        pixel cor = trace(raio);
trace(raio)
{
    encontre o primeiro objeto interceptado;
   calcule o ponto de interseção e a normal neste ponto;
    cor = shade(ponto, normal);
   retorne cor;
}
shade(ponto, normal)
   para cada fonte de luz
       trace um raio de sombra para testar a visibilidade do ponto;
          se o ponto é visível
           cor = cor + iluminação direta;
    se a superfície é especular
       cor = cor + trace(raio refletido/refratado);
   retorne cor;
}
```

Algortimo 4.2.1: Traçado de Raios Clássico.

É notável a simplicidade e elegância do algoritmo de traçado de raios em comparação a complexidade dos algoritmos tradicionais de síntese de imagens por emulação. Porém, apesar dessas características, o traçado de raios possui uma desvangatem: alta complexidade computacional, o que impede a sua utilização em aplicações em tempo real. Isto se deve principalmente ao cálculo das interseções dos raios com os objetos da cena.

### 4.3 Cálculo das Interseções

Esta seção tem o objetivo de apresentar como é realizado o cálculo das interseções, especificamente com formas geométricas simples, as quais chamamos de **primitivas**.

Primitivas são utilizadas na sua forma simples ou combinadas para representar os objetos do mundo real que estamos querendo visualizar. São exemplos comuns de primitivas: esfera, triângulo, plano, disco, caixa e cilindro.

### **4.3.1** Esfera

Seja  $E_c$  o centro da esfera com coordenadas  $(x_c, y_c, z_c)$ , r o raio da esfera e P um ponto qualquer da superfície esférica, cujas coordenadas são (x, y, z). A superfície esférica é o lugar geométrico dos pontos tais que:

$$|E_c - P| - r = 0$$

ou na sua equação implícita:

$$(x - x_c)^2 + (y - y_c)^2 + (z - z_c)^2 - r^2 = 0$$
(4.4)

Desenvolvendo a Equação 4.4 temos:

$$(x^{2} - 2x_{c}x + x_{c}^{2}) + (y^{2} - 2y_{c}y + y_{c}^{2}) + (z^{2} - 2z_{c}z + z_{c}^{2}) - r^{2} = 0$$

$$(4.5)$$

O que devemos fazer agora é substituir a equação do raio na Equação 4.5 e calcularmos o valor do parâmetro t que satisfaz a equação. Assim, podemos encontrar, se existe, o ponto de interseção.

Introduzindo a Equação 4.2, que nos dá as expressões paramétricas para cada uma das coordenadas dos pontos sobre a reta que o raio pertence, temos:

$$(x_o + x_d t)^2 - 2x_c(x_o + x_d t) + x_c^2 + (y_o + y_d t)^2 - 2y_c(y_o + y_d t) + y_c^2 + (z_o + z_d t)^2 - 2z_c(z_o + z_d t) + z_c^2 - r^2 = 0$$

$$(4.6)$$

Desenvolvendo:

$$(x_o^2 + 2x_o \ x_d t + x_d^2 \ t^2) - 2x_c(x_o + x_d t) + x_c^2 +$$

$$(y_o^2 + 2y_o \ y_d t + y_d^2 \ t^2) - 2y_c(y_o + y_d t) + y_c^2 +$$

$$(z_o^2 + 2z_o \ z_d t + z_d^2 \ t^2) - 2z_c(z_o + z_d t) + z_c^2 - r^2 = 0$$

$$(4.7)$$

e reagrupando:

$$t^{2} (x_{d}^{2} + y_{d}^{2} + z_{d}^{2}) +$$

$$2t [x_{d}(x_{o} - x_{c}) + y_{d}(y_{o} - y_{c}) + z_{d}(z_{o} - z_{c})] +$$

$$(x_{o} - x_{c})^{2} + (y_{o} - y_{c})^{2} + (z_{o} - z_{c})^{2} - r^{2} = 0$$

$$(4.8)$$

Chegamos a uma equação do  $2^{\circ}$  grau em t, da forma:

$$At^2 + 2Bt + C = 0 (4.9)$$

onde os coeficientes  $A, B \in C$  são:

$$A = x_d^2 + y_d^2 + z_d^2$$

$$B = x_d(x_o - x_c) + y_d(y_o - y_c) + z_d(z_o - z_c)$$

$$C = (x_o - x_c)^2 + (y_o - y_c)^2 + (z_o - z_c)^2 - r^2$$
(4.10)

A solução da Equação 4.9 é:

$$t = \frac{-B \pm \sqrt{B^2 - 4AC}}{2A}$$

que nos leva as seguintes interpretações:

- Se  $B^2 4AC < 0$ , as duas raízes da equação são imaginárias e, portanto, o raio não intercepta a esfera em nenhum ponto.
- Se  $B^2 4AC = 0$ , existe uma raiz dupla e o raio intercepta a esfera em apenas um ponto, ou seja, o raio é tangente a esfera.
- Se B<sup>2</sup> 4AC > 0, a linha reta sobre a qual o raio existe intercepta a esfera em
  dois pontos correspondentes aos dois valores de t. Nos interessa determinar o ponto
  de interseção mais próximo da origem do raio, que corresponde ao menor valor
  positivo de t.

#### 4.3.2 Plano

A equação implícita de um plano é dada por:

$$A x + B y + C z + D = 0 (4.11)$$

em que  $A^2 + B^2 + C^2 = 1$  e D é a menor distância da origem até o plano.

Substituindo x, y, z na Equação 4.11 pela expressão paramétrica de cada coordenada (Eq. 4.2), temos:

$$A(x_o + x_d t) + B(y_o + y_d t) + C(z_o + z_d t) + D = 0$$
(4.12)

Explicitando o parâmetro t, obtemos a equação que nos dá o seu valor no ponto de interseção de um raio com um plano:

$$t = \frac{-(A x_o + B y_o + C z_o + D)}{A x_d + B y_d + C z_d}$$
(4.13)

Primeiro, devemos calcular o produto escalar  $\vec{N} \cdot \vec{\omega}$ , onde  $\vec{N}$  é a normal do plano dada por  $\vec{N} = (A, B, C)$  e  $\vec{\omega}$  a direção do raio. Se o resultado for nulo, o raio é paralelo ao plano e não o intercepta. Caso contrário, existe interseção entre a linha reta que o raio pertence e o plano. Calculamos então o valor de t, o que nos leva s seguintes interpretações:

- Se t < 0, o plano está no sentido contrário ao sentido do raio e não existe interseção.
- ullet Se t=0, a origem do raio se encontra sobre o plano e não existe interseção.
- Se t > 0, o raio intercepta o plano. Substituímos o valor de t na equação do raio e encontramos assim o ponto onde ocorre a interseção.

### 4.3.3 Triângulo

Para calcularmos a interseção de um raio com um triângulo de vértices  $a, b \in c$ , devemos primeiro verificar se existe interseção entre o raio e o plano que contêm o triângulo.

Se existe a interseção, o próximo passo agora é verificar se o ponto onde a interseção ocorre encontra-se dentro do triângulo. Um método é utilizar as coordenadas baricêntricas (SHIRLEY, 2000).

Podemos descrever o plano que contém o triângulo utilizando coordenadas baricêntricas:

$$\mathbf{p}(\alpha, \beta, \gamma) = \alpha a + \beta b + \gamma c$$

onde  $\alpha + \beta + \gamma = 1$ .

As coordenadas baricêntricas são definidas para todos os pontos do plano e o ponto **p** se encontra dentro do triângulo se e somente se:

$$0 < \alpha, \beta, \gamma < 1$$

### 4.3.4 Superfícies Quádricas

A equação implícita de uma superfície quádrica é dada por:

$$F(x,y,z) = ax^{2} + 2bxy + 2cxz + 2dx + ey^{2} + 2fyz + 2gy + hz^{2} + 2iz + j = 0$$
 (4.14)

a qual podemos expressar de uma forma mais compacta, através de uma matriz 4 x 4, Q, com os coeficientes:

$$F(x, y, z) = \begin{bmatrix} x & y & z & 1 \end{bmatrix} \begin{bmatrix} a & b & c & d \\ b & e & f & g \\ c & f & h & i \\ d & g & i & j \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = X Q X^{T} = 0$$

Abaixo, definimos algumas superfícies quádricas comumente utilizadas (Fig.14), assumindo a posição padrão, ou seja, origem em (0,0,0) e tendo o eixo y como linha de simetria. Definimos os valores dos coeficientes da matriz Q a partir das equações implícitas (KIRK, 1992).

**Elipsóide:** com eixos de tamanho 2a, 2b e 2c ao longo das principais direções. Uma esfera de raio r é simplesmente um caso especial de elipsóide, onde r = a = b = c.

$$F(x,y,z) = \frac{x^2}{a^2} + \frac{y^2}{b^2} + \frac{z^2}{c^2} - 1 = X \begin{bmatrix} \frac{1}{a^2} & 0 & 0 & 0\\ 0 & \frac{1}{b^2} & 0 & 0\\ 0 & 0 & \frac{1}{c^2} & 0\\ 0 & 0 & 0 & -1 \end{bmatrix} X^T = 0$$

Cilindro elíptico: com eixos principais de tamanho 2a e 2c. Um cilindro circular de

raio r é simplesmente um caso especial em que r = a = c.

$$F(x,y,z) = \frac{x^2}{a^2} + \frac{z^2}{c^2} - 1 = X \begin{bmatrix} \frac{1}{a^2} & 0 & 0 & 0\\ 0 & 0 & 0 & 0\\ 0 & 0 & \frac{1}{c^2} & 0\\ 0 & 0 & 0 & -1 \end{bmatrix} X^T = 0$$

Cone elíptico: com eixos principais de tamanho 2a e 2c e uma distância unitária do ápice. Um cone circular é um caso especial, onde r = a = c.

$$F(x, y, z) = \frac{x^2}{a^2} - y^2 + \frac{z^2}{c^2} = X \begin{bmatrix} \frac{1}{a^2} & 0 & 0 & 0\\ 0 & -1 & 0 & 0\\ 0 & 0 & \frac{1}{c^2} & 0\\ 0 & 0 & 0 & 0 \end{bmatrix} X^T = 0$$

Parabolóide elíptico: com eixos principais de tamanho 2a e 2c e duas vezes a distância focal f.

$$F(x,y,z) = \frac{x^2}{a^2} + \frac{z^2}{c^2} - 4fy = X \begin{bmatrix} \frac{1}{a^2} & 0 & 0 & 0\\ 0 & 0 & 0 & 0\\ 0 & 0 & \frac{1}{c^2} & 0\\ 0 & -2f & 0 & 0 \end{bmatrix} X^T = 0$$

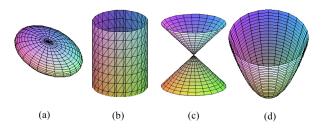


Figura 14: Superfícies Quádricas: (a) Elipsóide (b) Cilindro elíptico (c) Cone elíptico (d) Parabolóide elíptico.

Para encontrarmos a interseção de um raio com um superfícies quádrica basta substituírmos a equação que nos dá as expressões paramétricas para cada uma das coordenadas dos pontos sobre a reta que o raio pertence (Eq. 4.2) na equação implícita da superfície quádrica (Eq. 4.14). O resultado é uma equação de  $2^{\circ}$  grau em t, como acontece nos cálculos da interseção entre um raio e uma esfera. Devemos proceder da mesma forma, ou seja, calculamos o discriminante  $B^2 - 4AC$  e fazemos as análises em cima desse resultado. Se o discriminante é zero ou positivo, calculamos o valor de t, o que nos dá

o(s) ponto(s) onde ocorre a interseção. Abaixo, os cálculos dos coeficientes A, B e C.

Primeiro, escrevemos a Equação 4.2 na forma matricial:

$$X = X_0 + Rt$$

onde  $X_o = [x_o, y_o, y_o]$ , coordenadas da origem do raio e  $R = [x_d, y_d, z_d]$ , coordenadas da direção do raio.

Substituíndo X na equação implícita de uma superfície quádrica em sua forma matricial temos:

$$F(x, y, z) = XQX^{T} = [X_o + Rt]Q[X_o + Rt]^{T}$$

$$= X_oQX_o^{T} + t(X_oQR^{T} + RQX_o^{T}) + t^{2}RQR^{T}$$

$$= At^{2} + Bt + C = 0$$

onde:

$$A = x_d (x_d Q_{11} + y_d (Q_{12} + Q_{21}))$$

$$+ y_d (y_d Q_{22} + z_d (Q_{23} + Q_{32}))$$

$$+ z_d (z_d Q_{33} + x_d (Q_{13} + Q_{31}))$$

$$B = x_d ((z_o(Q_{13} + Q_{31}) + y_o(Q_{12} + Q_{21}) + x_o(2Q_{11})) + (Q_{14} + Q_{41})$$

$$y_d ((z_o(Q_{23} + Q_{32}) + x_o(Q_{12} + Q_{21}) + y_o(2Q_{22})) + (Q_{24} + Q_{42})$$

$$z_d ((y_o(Q_{23} + Q_{32}) + x_o(Q_{13} + Q_{31}) + z_o(2Q_{33})) + (Q_{34} + Q_{43})$$

$$C = x_o (x_o Q_{11} + y_o (Q_{12} + Q_{21}) + z_o (Q_{13} + Q_{31}) + (Q_{14} + Q_{41}))$$

$$y_o (y_o Q_{22} + z_o (Q_{23} + Q_{32}) + (Q_{24} + Q_{42}))$$

$$z_o (z_o Q_{33} + (Q_{34} + Q_{43}))$$

# 4.4 Limitações do Traçado de Raios Clássico

Vimos anteriormente que a alta demanda computacional, devida principalmente ao cálculo das interseções, é um ponto negativo no algoritmo de traçado de raios. Isso limita sua utilização em aplicações que necessitem de um certo grau de interatividade, já que as imagens para serem geradas, requerem uma parcela de tempo considerável. Porém, outras limitações são encontradas no algoritmo clássico e serão discutidas nesta seção.

Um problema bastante visível em imagens sintetizadas através do algoritmo clássico é a presença do que chamamos em Computação Gráfica de aliasing, que é causado pela baixa frequência de amostragem relativa as altas frequências da cena que estamos querendo visualizar. Aqui, a frequência de amostragem nada mais é que a frequência de raios lançados por pixel, que pode ser muito inferior à que seria necessária para capturar todos os detalhes da cena. Isso gera um aspecto serrilhado na imagem, em lugar dos objetos materem seu aspecto retilínio ou curvo.

Outro problema é quanto a precisão direcional dos raios, o que resulta em imagens possuindo sombras e reflexos com contornos extremamente bem definidos e precisos. Na realidade, esses contornos não são verdadeiramente tão bem definidos e as sombras possuem graduações, as quais chamamos de umbra e penumbra. Isto se deve novamente a baixa frequência de amostragem, aqui ligada a quantidade de raios sombra e de reflexão.

No caso da reflexão, esse problema é maior. Lembramos que para superfícies perfeitamente especulares, o algortimo traça um raio na direção de reflexão especular ideal. Dessa forma, apenas nessas superfícies a iluminação indireta pode ser calculada, o que caracteriza o traçado de raios clássico como não sendo um algoritmo de iluminação global completo. Isso porque a iluminação indireta em superfícies difusas não pode ser avaliada.

As limitações do traçado de raios clássico estão, em sua maioria, ligadas a baixa frequência de amostras, ou seja, o número de raios, tanto raios primários, secundários ou raios de sombra. O aumento dessas amostras é possível e é feito utilizando-se o método de Monte Carlo.

# 5 Traçado de Raios Monte Carlo

Neste capítulo, apresentamos o método de Monte Carlo e o conceito de integração Monte Carlo, fundamentos básicos para a compreensão do Traçado de Raios Monte Carlo. Em seguida, apresentamos a forma que o método é utilizado no nosso contexto e dois algoritmos, extensões do traçado de raios que fazem uso do método: **traçado de raios distribuído** (distributed ray tracing) e **traçado de caminhos** (path tracing).

#### 5.1 Método de Monte Carlo

Método Monte Carlo é um método estatístico utilizado em simulações estocásticas com diversas aplicações em áreas como a física, matemática, biologia e computação. O nome Monte Carlo é uma alusão ao famoso cassino de Monte Carlo, devido a natureza aleatória e probabilística do método. Embora existam registros da utilização de técnicas Monte Carlo há muito tempo, o crédito pela criação cabe ao matemático Stanislaw Ulam, que trabalhou junto com John von Neumann no projeto Manhattan, durante a Segunda Guerra Mundial. As aplicações mais comuns são para a avaliação de integrais, que veremos a seguir.

## 5.1.1 Integração Monte Carlo

Dada uma função f(x) que desejamos integrar sobre o domínio unidimensional de a até b:

$$I = \int_{a}^{b} f(x)dx$$

Um caminho para avaliar essa intergal é calcular o valor médio da função sobre o intervalo [a,b] e multiplicá-lo pelo comprimento do intervalo, dado por b-a. Desde modo, podemos calcular a média dos valores de f(x) para N amostras  $\xi_1, \xi_2, ..., \xi_N$ , onde  $\xi_{1...N}$  são números randômicos distribuídos uniformemente no intervalo [a,b]. Assim, a média de f(x) pode ser estimada pela média das amostras e podemos escrever:

$$I_m = \frac{(b-a)}{N} \sum_{i=1}^{N} f(\xi_i)$$
 (5.1)

onde  $I_m$  é a estimativa Monte Carlo para a integral. Quando aumentamos o número de amostras N, mais exata se torna essa estimativa e o limite é dado por:

$$\lim_{N \to \infty} I_m = I$$

ou seja, quanto maior o núemro de amostras, a estimativa Monte Carlo para a integral, Im, converge para o valor real, I.

A convergência é de ordem  $\frac{1}{\sqrt{N}}$ , porém para integrais de alta dimensão a convergência se mantém a mesma e é melhor que de outros métodos.

## 5.2 Equação de Rendering

Kajiya formulou a equação integral que generaliza o problema de todo algoritmo de iluminação global: o cálculo do transporte da luz. Essa equação é chamada **equação de** *rendering* (KAJIYA, 1986), dada por:

$$L_T(x,\vec{\omega}) = L_e(x,\vec{\omega}) + L_r(x,\vec{\omega}) \tag{5.2}$$

e diz que a radiância total emitida,  $L_T$ , por um ponto x na direção  $\vec{\omega}$  é igual a radiância auto-emitida,  $L_e$ , pelo ponto nessa direção mais a radiância refletida,  $L_r$  pelo ponto nessa direção.

Da seção sobre BRDF, sabemos que a radiância refletida por um ponto em uma direção pode ser calculada pela Equação 2.10. Assim, podemos reescrever a Equação 5.2 da seguinte forma:

$$L_T(x,\vec{\omega}) = L_e(x,\vec{\omega}) + \int_{\Omega} f_r(x,\vec{\omega'},\vec{\omega}) \ L_i(x,\vec{\omega'}) \ cos\theta \ d\vec{\omega'}$$

Logo, o que os algortimos de iluminação global fazem é obter uma solução aproximada para essa equação. A solução proposta por Kajiya é o algoritmo de traçado de caminhos (path tracing), que será apresentado mais adiante.

## 5.3 Traçado de Raios Distribuído

O traçado de raios distribuído foi introduzido por (COOK; PORTER; CARPENTER, 1984). Consiste em distribuir estocasticamente os raios ao longo do ângulo sólido que envolve a direção de interesse. Isso equivale a um aumento da frequência de amostragem, que resulta em uma melhor aproximação da equação integral que descreve o transporte da luz, apresentada anteriormente.

Por exemplo, para calcular a iluminação indireta de uma superfície especular, o método não considera apenas um raio secundário traçado na direção de reflexão especular ideal. Em vez disso, distribui estocasticamente vários raios ao redor dessa direção e leva em consideração a radiância retornada por cada um deles. Dessa forma, conseguimos capturar uma reflexão especular borrada, a qual chamamos de glossy e não mais apenas um reflexão especular com contornos extremamente bem definidos (Fig.15). Essa distribuição também é utilizada em torno da direção de refração, o que permite capturar a translucência (Fig.16).

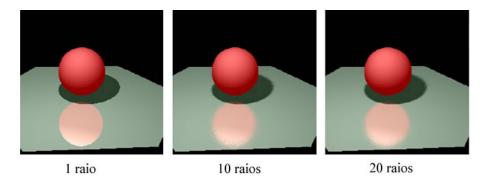


Figura 15: Reflexão borrada para diferentes números de raios distribuídos em torno da direção de reflexão ideal. Fonte: (MARTIN, 2007).

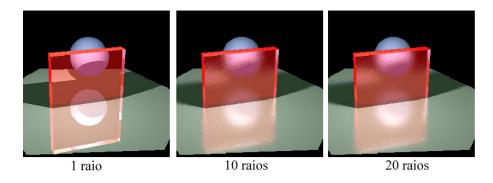


Figura 16: Translucência para diferentes números de raios distribuídos em torno na direção de refração. Fonte: (MARTIN, 2007).

Os raios de sombra também são distribuídos sobre a área das fontes de luz. Assim, ao

invés de traçarmos um único raio de sombra para avaliarmos a visibilidade de um ponto, vários raios são estocasticamente distribuídos. Esses raios são direcionados para posições aleatórias nas fontes de luz. Desde modo, as diferentes graduações de sombra (umbra e penumbra) podem ser calculadas. A umbra ocorre quando um ponto não é visível por determinada fonte de luz e temos total ausência de energia luminosa. Já a penumbra ocorre quando o ponto é parcialmente visível. A intensidade luminosa devido a uma fonte de luz neste estado é proporcional ao angulo sólido da porção visível da luz.

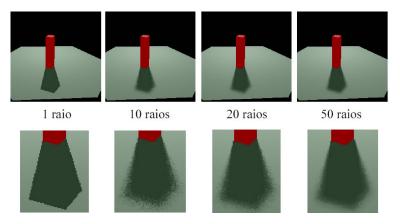


Figura 17: Penumbra para diferentes números de raios de sombra. Fonte: (MARTIN, 2007).

Outra importante caracterísitica do método é a utilização de múltiplos raios distribuídos no espaço do *pixel*. Esses raios são utilizados para amostrar as fontes de luz presentes na cena. A radiância total para um *pixel* é a média da radiância retornada por cada raio lançado através dele. Isso pode ser visto como uma técnica *anti-aliasing*.

Considerando um modelo de câmera de lente fina, Cook inclui, por meio desta técnica, efeitos utilizados em fotografia como profundidade de campo (depth field) e borrão de movimento (motion blur).

O efeito de profundidade de campo pode ser conseguido distribuindo os raios através de uma lente colocada em frente ao plano da imagem. Primeiro é traçada uma linha que vai do *pixel* a um ponto escolhido no círculo definido pela lente. O raio primário é traçado a partir deste ponto da lente, passando pelo ponto focal (Fig.18). Esse processo é repetido N vezes para cada *pixel*.

O efeito borrão de movimento ocorre em uma fotografia quando os objetos da cena estão em movimento. A quantidade de borramento é proporcional à velocidade que se encontra o objeto. No traçado de raios destribuído, o efeito é conseguido distribuíndo os raios primários no tempo. Para cada raio, em um determinado instante de tempo, é necessário determinar a posição dos objetos em movimento.

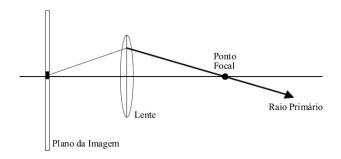


Figura 18: Profundidade de campo: lente colocada em frente ao plano da imagem. O raio primário tem origem em um ponto no círculo definido pela lente e passa pelo ponto focal.

Abaixo, exemplos de imagens sintetizadas contendo os efeitos de profundiade de campo (Fig.19) e borrão de movimento (Fig.20).

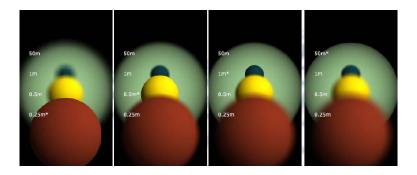


Figura 19: Profundidade de campo para diferentes valores de distância focal. Fonte: (BUSS, 2003).



Figura 20: Imagen com borrão de movimento. Os raios são distribuídos em instantes de tempo diferentes. Fonte: (BUSS, 2003).

Vimos que o traçado de raios distribuído (Fig.21) torna possível a adição de vários efeitos nas imagens, o que contribui para o aumento do realismo, além de ser considerado um método *anti-aliasing*. Porém, continuamos com o mesmo problema observado no traçado de raios clássico, a iluminação indireta entre superfícies difusas não pode ser calculada. O método proposto por Kajiya torna isso possível.

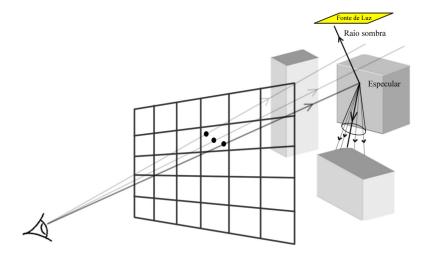


Figura 21: Algoritmo de traçado de raios distribuído. Vários raios são lançados por *pixel*. Os raios de sombra são distribuídos sobre a área da fonte de luz. Para superfícies especulares, são distribuídos raios de reflexão em torno da direção de reflexão ideal.

# 5.4 Traçado de Caminhos

O método de traçado de caminhos foi apresentado por (KAJIYA, 1986) com o propósito de aproximar a solução da equação de rendering utilizando métodos de Monte Carlo para a avaliação da integral que define o transporte da luz. É baseado no algoritmo de traçado de raios distribuído, apresentado anteriormente, com a diferença que usa amostras estocásticas para amostrar todos os caminhos possíveis realizados pela luz. É considerado um algoritmo de iluminação global completo.

Uma importante característica no traçado de caminhos é a utilização de apenas um raio refletido para estimar a iluminação indireta, enquanto um ou mais raios são utilizados para amostrar as fontes de luz. Segundo Kajiya, é melhor focalizar os cálculos em eventos que se submetem a poucas reflexões. Traçando apenas um raio para calcular a iluminação indireta, o método assegura, no mínimo, o mesmo esforço que é investido em superfícies vistas diretamente pelo observador. Para calcular uma estimativa precisa para o pixel, é necessário a utilização de vários raios primários.

Isso se torna um problema, pois a variação do número de amostras gera ruídos nas imagens sintetizadas. A presença de ruído está diretamente ligada a utilização de poucos raios por pixel (Fig.22).

Com certeza, esse é o maior problema do traçado de caminhos e dos algoritmos que utilizam integração Monte Carlo para aproximar o cálculo da iluminação. Lembramos que a ordem de convergência do método de integração Monte Carlo é de  $\frac{1}{\sqrt{N}}$ , onde N

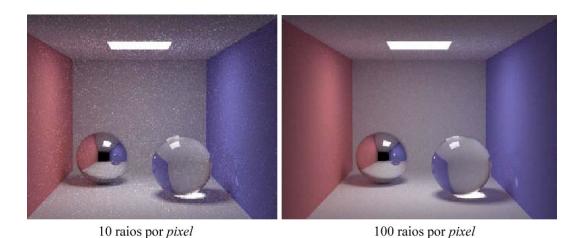


Figura 22: Presença de ruído em imagens geradas através do traçado de caminhos. A utilização de 100 raios/pixel ainda assim não é suficiente para eliminar por completo o ruído na imagem. Fonte: (JENSEN, 2001).

é o número de amostas. Por exemplo, para reduzirmos o erro pela metade, precisamos quadruplicar o número de amostras. O erro aqui nada mais é que o ruído presente na imagem. Porém, a quantidade de raios primários a serem utilizados para eliminação do ruído depende da complexidade da iluminação e dos modelos presentes na cena.

A seguir, apresentamos o traçado de caminhos em termos algorítmicos e a estratégia utilizada para a geração dos raios de reflexão.

## 5.4.1 Algoritmo

O algoritmo de traçado de caminhos é muito semelhante ao algoritmo de traçado de raios clássico e ao traçado de raios distribuído. A diferença principal está na função shade(), onde todos os raios de reflexão são traçados e não apenas os de reflexão especular. Outra diferença que deve ser observada em comparação ao traçado de raios clássico é a utilização de mais de um raio por *pixel*, que podem ser distribuídos estocasticamente no tempo e na área do *pixel*. Os raios sombra são traçados em direção a uma posição aleatória nas fontes de luz, para calcular regiões de umbra e penumbra.

A cada interseção, precisamos gerar um raio de reflexão aleatório. Esse raio pode ser de três tipos: difuso, especular ou de transmissão. Para definirmos que tipo de raio deve ser gerado, utilizamos uma técnica chamada Roleta Russa (SPANIER; GELBARD, 1969), introduzida na Computação Gráfica por (ARVO; KIRK, 1990). A idéia da técnica é permitir que sejamos capazes de evitar aleatoriamente a avaliação de direções de reflexão e mesmo assim, obter o resultado o correto.

Para isso, utilizamos as propriedades do material que consitute a superfície. Seja  $k_d, k_s, k_t$  coeficientes de reflexão difusa, especular e de transparência respectivamente. Seja  $k_{total} = k_s + k_s + k_t$  e  $\xi$  um número aleatório entre  $[0, k_{total}]$ .

- Se  $\xi < k_d$ , é gerado um raio de reflexão difusa. A direção é calculada utilizando a Equação 3.3;
- Se  $\xi < k_d + k_s$ , é gerado um raio de reflexão especular, e a direção é dada pela Equação 3.4
- Senão, é gerado um raio de transmissão.

```
início
   para cada pixel
    cor = 0;
     para cada amostra
        selecione um raio com origem no observador até uma posição aleatória no pixel;
        selecione um tempo aleatório e uma posição na lente para o raio;
        cor = cor + trace(raio);
     pixel-cor = cor / número de amostras
trace(raio)
{
   encontre o primeiro objeto interceptado;
   calcule o ponto de interseção e a normal neste ponto;
   cor = shade(ponto, normal);
   retorne cor;
}
shade(ponto, normal)
   cor = 0;
   para cada fonte de luz
     trace um raio de sombra até uma posição aleatória na fonte de luz
       se o ponto é visível
         cor = cor + iluminação direta;
    cor = cor + trace(raio aleatório);
    retorne cor;
}
```

Algortimo 5.4.1: Traçado de caminhos.

# 6 Implementação e Resultados

Neste capítulo, serão discutidos os detalhes da construção de um sintetizador de imagens que utiliza traçado de raios. Veremos as estuturas necessárias, seus relacionamentos, o funcionamento geral da implementação e os métodos utilizados. Em seguida, o capítulo apresenta um conjunto de imagens que demonstram os resultados obtidos e uma descrição dos problemas e limitações encontrados.

# 6.1 Implementação

Para a implementação do sintetizador, a linguagem de programação utilizada foi o JAVA, por se tratar de uma linguagem orientada a objetos, livre e independente de plataforma. A divisão das estuturas em classes, além de facilitar o desenvolvimento, permite a inclusão de novas funcionalidades de forma bastante simples.

#### 6.1.1 Estruturas básicas

Devemos criar uma classe para representar vetores em três dimensões, também utilizada para representar pontos no espaço 3D (classe Vector3D). A classe deve prover todos métodos para a realização de operações envolvendo vetores, como adição, subtração, produto vetorial, produto escalar, norma etc. Os atributos são as coordenadas  $x, y \in z$  do vetor.

Outra classe importante a ser criada é uma classe que represente um espaço de cor. O mais utilizado é o espaço **RGB**, adotado neste trabalho e definido pela classe **RGBColor**. Também deve prover todos os métodos para a realização de operaões aritméticas envolvendo cores, como adição, subtração, multiplicação e divisão. Têm como atributos números (double) que representam as três componentes: R, G e B.

Uma classe muito importante para o traçado de raios, porém bastante simples, é a

que define um raio (classe Ray). Possui como atributos:

- Vector3D origem do raio;
- Vector3D direção do raio.

e deve provêr um método, que dado um escalar t, retorne o ponto destino para o raio.

```
public Vector3D pointAtParameter(double t){
         retorne origem + direção * t;
}
```

#### 6.1.2 Câmera

A classe Camera define uma câmera (observador) para a cena a ser sintetizada. Possui como atributos básicos:

- Vector3D posição;
- Vector3D direção de visualização.

No traçado de raios, além da câmera definir um observador com posição e direção de visualização, é interessante definirmos nessa classe o plano onde a imagem será projetada. Isso é feito através de dois atributos: altura e largura do plano, dados em coordenadas do mundo. Conhecendo-se a resolução da imagem a ser gerada, o que deve ser feito é um mapeamento das coordenadas de tela para coordenadas do mundo. Desse modo, definimos os respectivos *pixels* da imagem no plano de projeção.

A classe Camera deve possuir um método, que dado dois números, x e y, índices do pixel na imagem, realize essa transformação e retorne um raio que passa por esse pixel.

```
public Ray getRay(double x, double y){
    encontre o ponto no plano correspondente a x e y;
    calcule o raio com origem no observador e que passa por esse pixel;
    retorne raio;
}
```

#### 6.1.3 Primitivas e Materiais

Cada primitiva será representada por uma classe, que deve possuir um método para o cálculo da interseção com um raio e outro para o cálculo da normal. Isso nos permite

adicionar qualquer primitiva futuramente, bastando apenas a criação de uma nova classe, que contenha os dois métodos acima citados, além é claro, de seus atributos específicos.

Um importante atribuído de uma primitiva é o **material** que a constitui. Sendo assim, devemos criar uma classe Material, onde os atributos descrevem suas propriedades de reflexão. Alguns atributos devem ser criados em função dos modelos de BRDF que serão utilizados. Por exemplo, o modelo de Phong Modificado utiliza três parâmetros, enquanto o modelo de Schilick possui seis no total.

Para uma primitiva, além do material, devemos associar uma BRDF.

#### 6.1.4 BRDF

Para a implementação dos modelos de BRDF, criamos uma interface BRDF, que define o método eval(), utilizado para avaliar a BRDF em um ponto, dada uma direção de incidência, uma direção de reflexão e o material da superfície.

```
public double eval(Material m, Vector3D in, Vector3D out, Vector3D normal);
```

A implementação do método varia para cada modelo de BRDF, representado pela sua respectiva classe. Inserir um novo modelo de BRDF acarreta na criação de uma nova classe que implementa a interface BRDF e a inclusão de novos atributos à classe Material, se necessário.

#### 6.1.5 Fontes de Luz

Fontes de luz podem ser simplificadamente classificadas como: pontuais ou com áreas finitas. Vimos que o traçado de raios clássico considera as fontes de luz como pontuais e o distribuído permite modelar as fontes de luz com área finita. Devemos criar duas classes para representar as diferentes fontes de luz (classe PointLight e classe AreaLight). Os atributos comuns são:

- RGBColor cor da luz emitida:
- double potência em watts.

Na classe PointLight deve existir, além desses, o atributo que indica sua posição no espaço 3D. Na classe AreaLight, devemos associar uma geometria, por exemplo um

retângulo, que define a área da fonte de luz. Lembramos que os raios de sombra, tanto no traçado de raios dsitribuído quanto no traçado de caminhos, são traçados na direção de pontos aleatórios sobre a área da fonte de luz. Devemos ter nessa classe um método que nos retorne este ponto aleatório.

Geralmente, isso é feito utilizando-se dois números aleatórios,  $x \in y$ , no intervalo [0, 1].

```
public Vector3D randomPointLight(double randomX, double randomY) {
         retorne um ponto aleatório sobre a área da fonte de luz;
}
```

Logicamente, a implementação do método depende da geometria associada. Um método para gerar números aleatórios com distribuição uniforme em um triângulo pode ser encontrado em (GLASSNER, 1990).

#### 6.1.6 Cena

Apresentado as estruturas básicas, bem como as primitivas, materiais, BRDF e câmera, podemos agora definir um cena (classe Cene). Uma cena é que um conjunto de primitivas, com seus respectivos materiais e BRDF, um conjunto de fontes luminosas e uma câmera que define o observador. Com isso, estamos prontos para utilizar os métodos de traçado de raios para sintetizar a cena em uma imagem.

### 6.2 Resultados

Esta seção apresenta uma série de imagens obtidas com o sintetizador implementado, que ilustram a aplicação das várias técnicas discutidas.

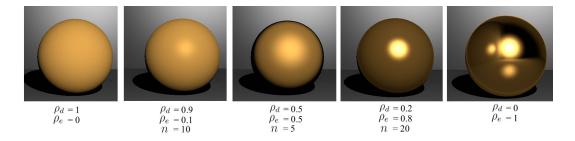


Figura 23: Esfera gerada com diferentes valores para os parâmetros do modelo de Phong Modificado. A última é perfeitamente especular e tem a iluminação indireta calculada.

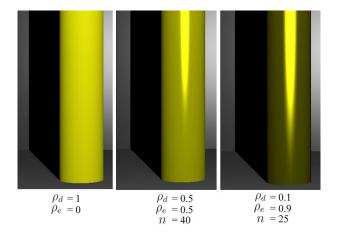


Figura 24: Superfície quádrica: Cilindro gerado com diferentes valores para os parâmetros do modelo de Phong Modificado.

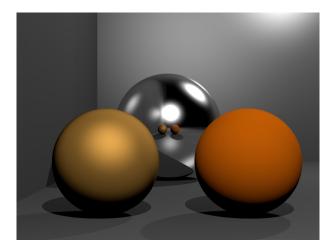


Figura 25: Esferas e plano: iluminação indireta calculada para uma esfera perfeitamente especular. Utilização de 50 raios/pixel.

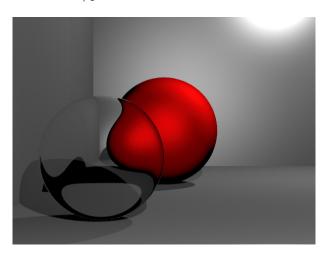


Figura 26: Refração: o índice de refração da cena é igual a 1,003 (ar), enquanto o da esfera é 1,59 (vidro).

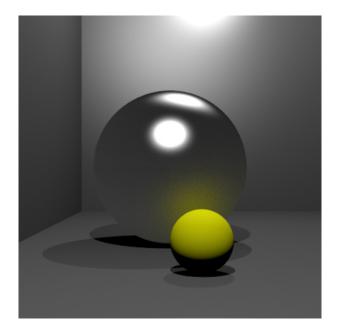


Figura 27: Reflexão borrada (glossy): utilizando 20 raios em torno da direção de reflexão ideal.

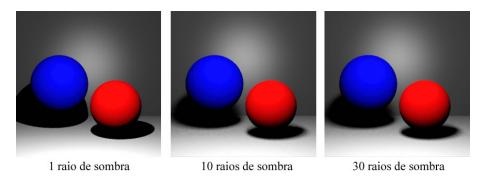


Figura 28: Sombras suaves (soft shadows) para diferentes números de raios de sombra.

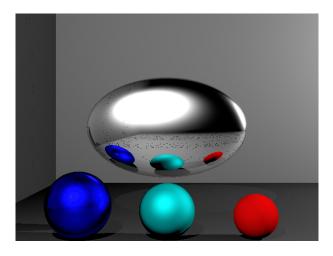


Figura 29: Superfície quádrica especular (elipsóide). Devido a utilização de apenas 1 raio por *pixel*, temos o problema de *jittering*.

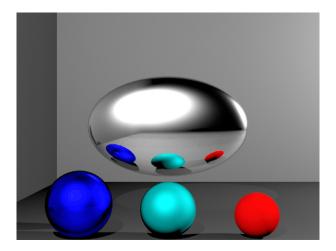


Figura 30: Utilização de 10 raios por pixel com distribuição uniforme.

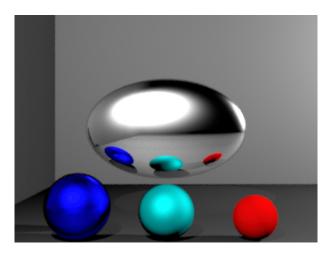


Figura 31: Utilização de 10 raios por pixel com distribuição normal.

# 6.3 Dificuldades e Limitações

Sem dúvida, a maior limitação encontrada foi o elevado tempo computacional para a geração das imagens. O processo de síntese de imagens envolve, além de aspectos técnicos, fases em que é necessário a utilização de um senso estético. A escolha dos atributos dos materiais, posição das primitivas, as fontes de luz, sua disposição no ambiente, entre outros detalhes são realizadas nestas fases. Isto deve ocorrer de forma iterativa e interativa, mas devido a lentidão para a geração das imagens, esse processo torna-se difícil e um tanto quanto tedioso.

Mais importante, isto dificulta a realização de testes e limita o número de iterações necessárias para a obtenção de resultados satisfatórios, já que após a modificação de alguns parâmetros ou na implementação, é necessário uma grande parcela de tempo para a verificação dos resultados.

# 7 Conclusão e Trabalhos Futuros

Neste trabalho, foram apresentados os conceitos e métodos utilizados na síntese de imagens realistas. Vimos a necessidade do estudo da física natural da luz, sua interação com superfícies e o modelo matemático utilizado para descrevê-la. Discutimos as simplificações adotadas, tanto no modelo utilizado por nós para representar a luz, o da ótica geométrica, quanto na função que descreve as propriedades de reflexão de uma superfície, a BRDF. A perda de informações devido a essas simplificações é considerável, porém, os resultados obtidos se aproximam da realidade. Modelos mais complexos de BDRF têm sido desenvolvidos, de forma a representar melhor a aparência dos materiais.

Apresentamos o método de traçado de raios e suas extensões. Esses métodos reduzem o problema relacionado com o transporte da luz, matematicamente expresso pela equação de rendering. O traçado de raios clássico foi o primeiro grande avanço para a síntese de imagens realistas, sendo um algoritmo simples capaz de avaliar parcialmente a iluminação global. A distribuição estocástica dos raios permitiu iniciamente a adição de vários efeitos, como sombras suaves e reflexão borrada, para depois ser utilizada no traçado de caminhos, tornando possível o cálculo da iluminação global completa de uma cena. Constatamos também que os métodos de traçado de raios, em geral, são lentos. Em 2001, um algoritmo foi proposto por Henrik Wann Jensen para acelerar os algoritmos de traçado de raios baseado em métodos de Monte Carlo, o mapeamento de fótons (photon mapping) (JENSEN, 2001).

Como trabalhos futuros, podem ser realizados estudos sobre outros modelos de BRDF, mais complexos que o de Phong e de Schlick. Uma outra questão importante envolvendo traçado de raios são as técnicas de otimização, que se concentram principalmente na aceleração dos cálculos das interseções. Outra abordagem interessante é quanto a utilização da Computação Paralela para aceleração do método. Apesar da grande redução no tempo de processamento, proporcionada pelas técnicas de aceleração, o processamento paralelo ainda é necessário para atingir níveis interativos. (MCNEILL et al., 1999). Por último, sugerimos um estudo sobre o mapeamento de fótons. O algoritmo permite que a ilu-

minação global seja calculada de maneira eficiente e robusta, além de capturar efeitos como cáusticas, que dificilmente são exibidas corretamente pela maior parte dos outros algoritmos.

# Referências

JENSEN, H. W. Realistic Image Synthesis Using Photon Mapping. Natick, MA: AK Peters, 2001.

MARTIN, A. Distributed Ray Tracing. 2007. Disponível em: <a href="http://web.cs.wpi.edu/~matt/courses/cs563/talks/dist\_ray/dist.html">http://web.cs.wpi.edu/~matt/courses/cs563/talks/dist\_ray/dist.html</a>.

BUSS, S. R. 3-D Computer Graphics - A Mathematical Introduction with OpenGL. New York: Cambridge University Press, 2003. 371 p.

NICODEMUS, F. E. et al. Geometric Considerations and Nomenclature for Reflectance. [S.l.], out. 1977.

PHONG, B.-T. Illumination for computer generated pictures. *Communications of the ACM*, v. 18, n. 6, p. 311–317, jun. 1975.

BLINN, J. F. Models of light reflection for computer synthesized pictures. In: . [S.l.: s.n.], 1977. v. 11, n. 2, p. 192–198.

TORRANCE, K. E.; SPARROW, E. M. Theory for off-specular reflection from roughened surfaces. *Journal of Optical Society of America*, v. 57, n. 9, 1967.

COOK, R. L.; TORRANCE, K. E. A reflectance model for computer graphics. In: . [S.l.: s.n.], 1981. v. 15, n. 3, p. 307–316.

OREN, M.; NAYAR, S. K. Generalization of lambert's reflectance model. In: GLASSNER, A. (Ed.). *Proceedings of SIGGRAPH '94*. [S.l.]: ACM Press, 1994. p. 239–246.

WARD, G. J. Measuring and modeling anisotropic reflection. In: ACM SIGGRAPH. *Proceedings SIGGRAPH '92.* [S.l.]: ACM Press, 1992. v. 26, p. 265–272.

POULIN, P.; FOURNIER, A. A model for anisotropic reflection. In: ACM SIGGRAPH. *Proceedings SIGGRAPH '90.* [S.l.]: ACM Press, 1990. v. 24, p. 273–282.

LAFORTUNE, E. P.; WILLEMS, Y. D. Using the Modified Phong BRDF for Physically Based Rendering. Leuven, Belgium, 1994.

SCHLICK, C. An inexpensive BRDF model for physically-based rendering. In: EUROGRAPHICS. *Computer Graphics Forum.* [S.l.], 1994. v. 13, n. 3, p. 233–246. Eurographics '94 Conference issue.

PALIK, E. D. Handbook of Optical Constants of Solids. [S.l.]: Academic Press, 1985.

APPEL, A. Some techniques for shading machine renderings of solids. In: . [S.l.: s.n.], 1968. v. 32, p. 37–45.

WHITTED, T. An improved illumination model for shaded display. *Communications of the ACM*, v. 23, n. 6, p. 343–349, jun. 1980.

SHIRLEY, P. Realistic Ray Tracing. Natick, MA: AK Peters, 2000.

KIRK, D. (Ed.). Graphics Gems III. [S.l.]: Academic Press, 1992. 275–283 p.

KAJIYA, J. T. The rendering equation. In: ACM SIGGRAPH. *Proceedings SIGGRAPH* '86. [S.l.]: ACM Press, 1986. v. 20, p. 143–150.

COOK, R. L.; PORTER, T.; CARPENTER, L. Distributed ray tracing. In: ACM SIGGRAPH. *Proceedings SIGGRAPH '84*. [S.l.]: ACM Press, 1984. v. 18, n. 3, p. 137–45.

SPANIER, J.; GELBARD, E. M. Monte Carlo Principles and Neutron Transport Problems. [S.l.]: Addison-Wesley Publishing Co., 1969.

ARVO, J.; KIRK, D. B. Particle transport and image synthesis. In: ACM SIGGRAPH. *Proceedings SIGGRAPH '90.* [S.l.]: ACM Press, 1990. v. 24, p. 63–66.

GLASSNER, A. S. (Ed.). Graphics Gems I. [S.l.]: Academic Press, 1990. 24-28 p.

MCNEILL, M. et al. Performance of space subdivison techniques in ray tracing. Computer Graphics Forum, v. 11, n. 4, p. 213–222, 1999.

HAINES, E. Ray Tracing News Guide. ACM Press, 2007. Disponível em: <a href="http://www.acm.org/tog/resources/RTNews/html/">http://www.acm.org/tog/resources/RTNews/html/</a>>.

JOHN, M. Focus On Photon Mapping. Cincinnati, Ohio: Premier Press, 2003. 330 p.