

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL  
INSTITUTO DE INFORMÁTICA  
CURSO DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

269447

**Aplicação da Técnica de Rastreamento  
Bidirecional à Síntese de Objetos Transparentes**

por

**GILDA APARECIDA DE ASSIS**

Dissertação submetida à avaliação, como requisito parcial  
para a obtenção do grau de  
Mestre em Ciência da Computação

**Prof. Anatólio Laschuk**  
Orientador



UFRGS

SABi



85230119

Porto Alegre, outubro de 1998

**UFRGS**  
INSTITUTO DE INFORMÁTICA  
RIBEIRÃO

## CIP – CATALOGAÇÃO NA PUBLICAÇÃO

Assis, Gilda Aparecida de

Aplicação da Técnica de Rastreamento Bidirecional à Síntese de Objetos Transparentes / por Gilda Aparecida de Assis – Porto Alegre: CPGCC da UFRGS, 1998:

101f. : il.

Dissertação (mestrado)- Universidade Federal do Rio Grande do Sul. Curso de Pós-Graduação em Ciência da Computação, Porto Alegre, BR-RS, 1998. Orientador: Laschuk, Anatólio.

1. Realismo. 2. Refração. 3. Transparência. 4. Ray-Tracing Bidirecional. I. Laschuk, Anatólio. II. Título.

*Aplicação dos computadores em*  
*Computação*  
*gráfica*  
*Ray tracing*  
*Refração luminosa*

INSTITUTO DE INFORMÁTICA BIBLIOTECA		
N.º CHAMADA 681.327.16(043) A848a		N.º REG.: 38812
ORIGEM: D		DATA: 10/05/00
PUNDO: II		CHECO: R\$ 30,00
FORN.: II		
27,06,00		

CNPq 1.03.04.00-2

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitora: Profa. Wrana Panizzi

Pró-Reitor de Pós-Graduação: Prof. José Carlos Ferraz Hennemann

Diretor do Instituto de Informática: Prof. Philippe Oliver Alexandre Navaux

Coordenadora do CPGCC: Profa. Carla Maria Dal Sasso Freitas

Bibliotecária – Chefe do Instituto de Informática: Beatriz Regina Bastos Haro

*“O mistério não é um muro onde a inteligência esbarra, mas um oceano onde o homem  
mergulha.”*  
*(G. Thibon)*

*“Que Deus me dê serenidade para aceitar as coisas que não posso mudar, coragem  
para mudar as que posso e sabedoria para distinguir entre elas.”*  
*(R. Niebuhr)*

## Agradecimentos

Ao meu orientador, prof. Anatólio Laschuk, cuja confiança, sugestões e sábios conselhos foram imprescindíveis para a realização desta dissertação.

Aos colegas e amigos do Grupo de Computação Gráfica, especialmente Christian e Mauro, cujas idéias, bibliografia e sugestões muito contribuíram para a concretização deste trabalho.

Aos amigos do CPGCC, Duda, Eunice, Bello, Cadinho e Marco, que tanto me incentivaram nos momentos finais do trabalho.

Aos demais colegas do CPGCC, em especial aos amigos Rita, Coelho e Érika, sempre dispostos a auxiliar nas dúvidas sobre a linguagem C.

À prof. Carla, pelo apoio e interesse no início do curso.

Aos demais colegas, professores e funcionários do Instituto de Informática da UFRGS, que direta ou indiretamente ajudaram na realização deste trabalho.

Ao Clésio, meu namorado, pela compreensão, carinho e auxílio nos momentos críticos.

Ao meu pai José Fagundes (in memorian), pelo amor, carinho, compreensão e apoio à minha decisão de cursar o mestrado longe da família.

À Biblioteca do CPGCC e da Física.

À CAPES pelo importante apoio financeiro.

A todos que, de alguma forma, contribuíram para a realização desta dissertação.

*Aos meus pais José (in memoriam) e Maria José.  
Aos meus irmãos Gilson, Márcia e Márcio.  
À minha sobrinha Leticia.*

## Sumário

<b>Lista de Abreviaturas</b> .....	9
<b>Lista de Símbolos</b> .....	10
<b>Lista de Figuras</b> .....	11
<b>Lista de Tabelas</b> .....	13
<b>Resumo</b> .....	14
<b>Abstract</b> .....	15
<b>1 Introdução</b> .....	16
<b>1.1 Motivação</b> .....	16
<b>1.2 Objetivos</b> .....	17
<b>1.3 Organização desta Dissertação</b> .....	17
<b>2 Fundamentos da Óptica</b> .....	19
<b>2.1 Introdução</b> .....	19
<b>2.2 A Luz</b> .....	19
2.2.1 A Natureza da Luz.....	19
2.2.2 Geometria das Fontes de Luz.....	20
2.2.3 A Propagação da Luz.....	20
<b>2.3 Fenômenos Ópticos Resultantes da Interação Luz-Objeto</b> .....	20
<b>2.3.1 Reflexão</b> .....	21
2.3.1.1 Reflexão Especular .....	21
2.3.1.2 Reflexão Difusa.....	22
<b>2.3.2 Absorção</b> .....	23
<b>2.3.3 Refração</b> .....	23
2.3.3.1 Meios de Propagação da Luz .....	23
2.3.3.2 Leis da Refração.....	23
2.3.3.3 Índices de Refração .....	24
2.3.3.4 Reflexão Total.....	25
<b>2.3.4 Cálculo dos Vetores Reflexão e Refração</b> .....	25
<b>2.4 Lentes</b> .....	27

<b>2.5 Conclusão</b> .....	30
<b>3 Modelos de Iluminação Global</b> .....	31
<b>3.1 Introdução</b> .....	31
<b>3.2 Rastreamento de Raios</b> .....	31
3.2.1 Traçado de Raios A Partir do Observador.....	32
3.2.2 Traçado de Raios A Partir das Fontes de Luz.....	34
<b>3.3 Conclusão</b> .....	34
<b>4 A Síntese de Objetos Transparentes</b> .....	35
<b>4.1 Introdução</b> .....	35
<b>4.2 A Síntese de Objetos Transparentes Sem Efeitos de Refração</b> .....	35
<b>4.3 A Síntese de Objetos Transparentes Com Refração</b> .....	36
4.3.1 Método de Kay-Greenberg .....	37
4.3.2 Rastreamento de Raios .....	37
<b>5 Bitracer - Um Modelo para a Síntese de Lentes Esféricas</b> .....	38
<b>5.1 Introdução</b> .....	38
<b>5.2 Esferas</b> .....	38
<b>5.3 Espalhamento Luminoso</b> .....	39
<b>5.4 Calota Visível</b> .....	40
<b>5.5 Identificação de Obstáculos</b> .....	41
<b>5.6 Fonte Secundária de Luz</b> .....	44
<b>5.7 Estrutura do Algoritmo de Traçado Bidirecional de Raios Proposto</b> .....	53
5.7.1 Fase 1 do Traçado Bidirecional.....	53
5.7.2 Fase 2 do Traçado Bidirecional.....	55
5.7.3 Modelo de Iluminação .....	59
<b>5.8 Conclusão</b> .....	61
<b>6 Implementação do Algoritmo Proposto</b> .....	62
<b>6.1 Introdução</b> .....	62
<b>6.2 Formatos de Arquivos Utilizados</b> .....	62
6.2.1 Arquivos de Entrada.....	62
6.2.2 Arquivos de Saida .....	63
<b>6.3 Funcionamento do Protótipo</b> .....	63
<b>6.4 Estruturas de Dados</b> .....	65
<b>6.5 Sumário</b> .....	68
<b>7 Geração de Imagens e Análise dos Resultados</b> .....	69
<b>7.1 Introdução</b> .....	69

<b>7.2 Ambiente Onde as Imagens Foram Geradas e Exibidas</b> .....	69
<b>7.3 Avaliação quanto ao Realismo</b> .....	69
<b>7.4 Avaliação quanto à Complexidade e Tempo de Processamento</b> .....	76
<b>7.5 Comparação entre o Método Bitracer e o Método de Rastreamento de Raios Recursivo</b> .....	82
<b>7.6 Conclusão</b> .....	83
<b>8 Conclusões</b> .....	84
<b>8.1 Conclusões Gerais</b> .....	84
<b>8.2 Desdobramentos deste Trabalho</b> .....	85
<b>Anexo A</b> .....	86
<b>Anexo B</b> .....	91
<b>Bibliografia</b> .....	100

## **Lista de Abreviaturas**

RGB	Red, Green, Blue
SRU	Sistema de Referência do Universo
ART	Árvore de Raios Refletidos e Transmitidos
TGA	Truevision Targa

## Lista de Símbolos

$\leq$	Menor ou igual a
$\geq$	Maior ou igual a
$=$	Igual a
$\neq$	Diferente de
$\alpha$	Letra grega alfa
$\beta$	Letra grega beta
$\gamma$	Letra grega gama
$\theta$	Letra grega teta
$  $	Módulo ou valor absoluto
$   $	Norma ou tamanho de um vetor
$\cup$	União
$\emptyset$	Conjunto vazio
$\bullet$	Produto interno de dois vetores
$\Leftrightarrow$	Se e somente se
$:=$	Atribuição
$\infty$	Infinito
$\Delta$	Triângulo

## Lista de Figuras

FIGURA 2.1- Seção de corte de uma onda de luz.....	19
FIGURA 2.2- Fonte de luz puntiforme.....	20
FIGURA 2.3- Reflexão especular .....	21
FIGURA 2.4- Elementos básicos para o cálculo da componente de reflexão especular.....	21
FIGURA 2.5- Reflexão difusa [KRU 92] .....	22
FIGURA 2.6- Elementos básicos para o cálculo da componente de reflexão difusa ..	22
FIGURA 2.7- Refração da luz entre meios transparentes.....	23
FIGURA 2.8- Raio de luz que passa do meio A para o meio B ( $\theta_i > \theta_t$ ) .....	24
FIGURA 2.9- Reflexão e transmissão para o modelo de Whitted .....	26
FIGURA 2.10- Elementos de uma lente esférica convergente.....	27
FIGURA 2.11- Elementos utilizados para determinar o ponto imagem ( $I$ ) numa lente espessa .....	29
FIGURA 3.1- Modelo básico do algoritmo de traçado de raios tradicional .....	32
FIGURA 3.2- Sombra em rastreamento de raios convencional .....	33
FIGURA 4.1- Linhas de visada óptica e geométrica .....	35
FIGURA 5.1- Fontes luminosas secundárias .....	38
FIGURA 5.2- Tipos de fontes secundárias quanto à posição da fonte primária $L$ em relação a esfera.....	40
FIGURA 5.3- Determinação da calota da esfera $i$ visível a partir da fonte luminosa $L_j$ .....	40
FIGURA 5.4- Relações trigonométricas para determinar $R_T$ e $\alpha$ .....	41
FIGURA 5.5- Tipos de fontes secundárias: cheia e não-cheia.....	42
FIGURA 5.6- Testes de visibilidade.....	43
FIGURA 5.7- Exemplo de obstrução parcial .....	44
FIGURA 5.8- Exemplo de obstrução total.....	44
FIGURA 5.9- Exemplo da influência de uma fonte secundária de luz na iluminação da cena.....	46
FIGURA 5.10- Fonte secundária real e cheia, localizada no ponto $I$ .....	47
FIGURA 5.11- Cálculo do ângulo de espalhamento para uma fonte secundária real localizada em $I$ .....	49
FIGURA 5.12- Cálculo do ângulo de espalhamento para uma fonte secundária virtual localizada em $I$ .....	50

FIGURA 5.13- Elementos para o cálculo do raio $R_C$ do cilindro transmissor de luz..	52
FIGURA 5.14- Ponto $P$ está na sombra da fonte secundária não-cheia $L_{ji}$ .....	59
FIGURA 6.1- Arquivo de entrada <i>esfera.bi</i> .....	63
FIGURA 6.2- Fluxo de funcionamento do protótipo “BITRACER” .....	64
FIGURA 6.3- Cálculo do raio de sombra para uma fonte secundária cilíndrica.....	65
FIGURA 7.1- Imagem gerada por um algoritmo de <i>ray-tracing</i> convencional para os parâmetros da TABELA 7.1 .....	71
FIGURA 7.2- Imagem gerada pelo algoritmo <i>bitracer</i> para os parâmetros da TABELA 7.1.....	71
FIGURA 7.3- Imagem gerada por um algoritmo de <i>ray-tracing</i> convencional para os parâmetros da TABELA 7.2 .....	73
FIGURA 7.4- Imagem gerada pelo algoritmo <i>bitracer</i> para os parâmetros da TABELA 7.2.....	73
FIGURA 7.5- Imagem gerada por um algoritmo de <i>ray-tracing</i> convencional para os parâmetros da TABELA 7.3 .....	76
FIGURA 7.6- Imagem gerada pelo algoritmo <i>bitracer</i> para a TABELA 7.3 .....	76
FIGURA 7.7- Gráfico: Tempo de processamento x cena gerada.....	81
FIGURA 7.8- Gráfico: Porcentagem de $\frac{(\text{Tempo}_{\text{Bitracer}} - \text{Tempo}_{\text{Tracer}})}{\text{Tempo}_{\text{Tracer}}}$ x número de esferas transparentes.....	82
FIGURA A.1 .....	87
FIGURA A.2 .....	87
FIGURA A.3 .....	89

## Lista de Tabelas

TABELA 7.1- Parâmetros do arquivo de entrada " <i>fig1.bi</i> " .....	70
TABELA 7.2- Parâmetros do arquivo de entrada " <i>fig2.bi</i> " .....	72
TABELA 7.3- Parâmetros do arquivo de entrada " <i>fig3.bi</i> " .....	74
TABELA 7.4- Tempos de geração de imagens no protótipo <i>bitracer</i> .....	79
TABELA 7.5- Tempos de geração de imagens para as amostras da TABELA 7.4 ....	80

## Resumo

Este trabalho apresenta uma proposta de aplicação da técnica de *ray-tracing* bidirecional em ambientes esféricos, contendo fontes luminosas puntiformes. Trata-se de um trabalho que discorre no contexto da área de Síntese de Imagens Realísticas dentro da Computação Gráfica. O trabalho tem como principal contribuição a definição e o desenvolvimento de uma técnica para simular o fenômeno físico de refração da luz proveniente das fontes luminosas puntiformes da cena.

A Síntese de Imagens Realísticas é uma das principais áreas de aplicação e pesquisa da Computação Gráfica. Uma imagem realística é uma imagem que incorpora os efeitos da luz que interage com objetos fisicamente reais.

A dificuldade fundamental para a síntese de imagens realísticas se encontra na complexidade do mundo real, que apresenta uma infinidade de gradações de cores, texturas, reflexões, sombras, etc. Para a criação destas imagens realísticas, percorre-se um grande número de estágios, englobando métodos de modelagem, definição da posição de visualização, remoção de elementos ocultos, efeitos de reflexão e refração, e assim por diante.

O trabalho está organizado da seguinte forma. Inicialmente, faz-se um estudo aprofundado da Óptica, área da Física que estuda o comportamento da luz no mundo real. A seguir, são apresentados algoritmos que simulam este comportamento da luz, enfatizando-se o algoritmo de rastreamento de raios (*ray-tracing*).

Discute-se a seguir os principais problemas relacionados à simulação de objetos transparentes na Computação Gráfica. Então, uma proposta para a simulação em ambientes esféricos da refração da luz que provém diretamente das fontes luminosas puntiformes da cena é apresentada. Esta proposta baseia-se na utilização do algoritmo de rastreamento bidirecional de raios.

O algoritmo apresentado é composto de duas fases. Na primeira etapa, são geradas as fontes secundárias de luz. Na segunda etapa, utiliza-se a informação obtida na primeira etapa para simular a refração da luz que provém diretamente das fontes luminosas da cena.

As fontes secundárias de luz têm sua origem na utilização dos objetos esféricos transparentes como lentes esféricas convergentes. A fonte luminosa secundária localiza-se no ponto imagem da lente, considerando-se como ponto objeto a fonte luminosa puntiforme original. A localização da fonte luminosa secundária é obtida através da equação dos pontos conjugados. Também armazena-se uma informação relacionada com a área de atuação da fonte luminosa secundária (ângulo de espalhamento). O ângulo de espalhamento é essencial para que, na segunda fase do algoritmo, seja possível identificar se o ponto atual é iluminado ou não pela fonte secundária em questão.

Finalizando, são geradas imagens tanto no protótipo implementado quanto em um algoritmo de *ray-tracing* convencional. Os resultados obtidos são comparados em nível de realismo e tempo de execução.

**Palavras-chave:** Realismo, Refração, Transparência, *Ray-Tracing* bidirecional.

**Title:** "APPLICATION OF THE BIDIRECTIONAL RAY TRACING METHOD IN RENDERING OF THE TRANSPARENT OBJECTS"

## **Abstract**

This work presents a proposal of using the bidirectional ray tracing method in spherical modeling environments containing punctual light sources. This project was developed within the field of Computer Graphics, more precisely in the area of synthesis of realistic images. The main contribution of this work is the definition and the development of a method that simulates the light refraction proceeding from localized light sources in the scene.

The synthesis of realistic images is one of the main areas of application and research in Computer Graphics. A realistic image is an image that contains light effects interacting with physically real objects.

The major difficulty for rendering realistic images is the complexity of the real world, with several color graduations, textures, reflections, shadows, *etc.* For this rendering, many steps like modeling methods, definition of visualization position, hidden-surface algorithms, reflection and refraction effects, and so forth, are developed.

At first, this work presents a study about Optics, the area of Physics that studies the behaviour of light in the real world. In sequence, algorithms that simulate that behaviour are presented, with special attention to ray tracing method. After that, the principal problems of the simulation of transparency in Computer Graphics are discussed. So, a proposal for simulation of the light refraction proceeding from light source in spherical modeling environment, is presented. This proposal is based on use of the bidirectional ray tracing algorithm.

This algorithm is divided in two main stages. In the first stage, the secondary light sources are generated. In the second stage, the information about the secondary light sources is utilized to simulate the light refraction directly proceeding from light sources of the scene.

The secondary light sources are originated from transparent spherical objects like convergent spherical lenses. The position of the secondary light source is the image point of the lens, corresponding to punctual light source like object point. The position of the secondary light source is calculated by the equation of the conjugated points. Also the information about the scattering angle of the secondary light source is stored. The scattering angle is essential, in first stage of algorithm, to establish if the current point is illuminated by any secondary light source.

Finally, images are generated both in the implemented prototype as in conventional ray tracing. The final results of this work are evaluated based on realism and runtime.

**Keywords:** Realism, Refraction, Transparency, Bidirectional Ray Tracing.

# 1 Introdução

A criação de imagens realísticas é um importante objetivo em áreas como simulação, projeto, entretenimento e publicidade, pesquisa e educação, e comando e controle. Por imagem realística entende-se uma imagem sintetizada em computador e que incorpora boa parte dos efeitos ópticos de interação da luz com objetos fisicamente reais [FOL 90].

Segundo Newman [NEW 79], uma grande dificuldade para a síntese de imagens realísticas se encontra na complexidade do mundo real, que possui uma infinidade de graduações de cores, inúmeras texturas, sombras, reflexões e outras características próprias. Assim, a criação de imagens que simulem todas estas características torna-se um desafio, onde é percorrido um grande número de estágios, desde a utilização dos métodos de modelagem e definição de parâmetros de visualização até a remoção de elementos ocultos e efeitos de sombras, reflexão, refração, etc [FOL 90].

O desenvolvimento da área de Síntese de Imagens Realísticas na Computação Gráfica está ligado aos surgimento e progresso dos modelos de iluminação global, rastreamento de raios (1968) e radiosidade (1984). Entretanto, ainda hoje estudos e projetos de pesquisa são desenvolvidos nesta área por profissionais e pesquisadores de Computação Gráfica, Física, Fisiologia e Psicologia, entre outros.

## 1.1 Motivação

A motivação para este trabalho fundamenta-se na geração de sombras em imagens foto-realísticas. Segundo Gomes [GOM 90], as sombras desempenham um importante papel na obtenção de realismo em imagens sintetizadas por computador. Elas auxiliam na compreensão do relacionamento espacial entre os objetos da cena, estabelecem diversos níveis de profundidade e fazem uma pontuação geométrica de modo a evitar que os objetos pareçam estar flutuando no ar. Além disso, causam um dos maiores contrastes de intensidade nas imagens sintetizadas por computador. E, fornecem fortes indicações sobre as formas, posições relativas e características das superfícies dos objetos, além de indicar a localização aproximada, forma e tamanho da fonte de luz.

Especificamente, pretende-se aplicar o método de rastreamento bidirecional à síntese de cenas contendo tanto superfícies difusas quanto especulares, transparentes ou opacas. A utilização do método de rastreamento bidirecional determina um mecanismo para aproximar os efeitos de refração resultantes da iluminação direta em objetos esféricos transparentes (lentes esféricas convergentes).

Por outro lado, a motivação também fundamenta-se na utilização e aprimoramento da técnica de rastreamento inverso dos raios, método adequado para a simulação dos efeitos de refração da luz. Este método representa uma idéia poderosa para os sistemas de visualização e pode produzir resultados compensadores quanto ao nível de realismo das imagens geradas [WAT 90]. Desta forma, direcionam-se esforços para identificar possíveis técnicas de otimização e/ou aproximação do método de rastreamento inverso em ambientes contendo objetos esféricos. O método de rastreamento inverso dos raios adaptado é utilizado na primeira fase do algoritmo de rastreamento bidirecional, a etapa independente da visão. A informação que resulta do rastreamento inverso é combinada na segunda etapa, rastreamento de raios a partir do observador, de forma a incorporar parte dos efeitos de refração da luz que provém diretamente das fontes luminosas puntiformes da cena na imagem sintetizada.

## 1.2 Objetivos

O objetivo global deste trabalho de Mestrado é um estudo sobre a aplicabilidade da técnica de rastreamento bidirecional de raios à síntese de objetos transparentes e uma análise comparativa deste método em relação a uma técnica de rastreamento recursivo de raios.

Como objetivos específicos, podem-se citar:

- Estudo aprofundado da área da Física que trata dos fenômenos relacionados ao comportamento da luz no mundo real: a Óptica. Este estudo proporciona um embasamento teórico para se entender posteriormente o modelo de iluminação e os mecanismos utilizados para a simulação dos efeitos ópticos em Computação Gráfica;
- estudo dos algoritmos de rastreamento de raios quanto aos aspectos da visualização de objetos transparentes;
- aprimoramento do método de rastreamento bidirecional, de forma a aproximar imagens com efeitos de refração provenientes da iluminação direta (a partir das fontes de luz da cena).

## 1.3 Organização desta Dissertação

Esta dissertação está dividida em 8 capítulos, incluindo esta introdução.

O capítulo 2 (Fundamentos da Óptica) retrata um estudo sobre a Óptica. Apresentam-se algumas características das fontes de luz e os efeitos produzidos por sua interação com os objetos do mundo real (reflexão, refração e absorção). Também inclui-se um estudo sobre lentes esféricas convergentes. Este capítulo produz um embasamento teórico para o entendimento futuro da aplicação destes conceitos em procedimentos da Computação Gráfica.

O capítulo 3 (Modelos de Iluminação Global) apresenta os principais modelos de iluminação global. Enfatizam-se os pontos fortes de cada um dos métodos e faz-se uma avaliação de sua aplicabilidade à síntese de transparência.

O capítulo 4 (A Síntese de Objetos Transparentes) trata das técnicas existentes para a síntese de objetos transparentes, diferenciando os métodos em duas categorias: sem efeitos de refração e com efeitos de refração. A segunda categoria é fisicamente embasada e a primeira representa uma aproximação do que ocorre realmente no mundo físico.

O capítulo 5 (Bitracer - Um Modelo para a Síntese de Objetos Transparentes) descreve o modelo proposto para a geração de imagens. O capítulo inicialmente apresenta a estrutura da primeira fase da proposta (rastreamento inverso de raios) quando são geradas as fontes secundárias de luz. Na sequência, tem-se uma descrição da segunda fase do algoritmo (raios partem do observador para a cena) com ênfase na utilização da informação das fontes secundárias de luz no cálculo dos componentes locais do modelo de iluminação.

O capítulo 6 (Implementação do Algoritmo Proposto) descreve em linhas gerais a implementação do algoritmo *bitracer*, apresentando as principais estruturas de memória, a organização modular do protótipo e os formatos dos arquivos de entrada e de saída.

O capítulo 7 (Geração de Imagens e Análise dos Resultados) apresenta os resultados desta dissertação. Faz-se uma análise comparativa entre a proposta *Bitracer* e um método de rastreamento recursivo de raios quanto ao tempo de processamento e grau de realismo das imagens geradas. Também faz-se uma análise de complexidade da proposta *Bitracer*.

As conclusões desta dissertação de mestrado, juntamente com propostas para a continuidade do trabalho são abordadas no capítulo 8 (Conclusões).

## 2 Fundamentos da Óptica

### 2.1 Introdução

Visando fundamentar as discussões posteriores, este capítulo desenvolve uma abordagem introdutória à área da Física que trata da luz e suas aplicações: a Óptica. Inicialmente apresenta-se uma seção sobre a luz, tratando de aspectos referentes à sua natureza, propagação, velocidade e geometria. A seguir, discutem-se os principais fenômenos ópticos resultantes da interação luz-objeto: reflexão, absorção e refração. Na seqüência, apresentam-se as leis que regem a refração e as bases matemáticas e geométricas para o seu cálculo. Por fim, descrevem-se as lentes esféricas convergentes como objetos refratores.

### 2.2 A Luz

O modelo da fonte de luz é um fator importante a ser considerado na síntese de imagens no computador e tem grande influência na formação das sombras. A fonte de luz é mais um objeto da cena, diferenciado dos demais objetos pelo fato de emitir luz [NAS 92].

Segundo Tipler [TIP 84], luz é a radiação eletromagnética com comprimentos de onda na faixa dos 380nm a 780nm para os quais o olho humano possui sensibilidade. Qualitativamente, a luz visível é uma forma de energia similar a outros fenômenos eletromagnéticos como Raios-X, Radar, Raios Gamma e Ondas de Rádio.

Uma vez que o olho humano apresenta três tipos de receptores cromáticos diferentes, espera-se obter três espécies de respostas, de acordo com o estímulo provocado. Assim, em Computação Gráfica, as cores são comumente representadas por um padrão RGB (Vermelho, Verde, Azul), de forma que a aproximação dos feixes de luz corresponda aos estímulos predominantes para cada um dos três tipos de receptores cromáticos existentes no olho humano.

#### 2.2.1 A Natureza da Luz

Há duas teorias aparentemente diferentes sobre a natureza da luz [LEV 81]. Na primeira, considera-se a luz como um conjunto de partículas individuais, indivisíveis, chamadas *fótons*. Os *fótons* se propagam em uma linha reta a uma velocidade que depende do meio onde eles estão.

Na segunda teoria, considera-se a luz como um fenômeno ondulatório. A FIGURA 2.1 apresenta o esquema da seção de corte de uma onda de luz.

Um feixe de luz é composto por diferentes comprimentos de onda, cada qual com características próprias de velocidade e intensidade.

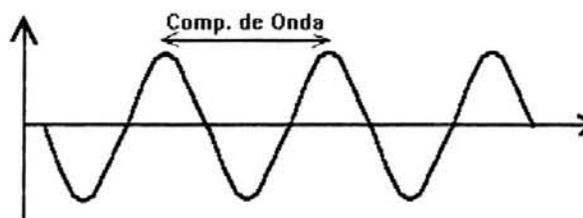


FIGURA 2.1- Seção de corte de uma onda de luz

Ambas as teorias (ondulatória e corpuscular) são consistentes uma com a outra. A existência de *fótons* indivisíveis é universalmente aceita pelos físicos contemporâneos e a teoria ondulatória continua útil como uma forma de descrever o movimento de um grande número de *fótons*.

### 2.2.2 Geometria das Fontes de Luz

Deve-se levar em consideração a geometria de uma fonte de luz nos cálculos de sombras projetadas, uma vez que porções de luz podem estar parcialmente obstruídas, produzindo assim um efeito de penumbra na cena.

Quanto à geometria, as fontes de luz são classificadas em [LEV 81]:

- Direcionais;
- puntiformes;
- lineares e
- fontes de área.

As fontes direcionais foram as primeiras a serem utilizadas na síntese de imagens. Para definir a geometria de uma fonte direcional especifica-se um vetor unitário, que define a direção de propagação da luz. Utilizam-se fontes direcionais para simular objetos luminosos que estão no infinito (emitem raios paralelos).

As fontes puntiformes (FIGURA 2.2), por sua vez, são definidas através de um ponto no espaço. São fontes luminosas que emitem luz em todas as direções.

Já as fontes lineares são representadas por um segmento de reta e um vetor direção de iluminação.

Por fim, as fontes de área possuem uma superfície emissiva não puntiforme, semelhantemente às fontes lineares, sendo definidas por um polígono (superfície emissora) e por um vetor direção de iluminação.

### 2.2.3 A Propagação da Luz

A onda de luz emitida por um objeto luminoso e que se propaga em um meio homogêneo é representada por um raio de luz (segmento de reta orientado). A noção de raio de luz é essencial. Para exemplificar, a Óptica Geométrica estuda os fenômenos luminosos sem levar em consideração previamente a natureza da luz, usando apenas a noção de raio de luz, alguns princípios fundamentais e considerações de geometria.

A velocidade de propagação da luz no vácuo é uma constante igual a

$$c = 2,9979925 \pm 0,000003 \times 10^8 \text{ m/s.} \quad (2.1)$$

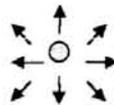


FIGURA 2.2- Fonte de luz puntiforme

## 2.3 Fenômenos Ópticos Resultantes da Interação Luz-Objeto

Segundo Padgham [PAD 55], quando um raio de luz incide sobre a superfície de um objeto, parte da energia armazenada pelo raio é refletida, parte é transmitida (refratada) e parte é absorvida. A quantidade de energia refletida, refratada ou absorvida depende das características físicas do objeto (cor, rugosidade, grau de transparência); das características da luz incidente (cor, ângulo de incidência) e das características dos meios onde a luz está se propagando (índices de refração).

## 2.3.1 Reflexão

### 2.3.1.1 Reflexão Especular

Identifica-se a reflexão especular em qualquer superfície polida. Num espelho perfeito, a reflexão especular ocorre somente numa direção, respeitando-se a condição de que o ângulo de incidência é igual ao ângulo de reflexão, ambos considerados em relação ao vetor normal à superfície no ponto. Em superfícies não perfeitamente espelhadas, a luz refletida distribui-se numa faixa, nas proximidades da direção de reflexão. Neste caso, dependendo do ângulo de observação notam-se variações no efeito especular [FOL 90].

A FIGURA 2.3 ilustra a reflexão especular. O feixe de raios paralelos que se propaga no meio (1) incide sobre a superfície S e retorna ao meio (1) mantendo o paralelismo. A superfície S é plana e polida e apresenta um coeficiente de reflexão especular  $k_s$  diferente de zero ( $k_s \neq 0$ ). Os ângulos  $\theta_i$  e  $\theta_r$  são, respectivamente, os ângulos de incidência e reflexão no ponto em questão.

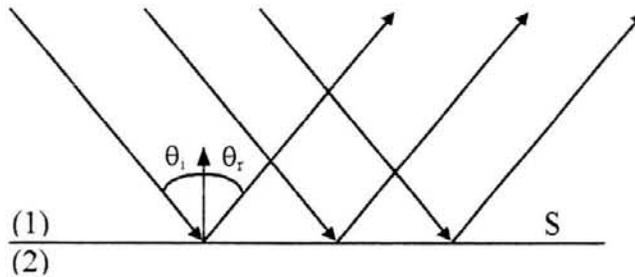


FIGURA 2.3- Reflexão especular

O modelo de iluminação de Phong para refletores não perfeitos [BUI 75], aproxima a contribuição especular para cada um dos componentes da representação de cores RGB de um ponto qualquer da superfície por

$$\sum_{j=1}^m \frac{I_{lj}}{d_j + k} (k_{sc} \cos^n \alpha_j) \quad (2.2)$$

onde,

$m$  é o número de fontes de luz presentes na cena;

$I_{lj}$  é a intensidade da fonte de luz  $j$  em termos de R, G ou B;

$k_{sc}$  é o coeficiente especular da superfície, para cada cor primitiva  $c$  (R, G ou B);

$d_j$  é a distância entre a fonte de luz e o ponto na superfície;

$k$  é uma constante arbitrária utilizada para se obter resultados mais realísticos na atenuação pela distância;

$\cos \alpha_j$  é o termo que modela a queda na contribuição especular devido ao afastamento do observador em relação à direção de reflexão especular máxima  $\vec{R}$  (FIGURA 2.4); e

$n$  é o expoente de reflexão especular. Para espelhos perfeitos  $n$  tende a infinito.

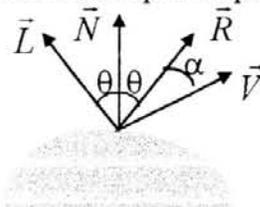


FIGURA 2.4- Elementos básicos para o cálculo da componente de reflexão especular

Se os vetores  $\vec{R}$  (reflexão) e  $\vec{V}$  (visada) estão normalizados, então

$$\cos_j \alpha = \vec{R} \cdot \vec{V}$$

E a contribuição especular pode ser aproximada, com base na equação (2.2), por

$$\sum_{j=1}^m \frac{I_{lj}}{d_j + k} (k_s (\vec{R} \cdot \vec{V})^n) \quad (2.3)$$

### 2.3.1.2 Reflexão Difusa

Um objeto é visível porque reflete difusamente a luz que recebe do ambiente. O objeto é visto porque parte desta luz refletida difusamente atinge o globo ocular do observador.

A FIGURA 2.5 esquematiza o fenômeno de reflexão difusa. Um feixe de raios paralelos proveniente do meio (1) incide sobre a superfície S e retorna ao meio (1), perdendo o paralelismo e espalhando-se em todas as direções.

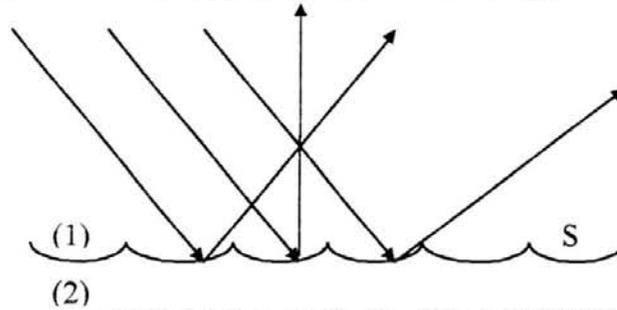


FIGURA 2.5- Reflexão difusa [KRU 92]

A quantidade de luz vista pelo observador é independente da direção de visada e é proporcional ao co-seno do ângulo que o raio de luz incidente forma com a normal (FIGURA 2.6). A equação da componente difusa, segundo o modelo de iluminação apresentado em [ROG 85], é

$$\sum_{j=1}^m \frac{I_{lj}}{d_j + k} (k_{dc} O_{d\lambda} \cos \theta_j) \quad (2.4)$$

onde,

$m$ ,  $I_{lj}$ ,  $d_j$  e  $k$  são os mesmos da equação (2.3);

$k_{dc}$  é o coeficiente de reflexão difusa, em termos da cor primitiva  $c$  R, G e B (constante para toda a superfície);

$O_{d\lambda}$  é a intensidade da luz difusa da superfície, em termos de R, G e B; e

$\cos \theta_j$  é o co-seno do ângulo  $\theta$  formado pelo raio de luz incidente e o vetor normal no ponto.

Como constata-se na equação (2.4), a iluminação difusa independe da direção de visada (vetor  $\vec{V}$  na FIGURA 2.4).

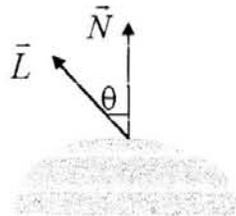


FIGURA 2.6- Elementos básicos para o cálculo da componente de reflexão difusa [FOL 90]

### 2.3.2 Absorção

Como foi dito anteriormente, parte da energia armazenada por um raio de luz pode ser absorvida pelo objeto sobre o qual a luz incide.

A luz branca é constituída de uma infinidade de luzes monocromáticas com intensidades relativas específicas, divididas em sete faixas cromáticas principais, a saber, vermelho, alaranjado, amarelo, verde, azul, anil e violeta.

Determina-se a cor de um corpo pelo tipo de luz que ele reflete difusamente. Assim, um corpo ao ser iluminado por luz branca se apresenta azul porque reflete difusamente a luz azul e absorve as demais. Um corpo se apresenta branco porque reflete difusamente todas as luzes componentes da luz branca. Em contrapartida, um corpo negro absorve todas as luzes componentes da luz branca.

### 2.3.3 Refração

Ao passar obliquamente de um meio para outro, o raio de luz muda de direção na superfície de separação, ou seja, refrata-se. Segundo Ramalho [RAM 84], a refração da luz pode ser entendida como a variação de velocidade sofrida pela luz ao mudar de meio. Neste âmbito, os fatores que influenciam a refração da luz são principalmente a razão entre as velocidades de propagação da luz nos meios e, em menor escala, a cor da luz incidente, dado que cada comprimento de onda tem uma velocidade própria.

A FIGURA 2.7 esquematiza o fenômeno de refração da luz. O feixe de raios paralelos que se propaga no meio (1) incide sobre a superfície S e é transmitido para o meio (2).

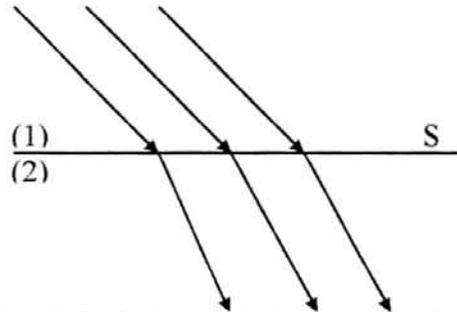


FIGURA 2.7- Refração da luz entre meios transparentes

#### 2.3.3.1 Meios de Propagação da Luz

Segundo a propagação da luz, os materiais classificam-se em:

- Transparentes;
- translúcidos; e
- opacos.

Os meios onde os objetos são vistos com nitidez como o vidro comum, a água e o ar são denominados meios transparentes (trajetória regular da luz). Os meios transparentes têm a capacidade de transmitir a luz sem mudar a direção dos raios. Já o vidro fosco, o papel de seda e o papel vegetal permitem que os objetos sejam vistos através deles mas sem nitidez. São chamados meios translúcidos (trajetória irregular da luz). Materiais como a madeira, por exemplo, não permitem que visualizem-se objetos através deles e por isso são denominados meios opacos (não permitem a propagação da luz).

#### 2.3.3.2 Leis da Refração

A refração da luz segue três leis básicas, válidas para qualquer luz incidente monocromática [KRU 94]. São elas:

- 1ª Lei de Snell-Descartes: O raio incidente, a normal no ponto de incidência e o raio refratado pertencem a um mesmo plano;
- 2ª Lei de Snell-Descartes: A razão entre os senos dos ângulos de incidência e refração é uma constante ( $n$ ) para cada par de meios; e
- 3ª Lei de Foucault: A constante  $n$  da 2ª Lei de Snell-Descartes também pode ser definida como a razão entre as velocidades da luz no meio de incidência e meio de refração.

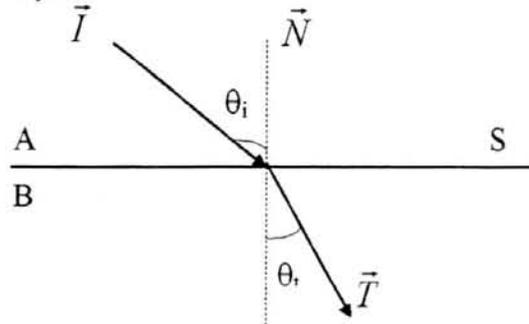


FIGURA 2.8- Raio de luz que passa do meio A para o meio B ( $\theta_i > \theta_r$ )

De acordo com a 2ª Lei de Snell-Descartes, na FIGURA 2.8 tem-se que

$$\frac{\text{sen}(\theta_i)}{\text{sen}(\theta_r)} = n. \quad (2.5)$$

E, levando em consideração a 3ª Lei de Foucault, a equação (2.5) pode ser reescrita como

$$\frac{\text{sen}(\theta_i)}{\text{sen}(\theta_r)} = \frac{v_A}{v_B}, \quad (2.6)$$

onde  $v_A$  e  $v_B$  são as velocidades da luz no meio de incidência (A) e no meio de transmissão (B), respectivamente.

A FIGURA 2.8 ilustra a mudança de direção do raio de luz  $\vec{I}$  ao atingir a superfície que separa os meios A e B. Observe que  $\theta_i$  é maior que  $\theta_r$  e então pela equação (2.6)  $v_A$  é maior que  $v_B$ . Desta forma, a constante  $n$  é maior que 1.

### 2.3.3.3 Índices de Refração

À constante  $n$  da equação (2.5) dá-se o nome de índice de refração relativo do segundo meio em relação ao primeiro. Na FIGURA 2.8,  $n$  é o índice de refração de B em relação a A. O índice de refração relativo indica em que proporção o raio refratado se aproxima ou se afasta da normal.

A razão  $\frac{c}{v_A}$ , onde  $c$  é a velocidade da luz no vácuo dada pela equação (2.1) e  $v_A$  é a velocidade da luz no meio A, é denominada índice de refração absoluto do meio A ( $n_A$ ).

Substituindo na equação (2.6)

$$v_A = \frac{c}{n_A} \quad \text{e}$$

$$v_B = \frac{c}{n_B}$$

e, após algumas operações matemáticas básicas tem-se que

$$\frac{\text{sen}(\theta_i)}{\text{sen}(\theta_r)} = \frac{n_B}{n_A} = n. \quad (2.7)$$

Desta forma, o índice de refração relativo do meio  $B$  em relação ao meio  $A$  é dado pela razão entre os índices de refração absolutos do meio  $B$  e do meio  $A$ , tomados nesta ordem  $\left(\frac{n_A}{n_B}\right)$ .

Os índices de refração absolutos variam com o comprimento de onda da luz ( $\lambda$ ) e com a temperatura no meio de propagação, ou seja,  $n=f(\lambda,t)$  [ASS 96]. A variação do índice de refração em função da temperatura é interessante para simulações de alta precisão. Como este trabalho fundamenta-se no aprimoramento de uma técnica de síntese de imagens, o aspecto temperatura não é considerado na modelagem dos índices de refração.

Dado que o índice de refração varia com o comprimento de onda da luz incidente, uma modelagem fisicamente correta dos índices de refração utilizaria um valor do índice de refração para cada comprimento de onda. Como os feixes de luz estão sendo aproximados pela representação RGB, uma aproximação dos índices de refração como função do comprimento de onda poderia ser obtida através da utilização de valores de  $n$  para cada componente ( $n_r, n_g, n_b$ ). Entretanto, ao utilizar esta aproximação, cada raio refratado originará três novos raios, os quais podem ser refletidos ou transmitidos, ampliando a árvore de raios e, por consequência, o custo computacional. Assim, o índice de refração absoluto para cada meio será tratado como um valor constante, independente do comprimento de onda da luz incidente, ou seja, desconsiderando a refração dispersiva.

#### 2.3.3.4 Reflexão Total

Quando um raio passa de um meio com um índice de refração menor para um meio com um índice de refração maior ( $n_i < n_t$ ), não há restrição quanto à ocorrência de refração, pois  $\theta_t > \theta_i$  para qualquer que seja  $\theta_i$ .

Entretanto, quando um raio se propaga do meio mais refringente para o meio menos refringente ( $n_i > n_t$ ), o raio transmitido se afasta da normal ( $\theta_t < \theta_i$ ). Se o ângulo de refração é maior que  $90^\circ$ , ocorre reflexão ao invés de refração. Este fenômeno surge para ângulos de incidência maiores que o ângulo limite de refração  $L$ , o qual corresponde a um ângulo de refração igual a  $90^\circ$  [KRU 94]. O ângulo  $L$  é determinado por

$$\text{sen}^{-1}\left(\frac{n_t}{n_i}\right).$$

#### 2.3.4 Cálculo dos Vetores Reflexão e Refração

No modelo de iluminação global de Whitted [WHI 80], os termos globais da reflexão especular ( $k_s I_s$ ) e da transmissão ( $k_t I_t$ ) são obtidos com base no esquema da FIGURA 2.9, para cada uma das componentes R, G e B.



Elevando ambos os lados da equação anterior ao quadrado chega-se a

$$\text{sen } \theta', 2 = \frac{k_f^2 |\vec{X}|^2}{(k_f^2 |\vec{X}|^2 + 1)}.$$

Da 2ª Lei de Snell-Descartes sabe-se que  $\text{sen } \theta' = \frac{n_1}{n_2} \text{sen } \theta$ . Substituindo esta relação na equação anterior, juntamente com a equação (2.10) obtém-se

$$\left(\frac{n_1}{n_2}\right)^2 \frac{|\vec{X}|^2}{|\vec{V}'|^2} = \frac{k_f^2 |\vec{X}|^2}{(k_f^2 |\vec{X}|^2 + 1)}.$$

Eliminando  $|\vec{X}|^2$  do numerador de ambos os lados da igualdade, substituindo  $\vec{X} = \vec{V}' + \vec{N}$  e isolando  $k_f$  chega-se a

$$k_f = \left( \left( \frac{n_2}{n_1} \right) |\vec{V}'|^2 - |\vec{V}' + \vec{N}|^2 \right)^{-\frac{1}{2}}.$$

Uma situação de reflexão interna total (ângulo incidente > ângulo limite) é detectada quando o denominador de  $k_f$  for imaginário e, neste caso, a contribuição do raio refratado ( $I_t$ ) será zero.

Para calcular a contribuição especular de Phong no modelo de iluminação local é preciso determinar o vetor reflexão local ( $\vec{R}$ ). Obtém-se este vetor a partir da verificação de duas condições básicas envolvendo os seguintes vetores: o vetor reflexão local ( $\vec{R}$ ), o vetor normal no ponto ( $\vec{N}$ ) e o vetor que parte do ponto para a fonte de luz em questão ( $\vec{L}$ ). As condições são:

- i) Os três vetores ( $\vec{R}$ ,  $\vec{N}$  e  $\vec{L}$ ) estão no mesmo plano, ou seja,  $\vec{N} \times \vec{L} = \vec{R} \times \vec{N}$ ;  
e
- ii) ângulo de incidência igual ao ângulo de reflexão ( $\vec{N} \cdot \vec{L} = \vec{R} \cdot \vec{N}$ ).

## 2.4 Lentes

Uma lente é um objeto transparente com duas superfícies refratoras cujos eixos centrais coincidem [RES 96]. O eixo comum é chamado eixo central da lente e as superfícies refratoras são denominadas faces da lente. Uma lente esférica é determinada pela intersecção de duas superfícies de natureza esférica.

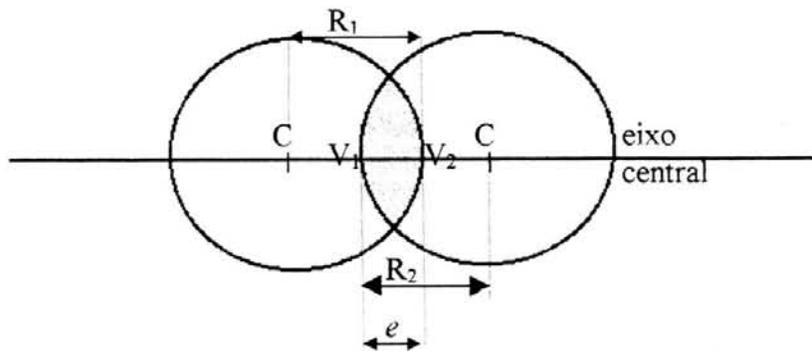


FIGURA 2.10- Elementos de uma lente esférica convergente

As lentes podem ser convergentes ou divergentes. Uma lente é convergente se os raios que incidem paralelos ao eixo central da lente convergem para um mesmo ponto. Ao contrário, se ela os faz divergir, então é uma lente divergente. Deste ponto em diante, todas as considerações sobre lentes serão referentes a lentes esféricas convergentes.

A FIGURA 2.10 apresenta os componentes de uma lente esférica convergente. São eles:

- Centros de curvatura das faces da lente ( $C_1$  e  $C_2$ );
- raios de curvatura das faces da lente ( $R_1$  e  $R_2$ );
- eixo central ou principal da lente. É uma linha reta normal às faces da lente e que passa pelos centros de curvatura da lente;
- vértices das faces da lente ( $V_1$  e  $V_2$ ). São os pontos de intersecção do eixo central com cada uma das faces da lente; e
- espessura da lente ( $e$ ). É a distância entre os vértices da lente.

Segundo Jenkins [JEN 76], considerando-se a óptica gaussiana e a aproximação dos raios centrais, é possível determinar a distância  $i$  do ponto imagem de um objeto colocado sobre o eixo principal da lente ao segundo vértice da lente ( $V_2$ ) a partir dos elementos básicos da lente, apresentados na FIGURA 2.10. Uma forma de determinar a distância  $i$  é através da fórmula dos pontos conjugados. Seja a lente esférica da FIGURA 2.11 de raio de curvatura  $R$  (mesmo para as duas faces), índice de refração  $n$ , imersa no ar e com espessura  $e$ . Um ponto luminoso  $P$  é colocado sobre o eixo da lente a uma distância  $p$  do primeiro vértice da lente. A distância  $i$  do ponto imagem  $I$  ao segundo vértice da lente é determinada pelo seguinte sistema de equações (fórmula dos pontos conjugados)

$$\begin{cases} \frac{1}{p} + \frac{n}{p_1 + e} = \frac{n-1}{R} \\ \frac{1}{i} - \frac{n}{p_1} = \frac{n-1}{R} \end{cases} \quad (2.11)$$

onde,

$p_1$  é a distância da imagem formada considerando-se apenas a primeira face da lente. A distância  $p_1$  é medida até o segundo vértice da lente, ao longo do eixo central.

Para maiores detalhes sobre a fórmula dos pontos conjugados, a equação (2.11), consulte o Anexo A.

Observe que a segunda igualdade da equação (2.11) equivale à primeira igualdade. Esta segunda igualdade tem como características um índice de refração para o meio de incidência igual a  $n$  ( $n_{lente}$ ); um índice de refração para o meio emergente igual 1 ( $n_{ar}$ ); distância do objeto à segunda face igual a  $p_1$ ; distância da imagem à segunda face igual a  $i$  e raio de curvatura igual a  $R$ . Ao substituir estes parâmetros na equação que originou a primeira igualdade, ou seja,

$$\frac{n_{incidente}}{dist\_objeto} + \frac{n_{emergente}}{dist\_imagem} = \frac{n_{emergente} - n_{incidente}}{Raio},$$

obtém-se que

$$\frac{n}{p_1} + \frac{1}{i} = \frac{1-n}{R}.$$

Pela convenção de sinais adotada, na segunda face da lente a variável  $p_1$  e o raio de curvatura  $R$  devem ser negativos. Como a variável  $p_1$  e o raio de curvatura são positivos para a primeira face da lente, substitui-se  $p_1$  por  $-p_1$  e  $R$  por  $-R$  na equação. Assim, a equação resultante é

$$\frac{1}{i} - \frac{n}{p_1} = \frac{n-1}{R},$$

isto é, a própria segunda igualdade da equação (2.11).

A fórmula dos pontos conjugados considera a lente espessa como um sistema de duas lentes. Desta forma, o ponto imagem resultante da refração dos raios na primeira face da lente funciona como o ponto objeto para a segunda face da lente.

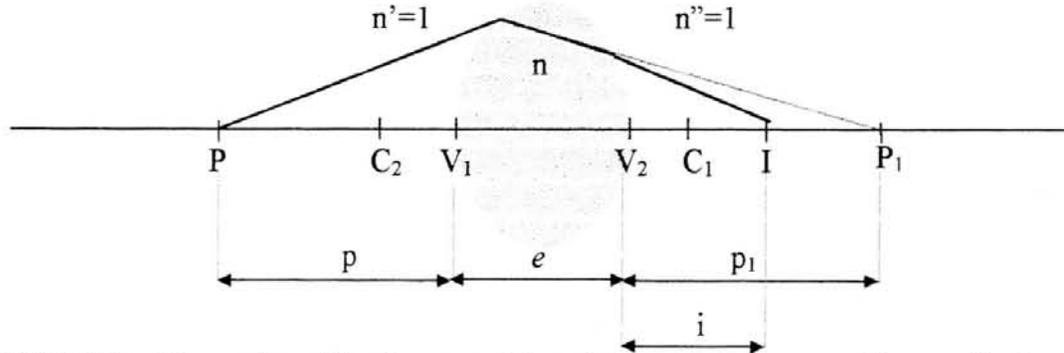


FIGURA 2.11- Elementos utilizados para determinar o ponto imagem ( $I$ ) numa lente espessa

Os pontos focais de uma lente convergente são dois pontos ao longo do eixo central para os quais os respectivos pontos imagens estão no infinito, ou seja, todo raio incidente partindo de um ponto focal refrata paralelo ao eixo central da lente. Analogamente, se o ponto luminoso está no infinito, os raios que partem do ponto incidem paralelos ao eixo central e, portanto, refratam-se passando por um ponto focal da lente.

A distância focal de uma lente espessa imersa num meio com índice de refração  $n_{meio}$  é a distância de qualquer um dos pontos focais ao vértice mais próximo da lente, ao longo do eixo central. Neste caso, a distância do ponto focal 1 ( $f$ ) e a distância do ponto focal 2 ( $f'$ ) são iguais pois o índice de refração do meio que engloba a parte externa da primeira face da lente ( $n'$ ) é igual ao índice de refração que engloba a parte externa da segunda face da lente ( $n''$ ). Para encontrar a distância focal, basta fazer  $p$  na equação (2.11) tender a infinito e a variável  $i$  dará a distância focal ( $f$ ). Assim, a equação (2.11) fica

$$\begin{cases} \lim_{p \rightarrow \infty} \left( \frac{1}{p} + \frac{n}{p_1 + e} \right) = \frac{n-1}{R} \\ \frac{1}{f} - \frac{n}{p_1} = \frac{n-1}{R} \end{cases} \quad (2.12)$$

Como o limite é

$$\lim_{p \rightarrow \infty} \left( \frac{1}{p} + \frac{n}{p_1 + e} \right) = 0 + \frac{n}{p_1 + e}$$

a variável  $p_1$  é obtida da primeira equação por

$$-e + \frac{nR}{(n-1)}.$$

E, a partir da segunda equação tem-se que

$$\frac{1}{f} = \frac{n-1}{R} + \frac{n}{p_1}, \text{ e portanto} \quad (2.13)$$

$$f = \frac{Rp_1}{p_1(n-1) + nR}.$$

Determinar a distância focal da lente é fundamental para identificar onde o ponto imagem situa-se. Dependendo da distância  $p$  do objeto ao primeiro vértice da lente ( $V_1$ ), três situações podem ocorrer. São elas:

- i) O objeto está a uma distância maior que a distância focal da lente ( $p > f$ ); ou
- ii) o objeto está a uma distância igual à distância focal da lente ( $p = f$ ); ou
- iii) o objeto está a uma distância menor que a distância focal ( $p < f$ ).

No caso (i) a lente forma uma imagem  $I$  real, ou seja, no lado oposto à superfície por onde os raios incidiram.

No caso (ii) a imagem do objeto se forma no infinito, ou seja, os raios emergem paralelos ao eixo central da lente.

No caso (iii) a lente forma uma imagem  $I$  virtual, ou seja, no mesmo lado em que está o objeto.

## 2.5 Conclusão

Segundo o modelo de iluminação de Phong, a intensidade num ponto qualquer é determinada a partir das componentes especular, difusa, reflexão global e refração global. Para determinar a componente especular, é preciso conhecer as características de especularidade da superfície, a direção de visada, o vetor normal no ponto e as informações sobre posição e intensidade da fonte de luz, como mostra a equação (2.3). O cálculo da componente difusa, por sua vez, não utiliza a direção de visada, como mostra a equação (2.4). Já a componente de reflexão global é determinada a partir do vetor normal no ponto e do vetor direção de visada. O vetor refração global é calculado pela equação (2.9) levando em consideração a relação entre os índices de refração dos meios de propagação.

Os raios que incidem numa lente esférica convergente partindo de um ponto de luz colocado sobre o eixo central da lente convergem para o ponto imagem, e, a partir deste ponto espalham-se na cena. As lentes esféricas convergentes permitem determinar a distância deste ponto imagem a partir dos seguintes elementos:

- distância da fonte luminosa puntiforme (ponto de luz) à lente (distância  $\gg \gg$  raio);
- índice de refração do meio;
- índice de refração da lente; e
- raios de curvatura da lente.

Para o cálculo do ponto imagem em lentes espessas utiliza-se a óptica gaussiana com aproximação dos raios paraxiais. A óptica gaussiana descreve uma situação ideal, desconsiderando as aberrações das lentes. O cálculo baseia-se na determinação do ponto conjugado do objeto luminoso colocado sobre o eixo central da lente.

A partir dos conceitos básicos aqui apresentados, tem-se um embasamento de Óptica que possibilita dar seqüência ao trabalho.

## 3 Modelos de Iluminação Global

### 3.1 Introdução

Um modelo de iluminação descreve o cálculo da cor num ponto qualquer da superfície. Este modelo pode ser local ou global. Um modelo de iluminação local leva em consideração apenas a luz incidente que provém das fontes de luz. Este modelo trata a luz proveniente da reflexão ou transmissão através da própria superfície ou de outras superfícies como uma constante (comumente denominada luz ambiente). Um modelo de iluminação global, por sua vez, considera a interação entre os diversos objetos presentes na cena. Assim, a luz que incide num ponto pode ter sua origem na iluminação direta (fontes de luz) ou indireta (reflexão ou transmissão a partir de outras superfícies da cena) [ROG 85]. De modo geral, a obtenção de imagens realísticas, que incorporem os efeitos ópticos da interação luz-objeto, exige a utilização de um modelo de iluminação global [CUN 88].

Há duas classes básicas de algoritmos para gerar imagens com um modelo de iluminação global: rastreamento de raios e radiosidade.

O método da radiosidade é independente do ponto de vista do observador. Esta técnica adequa-se à modelagem da componente de reflexão difusa. Entretanto, não se adequa à modelagem da componente de reflexão especular, a qual é totalmente dependente da direção de visada.

Por outro lado, a técnica de rastreamento de raios é dependente do ponto de vista do observador. Este método explicita as componentes de iluminação que representam a luz refletida por outros objetos e a luz transmitida por outros objetos. A componente especular é bem modelada por esta técnica, ao contrário do fenômeno de difusão da luz.

Neste trabalho pretende-se tratar a luz transmitida (refratada) através dos objetos transparentes, um fenômeno fundamentalmente direcional. Assim, o método de síntese utilizado neste trabalho está fundamentado na metodologia de rastreamento de raios. Trata-se de uma extensão do algoritmo de rastreamento de raios convencional, o qual será descrito em detalhes nas seções seguintes.

### 3.2 Rastreamento de Raios

O algoritmo de traçado de raios foi apresentado inicialmente por Apple em [APP 68] como um método para a remoção de superfícies ocultas.

Whitted, em [WHI 80], aprimorou o algoritmo para incluir um modelo global de iluminação, baseado no modelo proposto por Phong em [BUI 75]. Desde então, diferentes algoritmos de rastreamento de raios vêm sendo desenvolvidos para a síntese de imagens realísticas.

Inicialmente, o principal objetivo de tais algoritmos era a redução dos tempos de execução e, para tanto, desenvolveram-se técnicas para:

- A melhoria dos métodos de cálculo das intersecções. Neste âmbito, destacam-se os trabalhos apresentados em [WHI 80], [RUB 80], [BOU 85], [TOT 85], [KAY 86], [HAI 89];
- redução do número de objetos a serem testados na identificação da intersecção mais próxima da origem do raio. Os trabalhos apresentados em [RUB 80], [WEG 84], [DIP 84], [GLA 84], [FUJ 85], [KAY 86], [GOL 87],

[AMA 87], [SNY 87] e [SAM 89] concentraram seus esforços neste objetivo.

Também encontram-se na literatura vários trabalhos objetivando melhorar a qualidade das imagens geradas. Neste contexto, desenvolveram-se técnicas para:

- redução do “aliasing”<sup>1</sup> das imagens. Neste âmbito, destacam-se entre outros os trabalhos de [WHI 80], [AMA 84], [COO 84], [HEC 84], [DIP 85], [LEE 85], [COO 86], [KAY 86], [MIT 87], [SHI 87];
- melhoria do modelo de iluminação utilizado. Trabalhos apresentados em [HAL 83], [KAJ 85], [WOL 90] tratam deste aspecto; e
- produção de efeitos mais complexos nas imagens. Como exemplos citam-se os efeitos de profundidade de campo ([POT 82], [KRU 94]); dispersão da luz ([THO 86]); *motion blur* ([KOR 83], [POT 83]); efeitos cáusticos ([WAT 90]); cor de interferência ([DIA 94]); sombras de objetos transparentes ([KIM 92]); e assim por diante.

O trabalho aqui apresentado fundamenta-se na utilização de uma técnica conhecida como rastreamento bidirecional de raios. Este algoritmo de rastreamento de raios é composto de duas etapas. São elas:

- primeira etapa onde os raios são disparados a partir das fontes de luz para o ambiente; e
- segunda etapa quando os raios são disparados a partir do observador para a cena (traçado convencional de raios).

As próximas seções apresentam as duas técnicas que compõem o rastreamento bidirecional de raios. Primeiramente, o método de rastreamento dos raios disparados do observador para a cena é descrito. Na sequência, o método de rastreamento dos raios disparados das fontes de luz para a cena é apresentado.

### 3.2.1 Traçado de Raios A Partir do Observador

Esta é a forma tradicional de traçado dos raios, apresentada pela primeira vez como algoritmo de iluminação global por Whitted em [WHI 80]. Neste método, o plano onde se forma a imagem é discretizado, e são lançados raios de visada a partir do observador (centro de projeção da câmera sintética) para a cena, como ilustra a FIGURA 3.1.

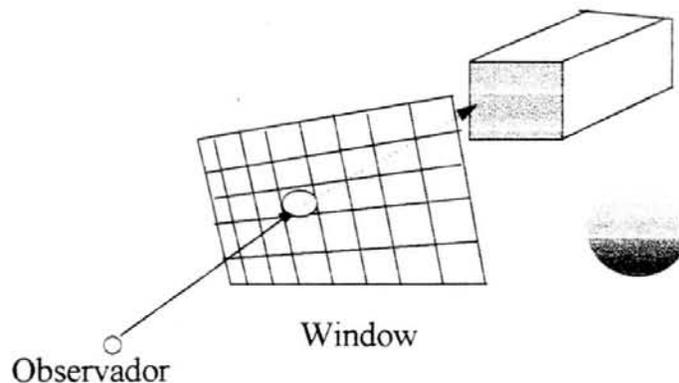


FIGURA 3.1- Modelo básico do algoritmo de traçado de raios tradicional

No algoritmo de Whitted, disparam-se raios para cada pixel particular da imagem. Desta forma, um raio ao atingir um objeto da cena determina um ponto onde a equação de iluminação deve ser avaliada. Neste ponto de intersecção, calcula-se a

<sup>1</sup> Efeitos indesejáveis na imagem final. Resultado da amostragem discreta de pontos.

iluminação local, considerando-se as componentes de reflexão especular e difusa para cada fonte de luz. Já a iluminação global é obtida através da introdução das componentes reflexão e transmissão globais. Estes termos representam a luz “emitida” pelo objeto, originada de outros pontos da cena. Para a determinação dos termos globais (reflexão e transmissão), geram-se dois novos raios, a saber:

- Um raio na direção de reflexão; e
- um raio na direção de refração.

O raio refletido é disparado quando o objeto apresenta uma superfície refletora (coeficiente de reflexão especular diferente de zero). O raio transmitido, por sua vez, é disparado quando o objeto tem certo grau de transparência (coeficiente de transmissão diferente de zero). Estes novos raios “percorrem” a cena, trazendo para o ponto de intersecção a contribuição de outros objetos atingidos direta ou indiretamente por eles. Assim, finaliza-se o cálculo da iluminação em um ponto quando cada um dos raios gerados para aquele ponto (refletido e transmitido) retornam, trazendo a contribuição dos outros objetos da cena. É esta característica recursiva do algoritmo de rastreamento de raios a responsável por seu elevado custo computacional.

Um problema com o rastreamento de raios convencional é causado por se traçar os raios a partir do observador [FOL 90]. O algoritmo não reproduz apropriadamente os efeitos das fontes de luz refletidas e refratadas. Um exemplo é o caso das sombras de objetos transparentes que não evidenciam refração. Isto ocorre porque os raios de sombra são traçados considerando a linha geométrica que liga o ponto à fonte luminosa e não a linha óptica. Observe-se a FIGURA 3.2. O raio incidente,  $\vec{I}$ , que pode ter seu ponto de origem no observador ou em algum objeto da cena, atinge o objeto transparente que “deixa” parte da luz atravessá-lo, na direção de refração,  $\vec{I}_{refratado}$ . O raio atravessa a superfície do objeto e refrata-se na direção  $\vec{I}_{refratado}$ . Em um algoritmo de rastreamento de raios convencional, o raio de sombra é traçado como a linha geométrica que liga a fonte de luz ao ponto de incidência e não como a linha óptica ( $\vec{I}_{refratado}$ ).

O método de traçar raios do observador para a cena utiliza a linha de visada geométrica no cálculo da iluminação direta para garantir que o raio atingirá a fonte de luz. Caso o raio fosse refratado ao atravessar um objeto transparente, o novo raio provavelmente não atingiria a fonte luminosa. E, neste caso, novos raios seriam gerados até que um deles atingisse a fonte luminosa ou fosse possível concluir que o ponto está em sombra.

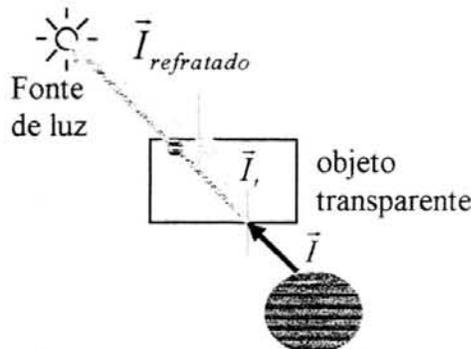


FIGURA 3.2- Sombra em rastreamento de raios convencional

Uma solução para o problema descrito acima é o traçado de raios a partir das fontes de luz. Esta técnica é apresentada na próxima seção.

### 3.2.2 Traçado de Raios A Partir das Fontes de Luz

Este método, também denominado rastreamento de raios inverso, fundamenta-se numa idéia bastante natural, que vai de encontro ao comportamento físico real de um modelo de iluminação. Os feixes de luz são disparados a partir das fontes de luz na direção da cena.

Ainda que a idéia seja bastante simples, um problema da técnica está relacionado com a amostragem da cena. Como determinar as direções de disparos iniciais de forma a se obter uma amostragem suficiente e adequada para cada pixel da tela? Dependendo da posição relativa dos objetos na cena tais direções podem ser estimadas. Entretanto, para uma situação genérica, não há como determiná-las antecipadamente com confiabilidade e precisão.

Assim, a utilização de uma técnica de rastreamento de raios inverso “pura” exige um grande número de disparos iniciais para a obtenção de uma amostragem com probabilidade de ser adequada. Por este motivo, este método tem sido utilizado para suplementar a informação obtida com rastreamento de raios convencional. As propostas apresentadas em [HEC 84] e [ARV 86] exemplificam a utilização do método híbrido, no qual o traçado de raios a partir das fontes de luz é utilizado para enriquecer o método tradicional de traçar os raios a partir do observador.

## 3.3 Conclusão

Este capítulo apresentou uma descrição do principal algoritmo para a síntese de imagens realísticas, o rastreamento de raios. Foram identificados os principais problemas relacionados com o método.

O rastreamento de raios convencional, onde os raios iniciais são disparados a partir do observador, não reproduz adequadamente a refração da luz que provém diretamente das fontes de luz. Então, pretende-se utilizar um método híbrido para modelar a refração da luz que provém das fontes luminosas da cena. Este método, denominado de rastreamento bidirecional de raios, consiste em duas etapas de disparo dos raios. Na primeira etapa os raios são traçados a partir das fontes de luz. Na segunda etapa os raios são traçados a partir do observador, considerando-se a informação obtida pela primeira etapa do algoritmo.

## 4 A Síntese de Objetos Transparentes

### 4.1 Introdução

Um aspecto importante na visualização de objetos com realismo é a simulação de transparência [KRU 94]. Com a modelagem desta característica é possível simular materiais como vidro, água e plásticos.

Este capítulo apresenta as principais técnicas existentes para a síntese de objetos com certo grau de transparência<sup>2</sup>. Inicialmente, é apresentada a simulação de transparência sem os efeitos de refração. Na sequência, são descritos os principais métodos para a síntese de objetos transparentes considerando o fenômeno de refração da luz.

### 4.2 A Síntese de Objetos Transparentes Sem Efeitos de Refração

O método mais simples para a síntese de objetos transparentes ignora o fenômeno da refração. Desta forma, os objetos visíveis através de um objeto transparente estão localizados geometricamente na linha de visada (observação). Para exemplificar, veja a FIGURA 4.1. Se o fenômeno óptico da refração for considerado, o objeto *A* será visível no ponto determinado pela intersecção do raio que parte do observador e atinge a superfície (*P*). Entretanto, se o fenômeno de refração não for considerado, o objeto *B* será visível neste ponto.

A síntese de objetos transparentes sem os efeitos de refração não é congruente com realismo, porém, é útil em aplicações específicas, como por exemplo visualização do interior de objetos complexos [FOL 90].

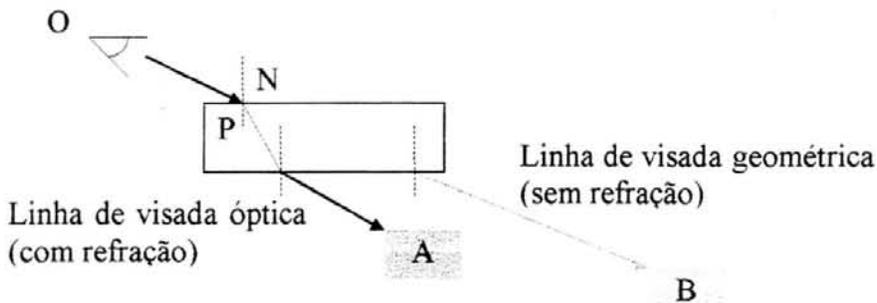


FIGURA 4.1- Linhas de visada óptica e geométrica

Krug, em [KRU 92], descreve três métodos para aproximar a forma como as cores de dois objetos podem ser combinadas quando um dos objetos é visualizado através do outro, sem os efeitos de refração. São eles:

- i) Interpolação;
- ii) Filtragem; e
- iii) Aleatoriedade.

O método de interpolação calcula a intensidade do *pixel* na intersecção das projeções das faces correspondentes de dois objetos pela interpolação linear das intensidades individuais de cada face. Então,

$$I = (1 - k_{t1})I_1 + k_{t1}I_2.$$

<sup>2</sup> Grau de transparência é a porção de luz que atravessa o objeto por unidade de distância percorrida no meio material do objeto.

Nesta equação,  $I_1$  e  $I_2$  são, respectivamente, as intensidades das superfícies dos objetos 1 e 2, onde o objeto 1 está mais próximo do observador e  $k_{t1}$  é o coeficiente de transparência do objeto 1 e assume valores no intervalo  $[0,1]$ . Pela equação de interpolação das intensidades, quando o objeto 1 é transparente ( $k_{t1} \neq 0$ ), calcula-se a intensidade  $I$  em  $P$  como a fração de  $I_2$  que atravessa o objeto 1 acrescida da intensidade restante que provém do próprio objeto 1. Porém, se o objeto 1 é opaco ( $k_{t1} = 0$ ), então a intensidade  $I$  é igual à intensidade do próprio objeto 1 ( $I_1$ ).

O método de filtragem considera o objeto como um filtro transparente que permite a passagem seletiva de determinados comprimentos de onda. Este método é modelado pela equação

$$I = I_1 + k_{t1} O_t I_2.$$

Analogamente ao método de interpolação,  $I_1$  e  $I_2$  são as intensidades das superfícies correspondentes dos objetos 1 e 2 e o objeto 1 está mais próximo do observador,  $k_{t1}$  é o coeficiente de transparência do objeto 1 e  $O_t$  é a cor de transparência deste objeto, isto é, uma terna de valores, onde cada valor corresponde a um componente da representação RGB. É a cor de transparência que determina os comprimentos de onda transmitidos através do objeto transparente, ou seja, é a responsável pela filtragem de cores. Por causa do caráter ondulatório da luz, a transmissão de luz através de um objeto transparente não é um fenômeno determinístico. Esta característica dá origem a efeitos como a cor de interferência [DIA 94]. Semelhantemente ao método de interpolação, o cálculo por filtragem também é uma aproximação da forma como as cores podem ser combinadas.

O método de aleatoriedade utiliza uma função de geração de números aleatórios, que produz um valor  $r$  no intervalo  $[0,1]$ . O cálculo da intensidade é então descrito pela função

$$I = \begin{cases} I_2, & \text{se } r < k_{t1} \\ I_1, & \text{se } r \geq k_{t1} \end{cases}$$

O método compara o valor  $r$  com o coeficiente de transparência  $k_{t1}$  do objeto 1. Dependendo do resultado desta comparação, a intensidade no ponto de intersecção das projeções dos dois objetos é determinada como sendo  $I_1$  ou  $I_2$ . É um método interessante para modelar objetos constituídos por uma malha de retículos.

### 4.3 A Síntese de Objetos Transparentes Com Refração

Uma vez que as linhas de visada óptica e geométrica são diferentes quando considera-se o fenômeno refração, a modelagem de transparência com refração se torna mais complexa que a modelagem sem refração. Veja a FIGURA 4.1. No cálculo da intensidade de um ponto em um objeto transparente com refração, inicialmente determina-se o objeto mais próximo, localizado na linha de visada óptica (objeto  $A$ ). Após, faz-se a combinação das cores dos dois objetos, podendo-se inclusive utilizar qualquer dos métodos descritos na seção anterior neste processo.

Na literatura, identificaram-se dois métodos principais para a modelagem de objetos transparentes com refração. São eles:

- i) Método de Kay-Greenberg; e
- ii) Método de transparência com rastreamento de raios.

Nas próximas seções, estes métodos são descritos em linhas gerais.

### 4.3.1 Método de Kay-Greenberg

Este método, proposto por Kay e Greenberg em [GRE 79], faz duas suposições para reduzir a carga computacional de simulação da refração. O custo computacional de simular a refração concentra-se no cálculo das intersecções do raio de visada com as superfícies da cena em cada mudança de direção do raio. O método supõe que:

- Os raios de visada sempre viajam paralelos ao eixo  $z$ , independente do meio de propagação. Desta forma, a posição atual do objeto  $(x,y)$  é determinante na identificação da próxima superfície a ser interseccionada; e
- As fontes de luz podem ser vistas diretamente a partir de qualquer superfície da cena.

O algoritmo pode ser implementado utilizando-se qualquer método de sombreado. Ele utiliza um vetor de refração, determinado em cada mudança de meio de propagação. Com o vetor de refração e as características físicas do material transparente, calculam-se os incrementos para  $x$  e  $y$  de forma a desviar o raio, mantendo-o sempre paralelo ao eixo  $z$ .

### 4.3.2 Rastreamento de Raios

Rastreamento de raios é uma técnica para a síntese de imagens, cuja fundamentação teórica está na simulação numérica da óptica geométrica. Trata-se de um método que determina a visibilidade das superfícies pelo traçado de raios de luz imaginários, partindo do observador até os objetos da cena [FOL 90]. A intensidade de cada ponto visível é determinada pela equação de iluminação utilizada, onde são modelados os efeitos ópticos de reflexão especular, reflexão difusa e refração. Para obter maiores informações sobre a técnica rastreamento de raios consulte a seção “Rastreamento de Raios”, no capítulo “Modelos de Iluminação Global” desta dissertação.

O rastreamento de raios recursivo de Whitted é adequado para simular a linha de visada óptica da FIGURA 4.1. Com este método, o observador veria o objeto  $A$  em  $P$ , como esperado.

Neste algoritmo verifica-se se um ponto está ou não recebendo luz diretamente de uma fonte luminosa da cena através do traçado de um “raio de sombra” para aquela fonte de luz, o qual é representado pela linha geométrica que liga o ponto à fonte luminosa. Se este “raio de sombra” atinge um objeto opaco, conclui-se que o ponto está em sombra (não recebe luz) para aquela fonte luminosa. Entretanto, se este raio atingir um objeto transparente, a intensidade da luz da fonte luminosa que chega ao ponto é calculada como uma fração da intensidade da fonte de luz. Esta fração é ponderada pelo coeficiente de transparência do objeto atingido e pela distância percorrida pela luz. Observe-se que este tratamento não leva em consideração o fenômeno óptico de refração da luz. Todavia, como foi visto na seção “Traçado de Raios a Partir das Fontes de Luz”, uma fase inicial com o traçado dos raios a partir das fontes de luz pode suprir esta deficiência de realismo.

## 5 Bitracer - Um Modelo para a Síntese de Lentes Esféricas

### 5.1 Introdução

Este capítulo apresenta um modelo para a simulação de lentes esféricas convergentes. O trabalho baseia-se na geração de fontes secundárias de luz, cuja origem está relacionada com o fenômeno de refração da luz nas lentes. Os raios de luz que proporcionam este fenômeno de refração provêm diretamente das fontes luminosas puntiformes da cena. Desta forma, as fontes secundárias são obtidas como imagens das fontes primárias através do traçado de raios que partem destas (FIGURA 5.1).

As fontes secundárias podem originar fontes terciárias que, por sua vez, podem originar fontes quaternárias, e assim por diante. Neste trabalho foi estabelecida a seguinte condição de contorno: restringir as fontes-imagem a fontes luminosas geradas pelo primeiro nível de traçado, isto é, fontes luminosas secundárias, originadas diretamente das fontes luminosas definidas no arquivo descritor da cena.

Este capítulo está estruturado como se verá a seguir. Inicialmente, o ambiente de modelagem é apresentado. A seguir, discutem-se elementos essenciais ao algoritmo, como o espalhamento luminoso, a determinação da calota visível, a identificação de obstáculos e a definição das fontes secundárias de luz. Na seqüência, descreve-se em detalhes a primeira fase do algoritmo de traçado de raios bidirecional. A seguir, a segunda etapa do algoritmo é descrita, com ênfase na utilização das informações obtidas na primeira fase do algoritmo. Finalmente, são apresentadas as principais conclusões do capítulo.

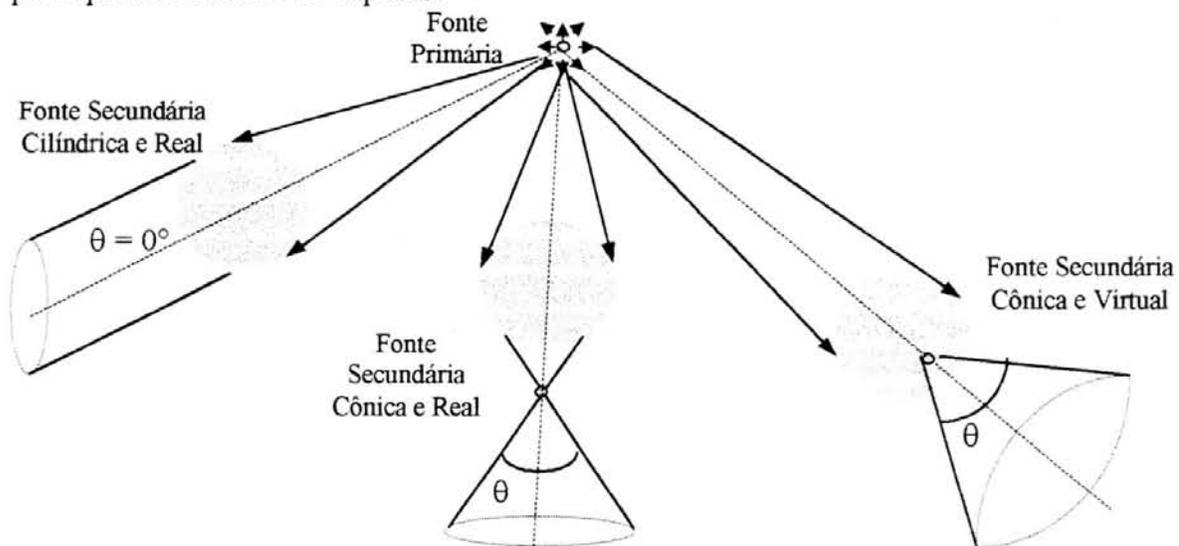


FIGURA 5.1- Fontes luminosas secundárias. O ângulo  $\theta$  é o espalhamento da fonte luminosa secundária.

### 5.2 Esferas

O ambiente de modelagem é constituído de esferas. O objeto esfera, largamente utilizado em Computação Gráfica, permite a modelagem de objetos transparentes como lentes esféricas convergentes. Além disso, a esfera apresenta facilidades de modelagem e de cálculos de intersecção.

Define-se uma esfera de centro  $C_E(x_c, y_c, z_c)$  e raio  $r$  como sendo o conjunto de pontos  $P(x, y, z) \in \mathbb{R}^3$  tais que

$$(x - x_c)^2 + (y - y_c)^2 + (z - z_c)^2 \leq r^2,$$

onde

$C_E(x_c, y_c, z_c)$  é definido no sistema de referência do universo de modelagem.

Para o cálculo da intersecção de um raio luminoso com a superfície da esfera, torna-se necessário solucionar um sistema de equações que envolve a equação que define a superfície da esfera e a equação do raio, tendo-se o sistema

$$(x - x_c)^2 + (y - y_c)^2 + (z - z_c)^2 = r^2, \text{ e} \quad (5.1)$$

$$\begin{cases} x = x_0 + tx_r \\ y = y_0 + ty_r \\ z = z_0 + tz_r \end{cases} \quad (5.2)$$

Na equação (5.2),  $(x_0, y_0, z_0)$  é o ponto de origem do raio e  $(x_r, y_r, z_r)$  são as componentes do vetor direção do raio.

Usando para substituição a equação (5.2) na equação da superfície da esfera (5.1), obtém-se como solução

$$t = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a},$$

onde

$$\begin{aligned} a &= (x_r^2 + y_r^2 + z_r^2), \\ b &= 2[x_r(x_0 - x_c) + y_r(y_0 - y_c) + z_r(z_0 - z_c)], \text{ e} \\ c &= [(x_0 - x_c)^2 + (y_0 - y_c)^2 + (z_0 - z_c)^2 - r^2]. \end{aligned}$$

Se o vetor direção está normalizado, então  $a = 1$ , e

$$t = -d \pm \sqrt{d^2 - c}, \quad (5.3)$$

onde  $d = \frac{b}{2}$ .

Se não há solução real para a equação (5.3), ou seja,  $d^2 - c < 0$ , o raio não intercepta a esfera. Se a equação (5.3) tem duas soluções reais,  $t_1$  e  $t_2$ , o raio intercepta a esfera desde que pelo menos uma das soluções esteja contida no intervalo  $(0, 1]$ .

### 5.3 Espalhamento Luminoso

A esfera transparente comporta-se como uma lente que ao receber raios luminosos de uma fonte cria uma imagem desta formando uma fonte secundária de luz.

De acordo com a posição da fonte primária em relação à esfera têm-se fontes secundárias reais ou virtuais (FIGURA 5.1). As fontes secundárias reais podem estar a uma distância finita (cônicas) ou infinita (cilíndricas) da esfera. Neste caso, os raios emergentes são paralelos entre si.

Para se obter uma fonte secundária real cilíndrica, a fonte primária de luz deve ser colocada no foco da esfera vista como uma lente.

Uma fonte secundária virtual é obtida colocando-se a fonte primária entre o foco e a esfera.

Finalmente, para se obter uma fonte secundária real cônica da esfera, coloca-se a fonte primária antes do foco da lente (FIGURA 5.2).

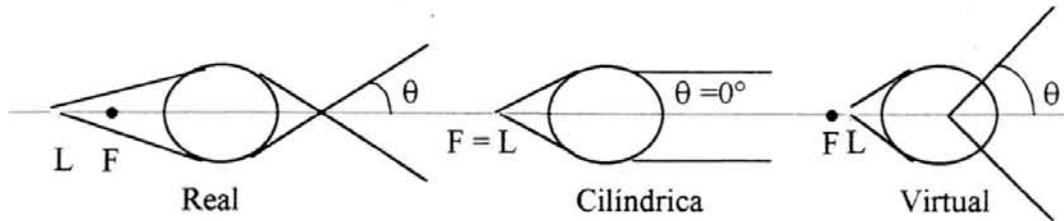


FIGURA 5.2- Tipos de fontes secundárias quanto à posição da fonte primária  $L$  em relação a esfera. O ponto  $F$  é o foco da lente.

O ângulo de abertura do cone resultante dos raios-imagem é denominado de espalhamento  $\theta$  da fonte secundária.

## 5.4 Calota Visível

Calota iluminada ou visível, impropriamente, é a região da esfera que pode ser iluminada ou impropriamente vista a partir de uma fonte de luz puntiforme. Ela é delimitada pelo chamado círculo de base.

Veja-se a FIGURA 5.3. Seja a fonte luminosa puntiforme localizada no ponto  $L_j(x_j, y_j, z_j)$  no sistema de referência do universo da cena. Seja a esfera  $i$  definida geometricamente pelo seu centro  $C_E(x_i, y_i, z_i)$  e pelo seu raio  $R_E$ . Para determinar a calota visível da esfera  $i$  vista a partir de  $L_j$ , traçam-se raios  $\vec{R}_T$  que partem de  $L_j$  e tangenciam a superfície da esfera num ponto genérico  $P$ . Estes raios tangentes à esfera são normais formando um ângulo de  $90^\circ$  com o vetor  $\vec{R}_E$  que parte do centro da esfera  $C_E$  até o ponto de intersecção  $P$ .

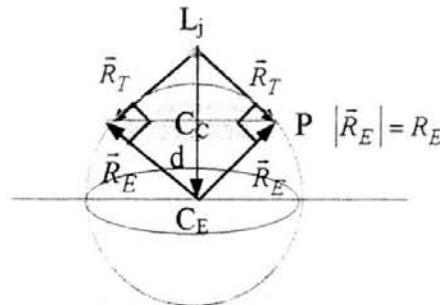


FIGURA 5.3- Determinação da calota da esfera  $i$  visível a partir da fonte luminosa  $L_j$

Deve-se extrair da estrutura da calota visível as seguintes informações:

- Ângulo radial da esfera, visto de  $L_j$ . Por ângulo radial da esfera  $i$  visto do ponto  $L_j$  entende-se o ângulo formado por dois vetores ligados a  $L_j$  tais que um dirige-se para o centro da esfera e o outro tangencia a superfície da esfera. São eles  $L_j C_E$  e  $\vec{R}_T$  na FIGURA 5.3; e
- distância  $R_T = |\vec{R}_T|$  da fonte luminosa  $L_j$  ao ponto  $P$  de tangência na superfície da calota visível.

O cálculo do ângulo radial é necessário para qualquer esfera, seja transparente ou não.

A distância  $R_T$  da fonte luminosa primária à circunferência de base da calota visível é utilizada de forma a restringir o número de objetos para os quais deve-se efetuar o teste de visibilidade angular, que será visto mais adiante. Calcula-se esta

distância para a esfera transparente em questão e para cada esfera candidata a obstáculo.

Na FIGURA 5.4, tem-se que  $\alpha$  é o ângulo radial da esfera  $i$ , visto a partir de  $L_j$  e  $R_T$  é o módulo do vetor que parte de  $L_j$  e é tangente à superfície da esfera  $i$  no ponto  $P$ .

No triângulo  $\Delta L_j C_E P$  tem-se

$$\cos \alpha = \frac{R_T}{|\vec{L_j C_E}|}. \quad (5.4)$$

Aplicando o teorema de Pitágoras ao  $\Delta L_j C_E P$ , obtém-se

$$R_T = \sqrt{|\vec{L_j C_E}|^2 - R_E^2}. \quad (5.5)$$

E o ângulo radial  $\alpha$  da esfera visto de  $L_j$ , por sua vez, é dado a partir da equação (5.4) por

$$\alpha = \cos^{-1} \left( \frac{R_T}{|\vec{L_j C_E}|} \right). \quad (5.6)$$

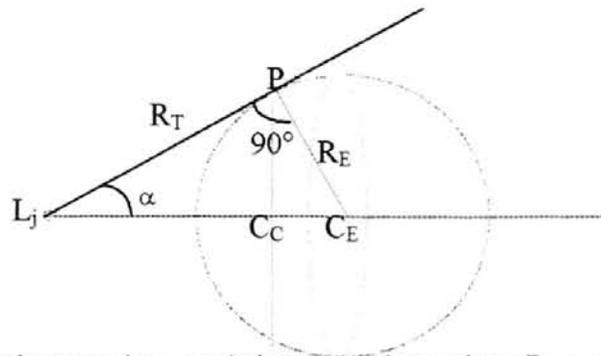


FIGURA 5.4- Relações trigonométricas para determinar  $R_T$  e  $\alpha$ . A medida  $R_T$  é o módulo do vetor tangente à esfera e  $\alpha$  é o ângulo radial da esfera  $i$  visto de  $L_j$ .

## 5.5 Identificação de Obstáculos

Uma fonte luminosa secundária pode ser cheia ou não-cheia. Uma fonte secundária é cheia quando apresenta uma distribuição contínua de luminosidade dentro de seu cone ou cilindro luminoso. Caso algum objeto esteja obstruindo parcialmente a luz que parte da fonte primária e chega à esfera transparente, tem-se uma fonte secundária não-cheia (FIGURA 5.5). Se há algum objeto que obstrui completamente a luz que parte da fonte luminosa primária e atinge a esfera transparente (calota visível), a fonte luminosa secundária correspondente não será criada.

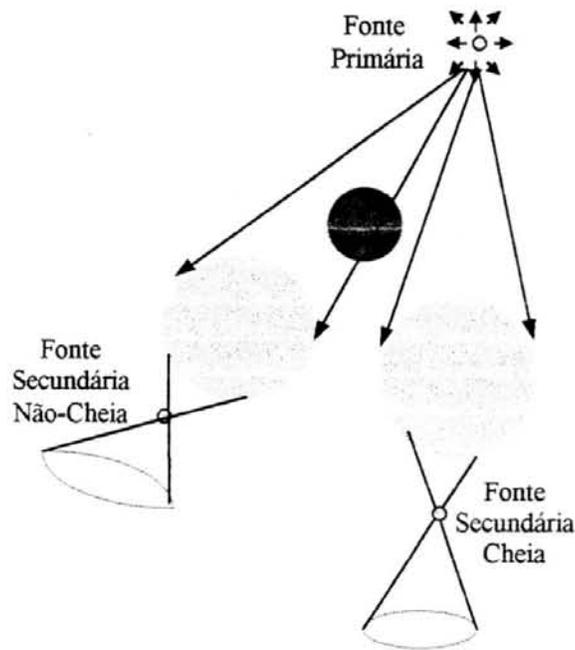


FIGURA 5.5- Tipos de fontes secundárias: cheia e não-cheia

Na segunda etapa do algoritmo de rastreamento bidirecional, as componentes especular e difusa do modelo de iluminação são obtidas traçando-se raios do ponto até cada uma das fontes de luz primárias e secundárias. Considere-se o raio  $\vec{L}$  sendo traçado de um ponto  $P$  qualquer para uma fonte secundária não-cheia  $L_{ji}$ . Assuma-se que  $\vec{L}$  esteja contido no cone ou cilindro de atuação da fonte  $L_{ji}$ . O raio  $\vec{L}$  deve ser refratado dentro da esfera transparente de origem e depois refratado da esfera para o meio (ar). Caso o raio final atinja algum obstáculo, o ponto  $P$  não estará recebendo luz da fonte secundária  $L_{ji}$ . Porém, se  $L_{ji}$  é uma fonte cheia, não é necessário refratar o raio  $\vec{L}$  dentro da esfera transparente de origem e depois da esfera para o ar, pois não há obstáculos que possam estar impedindo a luz refratada proveniente da fonte primária original de atingir o ponto  $P$ .

Buscando evitar um processamento adicional para detectar quando existem esferas que interceptam a luz que parte da fonte primária  $L_j$  para a esfera transparente  $i$ , cada uma obstruindo parcialmente a luz e todas em conjunto impedindo completamente que a luz de  $L_j$  chegue à esfera  $i$ , decidiu-se que a fonte secundária somente não será criada quando uma esfera estiver obstruindo totalmente a luz que parte de  $L_j$  para a esfera  $i$ .

O procedimento para determinar se a esfera  $k$  obstrui a luz que provém da fonte luminosa puntiforme  $L_j$  para a esfera transparente  $i$  consiste em verificar se a esfera  $k$  está entre a fonte luminosa puntiforme  $L_j$  e a esfera transparente  $i$  (FIGURA 5.6). O processo de detecção de obstáculos é composto de dois testes, os quais são descritos em detalhes a seguir.

O primeiro teste identifica se a esfera  $k$  está mais próxima da fonte luminosa  $L_j$  que a esfera transparente  $i$ . Para isso, faz-se uma comparação entre os módulos dos vetores oriundos da fonte luminosa primária  $L_j$  e que tangenciam a esfera transparente  $i$  e a candidata a obstáculo  $k$ . Se o módulo do vetor tangente à esfera candidata a obstáculo for menor que o módulo do vetor tangente à esfera transparente, então a esfera candidata pode estar projetando sombra na esfera transparente e, portanto, deve-se proceder ao segundo teste, a saber, o teste de visibilidade angular. Este primeiro teste é denominado teste das distâncias e pode ser definido como  $d_k < d_i$ ,

onde  $d_i$  é a medida do segmento tangente à esfera transparente  $i$  e  $d_k$  é a medida do segmento tangente à esfera candidata a obstáculo  $k$ , conforme ilustra a FIGURA 5.6. Calculam-se as distâncias  $d_k$  e  $d_i$  pela equação (5.5). Este teste é o critério de seleção do laço repetitivo do passo 1.2.6 da seção 5.7.1.

A seguir, faz-se o teste de visibilidade angular (segundo teste de visibilidade). Trata-se de verificar se a calota visível da esfera  $k$  candidata a obstáculo está entre a fonte luminosa primária  $L_j$  e a calota visível da esfera  $i$  transparente. Este teste trabalha no plano definido pelos seguintes pontos: Centro da esfera transparente ( $C_i$ ), centro da esfera candidata a obstáculo ( $C_k$ ) e posição da fonte luminosa primária ( $L_j$ ). O critério utilizado para assegurar que uma das esferas obstrui a outra, seja total ou parcialmente, é que a soma dos ângulos radiais das esferas tomados a partir de  $L_j$ , ( $\alpha + \beta$ ), seja maior ou igual à distância angular  $\gamma$  dos centros das esferas, também tomada em relação a  $L_j$ .

Os ângulos radiais  $\alpha$  e  $\beta$  são determinados através da equação (5.6). A distância angular  $\gamma$ , por sua vez, é obtida pelo produto escalar dos vetores  $\vec{L_j C_k}$  e  $\vec{L_j C_i}$ . Sabe-se que o co-seno do ângulo entre dois vetores é igual ao produto escalar dos vetores dividido pelo produto de seus módulos. Assim,  $\gamma$  é calculado por

$$\gamma = \cos^{-1} \left( \frac{(\vec{L_j C_k}) \cdot (\vec{L_j C_i})}{|\vec{L_j C_k}| |\vec{L_j C_i}|} \right) \quad (5.7)$$

Assumindo que a fonte luminosa puntiforme  $L_j$  não pertença à superfície de nenhum objeto da cena, ou seja,  $d_k > 0$ , para todo  $0 \leq k \leq n-1$ , onde  $n$  é o número de objetos da cena, é possível afirmar que os ângulos radiais  $\alpha$  e  $\beta$  da FIGURA 5.6 estão contidos no intervalo aberto ( $0^\circ, 90^\circ$ ). Desta forma, tem-se que a soma dos ângulos radiais ( $\alpha + \beta$ ) está contida no intervalo aberto ( $0^\circ, 180^\circ$ ).

Como a distância angular entre os centros das esferas ( $\gamma$ ) é o menor ângulo entre dois vetores de mesma origem no espaço, pode-se afirmar que  $0^\circ \leq \gamma \leq 180^\circ$ . O teste de visibilidade angular  $(\alpha + \beta) \geq \gamma$  é utilizado no critério de decisão do comando condicional 1.2.7.4. da seção 5.7.1.

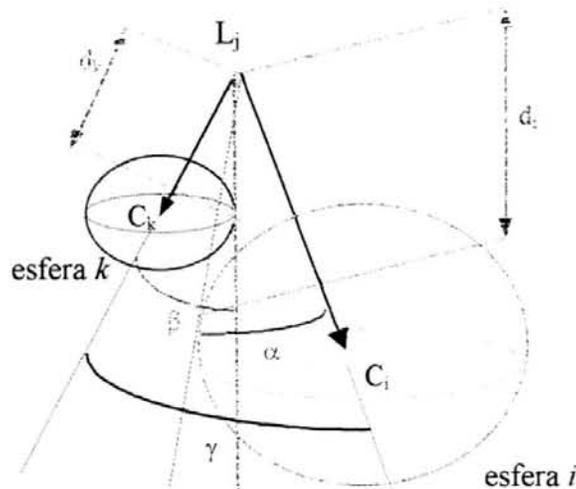


FIGURA 5.6- Testes de visibilidade

Se a esfera  $k$  está entre a fonte luminosa primária  $L_j$  e a esfera transparente  $i$ , ou seja, satisfaz a ambos os teste de visibilidade, o próximo passo é verificar se a esfera  $k$  obstrui completamente a esfera  $i$ . Caso isto ocorra, a fonte secundária  $L_{ji}$  não será criada pois a luz que parte de  $L_j$  não chega à esfera transparente  $i$ . O teste

correspondente à obstrução total da esfera transparente  $i$  pela esfera  $k$  (obstáculo) consiste em verificar se a calota visível da esfera  $k$  encobre a calota visível da esfera  $i$ . Este teste é descrito em detalhes a seguir.

Toda esfera  $k$  que obstrui completamente a luz que parte de  $L_j$  para a esfera  $i$  tem um ângulo radial visto de  $L_j$  maior ou igual ao ângulo radial da esfera  $i$ , também tomado a partir de  $L_j$ . Assim, o primeiro passo é verificar se o ângulo radial  $\beta$  da esfera  $k$  é maior ou igual ao ângulo radial  $\alpha$  da esfera  $i$ . Entretanto, esta condição não é suficiente para garantir que a esfera  $k$  obstrui totalmente a luz de  $L_j$  para a esfera  $i$ , como pode ser visto na FIGURA 5.7. Assim, torna-se necessário formular um teste mais abrangente para identificar a situação de obstrução total. Observe-se a FIGURA 5.8. Nesta figura tem-se que  $\beta \geq \gamma + \alpha$  e a esfera  $k$  obstrui totalmente a luz que parte de  $L_j$  para a esfera  $i$ . Portanto, a esfera  $k$  obstrui totalmente a luz que provém de  $L_j$  para a esfera transparente  $i$  quando o ângulo radial  $\beta$  da esfera obstáculo  $k$  for maior ou igual à soma do ângulo radial  $\alpha$  da esfera  $i$  com a distância angular  $\gamma$  entre os centros das esferas, todas tomadas a partir de  $L_j$ .

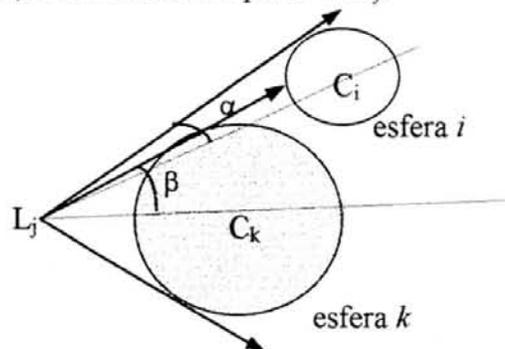


FIGURA 5.7- Exemplo de obstrução parcial. Nesta situação  $\beta > \alpha$  e a esfera  $k$  não obstrui completamente a esfera transparente  $i$ .

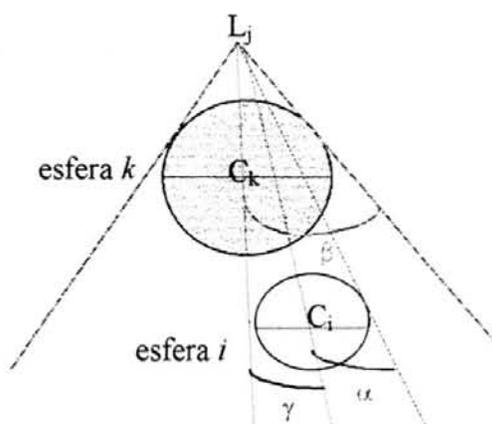


FIGURA 5.8- Exemplo de obstrução total. Nesta situação a esfera  $k$  obstrui completamente a esfera transparente  $i$ .

## 5.6 Fonte Secundária de Luz

Uma fonte luminosa secundária não está definida no arquivo descritor da cena. Ela tem sua origem no fenômeno de transmissão da luz através de uma esfera transparente.

Os objetos transparentes no modelo proposto são esferas e, portanto, comportam-se como lentes esféricas convergentes. Assim, a estrutura e os cálculos associados às fontes secundárias de luz baseiam-se nos princípios da Óptica clássica para lentes esféricas convergentes, apresentados na seção 2.4 desta dissertação.

Uma fonte secundária de luz é composta dos seguintes elementos:

- Posição;
- co-seno do ângulo de espalhamento ou ângulo de abertura;
- índice da esfera transparente que originou a fonte secundária;
- índice da fonte puntiforme (primária) que originou a fonte secundária;
- intensidade; e
- lista de sombras.

A posição da fonte secundária de luz  $L_{ji}$  é determinada através do ponto-imagem correspondente à fonte luminosa puntiforme original  $L_j$ . Na segunda etapa do algoritmo, utiliza-se esta posição na determinação do “raio de sombra” que parte de um ponto qualquer da cena para a posição da fonte secundária  $L_{ji}$ . Um raio de sombra é um raio disparado do ponto da cena para o qual está sendo feito o cálculo de intensidade até uma fonte luminosa, de forma a identificar se o ponto está recebendo luz ou se está na sombra da fonte em questão.

Com o índice  $i$  da esfera transparente, que originou a fonte secundária  $L_{ji}$ , obtém-se da lista de objetos da cena a esfera  $i$ . Da mesma forma, o índice  $j$  da fonte primária original possibilita identificar a fonte luminosa puntiforme  $L_j$  a partir da lista de fontes luminosas da cena. Outra condição de contorno do modelo é a consideração de que o vetor  $L_j\vec{C}_i$  que parte da fonte primária original para o centro da esfera transparente representa o vetor eixo central da lente esférica (material isotrópico). Utiliza-se o vetor eixo central juntamente com o co-seno do ângulo de espalhamento na segunda etapa do algoritmo para verificar se o raio de sombra está contido ou não na área iluminada pela fonte secundária de luz (FIGURA 5.9). Além disso, caso a fonte secundária seja uma fonte virtual, isto é, uma fonte luminosa localizada antes do segundo vértice da lente, um raio de sombra que esteja contido na área iluminada por esta fonte virtual poderá interseccionar a própria esfera transparente  $i$ . E, em se tratando de uma fonte secundária não-cheia, o raio de sombra intersecciona a esfera transparente  $i$ , sendo então refratado dentro da esfera e depois da esfera para o meio. Daí, surge a necessidade de armazenar tanto um indexador para a esfera transparente original quanto um indexador para a fonte luminosa puntiforme original.

A intensidade ou cor da fonte secundária de luz  $L_{ji}$  determina as componentes de reflexão especular e/ou difusa para a fonte secundária em questão.

A lista de sombras armazena um subconjunto de indexadores para objetos na lista de objetos da cena. Os índices armazenados informam quais os objetos que estão obstruindo parcialmente a passagem de luz da fonte primária para a esfera transparente. Se a fonte secundária é cheia, a lista de sombras está vazia. Ao traçar um raio de sombra para uma fonte secundária cheia é desnecessário refratá-lo, bastando verificar se ele pertence ao feixe de raios luminosos emitidos pela fonte secundária para concluir que o ponto está recebendo luz da fonte secundária. Se a fonte luminosa secundária é não-cheia, a lista de sombras não está vazia. Neste caso, além de verificar se o raio de sombra pertence ao feixe luminoso emitido pela fonte secundária, é necessário refratá-lo dentro da esfera transparente e depois da esfera transparente para o meio (raio refratado final), verificando se o raio atinge algum objeto indexado pela lista de sombras. O ponto está recebendo luz da fonte secundária não-cheia quando o raio refratado final não atinge qualquer objeto indexado na lista de sombras.

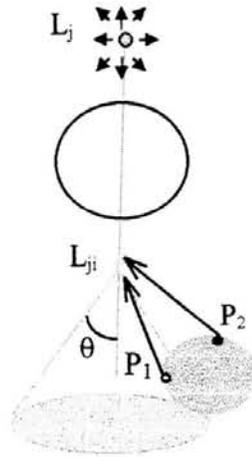


FIGURA 5.9- Exemplo da influência de uma fonte secundária de luz na iluminação da cena. O ponto  $P_1$  é iluminado por  $L_{ji}$  enquanto o ponto  $P_2$  está na sombra de  $L_{ji}$ .  $L_{ji}$  é uma fonte secundária real e cheia, originada pela fonte luminosa  $L_j$  e a esfera transparente  $i$  e com ângulo de espalhamento  $\theta$ .

Após esta descrição dos componentes de uma fonte secundária de luz, procede-se à obtenção destes elementos.

Para o cálculo da posição da fonte luminosa secundária considera-se a FIGURA 2.10, que ilustra os componentes básicos de uma lente esférica convergente.

Na FIGURA 5.10 os principais elementos da FIGURA 2.10 são facilmente identificados e calculados. O eixo central da lente é uma reta que passa por  $L_j$  (posição da fonte luminosa primária) e  $C_E$  (centro da esfera transparente). O objeto colocado sobre o eixo central da lente é a fonte luminosa  $L_j$ . A imagem de  $L_j$  localiza-se em  $I$ , ao longo do eixo central. Os vértices da lente,  $V_1$  e  $V_2$ , são determinados como

$$V_1 = C_E - R_1 \frac{\overrightarrow{L_j C_E}}{|\overrightarrow{L_j C_E}|} \text{ e}$$

$$V_2 = C_E - R_2 \frac{\overrightarrow{L_j C_E}}{|\overrightarrow{L_j C_E}|}.$$

Os raios das faces da lente ( $R_1$  e  $R_2$ ) têm mesmo módulo ( $R_E$ ), uma vez que a lente é uma esfera transparente. A convenção de sinais para lentes considera positiva a distância medida da esquerda para a direita e negativa a distância medida no sentido inverso. Assim, como  $R_1$  é a distância medida de  $V_1$  a  $C_E$  (esquerda→direita),  $R_1$  é igual a  $+R_E$ . Analogamente, como  $R_2$  é a distância medida de  $V_2$  a  $C_E$  (direita→esquerda),  $R_2$  é igual a  $-R_E$ . A espessura da lente ( $\epsilon$ ) é a distância entre os vértices da lente. Como trata-se de uma esfera de raio de curvatura  $R_E$ , a espessura da lente é igual ao diâmetro da esfera, ou seja,  $2R_E$ .

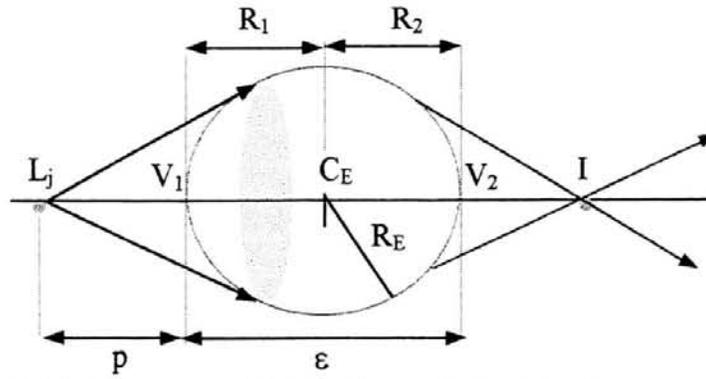


FIGURA 5.10- Fonte secundária real e cheia, localizada no ponto  $I$

Considerando a óptica gaussiana e a aproximação dos raios centrais pode-se localizar o ponto imagem  $I$  ao longo do eixo central a partir da equação (2.11), que trata uma lente espessa como um sistema de duas lentes. Esta equação, conhecida como fórmula dos pontos conjugados, determina a distância da imagem ao segundo vértice da lente ( $i$ ) como uma função da distância do objeto ao primeiro vértice da lente ( $p$ ), da espessura da lente ( $\varepsilon$ ), dos raios de curvatura da lente ( $R_1, R_2$ ), do índice de refração absoluto da lente ( $n$ ) e dos índices de refração do meio onde ela está imersa ( $n', n''$ ). Com base nas equações (2.11) e (2.13), o procedimento para determinar o ponto imagem  $I$  é:

- 1) Calcula-se a distância  $p$  da fonte de luz primária ao primeiro vértice da lente. Na FIGURA 5.10 tem-se que  $p = \text{dist}(L_j, V_1)$ , isto é

$$p = \left| \overrightarrow{L_j C_E} \right| - R_E;$$

- 2) calcula-se o termo  $a = n - 1$ , que será utilizado para simplificar/otimizar os cálculos efetuados nos passos 3, 4 e 5;
- 3) calcula-se a distância  $p_1$ , que representa a distância do “ponto imagem intermediário” ao segundo vértice da lente, considerando-se uma lente limitada à esquerda pela primeira face e ilimitada à direita. A distância  $p_1$  é calculada a partir da primeira igualdade do sistema de equações de (2.11), a saber,

$$\frac{1}{p} + \frac{n}{p_1 + \varepsilon} = \frac{n-1}{R_E}.$$

Após manipulações algébricas chega-se a

$$p_1 = \frac{R_E p n}{p a - R_E} - \varepsilon,$$

desde que  $(p a - R_E) \neq 0$ ;

- 4) calcula-se a distância  $i$  do ponto imagem ao segundo vértice da lente através da segunda igualdade da equação (2.11). Logo,

$$i = \frac{R_E p_1}{n p_1 - p_1 + n R_E}. \quad (5.8)$$

Colocando  $p_1$  no denominador em evidência, a equação é reescrita como

$$i = \frac{R_E p_1}{p_1 (n-1) + n R_E},$$

ou ainda com base no passo 2 acima,

$$i = \frac{R_E p_1}{p_1 a + n R_E};$$

5) calcula-se a distância focal da lente com base na equação (2.13). Assim,

$$f = \frac{R_E p_f}{p_f a + n R_E},$$

onde  $p_f$ , que equivale a  $p_l$ , é calculada por  $-\varepsilon + \frac{n R_E}{a}$ , desde que  $a = (n-1) \neq 0$ ;

6) efetua-se o teste descrito no final da seção 2.4, que relaciona a distância  $p$  do objeto ao primeiro vértice com a distância focal  $f$ . Se  $p=f$  então o ponto-imagem  $I$  está no infinito pois os raios emergem paralelos ao eixo central da lente. Neste caso, utiliza-se como posição da fonte de luz secundária  $I$  o vértice  $V_2$  da lente. Se  $p>f$  a fonte secundária é real e se  $p<f$  a fonte secundária é uma fonte virtual (FIGURA 5.2). Se a fonte for virtual, a distância  $i$  calculada no passo 4 será negativa. Em se tratando de uma fonte secundária real, o valor de  $i$  será positivo. Portanto, em ambos os casos, a localização da fonte secundária é dada por

$$I = V_2 + i \frac{\overrightarrow{L_j C_E}}{|\overrightarrow{L_j C_E}|},$$

onde

$$V_2 = C_E + R_E \frac{\overrightarrow{L_j C_E}}{|\overrightarrow{L_j C_E}|}.$$

Resumindo, tem-se que

$$I = \begin{cases} V_2 + i \frac{\overrightarrow{L_j C_E}}{|\overrightarrow{L_j C_E}|}, & p \neq f \\ V_2, & \text{caso contrário} \end{cases}.$$

Considere-se primeiramente as fontes secundárias reais para o cálculo do ângulo de espalhamento. A FIGURA 5.11 ilustra os elementos utilizados no cálculo do ângulo de espalhamento de uma fonte luminosa secundária real e cheia. Nesta figura,  $L_j$  é a posição da fonte de luz primária,  $C_E$  é o centro da esfera transparente e  $R_E$  o raio da esfera. O ponto  $T$  é um ponto da superfície da esfera tangenciado pelo vetor  $\vec{R}_T$  que parte de  $L_j$ . O vetor  $TR$  é o raio refratado dentro da esfera correspondente ao raio incidente  $\vec{R}_T$ . O ponto  $I$  é o ponto-imagem, localizado a uma distância  $i$  da esfera. O ângulo de espalhamento da fonte secundária de luz é o ângulo  $\theta$ .

Do triângulo  $\Delta DIR$  tem-se que

$$\operatorname{tg}(\theta) = \frac{\overline{RD}}{\overline{DI}} = \frac{d}{i + DV_2}. \quad (5.9)$$

Para obter  $d$  considere-se o triângulo  $\Delta C_E RD$ . Das relações trigonométricas neste triângulo pode-se escrever que

$$d = R_E \operatorname{sen}(\phi). \quad (5.10)$$

Assim, para determinar  $d$  primeiramente deve-se obter o ângulo  $\phi$ . Como o vetor  $\vec{TR}$  é secante à esfera, pode-se determinar um segmento  $C_E M$  que divide o segmento  $TR$  ao meio, ou seja,  $M$  é o ponto médio do segmento. Os triângulos  $\Delta TMC_E$  e  $\Delta RMC_E$  têm o lado  $C_E M$  em comum e os outros lados congruentes ( $TM=MR$  e  $TC_E=RC_E$ ) sendo portanto semelhantes. Desta forma, o ângulo  $\delta$  de

refração em  $T$  é igual ao ângulo de incidência em  $R$ . Logo, o ângulo formado no ponto  $C_E$  pelo triângulo  $\Delta TRC_E$  é igual a  $180^\circ - 2\delta$ . O ângulo  $\phi$ , por sua vez, é obtido como

$$\begin{aligned}\phi &= 180^\circ - (90^\circ - \alpha) - (180^\circ - 2\delta) \\ \phi &= 2\delta + \alpha - 90^\circ,\end{aligned}\quad (5.11)$$

onde  $\alpha$  é o ângulo radial da esfera transparente, calculado pela equação (5.6).

O ângulo  $\delta$  utilizado na equação (5.11) para o cálculo de  $\phi$  é o ângulo de refração correspondente a um ângulo de incidência de  $90^\circ$ . Assim, o ângulo  $\delta$  é obtido através da segunda lei de Snell-Descartes (lei dos senos) como

$$\delta = \text{sen}^{-1}\left(\frac{1}{n}\right).$$

Substituindo a equação acima na equação (5.11) obtém-se o ângulo  $\phi$  e por consequência  $d$ . Neste ponto para solucionar a equação (5.9) basta obter  $\overline{DV_2}$ . O segmento  $\overline{DV_2}$  é igual a  $R_E - \overline{C_E D}$ . Do  $\Delta C_E R D$  sabe-se que  $\overline{C_E D}$  é  $R_E \cos(\phi)$ . Portanto,

$$\overline{DV_2} = R_E - R_E \cos(\phi). \quad (5.12)$$

Como  $\theta$  sempre é diferente de  $90^\circ$ , pode-se escrever que

$$\cos(\theta) = \frac{1}{\sqrt{1 + \tan^2(\theta)}}. \quad (5.13)$$

Portanto, o co-seno do ângulo de espalhamento é

$$\cos(\theta) = \frac{1}{\sqrt{1 + \left(\frac{d}{i + R_E - R_E \cos(\phi)}\right)^2}} \quad (5.14)$$

onde  $i$  foi obtido anteriormente pela equação (5.8).

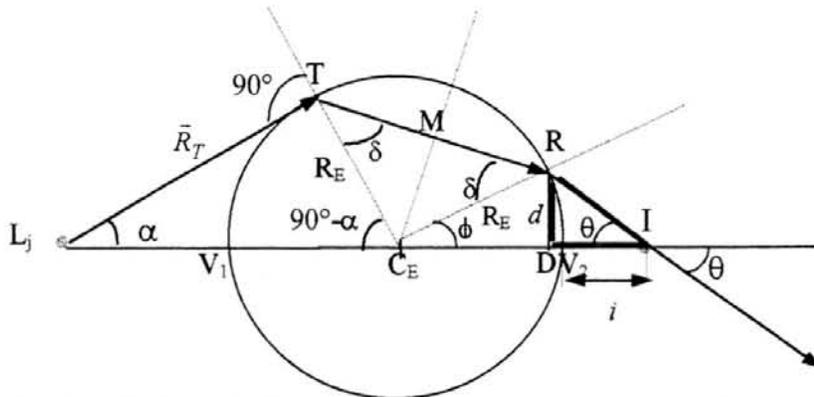


FIGURA 5.11- Cálculo do ângulo de espalhamento para uma fonte secundária real localizada em  $I$

Considere-se agora a obtenção do co-seno do ângulo de espalhamento para fontes secundárias virtuais. A FIGURA 5.12 ilustra esta situação. Nesta figura  $\theta$  é o ângulo de espalhamento da fonte secundária virtual localizada no ponto  $I$  ao longo do eixo central da lente. O ponto  $T$  é o ponto de tangência e  $R$  o ponto onde o raio refratado emerge da esfera para o meio. Das relações trigonométricas no  $\Delta I R D$  tem-se que

$$\text{sen}(\theta) = \frac{\overline{RD}}{\overline{IR}}. \quad (5.15)$$

Das relações trigonométricas no  $\Delta C_E R D$  obtém-se



- Intensidade da fonte luminosa puntiforme que a originou; e
- coeficiente de transmissão e diâmetro da esfera transparente que a originou.

A intensidade da fonte luminosa secundária é calculada da mesma maneira independente de se tratar de uma fonte secundária cheia ou não.

Seja  $I_j = (I_{jR}, I_{jB}, I_{jC})$  a intensidade da fonte de luz puntiforme  $L_j$  que originou a fonte luminosa secundária  $L_{ji}$ . Assume-se que o meio entre a fonte de luz  $L_j$  e a superfície da esfera é um meio homogêneo e com um índice de refração absoluto igual a 1 ( $n_{ar}$ ).

Segundo Lee e Uselton [LEE 85], o fenômeno de absorção ocorre quando as ondas de luz perdem energia ao atravessar um meio e a quantidade de energia absorvida aumenta com o aumento da distância percorrida. A quantidade de energia absorvida também depende do comprimento de onda da luz. Neste trabalho, entretanto, considera-se que a mesma quantidade de energia é absorvida para qualquer dos componentes RGB (diferentes comprimentos de onda).

Pode-se derivar a fórmula básica de absorção da seguinte forma.

Para uma espessura infinitamente pequena,  $dx$ , a fração de intensidade absorvida é proporcional a  $dx$  pela equação

$$\frac{dI}{I} = -A dx,$$

onde  $\frac{dI}{I}$  é a fração de intensidade absorvida na espessura  $dx$  e o fator  $A$  (absorção) é a fração de intensidade absorvida por unidade de espessura. Integrando ambos os lados da equação, a intensidade final para a distância  $x$  percorrida é obtida por

$$\int_{I_0}^{I_x} \frac{dI}{I} = - \int_0^x A dx.$$

Resolvendo a integral definida no lado esquerdo da igualdade tem-se que

$$\ln(I) \Big|_{I_0}^{I_x} = \ln(I_x) - \ln(I_0) = \ln\left(\frac{I_x}{I_0}\right).$$

Já a integral do lado direito da igualdade é

$$-Ax \Big|_0^x = -Ax.$$

Substituindo os resultados tem-se que

$$\ln\left(\frac{I_x}{I_0}\right) = -Ax, \text{ de onde tem-se } e^{-Ax} = \frac{I_x}{I_0}$$

ou ainda,

$$I_x = I_0 \frac{1}{e^{Ax}},$$

fórmula conhecida como lei lambertiana de absorção. Se  $k_t$  é o coeficiente de transmissão do meio então o coeficiente  $A$  de absorção do meio é  $1-k_t$ . Assim, a intensidade final após a luz percorrer uma distância  $x$  é dada por

$$I_x = I_0 e^{-(1-k_t)x}.$$

Para o caso do meio de transmissão ser igual ao ar ( $k_t = 1$ ), a atenuação somente será significativa quando a distância percorrida for consideravelmente grande. Neste trabalho considerou-se a atenuação pela distância somente dentro dos objetos transparentes. Considerando-se a atenuação sofrida pela luz proveniente da fonte primária  $L_j$ , a intensidade da fonte secundária  $L_{ji}$  é aproximada por

$$I_{Lji} = \frac{I_j}{e^{(1-k_i)\left(\frac{2R_E + \overline{TR}}{2}\right)}}, \quad (5.22)$$

onde

$I_j$  é a intensidade da fonte luminosa primária  $L_j$ ;

$1-k_i$  é o coeficiente de absorção da esfera transparente  $i$ ;

$2R_E$  é o diâmetro da esfera transparente  $i$ ; e

$\overline{TR}$  é a distância percorrida pelo raio dentro da esfera quando o raio incidente tangencia a esfera (vide figuras 5.11 e 5.12). Esta distância é calculada a partir da lei dos co-senos por

$$\overline{TR} = \sqrt{2R_E^2 - 2R_E^2 \cos(180^\circ - 2\delta)}.$$

Observe-se que o termo  $\frac{2R_E + \overline{TR}}{2}$  da equação (5.22) representa um valor médio de todas as distâncias percorridas por raios dentro da esfera. É uma aproximação dada pela média aritmética entre a maior e a menor distâncias percorridas pelos raios dentro da esfera. Note-se que desconsiderou-se a absorção sofrida no ar.

E, por fim, a lista de sombras da fonte secundária transparente é obtida da seguinte forma. Quando uma esfera  $k$  satisfaz a ambos os testes de visibilidade ( $1^\circ$  e  $2^\circ$ ) e não satisfaz o critério de obstrução total, os quais são apresentados na seção 5.5, a esfera  $k$  é inserida na lista de sombras da fonte secundária não-cheia que está sendo gerada.

Como uma fonte secundária cilíndrica tem um ângulo de espalhamento igual a  $0^\circ$ , a representação de sua propagação cilíndrica é feita através do raio de base do cilindro transmissor de luz. Utiliza-se a informação do raio do cilindro transmissor de luz no teste de pertinência, o qual verifica se o raio incidente proveniente de algum outro objeto da cena recebe luz da fonte secundária cilíndrica.

A FIGURA 5.13 ilustra os elementos necessários ao cálculo do raio do cilindro luminoso de uma fonte secundária cilíndrica.  $L_j$  é a fonte primária original localizada no foco da esfera e  $R_C$  é o raio do cilindro luminoso da fonte secundária cilíndrica. Observe-se que  $R_C$  é igual a  $d$  na FIGURA 5.11 e portanto  $R_C$  é calculado pela equação (5.10).

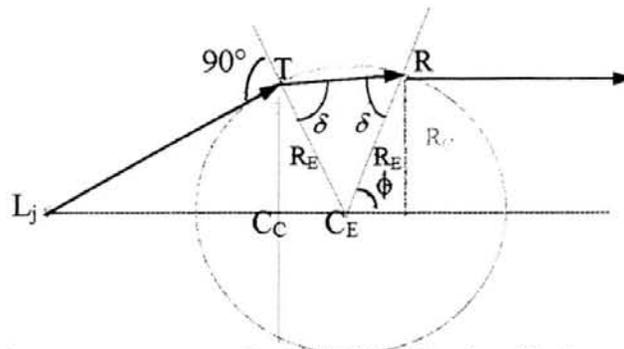


FIGURA 5.13- Elementos para o cálculo do raio  $R_C$  do cilindro transmissor de luz

## 5.7 Estrutura do Algoritmo de Traçado Bidirecional de Raios Proposto

O algoritmo proposto é composto de duas etapas. A primeira etapa produz informação sobre a luz que provém das fontes de luz presentes na cena (fontes primárias) e incide diretamente sobre os objetos transparentes. Esta informação é utilizada na segunda etapa do algoritmo, quando os raios são disparados da posição do observador para a cena.

As duas etapas do algoritmo (*fase 1* e *fase 2*) são descritas em detalhes nas próximas seções.

### 5.7.1 Fase 1 do Traçado Bidirecional

A primeira etapa do algoritmo produz informação sobre a luz que provém das fontes luminosas puntiformes e incide diretamente sobre as esferas transparentes.

Uma proposta de traçado bidirecional de raios é descrita por Watt em [WAT 90]. O trabalho de Watt insere o mecanismo de transmissão da luz de superfícies especulares para superfícies difusas em imagens sintetizadas por rastreamento de raios. É uma proposta para ambientes poligonais. O algoritmo de Watt utiliza polígonos de detalhe, chamados de polígonos cáusticos. Estes polígonos são determinados pelo traçado de raios das fontes de luz para os objetos especulares. Se a reflexão destes raios atinge uma superfície difusa, gera-se um polígono de detalhe através da projeção do “polígono refletido” na superfície difusa.

Outro trabalho de rastreamento bidirecional de raios está descrito em [KIM 92]. O trabalho de Kim aborda a geração das sombras de objetos transparentes. A proposta de Kim baseia-se na utilização de duas matrizes  $T$  e  $Z$ , cada uma com o mesmo número de elementos da tela do computador. A matriz  $Z$  armazena a profundidade dos pontos e a matriz  $T$ , a intensidade das sombras dos objetos transparentes. O principal problema do rastreamento bidirecional de Kim é a grande quantidade de memória necessária. Além disso, a contribuição da luz refratada para raios provenientes de outros pontos da cena que atinjam a “sombra” de um objeto transparente não é considerada.

O modelo apresentado nesta dissertação difere dos trabalhos descritos acima. Trata-se de um modelo que tira proveito da convergência da luz que atravessa uma esfera transparente, proveniente de uma fonte luminosa puntiforme da cena.

Considerando as esferas transparentes como lentes, o modelo cria uma fonte luminosa secundária para cada fonte luminosa puntiforme da cena (primária) que esteja iluminando uma esfera transparente. Se a fonte luminosa primária ilumina parcialmente a esfera transparente, a fonte secundária gerada é não-cheia. Caso a fonte luminosa primária ilumine completamente a esfera transparente, a fonte secundária gerada é cheia. A FIGURA 5.5 mostra a diferença entre uma fonte secundária cheia e uma fonte secundária não-cheia.

As fontes secundárias têm uma área de atuação limitada (cônica ou cilíndrica) e não fazem parte do arquivo descritor da cena. A informação destas “novas fontes luminosas” é utilizada para o cálculo das componentes de iluminação local, durante a segunda etapa do algoritmo.

Quando a fonte luminosa primária localiza-se no ponto focal de uma lente esférica, os raios de luz emergem paralelos ao eixo central da lente, ou seja, não há espalhamento da luz. Esta situação pode ser vislumbrada como um círculo luminoso, com centro no segundo vértice da lente e raio igual ao do cilindro luminoso, cujos pontos emitem raios de luz na direção e sentido do vetor eixo central da lente.

Com exceção da situação acima, as fontes secundárias, quer reais ou virtuais, têm uma área de atuação limitada por um cone. Optou-se neste trabalho por restringir as fontes luminosas obtidas na primeira etapa do algoritmo a fontes secundárias, ou seja, fontes luminosas com sua origem nos raios que partem de uma fonte luminosa primária da cena e incidem numa esfera transparente. Fontes luminosas terciárias, quaternárias, etc., podem ser tema para um trabalho futuro fora do escopo deste.

Pode-se descrever o algoritmo da primeira etapa do modelo proposto como segue. Os trechos em itálico delimitados por /\* \*/ são comentários.

1. **Para** cada fonte luminosa puntiforme  $L_j$  descrita no arquivo de entrada **faça**:
  - 1.1. Ordene as  $n$  esferas da cena em ordem crescente da distância a  $L_j$ . */\*Isto é feito mantendo-se um vetor  $buffer\_luz[0..n-1]$ , onde cada elemento do vetor contém o indexador do objeto esfera correspondente e a menor distância da esfera a  $L_j$ .\*/*
  - 1.2. **Para** cada esfera transparente  $i$  de  $buffer\_luz[0..n-1]$  **faça**:
    - 1.2.1. Calcule a distância  $p$  da fonte puntiforme  $L_j$  ao primeiro vértice da lente, ou seja,
 
$$p := |L_j - C_i| - R_i,$$
 onde  $C_i$  e  $R_i$  são, respectivamente, o centro e o raio da esfera transparente  $i$ .
    - 1.2.2. Calcule a distância  $f$  do ponto focal ao primeiro vértice da lente (esfera transparente  $i$ ). A variável  $f$  é calculada pela equação (2.13), ou seja,
 
$$f = \frac{R_i p_1}{p_1(n-1) + nR_i},$$
 onde  $R_i$  é o raio da esfera transparente  $i$ ,  $n$  é seu índice de refração absoluto e  $p_1$  é calculado por
 
$$p_1 = -e + \frac{nR_i}{(n-1)},$$
 onde a espessura  $e$  é igual ao diâmetro da esfera transparente  $i$ .
    - 1.2.3. **Se** ( $p = f$ ) */\*os raios emergem paralelos ao eixo central da lente, portanto o ângulo de espalhamento da fonte luminosa secundária é igual a  $0^\circ$ . Fonte secundária, se existir, será cilíndrica.\*/* **então**:
      - 1.2.3.1. Co-seno do ângulo de espalhamento  $\theta$  recebe  $\cos(0^\circ) = 1$ .
      - 1.2.3.2. Calcule o raio do cilindro luminoso  $R_C$ .
    - 1.2.4. **Se** ( $p > f$ ) **então**: */\*{ fonte secundária, se existir, será real}\*/*
      - 1.2.4.1. Calcule o co-seno do ângulo de espalhamento  $\theta$ ,
 
$$\cos(\theta) = \frac{1}{\sqrt{1 + \left( \frac{d}{i + R_E - R_E \cos(\phi)} \right)^2}},$$
 conforme a equação (5.14).
    - 1.2.5. **Se** ( $p < f$ ) **então**: */\*{ fonte secundária, se existir, será virtual}\*/*
      - 1.2.5.1. Calcule o co-seno do ângulo de espalhamento  $\theta$ ,
 
$$\cos(\theta) = \frac{1}{\sqrt{1 + \left( \frac{R_E \sin(\sigma)}{i - R_E - R_E \cos(\sigma)} \right)^2}},$$
 conforme a equação (5.21).
    - 1.2.6.  $lista\_sombras := \emptyset$ .
    - 1.2.7. **Para** cada esfera  $k$  em  $buffer\_luz[0..n-1]$  que esteja mais próxima de  $L_j$  que a "calota visível da esfera transparente  $i$ " */\*esfera  $k$  satisfaz o  $1^\circ$  teste de visibilidade apresentado na seção 5.5\*/* **faça**:

*/\*2º teste de visibilidade da seção 5.5\*/*

1.2.7.1. Calcule o ângulo radial da esfera transparente  $i$ , visto de  $L_j$ , ou seja,

$$\alpha := \cos^{-1} \left( \frac{R_{T_i}}{|C_i - L_j|} \right), \text{ de acordo a equação (5.6).}$$

1.2.7.2. Calcule o o ângulo radial da esfera  $k$ , visto de  $L_j$ , ou seja,

$$\beta := \cos^{-1} \left( \frac{R_{T_k}}{|C_k - L_j|} \right), \text{ com base na equação (5.6).}$$

1.2.7.3. Calcule a distância angular do centro da esfera  $i$  ao centro da esfera  $k$ , vista de  $L_j$ , ou seja,

$$\gamma := \cos^{-1} \left( \frac{(C_k - L_j) \cdot (C_i - L_j)}{|C_k - L_j| |C_i - L_j|} \right), \text{ de acordo com a equação (5.7).}$$

1.2.7.4. **Se** (Não( $\alpha + \beta \leq \gamma$ )) **então** */\*( $\alpha + \beta > \gamma$ )  $\Rightarrow$  a esfera  $k$  não obstrui a luz que parte de  $L_j$  e atinge a esfera  $i$ .\*/*

1.2.7.4.1. Próximo  $k$ .

1.2.7.4.2. Vá para 1.2.7.

1.2.7.5. **Senão**

1.2.7.5.1. **Se** (a esfera  $k$  obstrui completamente a esfera  $i$ ) **então**:

1.2.7.5.1.1. Próximo  $i$ . */\*Não criar  $L_{ji}$ \*/*

1.2.7.5.1.2. Vá para 1.2.

1.2.7.5.2. **Senão**

1.2.7.5.2.1. Inserir  $k$  na lista de sombras de  $L_{ji}$ , ou seja,  $lista\_sombras := lista\_sombras \cup \{k\}$ .

1.2.7.5.2.2. Próximo  $k$ .

1.2.7.5.2.3. Vá para 1.2.7.

1.2.7.6. **Se**  $i$  é um índice de objeto válido, ou seja, ( $i \leq n-1$ ) **então**:

1.2.7.6.1. Calcule o ponto imagem  $L_{ji}$  ao longo do eixo central da esfera  $i$ , tendo a fonte luminosa  $L_j$  como objeto. O eixo central é a reta que passa pelos pontos  $L_j$  e centro  $C_i$  da esfera transparente  $i$ . O cálculo do ponto  $L_{ji}$  é feito conforme exposto na seção 5.6.

1.2.7.6.2. Crie a fonte secundária de luz posicionada em  $L_{ji}$ , com co-seno do ângulo de espalhamento igual a  $\cos(\theta)$ , intensidade calculada pela equação (5.22), índice do objeto de origem igual a  $i$ , índice da fonte primária original igual a  $j$  e lista de objetos que obstruem a passagem da luz de  $L_j$  à esfera  $i$  igual a  $lista\_sombras$ .

## 5.7.2 Fase 2 do Traçado Bidirecional

A segunda fase do algoritmo proposto traça os raios a partir do observador para a cena. Disparam-se raios para pontos descritos no sistema de referência do universo de modelagem. Estes pontos são determinados através de uma janela imaginária. Faz-se uma correspondência entre os pontos na janela imaginária e os

pontos (*pixels*) na janela de exibição. A seção 3.2.1 descreve a filosofia e a estrutura do algoritmo de traçado de raios a partir do ponto de observação.

Pode-se descrever o algoritmo referente à segunda etapa do modelo proposto como segue. Os trechos em itálico delimitados por */\*\** são comentários.

1. **Para** cada linha de varredura da janela de exibição **faça**:
  - 1.1. **Para** cada pixel na linha de varredura corrente **faça**:
    - 1.1.1. Determine o vetor que parte do observador para o centro do pixel (*raio*).
    - 1.1.2. *profundidade := 1.*
    - 1.1.3. *Objeto\_fonte := -1. /\*raio parte do observador\*/*
    - 1.1.4. *origem := observador. /\*ponto de origem do raio\*/*
    - 1.1.5. *Cor\_pixel := RTtrace(objeto\_fonte, origem, raio, profundidade).*

A função *RTtrace(...)* chamada no passo 1.1.5. retorna a intensidade (cor) do *pixel*. Ela encontra a intersecção mais próxima do vetor *raio* com os objetos da cena. A seqüência de passos desta função é:

    1. **Se** (*profundidade ≤ MÁXIMA\_RECURSÃO*) **então**: */\*MÁXIMA\_RECURSÃO é uma constante definida no programa que limita o número de vezes que um raio é refletido ou refratado recursivamente.\*/*
      - 1.1. *distância\_mínima := MÁXIMO. /\*MÁXIMO é constante igual a idealmente infinito\*/*
      - 1.2. **Para** cada objeto *i* da cena **faça**:
        - 1.2.1. *distância := Calcula\_intersecção(origem, raio, i).*
        - 1.2.2. **Se** (*distância < distância\_mínima*) **então**:
          - 1.2.2.1. *distância\_mínima := distância.*
          - 1.2.2.2. *objeto\_mais\_próximo := i.*
      - 1.3. **Se** (*distância\_mínima < MÁXIMO*) **então**:
        - 1.3.1. *ponto\_intersecção := origem + raio\*distância\_mínima.*
        - 1.3.2. *normal := Calcula\_normal(objeto\_mais\_próximo, ponto\_intersecção).*
        - 1.3.3. *cor := RTshade(objeto\_mais\_próximo, raio, ponto\_intersecção, normal, profundidade).*
        - 1.3.4. Retorna (*cor*).
    2. Retorna(*cor\_fundo*).

A função *RTtrace(...)* descrita acima faz três chamadas de funções, a função *Calcula\_intersecção(...)* no passo 1.2.1, a função *Calcula\_normal(...)* no passo 1.3.2 e a função *Rtshade(...)* no passo 1.3.3.

A função *Calcula\_intersecção(...)* verifica se o *raio* intercepta o objeto *i*, onde ambos são parâmetros da função, e retorna a menor distância da intersecção à origem do raio.

A função *Calcula\_normal(...)* calcula o vetor normal no ponto do objeto mais próximo interseccionado pelo raio (*ponto\_intersecção*). Este vetor normal é utilizado no cálculo do modelo de iluminação em *ponto\_intersecção*.

E a função *RTshade(...)* retorna a cor para o *pixel* atual. *RTshade(...)* é descrita pelos seguintes passos:

1. *objeto\_atual := objeto\_mais\_próximo.*
2. Calcula a *componente ambiental*.
3. *Componente difusa := (0 0 0)<sub>RGB</sub>.*
4. *Componente especular := (0 0 0)<sub>RGB</sub>.*
5. *V := (-1)\*raio.*

6. **Se**  $(V \bullet normal) < 0$  **então**:
- 6.1.  $normal := (-1) * normal$ .
  - 6.2.  $k_n := n^{-1}$ .
  - 6.3.  $Refracção := 1$ .
7. **Senão**
- 7.1.  $k_n := n / *n$  é o índice de refração absoluto do objeto\_atual\*/.
8.  $V' := \frac{V}{|V \bullet normal|}$ .
- /\*cálculo do raio refletido\*/*
9.  $rRaio := V' + 2 * normal$ , conforme a equação (2.8).
- /\*cálculo do raio refratado\*/*
10. **Se** (objeto\_atual é transparente) **então**:
- 10.1.  $k_f := (k_n^2 * |V'|^2) - (|V' + normal|^2)$ , conforme seção 2.3.4.
  - 10.2. **Se**  $(k_f \leq 0)$  **então**: */\*reflexão interna\*/*
    - 10.2.1.  $tRaio := rRaio$ .
  - 10.3. **Senão**
    - 10.3.1.  $tRaio := (V' + normal) * k_f^{-1/2} - normal$ .
- /\*cálculo das componentes difusa e especular\*/*
11. **Para** cada fonte luminosa puntiforme  $j$  definida no arquivo descritor da cena **faça**:
- 11.1. Calcule o raio de sombra  $\vec{L}$  como um vetor que parte de *ponto\_intersecção* para a fonte luminosa  $L_j$ .
  - 11.2.  $s := 1$ ;
  - 11.3. **Para** cada objeto atingido por  $\vec{L}$  no caminho do *ponto\_intersecção* até  $L_j$  **faça**:
    - 11.3.1.  $s := s * \text{coeficiente de transmissão do objeto atingido}$ .
  - 11.4. Calcule *componente difusa*  $j$ .
  - 11.5. Calcule *componente especular*  $j$ .
  - 11.6.  $componente\ difusa := componente\ difusa + s * componente\ difusa\ j$ .
  - 11.7.  $componente\ especular := componente\ especular + s * componente\ especular\ j$ .
12. **Para** cada fonte luminosa secundária  $L_s$  da cena **faça**:
- 12.1. Calcule o raio de sombra  $\vec{S}$  como um vetor que parte de *ponto\_intersecção* para a fonte luminosa secundária  $L_s$ .
  - 12.2. **Se** (vetor  $-\vec{S}$  pertence ao feixe de raios emitidos pela fonte secundária  $L_s$ ) **então**:
    - 12.2.1.  $fator := 1$ .
    - 12.2.2. **Para** cada objeto  $k$  atingido por  $\vec{S}$  no caminho até  $L_s$  **faça**:
      - 12.2.2.1.  $fator := fator * kt_k$ . */\*  $kt_k$  é o coeficiente de transparência da esfera  $k$ \*/*
    - 12.2.3. **Se** (fonte secundária  $L_s$  é cheia) **então**:
      - 12.2.3.1. Calcule *componente difusa* de  $L_s$ .
      - 12.2.3.2. Calcule *componente especular* de  $L_s$ .
      - 12.2.3.3.  $Componente\ difusa := fator * componente\ difusa\ de\ L_s$ .
      - 12.2.3.4.  $Componente\ especular := fator * componente\ especular\ de\ L_s$ .
    - 12.2.4. **Senão** */\*não-cheia\*/*

- 12.2.4.1. Encontre a intersecção do vetor  $\vec{S}$  com a superfície da esfera que originou a fonte secundária  $L_s$ . Ponto  $Q$  na FIGURA 5.14.
- 12.2.4.2. Calcule o vetor refratado dentro da esfera correspondente ao vetor incidente  $\vec{S}$  no ponto  $Q$ , ou seja, vetor refratado  $\vec{S}_{T1}$ .
- 12.2.4.3. Encontre o ponto de emergência do raio  $\vec{S}_{T1}$  ao deixar a esfera. Ponto  $R$  na figura.
- 12.2.4.5. Calcule o vetor refratado da esfera para o meio externo correspondente ao vetor incidente  $\vec{S}_{T1}$  no ponto  $R$  (vetor refratado  $\vec{S}_{T2}$ ).
- 12.2.4.6. **Para** cada objeto  $k$  na lista de sombras da fonte secundária  $L_s$  **faça**:
- 12.2.4.6.1. *condição* :=  $\vec{S}_{T2}$  atinge esfera  $k$ .
- 12.2.4.6.2. **Se** (*condição*) **então**:
- 12.2.4.6.2.1. *Comp. difusa de  $L_s$*  :=  $(0\ 0\ 0)_{RGB}$ .
- 12.2.4.6.2.2. *Comp. especular de  $L_s$*  :=  $(0\ 0\ 0)_{RGB}$ .
- 12.2.4.6.2.3. Próxima  $L_s$ .
- 12.2.4.6.2.4. Vá para 12.1.
- 12.2.4.7. **Se** ((Não *condição*) E ( $L_s$  é uma fonte secundária válida)) **então**:
- 12.2.4.7.1. Calcula a *componente difusa de  $L_s$* .
- 12.2.4.7.2. Calcula a *componente especular de  $L_s$* .
- 12.2.4.7.3. *componente difusa* := *componente difusa* + (*fator*\**componente difusa de  $L_s$* ).
- 12.2.4.7.4. *componente especular* := *componente especular* + (*fator* \* *componente especular de  $L_s$* ).
- /\*reflexão\*/*
13. *rCor* :=  $(0\ 0\ 0)_{RGB}$ .
14. **Se** (*objeto\_atual* é reflexivo) **então**:
- 14.1. *rCor* := RTtrace(*objeto\_atual*, *ponto\_intersecção*, *rRaio*, *profundidade* + 1).
- /\*refração\*/*
15. *tCor* :=  $(0\ 0\ 0)_{RGB}$ .
16. **Se** ((*objeto\_atual* é transparente) E (*refração*=1)) **então**:
- 16.1. *tCor* := RTtrace\_refratado(*objeto\_atual*, *ponto\_intersecção*, *tRaio*, *profundidade*+1).
- /\*cálculo da cor final\*/*
17. *cor* := *componente ambiental* + *componente difusa* + *componente especular* + *coeficiente\_especular*\**rCor* + *coeficiente\_transparência*\**tCor*.
18. Retorna (*cor*).

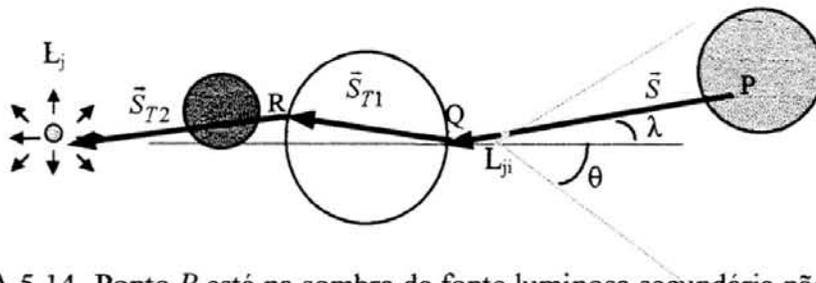


FIGURA 5.14- Ponto  $P$  está na sombra da fonte luminosa secundária não-cheia  $L_{ji}$

O passo 12.2 da rotina *Rtshade(...)* primeiramente verifica se trata-se de uma fonte luminosa secundária cônica ou cilíndrica. As fontes secundárias com ângulo de espalhamento diferente de  $0^\circ$  emitem um cone de luz. Já as fontes secundárias com ângulo de espalhamento igual a  $0^\circ$  emitem raios paralelos ao eixo central da esfera e, neste caso, a fonte secundária equivale a uma fonte cilíndrica de área direcional. A área desta fonte secundária é um círculo com raio igual ao raio do cilindro luminoso da esfera transparente e centro no segundo vértice da lente.

Após identificar o tipo de fonte secundária, faz-se o teste de pertinência angular. O teste de pertinência angular calcula o co-seno do ângulo  $\lambda$  entre os vetores  $\vec{S}$  e *eixo central* (produto escalar dos vetores) e compara o valor encontrado com o co-seno do ângulo  $\theta$  de espalhamento. Se o ângulo de espalhamento for menor que o ângulo entre os vetores então o raio  $-\vec{S}$  não pertence ao conjunto de raios emitidos pela fonte secundária e o processo é encerrado. Caso contrário, há duas situações a considerar. Se a fonte secundária tem espalhamento cônico então conclui-se que  $-\vec{S}$  está sendo emitido pela fonte  $L_s$ . Porém, caso a fonte luminosa secundária tenha um espalhamento cilíndrico é necessário efetuar mais um teste antes de concluir que  $-\vec{S}$  provém da fonte secundária  $L_s$ . Este teste consiste em verificar se a projeção do ponto  $P$  para o qual está sendo feito o cálculo da intensidade no plano do círculo transmissor da fonte  $L_s$  determina um segmento que está completamente contido no círculo de luz. Para isso, compara-se o comprimento do segmento projetado ( $x$ ) com o raio do cilindro luminoso ( $R_C$ ). A medida  $x$  é calculada como  $|\vec{S}|\sqrt{1 - \cos^2(\lambda)}$ , onde  $\lambda$  é ângulo formado pelos vetores  $\vec{S}$  e *eixo central*.

Se  $x \leq R_C$  então  $-\vec{S}$  provém da fonte luminosa secundária  $L_s$ .

### 5.7.3 Modelo de Iluminação

Esta proposta implementa basicamente o modelo de iluminação global de Whitted [WHI 80]. Dado que o objetivo deste trabalho é a síntese de objetos transparentes, o modelo implementado é estendido de forma a incorporar a atenuação sofrida pela luz ao atravessar um meio transparente.

O modelo de iluminação utilizado, baseado no modelo de iluminação de Whitted, é descrito pela equação

$$I = k_a I_a I_{obj} + \sum_{j=1}^{nl} \frac{S_j I_j}{(d_j + k)} \left[ k_d I_{obj} (\vec{N} \cdot \vec{L}_j) + k_s (\vec{R}_j \cdot \vec{V})^{esp} \right] + \sum_{k=1}^{ns} \frac{S_{k1} S_{k2} I_k}{(d_k + k)} \left[ k_d I_{obj} (\vec{N} \cdot \vec{L}_k) + k_s (\vec{R}_k \cdot \vec{V})^{esp} \right] + k_s I_r + k_t I_t \quad (5.23)$$

onde:

$I$  é a cor final do *pixel*;

$k_a$  é o coeficiente de iluminação ambiental;

$I_a$  é a cor da luz ambiente;  
 $I_{obj}$  é a cor do objeto;  
 $nl$  é o número de fontes luminosas puntiformes (primárias) da cena;  
 $ns$  é o número de fontes luminosas secundárias geradas pela primeira etapa do algoritmo (cheias e não-cheias);  
 $k_d$  é o coeficiente de reflexão difusa;  
 $k_s$  é o coeficiente de reflexão especular;  
 $k_t$  é o coeficiente de transparência;  
 $esp$  é o coeficiente de especularidade de Phong;  
 $\vec{N}$  é o versor normal à superfície do objeto no ponto;  
 $\vec{L}_j$  é o vetor sombra, que vai do ponto de intersecção até a j-ésima fonte de luz primária;  
 $\vec{L}_k$  é o vetor sombra, que vai do ponto de intersecção até a k-ésima fonte de luz secundária;  
 $I_j$  é a cor da j-ésima fonte de luz primária;  
 $I_k$  é a cor da k-ésima fonte luminosa secundária;  
 $d_j$  é a distância entre o ponto e a j-ésima fonte de luz primária;  
 $d_k$  é a distância entre o ponto e a k-ésima fonte de luz secundária;  
 $S_j$  é o produto dos coeficientes de transmissão ( $k_t$ ) dos objetos que estão no caminho do ponto atual até a j-ésima fonte de luz. Indica se o ponto está em sombra (0) ou não ( $S_j \neq 0$ ) em relação à j-ésima fonte de luz primária;  
 $S_{k1}$  determina a parcela da luz que chega ao ponto proveniente da fonte luminosa secundária  $k$ . É a variável *fator* da rotina *Rtshade(...)*;  
 $S_{k2}$  é igual a 1 se a fonte secundária  $k$  é cheia. Se a fonte secundária  $k$  é não-cheia,  $S_{k2}$  é igual a 1 se  $\vec{S}_{T2}$  não intercepta qualquer objeto da lista de sombras de  $k$  e é igual a 0 se  $\vec{S}_{T2}$  intercepta algum objeto da lista de sombras de  $k$ ;  
 $k$  é uma constante utilizada na definição do fator de atenuação pela distância. Para maiores detalhes, consultar [FOL 90];  
 $\vec{R}_j$  é o vetor reflexão local, ou seja, é o raio refletido correspondente ao raio incidente proveniente da j-ésima fonte de luz primária;  
 $\vec{R}_k$  é o vetor reflexão local, ou seja, é o raio refletido correspondente ao raio incidente proveniente da k-ésima fonte luminosa secundária;  
 $\vec{V}$  é o raio incidente no ponto. Pode ser proveniente do observador ou de outro ponto na cena;  
 $I_s$  é a intensidade retornada pelo raio refletido (reflexão global); e  
 $I_t$  é a intensidade retornada pelo raio transmitido (refração global).

Observe-se que todos os elementos do tipo cor têm um valor para cada componente da representação RGB.

A componente ambiental do modelo de iluminação ( $k_a I_a I_{obj}$ ) é utilizada para “substituir” os termos globais quando um modelo de iluminação local é aplicado. Uma vez que o protótipo *Bitracer* implementa um modelo de iluminação global, para a maioria das cenas o coeficiente  $k_a$  será nulo e, portanto, o termo ambiental não afetará a intensidade  $I$  final.

## 5.8 Conclusão

Este capítulo apresentou um modelo para simular a refração da luz que provém diretamente das fontes difusas da cena em ambientes esféricos.

O modelo proposto cria fontes secundárias de luz e as incorpora à cena. O processo de geração das fontes secundárias fundamenta-se nos princípios da Óptica Clássica para lentes esféricas convergentes.

O algoritmo proposto é um algoritmo de rastreamento bidirecional de raios, composto de duas etapas. Na primeira etapa, percorre-se a cena no sentido fontes de luz para as esferas transparentes. Quando uma esfera transparente é iluminada por uma fonte luminosa primária, cria-se uma fonte luminosa secundária para esta combinação de fonte luminosa primária e esfera transparente. Cada obstáculo identificado entre a fonte primária e a esfera transparente é incluído na lista de sombras da fonte secundária gerada. Na segunda etapa, a cena é percorrida do observador para os objetos e, a informação das fontes luminosas secundárias é utilizada para complementar o cálculo das componentes locais do modelo de iluminação.

## 6 Implementação do Algoritmo Proposto

### 6.1 Introdução

Com o propósito de analisar a proposta apresentada no capítulo anterior, tanto em termos do realismo das imagens geradas quanto do tempo computacional gasto na sua síntese, implementou-se um protótipo do modelo, denominado "BITRACER".

"BITRACER" foi desenvolvido no sistema operacional MS-DOS, utilizando a linguagem de programação C (ANSI C). Trata-se de um sistema que gera imagens através da aplicação do método de rastreamento de raios bidirecional a arquivos contendo uma descrição da cena a ser sintetizada.

### 6.2 Formatos de Arquivos Utilizados

#### 6.2.1 Arquivos de Entrada

Utiliza-se a extensão "BI" como padrão para os arquivos de entrada do protótipo. No arquivo descritor da cena são especificados:

- Parâmetros de visualização: Dimensões da janela de exibição (*viewport*), posição da câmera, centro de projeção, direção de observação e orientação e abertura da câmera;
- parâmetros de definição das fontes de luz: Intensidade para cada um dos três componentes da representação RGB, brilho e localização;
- parâmetros de definição dos objetos: Tipo, localização, ...; e
- parâmetros de definição das superfícies dos objetos: Intensidade para cada um dos três componentes da representação RGB, coeficientes de iluminação ambiental, difusa e especular, coeficiente de especularidade, coeficientes de reflexão e transparência globais e índice de refração.

A FIGURA 6.1 ilustra o formato de um arquivo de entrada do protótipo. À direita de cada parâmetro tem-se uma explicação de seu significado.

```

V 200 200 /*dimensão da janela de exibição*/
C 15 15 15 /*posição da janela de projeção*/
A 6 6 6/*alvo*/
U 0 1 0/*vetor de orientação da câmera sintética*/
B 20 20/*ângulos de abertura da câmera*/
L 1 /*número de fontes luminosas da cena*/
  /*Para cada fonte de luz tem-se:*/
C 1 1 1/*cor da luz (vermelho,verde,azul)*/
P 15 15 15 /*posição da fonte de luz*/
I 1 /*brilho*/
O 1 /*número de objetos da cena*/
  /*Para cada objeto tem-se:*/
T 0 /*tipo do objeto (esfera = 0 ou poliedro = 1)*/
  C 3 7 1 /*centro da esfera*/
  R 2 /*raio da esfera*/
  S /*definição da superfície do objeto*/
  #C 1 0 0/*cor da superfície (R, G, B)*/
  #A 0.1/*coeficiente da iluminação ambiental (ka)*/
  #D 0.5/*coeficiente de reflexão difusa (kd)*/
  #S 0.4/*coeficiente de reflexão especular (ks)*/
  #T 0 /*coeficiente de transparência (kt)*/
  #E 6 /*coeficiente da distribuição especular de Phong*/
  #N 1 /*índice de refração (n) */

```

FIGURA 6.1- Arquivo de entrada *esfera.bi*

### 6.2.2 Arquivos de Saída

Os arquivos de imagem gerados pelo protótipo são do tipo "TGA". Se o arquivo de saída já existe, ele é substituído pelo novo arquivo imagem independente de confirmação por parte do usuário.

Os arquivos gerados são visualizados em qualquer visualizador gráfico que suporte o formato "TGA".

### 6.3 Funcionamento do Protótipo

O funcionamento do protótipo implementado, "BITRACER", segue o fluxo descrito na FIGURA 6.2.



FIGURA 6.2- Fluxo de funcionamento do protótipo “BITRACER”

O protótipo primeiramente faz a leitura dos dados da cena a partir do arquivo de entrada (formato texto) e procede a inicialização das variáveis globais. Na seqüência, é criado um arquivo imagem com o nome especificado para o arquivo de saída na linha de comando. Um cabeçalho, que identifica o arquivo binário de saída como um arquivo imagem, é gravado no início do arquivo de saída. A seguir, executa-se a rotina *bitracer1*, que implementa a primeira etapa do algoritmo. Esta rotina é responsável pela criação das fontes secundárias de luz. Com o término da função *bitracer1*, a função *bitracer2*, que acompanha os raios disparados do centro de projeção (observador) para a cena, é executada.

Inicialmente, a função *bitracer2* dispara um raio para o centro de cada *pixel* da janela de exibição (*viewport*). Se o raio intersecciona algum objeto da cena, calcula-se o modelo de iluminação para o ponto interseccionado no objeto mais próximo. Senão, a intensidade do *pixel* é dada pela cor de fundo da cena.

O modelo de iluminação é composto de termos locais e globais. Os termos globais referentes a um ponto  $P$  de um objeto atingido pelo raio incidente  $\vec{I}$  são obtidos através do cálculo recursivo do modelo de iluminação para seus raios refletido e transmitido. Os componentes locais representam quanto de luz o ponto atual recebe diretamente das fontes luminosas da cena.

As fontes luminosas secundárias inseridas na primeira fase do programa (*bitracer1*) são tratadas pelo sistema de forma semelhante às fontes primárias. O teste realizado para identificar se o ponto atual está na sombra de uma fonte puntiforme (primária) consiste em verificar se o produto escalar dos vetores normal no ponto  $\vec{N}$  e vetor sombra  $\vec{L}$  (vetor que parte do ponto atual para a fonte luminosa) é maior que zero, ou seja, se o ângulo entre eles é menor que  $90^\circ$ . Em se tratando de uma fonte luminosa secundária, o teste do produto escalar  $\vec{N} \cdot \vec{L}$  não é suficiente, sendo necessário um teste adicional para verificar se o raio de sombra  $\vec{L}$  está contido no feixe de raios emitidos pela fonte secundária. Este teste é descrito como

$$\vec{S} \cdot \vec{V}_{eixo} \leq \cos\_espalhamento,$$

onde,

$$\vec{S} = -\vec{L},$$

$\vec{V}_{eixo}$  é o vetor eixo da lente esférica transparente que originou a fonte secundária (parte do centro da esfera transparente para a fonte secundária) e

$\cos\_espalhamento$  é o co-seno do ângulo de abertura/espalhamento da fonte secundária.

Caso a fonte secundária esteja emitindo os raios paralelos a  $\vec{V}_{eixo}$ , tem-se uma fonte secundária com área igual à área do círculo de base do cilindro transmissor de luz e direção dada por  $\vec{V}_{eixo}$ . Observe-se a FIGURA 6.3. Nesta situação, primeiramente obtém-se  $R_C$  (raio do cilindro luminoso). A seguir, calcula-se a medida  $d$  da projeção do ponto atual  $P$ , no plano do cilindro transmissor de luz. O ponto atual  $P$  é o ponto para o qual está sendo feito o cálculo do modelo de iluminação. Se  $d \leq R_C$ , um vetor  $\vec{X}$  auxiliar é definido como  $\vec{X} = -\vec{V}_{eixo} + \vec{L}$ . Calcula-se o ponto  $P'$ , projeção de  $P$  no plano do círculo transmissor de luz, como  $L_{ji} + \vec{X}$ , onde  $L_{ji}$  é a posição da fonte secundária cilíndrica. Dessa forma, o vetor raio de sombra é calculado como a diferença entre a projeção  $P'$  e o ponto atual  $P$ .

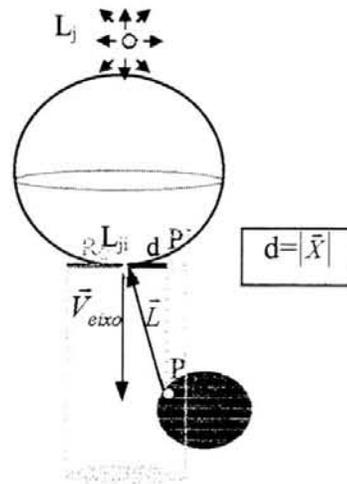


FIGURA 6.3- Cálculo do raio de sombra para uma fonte secundária cilíndrica

## 6.4 Estruturas de Dados

As principais estruturas de memória utilizadas para o armazenamento das informações referentes à primeira e segunda etapas do algoritmo proposto são apresentadas a seguir.

```

BUFFER_LUZ : Vetor [numero_objetos_cena] de
                Estrutura TIPO_BUFFER_LUZ {
                                distancia: real;
                                indice: inteiro;
                                }.

```

Utiliza-se a variável de memória BUFFER\_LUZ para ordenar os objetos da cena em ordem ascendente da distância do objeto à fonte de luz puntiforme corrente. Todos os objetos da cena têm uma entrada em BUFFER\_LUZ. O processo de criação das fontes luminosas secundárias consiste em um laço repetitivo que, a cada iteração, determina todas as fontes derivadas da mesma fonte luminosa primária. Desta forma, uma única variável de memória (BUFFER\_LUZ) é utilizada em todas as iterações. Cada elemento de BUFFER\_LUZ é composto de duas informações, a saber, a menor distância entre a fonte de luz primária e a superfície do objeto (distância) e a posição do objeto dentro do vetor de objetos (indice). O uso da variável BUFFER\_LUZ otimiza a

execução do primeiro teste de visibilidade apresentado na seção 5.5. A otimização proporcionada pela variável `BUFFER_LUZ` se dá da seguinte forma. O primeiro teste de visibilidade verifica se o objeto atual está a uma distância da fonte primária menor que a distância da fonte primária à calota visível da esfera transparente. Percorrendo o vetor `BUFFER_LUZ` da esquerda para a direita, a partir do momento em que um objeto deixa de satisfazer o primeiro teste de visibilidade, torna-se desnecessário testar os objetos seguintes no vetor `BUFFER_LUZ`.

```
SECUNDARIAS: ^ Estrutura TIPO_LUZ_SECUNDARIA {
    posicao: PONTO3D;
    intensidade: real;
    cos espalhamento: real;
    objeto origem: inteiro;
    fonte origem: inteiro;
    lista sombras: ^OBSTACULO;
    prox: ^TIPO_LUZ_SECUNDARIA;
}.
```

A variável de memória `SECUNDARIAS` é um apontador para a primeira fonte secundária da lista de fontes de luz geradas pela primeira etapa do algoritmo *Bitracer*. Ao determinar uma nova fonte secundária, aloca-se memória para outra estrutura de `TIPO_LUZ_SECUNDARIA` e o campo `prox` da última fonte secundária armazenada na lista passa a apontar para o endereço de memória da estrutura criada. Os seguintes elementos são armazenados para cada fonte secundária:

- Localização da fonte secundária no SRU (posição);
- brilho da fonte secundária (intensidade);
- co-seno do ângulo de espalhamento da fonte secundária;
- posição da esfera transparente, que originou a fonte secundária, dentro do vetor de objetos da cena (objeto origem);
- índice da fonte de luz primária, que originou a fonte secundária, dentro do vetor de fontes de luz da cena (fonte origem); e
- um subconjunto de indexadores de objetos dentro do vetor de objetos da cena (lista sombras). Este conjunto de índices identifica os objetos que obstruem parcialmente a luz proveniente da fonte primária (fonte origem) para a esfera transparente (objeto origem).

Observe que o campo *intensidade* da variável de memória `SECUNDARIAS` é

o fator  $\frac{1}{e^{(1-k_t)\left(\frac{2R_t + \overline{TR}}{2}\right)}}$  extraído da equação (5.19). Este escalar multiplica os três

componentes do vetor cor da fonte primária original. Desta forma, pode-se dizer que o campo *intensidade* é um indicador da fração de luz da fonte primária que é “transmitida” para a fonte secundária de luz.

```
Estrutura PONTO3D {
    x: real;
    y: real;
    z: real;
}.
```

O campo *posição* da variável de memória `SECUNDARIAS` é especificado como sendo do tipo `PONTO3D` (acima). O tipo `PONTO3D` é uma estrutura de memória que armazena um ponto no  $R^3$ . Esta estrutura contém três valores reais, correspondentes às coordenadas  $x$ ,  $y$  e  $z$ . É largamente utilizada para definir as variáveis de memória do programa.

```
Estrutura OBSTACULO {
    indice: inteiro;
    prox: ^Estrutura OBSTACULO;
}.
```

O campo *lista\_sombras* da variável SECUNDARIAS é especificado como sendo do tipo OBSTACULO (acima). O tipo OBSTACULO é uma estrutura de memória que armazena um elemento da lista de “sombras” de uma fonte secundária não-cheia. Esta estrutura contém o campo *indice* que identifica univocamente o objeto “sombra” dentro do vetor de objetos da cena. O campo *prox* aponta para o próximo objeto “sombra” na lista de sombras da fonte secundária não-cheia.

```
OBJETOS : Vetor [número_máximo_de_objetos] de
    Estrutura OBJETO {
        tipo: {0,1};
        obj: ESPECIFICACAO_OBJETO;
        coeficientes_face: SUPERFÍCIE;
    }.
```

A variável de memória OBJETOS (acima) armazena a lista de objetos da cena. O campo *tipo* indica a forma do objeto, ou seja, se trata-se de uma esfera (*tipo* = 0) ou poliedro (*tipo* = 1). Ainda que o trabalho esteja fundamentado em ambientes esféricos, a flexibilidade de especificar objetos em outros formatos é importante para futuras extensões do trabalho.

```
União ESPECIFICACAO_OBJETO {
    esfera: ^ ESFERA;
    poliedro: ^POLIEDRO;
}.
```

O campo *obj* da variável de memória OBJETOS é do tipo ESPECIFICACAO\_OBJETO e armazena os parâmetros do objeto, seja esfera ou poliedro. Se o objeto é esfera, então o campo *esfera* aponta para o endereço da variável de memória que armazena os elementos de definição do objeto esfera correspondente.

```
Estrutura ESFERA {
    centro: PONTO3D;
    raio: real;
}.
```

A estrutura ESFERA armazena um ponto especificado no SRU, que é o *centro* de curvatura da esfera e um valor real positivo, o *raio* da esfera.

```
Estrutura SUPERFÍCIE {
    cor: COR;
    ka: real;
    kd: real;
    ks: real;
    kt: real;
    n: real;
    distrib_espec_Phong: real;
}.
```

O campo *coeficientes\_face* da estrutura de memória OBJETOS é do tipo estrutura SUPERFÍCIE, especificado acima. Esta estrutura armazena os parâmetros que caracterizam a superfície quanto aos fenômenos ópticos. São eles: Cor difusa do objeto (*cor*), coeficiente ambiental (*ka*), coeficiente de reflexão difusa (*kd*), coeficiente de reflexão especular (*ks*), coeficiente de transmissividade (*kt*), índice de refração absoluto do objeto (*n*) e coeficiente da distribuição especular de Phong (*distrib\_espec\_Phong*). Os coeficientes *ka*, *kd*, *ks* e *kt* são valores entre 0 e 1, *n* é maior ou igual a 1 e *distrib\_espec\_Phong* é um valor inteiro não negativo.

```

Estrutura COR {
    r: real;
    g: real;
    b: real;
}.

```

O campo *cor* da estrutura de memória SUPERFÍCIE é do tipo estrutura COR, descrito acima. Esta estrutura armazena um valor real não negativo para cada componente RGB. No momento de gravar a cor de um *pixel* no arquivo de saída, cada componente da cor é convertido para um valor inteiro entre 0 e 255.

```

FONTES: Vetor [número_máximo_luz] de
Estrutura LUZ {
    cor: COR;
    posição: PONTO3D;
    intensidade: real;
}.

```

A variável de memória FONTES armazena a lista de fontes luminosas puntiformes da cena. Para cada fonte luminosa contida em FONTES tem-se a cor da fonte luminosa (*cor*), a posição da luz no SRU (*posição*) e seu brilho (*intensidade*).

## 6.5 Sumário

Este capítulo detalhou o protótipo implementado para a proposta *Bitracer*.

O ambiente de desenvolvimento, a linguagem de programação utilizada, o formato dos arquivos de entrada e de saída, a estrutura funcional e as estruturas de dados foram apresentados.

O código das rotinas *bitracer1*, *bitracer2* e *RT\_shade*, armazenadas no arquivo *bitracer.c*, está disponível no Anexo B, para maiores detalhes.

## 7 Geração de Imagens e Análise dos Resultados

### 7.1 Introdução

Este capítulo descreve o processo de validação do método *Bitracer*, descrito em detalhes no capítulo 5.

O processo de validação consiste na geração de imagens tanto no protótipo *Bitracer* quanto em um programa de rastreamento convencional de raios. A partir destas imagens, faz-se uma análise comparativa dos resultados obtidos em relação aos aspectos tempo de processamento e realismo das imagens geradas.

### 7.2 Ambiente Onde as Imagens Foram Geradas e Exibidas

As imagens apresentadas neste capítulo foram geradas em um microcomputador IBM-PC com processador 80486DX4 e velocidade de processamento 100 MHz.

Utilizou-se uma janela de exibição (*viewport*) padrão para todas as imagens de 640x480 *pixels*. As imagens foram exibidas em um monitor SVGA colorido, configurado para exibição simultânea de 16 milhões de cores diferentes.

### 7.3 Avaliação quanto ao Realismo

A seguir são exibidos alguns exemplos de imagens geradas pelo protótipo *Bitracer*, onde simula-se a refração da luz que provém diretamente das fontes luminosas da cena. Em todos os exemplos, o nível máximo de recursividade dos raios é igual a 5, a cor de fundo é a cor preta  $(0\ 0\ 0)_{\text{RGB}}$  e a cor ambiente é igual a  $(0,1\ 0,1\ 0,1)_{\text{RGB}}$ .

A FIGURA 7.1 e a FIGURA 7.2 foram geradas para o mesmo arquivo de entrada "*fig1.bi*". Na criação da FIGURA 7.1 utilizou-se o algoritmo de rastreamento de raios convencional. Já a FIGURA 7.2 foi gerada pelo protótipo *Bitracer*. Os parâmetros da cena, especificados no arquivo "*fig1.bi*" são apresentados na TABELA 7.1. Observe-se que a cena é constituída de duas esferas, uma transparente e uma opaca e há apenas uma fonte luminosa puntiforme na cena. A esfera transparente (esfera 1) apresenta um índice de refração igual ao do vidro *crown* ( $n = 1,5171$ ).

A FIGURA 7.1 foi sintetizada em um tempo total de 48 s<sup>3</sup> enquanto a FIGURA 7.2 foi sintetizada num tempo de 49 s.

Na FIGURA 7.2, identifica-se uma região da esfera opaca (vermelha) que está mais clara. Esta iluminação resulta da contribuição da fonte secundária real para a componente difusa da esfera opaca.

---

<sup>3</sup> Tempos obtidos através da função *time* da linguagem C. Esta função retorna o número de segundos do processo.

TABELA 7.1- Parâmetros do arquivo de entrada "fig1.bi"

Posição da câmera	(0 0 10)
Orientação da câmera (VUP)	(0 1 0)
Alvo	(1 1 -1)
Ângulo de abertura horizontal da câmera	45°
Ângulo de abertura vertical da câmera	45°
Número de fontes luminosas	1
Número de objetos esféricos	2
Cor da fonte luminosa	(100 100 100) <sub>RGB</sub>
Posição da fonte luminosa	(10 10 10)
Brilho da fonte luminosa	1
Centro da Esfera 1	(3 3 3)
Raio da Esfera 1	1
Cor da Esfera 1	(100 10 10) <sub>RGB</sub>
Coefficiente ambiental da Esfera 1	0,1
Coefficiente de reflexão difusa da Esfera 1	0,2
Coefficiente de reflexão especular da Esfera 1	0,0
Coefficiente de transmissão da Esfera 1	0,7
Coefficiente da distribuição especular de Phong da Esfera 1	80
Índice de refração da Esfera 1	1,5171
Centro da Esfera 2	(1 1 -1)
Raio da Esfera 2	2
Cor da Esfera 2	(250 0 0) <sub>RGB</sub>
Coefficiente ambiental da Esfera 2	0,1
Coefficiente de reflexão difusa da Esfera 2	0,9
Coefficiente de reflexão especular da Esfera 2	0,0
Coefficiente de transmissão da Esfera 2	0,0
Coefficiente da distribuição especular de Phong da Esfera 2	70
Índice de refração da Esfera 2	1,0

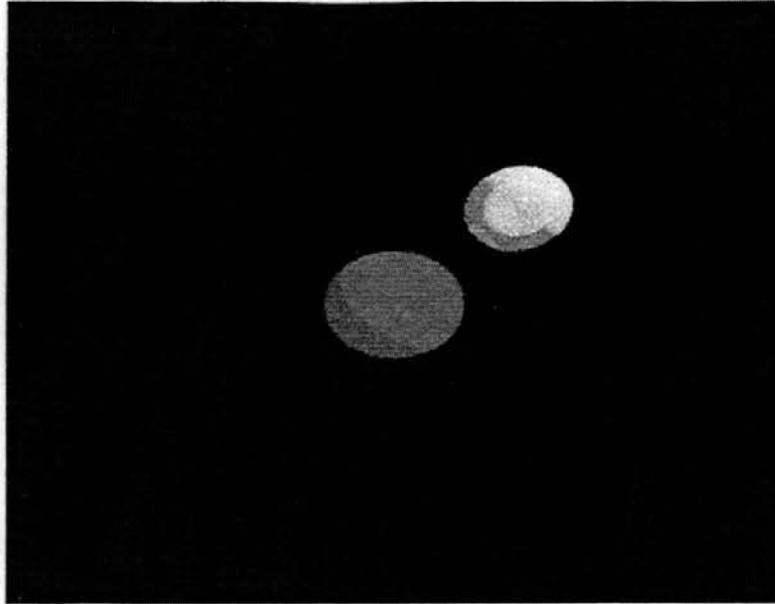


FIGURA 7.1- Imagem gerada por um algoritmo de *ray-tracing* convencional para os parâmetros da TABELA 7.1

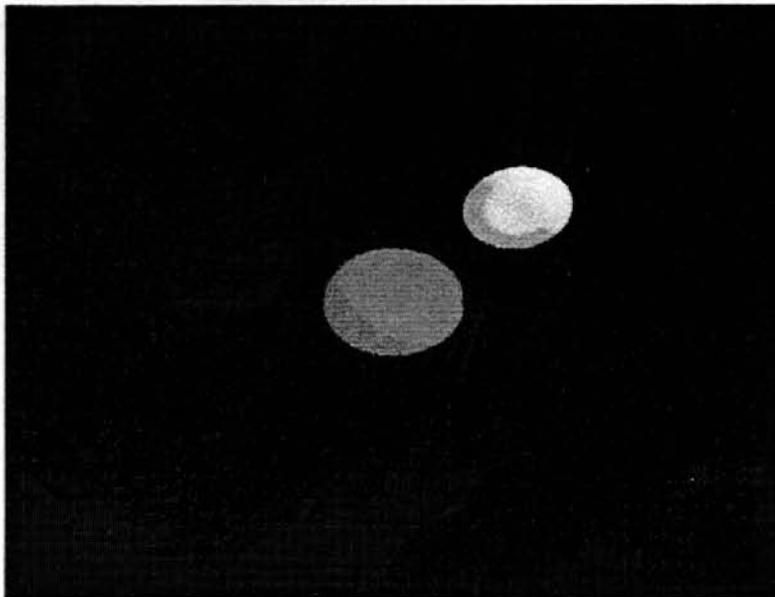


FIGURA 7.2- Imagem gerada pelo algoritmo *bitracer* para os parâmetros da TABELA 7.1. O algoritmo criou uma fonte secundária real e cheia, localizada em (2,04 2,04 2,04), com brilho 0,59 e ângulo de espalhamento de aproximadamente  $6,11^\circ$

A FIGURA 7.3 e a FIGURA 7.4 foram geradas para o arquivo de entrada “*fig2.bi*”. Os parâmetros do arquivo “*fig2.bi*” são apresentados na TABELA 7.2. Este arquivo de entrada descreve uma cena com duas esferas. A esfera 1, em termos de reflexão e transmitância, é especular e transparente. A esfera 2, quanto à reflexão e transmitância, é especular e opaca.

A FIGURA 7.3 foi gerada pelo algoritmo de rastreamento de raios convencional em um tempo total de 53 s. Já a FIGURA 7.4 foi sintetizada no protótipo *Bitracer* em um tempo total de 56 s.

A fonte secundária na FIGURA 7.4 contribuiu para a componente especular do modelo de iluminação 298 vezes. Entretanto, identificou-se uma pequena diferença

visual no *highlight*<sup>4</sup> da esfera azul (opaca) das Figuras 7.3 e 7.4. Tal ocorre pois a intensidade da fonte secundária é um escalar que multiplica a cor da fonte primária original, agindo portanto como um intensificador do *highlight* da fonte primária em determinados pontos, ou seja, na região iluminada pela fonte secundária de luz. Como a fonte secundária está situada ao longo do eixo central, ou seja, linha que conecta o centro da esfera transparente e a fonte luminosa primária originais, os pontos que recebem contribuição especular da fonte luminosa secundária também recebem contribuição especular da fonte luminosa primária original.

TABELA 7.2- Parâmetros do arquivo de entrada “fig2.bl”

Posição da câmera	(0 0 10)
Orientação da câmera	(0 1 0)
Alvo	(-2 2 1)
Ângulo de abertura horizontal da câmera	30°
Ângulo de abertura vertical da câmera	30°
Número de fontes luminosas	1
Número de objetos esféricos	2
Cor da fonte luminosa	(100 0 0) <sub>RGB</sub>
Posição da fonte luminosa	(-12 12 2)
Brilho da fonte luminosa	1
Centro da Esfera 1	(-4,5 4,5 1,5)
Raio da Esfera 1	1
Cor da Esfera 1	(10 0 100) <sub>RGB</sub>
Coefficiente ambiental da Esfera 1	0,03
Coefficiente de reflexão difusa da Esfera 1	0,02
Coefficiente de reflexão especular da Esfera 1	0,35
Coefficiente de transmissão da Esfera 1	0,6
Coefficiente da distribuição especular de Phong- Esfera 1	80
Índice de refração da Esfera 1	1,5
Centro da Esfera 2	(0 0 0)
Raio da Esfera 2	2
Cor da Esfera 2	(0 0 100) <sub>RGB</sub>
Coefficiente ambiental da Esfera 2	0,2
Coefficiente de reflexão difusa da Esfera 2	0
Coefficiente de reflexão especular da Esfera 2	0,8
Coefficiente de transmissão da Esfera 2	0,0
Coefficiente da distribuição especular de Phong- Esfera 2	70
Índice de refração da Esfera 2	1,0

<sup>4</sup> Região da superfície do objeto com um brilho intenso, resultante da reflexão especular da fonte luminosa. A cor desta região é determinada pela cor da fonte luminosa que a originou.

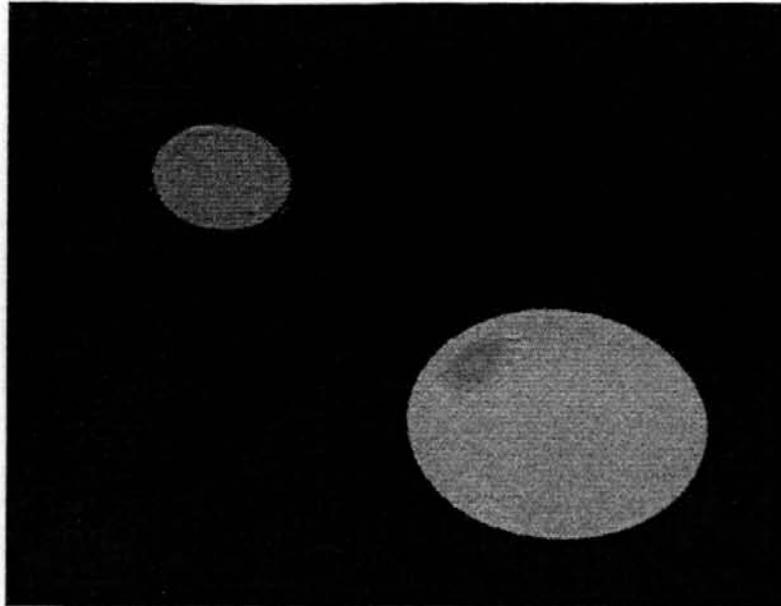


FIGURA 7.3- Imagem gerada por um algoritmo de *ray-tracing* convencional para os parâmetros da TABELA 7.2. A esfera 1, que apresenta coeficiente de transmitância diferente de zero, localiza-se no canto superior esquerdo da tela. E, a esfera 2, opaca, está localizada no canto inferior direito da tela.

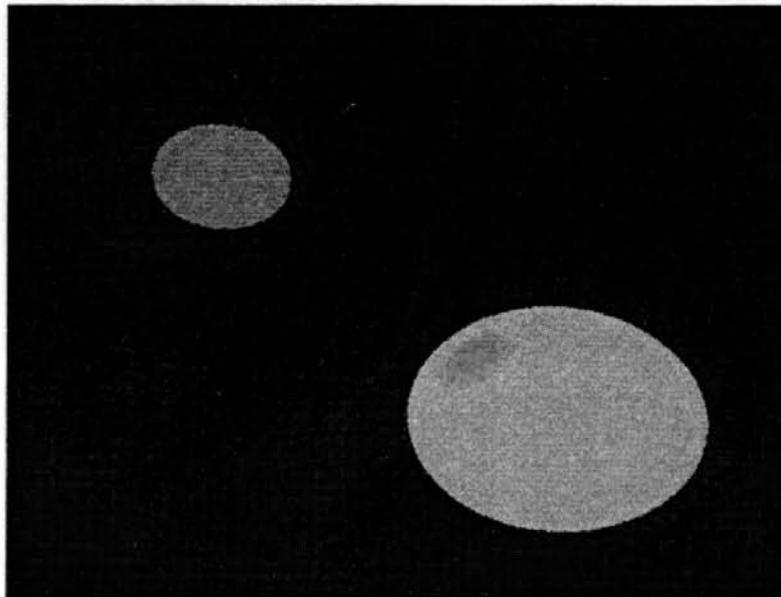


FIGURA 7.4- Imagem gerada pelo algoritmo *bitracer* para os parâmetros da TABELA 7.2. Foi gerada uma fonte secundária real e cheia para a cena, localizada em  $(-3,27 \ 3,27 \ 1,42)$ , com brilho 0,50 e ângulo de espalhamento de aproximadamente  $4,03^\circ$ .

As FIGURA 7.5 e FIGURA 7.6 foram geradas para o arquivo de entrada “*fig3.bi*”. Os parâmetros do arquivo “*fig3.bi*” são apresentados na TABELA 7.3. Este arquivo de entrada descreve uma cena com três esferas, as quais, em termos de transmitância são, transparente (esfera 2) e opacas (esferas 1 e 3).

A FIGURA 7.5 foi gerada pelo algoritmo de rastreamento de raios convencional em um tempo total de 57 s. Já a FIGURA 7.6 foi sintetizada pelo protótipo *Bitracer* em 60 s e contribuiu 75 vezes para a componente local de pontos pertencentes à superfície das esferas da cena.

A FIGURA 7.6 ilustra uma situação na qual foi gerada uma fonte secundária real e não-cheia. A esfera 1, opaca, oculta parcialmente a luz que provém da fonte luminosa puntiforme da cena e incide sobre a esfera transparente (esfera 2), produzindo uma fonte luminosa secundária localizada em (0,29 4,36 1,53), com uma intensidade de 0,50 e um ângulo de espalhamento de aproximadamente  $2,83^\circ$ .

Observando com atenção as figuras 7.5 e 7.6, verifica-se que as esferas 2 e 3 da FIGURA 7.6 têm uma tonalidade mais clara que as esferas equivalentes na FIGURA 7.5. Esta diferença de tons ocorre pelo acréscimo de luminosidade que a contribuição difusa da fonte luminosa secundária, gerada na primeira fase do algoritmo, produz. Observa-se também que o *highlight* produzido pela fonte luminosa puntiforme na esfera 3 da FIGURA 7.6 apresenta uma pequena diferença de tonalidade quando comparado ao da FIGURA 7.5. Esta diferença é resultado da contribuição especular da fonte luminosa secundária da cena.

TABELA 7.3- Parâmetros do arquivo de entrada "fig3.bi"

Posição da câmera	(6 10 4)
Orientação da câmera	(0 1 0)
Alvo	(0,5 3 0)
Ângulo de abertura horizontal da câmera	$37^\circ$
Ângulo de abertura vertical da câmera	$37^\circ$
Número de fontes luminosas	1
Número de objetos esféricos	3
Cor da fonte luminosa	(100 0 0) <sub>RGB</sub>
Posição da fonte luminosa	(4 13 4)
Brilho da fonte luminosa	1
Centro da Esfera 1	(1,5 7 2)
Raio da Esfera 1	0,25
Cor da Esfera 1	(0 0 0) <sub>RGB</sub>
Coefficiente ambiental da Esfera 1	0,2
Coefficiente de reflexão difusa da Esfera 1	0,4
Coefficiente de reflexão especular da Esfera 1	0,4
Coefficiente de transmissão da Esfera 1	0,0
Coefficiente da distribuição especular de Phong da Esfera 1	80
Índice de refração da Esfera 1	1,0
Centro da Esfera 2	(1 6 2)
Raio da Esfera 2	1
Cor da Esfera 2	(100 100 100) <sub>RGB</sub>
Coefficiente ambiental da Esfera 2	0,1
Coefficiente de reflexão difusa da Esfera 2	0,0

Coeficiente de reflexão especular da Esfera 2	0,4
Coeficiente de transmissão da Esfera 2	0,6
Coeficiente da distribuição especular de Phong da Esfera 2	80
Índice de refração da Esfera 2	1,5
Centro da Esfera 3	(-1 0 0)
Raio da Esfera 3	2
Cor da Esfera 3	(100 100 100) <sub>RGB</sub>
Coeficiente ambiental da Esfera 3	0,2
Coeficiente de reflexão difusa da Esfera 3	0,0
Coeficiente de reflexão especular da Esfera 3	0,8
Coeficiente de transmissão da Esfera 3	0,0
Coeficiente da distribuição especular de Phong da Esfera 3	70
Índice de refração da Esfera 3	1,0

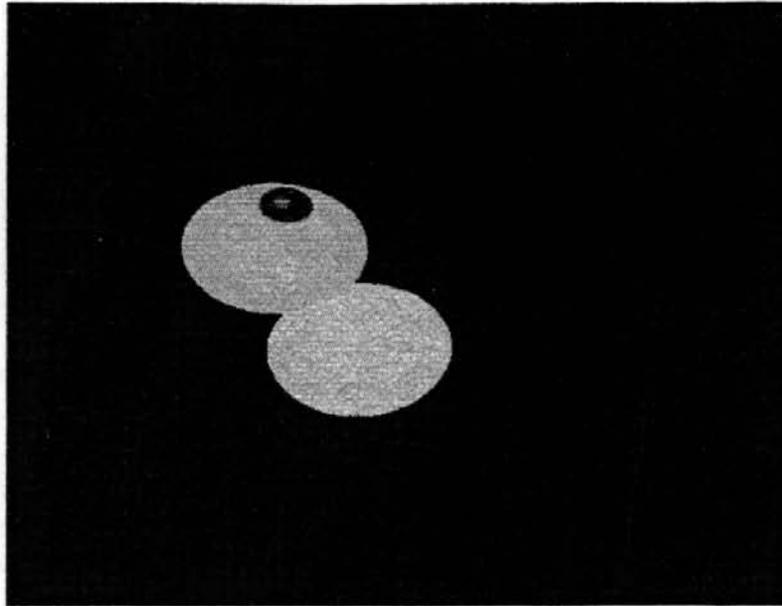


FIGURA 7.5- Imagem gerada por um algoritmo de *ray-tracing* convencional para os parâmetros da TABELA 7.3. A esfera 1 é a esfera negra localizada acima do centro da esfera 2 (cinza). A esfera 3, também cinza, é a esfera localizada à direita e abaixo da esfera 2.

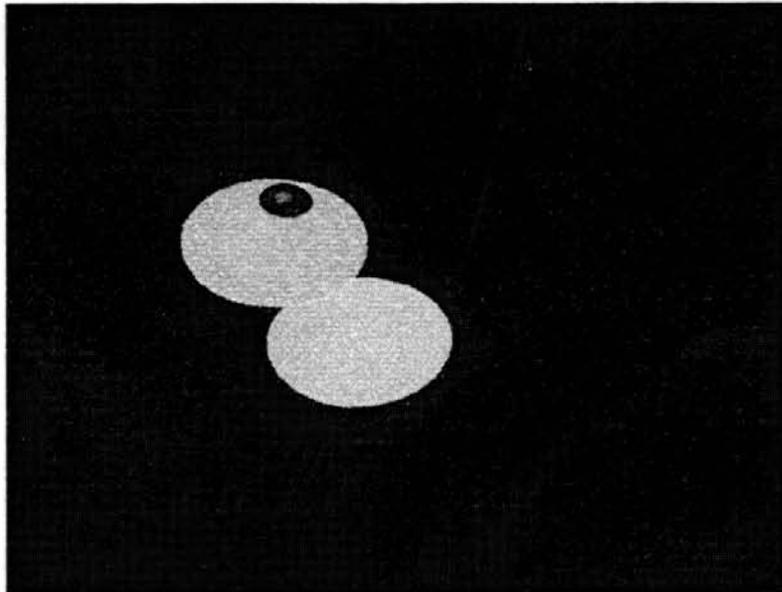


FIGURA 7.6- Imagem gerada pelo algoritmo *bitracer* para a TABELA 7.3

#### 7.4 Avaliação quanto à Complexidade e Tempo de Processamento

Um algoritmo de rastreamento de raios recursivo calcula a cor final de cada *pixel* após interseccionar o raio de visão, disparado do observador para o centro do *pixel*, com todos os objetos da cena, identificando, se existir, o objeto interseccionado mais próximo

do observador. Dependendo das características físicas do objeto atingido, o raio de visão pode gerar dois novos raios, um refletido e um refratado. Cada raio, por sua vez, é testado contra todos os objetos da cena à procura de alguma intersecção. Se esta ocorrer, são gerados novamente zero, um ou dois raios, e o processo é repetido, obtendo-se uma árvore de raios refletidos e transmitidos (ART). Neste algoritmo utiliza-se um valor máximo de iterações para impedir que a árvore de raios derivados cresça indefinidamente.

O algoritmo *Bitracer* é composto de duas fases. A primeira fase compreende a geração das fontes secundárias de luz e a segunda fase, a etapa de traçado dos raios. Para obter a função de complexidade da proposta *Bitracer*, primeiramente será feita a análise de complexidade de um algoritmo de traçado de raios convencional (*Tracer*). A seguir, far-se-á a análise de complexidade da Fase 1 – Criação das fontes secundárias. Por fim, será efetuada a análise de complexidade da Fase 2 – Traçado de Raios. As rotinas referenciadas na obtenção da função de complexidade de *Bitracer* correspondem às funções apresentadas no pseudocódigo das seções 5.7.1 e 5.7.2.

**i) Análise de um algoritmo de traçado de raios convencional (*Tracer*)**

Considere-se uma cena com  $p$  pixels,  $n$  objetos,  $l$  fontes luminosas puntiformes e um número máximo de iterações  $i$  (profundidade máxima da ART) sendo  $m$  a iteração corrente,  $0 \leq m \leq i$ . No pior caso, o número máximo de operações realizadas por um algoritmo de rastreamento de raios recursivo *Tracer* é dado por

$$O(\text{Tracer}) = p[c_1 + O(\text{RTtrace}(m))], \quad m = 0, \quad (7.1)$$

onde  $c_1$  é o custo de criação dos raios iniciais (do observador para o pixel) e  $O(\text{RTtrace}(0))$  é o número máximo de operações efetuadas para encontrar a cor de um pixel a partir do raio inicial.

O custo de  $\text{RTtrace}(0)$  é dada por

$$\begin{aligned} O(\text{RTtrace}(0)) = n[O(\text{Calcula\_intersecção})] + \\ O(\text{Calcula\_normal}) + \\ O(\text{RTshade}). \end{aligned} \quad (7.2)$$

Seja  $c_2$  o custo computacional da rotina *Calcula\_intersecção(...)* e  $c_3$  o custo computacional da rotina *Calcula\_normal(...)*. Portanto, pode-se reescrever a equação (7.2) como

$$O(\text{RTtrace}(0)) = nc_2 + c_3 + O(\text{RTshade}).$$

O custo computacional da rotina *RTshade(...)* é obtido por

$$O(\text{RTshade}) = c_4 + nc_2 + \sum_{j=1}^l (c_{5j} + c_{6j}) + O(\text{RTtrace}(1)) + O(\text{RTtrace}(1)),$$

ou ,

$$O(\text{RTshade}) = c_4 + nc_2 + l(c_5 + c_6) + 2O(\text{RTtrace}(1)), \quad (7.3)$$

onde  $c_4$  é o custo computacional de cálculo da componente ambiental,  $c_{5j}$  é o custo de cálculo da componente difusa e  $c_{6j}$  é o custo de cálculo da componente especular, ambos para a fonte  $j$ , onde  $1 \leq j \leq l$ . Portanto, tem-se que

$$\sum_{j=1}^l (c_{5j} + c_{6j}) = l(c_5 + c_6)$$

Como, potencialmente, uma fonte luminosa qualquer pode não contribuir para um ponto devido à presença de algum objeto que obstrui a passagem da luz, é necessário efetuar um teste de intersecção (custo  $c_2$ ) com os  $n$  objetos da cena. Isto é dado pelo produto  $nc_2$  que é o custo máximo para a obtenção do fator de sombra, ou seja, consiste no tempo gasto para interseccionar o raio de sombra com todos os objetos da cena. O fator de sombra é o produto de todos os coeficientes de transmitância dos objetos que estão no caminho do raio de sombra, entre o ponto atual e a fonte luminosa. E,  $2O(RTtrace(l))$  é o custo computacional dos raios refletido e transmitido de nível 1.

O próximo passo é obter  $O(RTtrace(l))$ . Sabe-se que

$$O(RTtrace(\text{máximo nível})) = O(RTtrace(i)) = 0, \text{ e}$$

$$O(RTtrace(m)) = nc_2 + c_3 + c_4 + nc_2 + l(c_5 + c_6) + 2O(RTtrace(m+1)).$$

Fazendo

$$g(n, l) = nc_2 + c_3 + c_4 + nc_2 + l(c_5 + c_6)$$

e substituindo em  $O(RTtrace(m))$  obtém-se

$$O(RTtrace(m)) = g(n, l) + 2O(RTtrace(m+1)).$$

Como  $O(RTtrace(m+1)) = g(n, l) + 2O(RTtrace(m+2))$ ,  $O(RTtrace(m))$  é reescrita como

$$O(RTtrace(m)) = g(n, l) + 2g(n, l) + 4O(RTtrace(m+2)).$$

Por indução matemática em  $m$  [CUN 88], tem-se

$$O(RTtrace(m)) = g(n, l)(2^{i-m} - 1).$$

Com  $m = 0$ , o resultado acima permite reescrever a equação (7.1) como

$$O(Tracer(p, n, l, i)) = p(c_1 + g(n, l)(2^i - 1)). \quad (7.4)$$

### ii) Análise da Fase 1 – criação das fontes secundárias

Seja  $O(\text{fase1})$  o custo computacional da primeira fase, quando são geradas as fontes secundárias de luz. No pior caso, o custo computacional de criação e verificação de obstrução de uma fonte secundária é  $c_8 + c_7(n-1)$ , onde  $c_8$  é o custo de criação de uma fonte secundária e  $c_7$  é o custo de se aplicar os testes de visibilidade a uma esfera da cena. O pior caso da *fase1* ocorre quando todos os  $n$  objetos são transparentes e originam fontes secundárias de luz para cada uma das  $l$  fontes puntiformes da cena. Além disso, todos os demais objetos da cena são candidatos a obstáculos na primeira etapa e, portanto, devem participar do teste de visibilidade. Assim, a função de complexidade de *fase1* é dada por

$$\begin{aligned} O(\text{fase1}) &= ln[c_8 + (n-1)c_7] \\ &= lnc_8 + ln^2c_7 - lnc_7. \end{aligned} \quad (7.5)$$

### iii) Análise da Fase 2 – traçado de raios

A função de complexidade da *fase2* é calculada de forma análoga à equação (7.4), com um custo adicional determinado pela inclusão das fontes secundárias de luz. Seja  $f$  o número de fontes secundárias geradas na *fase1*. No pior caso foram geradas  $nl$  fontes secundárias de luz, ou seja,  $f = nl$ . O custo da *fase2* é dado por

$$\begin{aligned} O(\text{fase2}) &= O(Tracer(p, n, l, i)) + O(Tracer(p, n, f, i)) \\ &= p[c_1 + g(n, l)(2^i - 1) + h(n, f)(2^i - 1)], \end{aligned} \quad (7.6)$$

onde  $h(n, f)$  é o custo computacional de cálculo da contribuição das fontes secundárias para a iluminação local. A função  $h(n, f)$  é

$$h(n, f) = f[c_9 + c_{10} + nc_2 + (c_5 + c_6)],$$

onde  $c_9$  é o custo computacional de criação de um raio de sombra para uma fonte secundária,  $c_{10}$  é custo computacional de verificação da pertinência do raio de sombra ao feixe de raios emitidos pela fonte secundária, o produto  $nc_2$  é o custo computacional da intersecção do raio de sombra com todos os objetos da cena e  $c_5$  e  $c_6$  são os custos de cálculo da contribuição difusa e especular da fonte secundária.

#### iv) Análise de Bitracer

Para o algoritmo *Bitracer* pode-se escrever que a função de complexidade é dada por

$$O(\text{Bitracer}) = O(\text{fase1}) + O(\text{fase2}) \quad (7.7)$$

onde,  $O(\text{fase1})$  é o custo computacional da primeira fase, quando são geradas as fontes secundárias de luz, dado pela equação (7.5) e,  $O(\text{fase2})$  é o custo computacional da segunda etapa, quando os raios são disparados do observador para a cena, obtido na equação (7.6).

Adicionando-se as equações (7.5) e (7.6), pode-se escrever que a ordem de complexidade de *Bitracer* é, para o pior caso,

$$O(\text{Bitracer}) = lnc_8 + ln^2c_7 - lnc_7 + p[c_1 + g(n, l)(2^i - 1) + h(n, f)(2^i - 1)]. \quad (7.8)$$

A TABELA 7.4 apresenta tempos de processamento obtidos com o protótipo *Bitracer*, variando o número de esferas transparentes da cena, seu posicionamento relativo, o número de fontes luminosas puntiformes da cena e, por consequência, o número de fontes secundárias geradas pela primeira etapa. São feitas combinações dos objetos transparentes e das fontes luminosas puntiformes da cena até um total de 12 amostras. As fontes luminosas puntiformes variam de 1 a 3 e as esferas transparentes de 1 a 4. Fez-se uma modificação no protótipo *Bitracer* implementado, de forma a considerar somente a etapa de geração das fontes secundárias, desconsiderando a contribuição destas fontes para a cena. Os mesmos exemplos utilizados na obtenção da TABELA 7.4 foram utilizados com o algoritmo de rastreamento de raios convencional. Os tempos de processamento obtidos com o algoritmo de rastreamento convencional são sumarizados na TABELA 7.5.

TABELA 7.4- Tempos de geração de imagens no protótipo *Bitracer*

Amostra	Número de Objetos (n)		Número de fontes luminosas		Tempo
	Opacos	Transparentes	Puntiformes (l)	Secundárias (f)	
1	0	1	1	1	55 s
2	0	1	2	2	58 s
3	0	1	3	3	64 s
4	0	2	1	2	46 s
5	0	2	2	4	47 s
6	0	2	3	6	47 s
7	0	3	1	3	51 s

8	0	3	2	6	51 s
9	0	3	3	9	53 s
10	0	4	1	4	62 s
11	0	4	2	8	63 s
12	0	4	3	12	65 s

TABELA 7.5- Tempos de geração de imagens para as amostras da TABELA 7.4, obtidos no programa *ray-tracing* convencional

<i>Amostra</i>	<i>Tempo</i>
1	55 s
2	58 s
3	64 s
4	46 s
5	47 s
6	47 s
7	51 s
8	51 s
9	53 s
10	59 s
11	61 s
12	62 s

A FIGURA 7.7 apresenta um gráfico que compara os tempos de processamento da TABELA 7.4, obtidos pelo protótipo *Bitracer* para as cenas de amostra, com os tempos de processamento da TABELA 7.5, obtidos por um *ray-tracing* recursivo convencional.

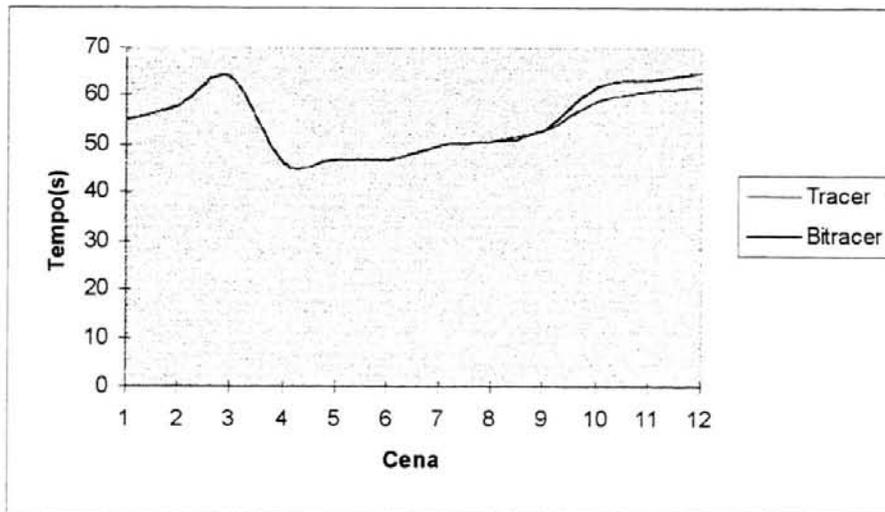


FIGURA 7.7- Gráfico: Tempo de processamento x cena gerada

Observando a FIGURA 7.7 identifica-se que o tempo de geração das fontes secundárias no protótipo *Bitracer* não foi significativo até a amostra 9, a qual contém 3 esferas transparentes. Da amostra 10 em diante o tempo de geração das fontes secundárias é diferente de zero. O tempo de geração das fontes secundárias é da ordem de  $O(\textit{fase1})$ , apresentada na equação (7.5). O termo predominante na equação (7.5) é  $\ln^2 c_7$  e, portanto a *fase1* é da ordem de  $O(\ln^2)$ . Observando-se os dados da TABELA 7.4 verifica-se que para um  $n$  fixo, o tempo de processamento cresce com o número de fontes puntiformes da cena. Este crescimento, pelo que sugere a equação (7.5), é possivelmente linear. Entretanto, para tornar estes resultados conclusivos seriam necessários testes extensivos, o que não foi possível por limitação de recursos.

A TABELA 7.6 apresenta resultados obtidos para cenas com apenas uma fonte luminosa primária. O número de esferas transparentes nos arquivos de teste variou de 0 a 4 e 8. E, as fontes secundárias geradas por *Bitracer* também sofreram uma variação de acordo com o número de esferas transparentes. Para gerar a primeira linha da tabela utilizou-se uma cena com uma esfera opaca e com uma fonte luminosa primária. O gráfico da FIGURA 7.9 ilustra os resultados apresentados na TABELA 7.6. Observe-se que o custo computacional da proposta *Bitracer* comparado ao custo do algoritmo de *ray-tracing* convencional aumenta conforme o número de esferas transparentes na cena cresce. Isto é esperado com base na análise de complexidade de *Bitracer* e *Tracer*. Na equação (7.8), que dá a ordem do custo computacional de *Bitracer* no pior caso, o termo predominante é  $p[h(n, f)(2^i - 1)]$ , onde  $p$  é o número de *pixels*,  $n$  é o número de objetos da cena,  $i$  é a profundidade máxima da ART e  $f$  é o número de fontes luminosas secundárias da cena. Uma vez que o custo computacional máximo de *Bitracer* ocorre quando todos os objetos são esferas transparentes, então, o número de objetos presentes na cena ( $n$ ) também é igual ao número de esferas transparentes na cena. Da mesma forma, o pior caso de *Bitracer* ocorre quando o número de fontes secundárias é igual a  $nl$ , onde  $l$  é o número de fontes puntiformes da cena. Portanto, o algoritmo *Bitracer* tem um custo computacional máximo da ordem de  $O(pn^2il)$ . Seguindo o mesmo raciocínio obtém-se que o algoritmo de *ray-tracing* recursivo tem um

custo computacional da ordem de  $O(pni)$ . Considerando-se os parâmetros  $p$ ,  $i$  e  $l$  constantes, conclui-se que *Bitracer* é  $O(n^2)$  e *Tracer* é  $O(n)$ .

Para um número pequeno de esferas transparentes observou-se que *Bitracer* apresenta um custo computacional comparável a *Tracer*, como pode ser visto na FIGURA 7.8.

TABELA 7.6- Tempos de geração de imagens para uma fonte primária obtidos em *Bitracer* e no programa *ray-tracing* convencional (*Tracer*)

$N^{\circ}$ fontes ( $f$ )	<i>Bitracer</i>	<i>Tracer</i>
0	47 s	47 s
1	55 s	53 s
2	53 s	50 s
3	55 s	52 s
4	63 s	58 s
8	88 s	83 s

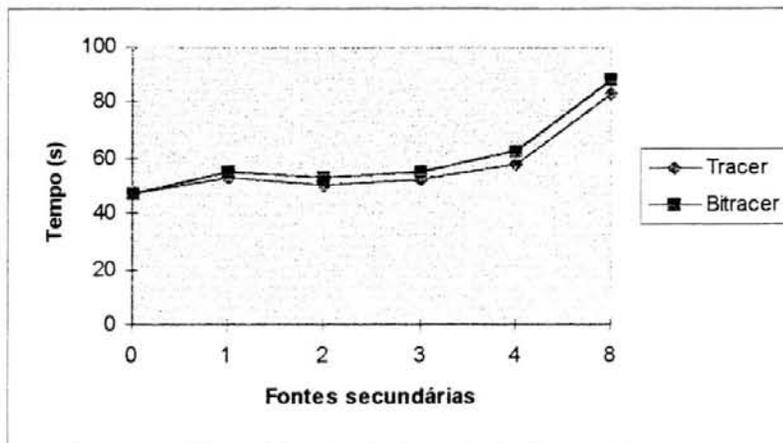


FIGURA 7.8- Gráfico: Tempo de geração de imagens em *Bitracer* x tempo de geração de imagens em um *ray-tracing* convencional (*Tracer*)

## 7.5 Comparação entre o Método *Bitracer* e o Método de Rastreamento de Raios Recursivo

Os dois métodos implementados apresentam diferenças quanto ao realismo e tempo de execução.

O custo computacional de *Bitracer* é proporcional ao quadrado do número de esferas transparentes da cena, enquanto o custo computacional de um algoritmo de *ray-tracing* recursivo convencional é proporcional ao número de objetos presentes na cena.

Observou-se, através dos exemplos gerados, que a componente especular das fontes secundárias não promove diferenças significativas nas cenas sintetizadas por *Bitracer*, em comparação com as cenas sintetizadas pelo algoritmo de *ray-tracing*

convencional. Como a componente especular é altamente dependente da visão e, dado que a fonte secundária de luz ilumina uma região bastante limitada da cena, pode-se reduzir o custo computacional de *Bitracer* eliminando os cálculos referentes à contribuição especular das fontes secundárias de luz.

Já a contribuição difusa da(s) fonte(s) secundária(s) de luz produziu, para alguns exemplos gerados, efeitos visuais que auxiliaram na identificação do posicionamento relativo da(s) fonte(s) de luz puntiforme(s) da cena.

Em última análise, pode-se afirmar que o método *Bitracer* proposto aumenta o realismo da cena.

## 7.6 Conclusão

Este capítulo abordou a geração de imagens no protótipo *Bitracer* e o processo de análise destas imagens quanto a realismo e tempo de processamento.

Fez-se uma análise comparativa entre os resultados obtidos com *Bitracer* e os resultados obtidos com um algoritmo de rastreamento de raios recursivo. As imagens produzidas pelo protótipo *Bitracer* podem apresentar efeitos de luminosidade localizados (FIGURA 7.2) ou difundidos por toda a cena (FIGURA 7.6). Entretanto, o algoritmo *Bitracer* apresenta, no pior caso, um custo computacional dado pelo quadrado do custo computacional de um algoritmo de *ray-tracing* convencional.

## 8. Conclusões

Este capítulo apresenta as principais conclusões obtidas a partir do trabalho aqui apresentado. Estas conclusões são relacionadas aos objetivos iniciais do trabalho, os quais estão descritos no capítulo “1-Introdução”.

Também são apresentadas propostas para a continuidade do trabalho de pesquisa desenvolvido.

### 8.1 Conclusões

A utilização da Computação Gráfica nas mais diversas áreas, como simulação, projeto, entretenimento, educação, comando e controle tem motivado o desenvolvimento de pesquisas para a obtenção de realismo visual.

O objetivo global deste trabalho, a saber, aplicação da técnica de rastreamento bidirecional de raios à síntese de objetos transparentes, foi alcançado.

Os objetivos específicos também foram alcançados, tendo sido desenvolvidos:

- Um estudo da Óptica. Neste estudo, identificaram-se os elementos fundamentais de um modelo de iluminação. Fez-se também um estudo das lentes esféricas convergentes, cujos elementos foram utilizados na concretização do objetivo global do trabalho. Neste contexto, o sistemas de equações dos pontos conjugados (equação 2.11) destacou-se por possibilitar a identificação de cada situação (imagem real, virtual ou no infinito) a partir dos elementos  $p$  e  $f$  conhecidos bem como localização da imagem  $i$  a partir de  $p$ ,  $f$  e  $n$ ;
- um estudo dos principais algoritmos de visualização de objetos transparentes. Este estudo possibilitou a escolha do método de *ray-tracing* como o mais apropriado para a síntese de imagens com efeitos de refração; e
- a criação de um modelo que utiliza a técnica de rastreamento bidirecional de raios em ambientes esféricos (*Bitracer*), de forma a simular a refração da luz que provém diretamente das fontes luminosas puntiformes da cena. Este modelo define uma nova fonte luminosa com uma área de atuação limitada (fonte secundária) para cada combinação de fonte luminosa puntiforme e esfera transparente vista, parcialmente ou completamente, pela fonte luminosa puntiforme em questão. Com a definição de novas fontes luminosas na cena (fontes secundárias) a segunda etapa do algoritmo de *ray-tracing* mantém sua estrutura funcional básica.

Com os objetivos específicos alcançados, o modelo desenvolvido foi implementado e aplicado a cenas contendo esferas transparentes e opacas.

A utilização do modelo desenvolvido gerou imagens com um grau de realismo maior que as imagens geradas por um algoritmo de *ray-tracing* convencional. Nos exemplos gerados, observou-se que o tempo de geração das imagens na proposta *Bitracer* é praticamente igual ao de geração das imagens em um algoritmo de *ray-tracing* convencional.

O cálculo do ângulo de espalhamento da fonte secundária de luz envolve várias variáveis, como pode-se observar nas equações (5.14) e (5.21). Constatou-se, através dos testes realizados, que um aumento na distância da fonte luminosa puntiforme à esfera transparente ( $p$ ), em se tratando de fontes secundárias reais, propicia um aumento no ângulo de espalhamento da fonte secundária. A variável  $p$  pode ser alterada diretamente nos arquivos de entrada modificando-se a posição da fonte luminosa primária na cena.

Outra observação feita a partir dos arquivos-imagem gerados pelo protótipo diz respeito à contribuição especular das fontes secundárias de luz. A contribuição especular de uma fonte luminosa para um ponto é determinada pelas características da fonte luminosa (cor e intensidade) e posição relativa do observador, fonte luminosa e ponto. Uma fonte secundária de luz tem a mesma cor da fonte primária original e sua intensidade é uma fração da intensidade da fonte primária. Ademais, como a fonte secundária localiza-se ao longo da linha que conecta a fonte luminosa primária e o centro da esfera transparente, seu *highlight* incide sobre o *highlight* da fonte luminosa primária. Notou-se que os arquivos-imagem gerados pelo protótipo não apresentaram mudanças significativas no *highlight* da fonte primária quando a fonte luminosa secundária correspondente contribuía especularmente para a cena.

O estudo desenvolvido gerou, ainda, propostas para novos desdobramentos, os quais serão abordados na próxima seção.

## 8.2 Desdobramentos deste Trabalho

Como trabalhos futuros de continuidade e melhoria da proposta apresentada, pode-se citar:

- Remoção dos cálculos referentes à contribuição especular da(s) fonte(s) secundária(s) e análise da influência desta modificação no custo computacional total de *Bitracer*;
- inclusão de limites inferiores para a intensidade e o ângulo de espalhamento de uma fonte secundária de luz, de forma a impedir a criação de fontes secundárias que não contribuem efetivamente para a cena; e
- obtenção de fontes secundárias para objetos transparentes não esféricos. Uma idéia é utilizar esferas inscritas ou circunscritas ao objeto para determinar as fontes secundárias de poliedros.

## Anexo A

Esta seção descreve a dedução da fórmula dos pontos conjugados para lentes espessas apresentada na equação (2.11).

Assuma uma óptica com aproximação gaussiana, ou seja, uma correspondência unívoca entre o ponto objeto e o ponto imagem. Veja o esquema da FIGURA A.1. Esta figura mostra a refração do raio incidente  $\overline{PI}$  que passa do ar para um meio mais refringente, o qual é limitado à esquerda por uma superfície esférica convexa e ilimitado à direita.  $P$  é o ponto luminoso colocado sobre o eixo central da superfície. O eixo central da superfície é determinado pelo raio  $\overline{PO}$ , que incide normal à superfície e não sofre desvio ao atravessá-la. O raio  $\overline{PI}$  é um raio incidente qualquer e que se refrata segundo  $\overline{IP_1}$ .  $P_1$  é um ponto sobre o eixo central conhecido como foco conjugado de  $P$  ou ponto imagem de  $P$ .

Considere os triângulos  $ICP$  e  $ICP_1$ , os quais têm suas bases situadas sobre o eixo central ( $CP$  e  $CP_1$ , respectivamente) e têm mesma altura ( $IO'$ ). A área de um triângulo é

$$\frac{\text{base} \times \text{altura}}{2}$$

Portanto, pode-se escrever que

$$\text{area}(ICP) = \frac{CP \times IO'}{2} \text{ e}$$

$$\text{area}(ICP_1) = \frac{CP_1 \times IO'}{2}$$

A razão destas áreas é

$$\frac{\text{area}(ICP)}{\text{area}(ICP_1)} = \frac{CP \times IO'}{2} \times \frac{2}{CP_1 \times IO'} = \frac{CP}{CP_1}$$

Substituindo

$$CP = p + R, \text{ e}$$

$$CP_1 = p'_1 - R$$

tem-se que

$$\frac{\text{area}(ICP)}{\text{area}(ICP_1)} = \frac{p + R}{p'_1 - R} \quad (\text{A.1})$$



Substituindo  $\text{sen}(P\hat{I}C)$  por  $\text{sen}(\hat{i})$  e  $IC$  por  $R$  na equação da área do triângulo  $PCI$  obtém-se

$$\text{area}(ICP) = \frac{IP \times R \times \text{sen}(\hat{i})}{2}.$$

Para a hipótese inicial de que os raios incidentes são centrais, isto é, o raio da calota da lente é muito menor em relação à sua distância em relação à fonte puntiforme, é possível aproximar  $IP$  por  $p$ . A equação da área do triângulo  $ICP$  pode então ser rescrita como

$$\text{area}(ICP) = \frac{p \times R \times \text{sen}(\hat{i})}{2}.$$

Por analogia, a área do triângulo  $ICP_1$  é dada por

$$\text{area}(ICP_1) = \frac{p'_1 \times R \times \text{sen}(\hat{r})}{2}.$$

E, a razão das áreas é

$$\frac{\text{area}(ICP)}{\text{area}(ICP_1)} = \frac{p \times R \times \text{sen}(\hat{i})}{2} \times \frac{2}{p'_1 \times R \times \text{sen}(\hat{r})} = \frac{p}{p'_1} \times \frac{\text{sen}(\hat{i})}{\text{sen}(\hat{r})}.$$

Como

$$\frac{\text{sen}(\hat{i})}{\text{sen}(\hat{r})} = \frac{n_r}{n_i} = \frac{n}{1} = n,$$

a razão das áreas é

$$\frac{\text{area}(ICP)}{\text{area}(ICP_1)} = \frac{p}{p'_1} \times n. \quad (\text{A.3})$$

Ao igualar a equação (A.1) e a equação (A.3) tem-se que

$$\frac{p+R}{p'_1-R} = \frac{p}{p'_1} \times n.$$

Eliminando o denominador de ambos os membros da igualdade obtém-se

$$p'_1 \times (p+R) = np \times (p'_1 - R).$$

Agora, basta dividir ambos os membros por  $Rpp'_1$  e a equação resultante é

$$\frac{1}{p} + \frac{n}{p'_1} = \frac{n-1}{R}. \quad (\text{A.4})$$

Veja o esquema da FIGURA A.3. Esta figura mostra a refração do raio incidente  $\overline{PI}$  que passa de um meio mais refringente, ilimitado à esquerda e limitado à direita, através de uma superfície esférica côncava.

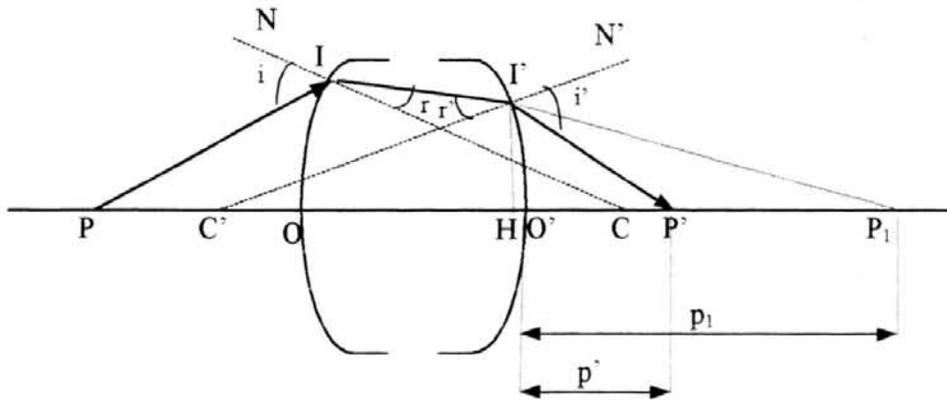


FIGURA A.3

O ponto  $P'$  é o ponto imagem conjugado do ponto luminoso  $P_1$ . Considere os triângulos  $I'C'P_1$  e  $I'C'P'$ , os quais compartilham o mesmo vértice  $I'$  e têm a mesma altura. A razão das áreas destes dois triângulos é dada por

$$\frac{\text{area}(I'C'P_1)}{\text{area}(I'C'P')} = \frac{\frac{C'P_1 \times I'H}{2}}{\frac{C'P' \times I'H}{2}} = \frac{C'P_1}{C'P'} = \frac{R' + p_1}{R' + p'}. \quad (\text{A.5})$$

Nesta equação  $R'$  é o raio de curvatura da superfície côncava. Aplicando a fórmula trigonométrica da equação (A.3) tem-se que a razão das áreas é

$$\frac{\text{area}(I'C'P_1)}{\text{area}(I'C'P')} = \frac{I'C' \times I'P_1 \times \text{sen}(r')}{I'C' \times I'P' \times \text{sen}(i')} = \frac{R' \times I'P_1 \times \text{sen}(r')}{R' \times I'P' \times \text{sen}(i')}.$$

Substituindo  $I'P_1$  por  $p_1$  e  $I'P'$  por  $p'$  obtém-se que

$$\frac{\text{area}(I'C'P_1)}{\text{area}(I'C'P')} = \frac{p_1 \times \text{sen}(r')}{p' \times \text{sen}(i')} = \frac{p_1}{p'} \times \frac{1}{n}. \quad (\text{A.6})$$

Igualando as equações (A.5) e (A.6) obtém-se a igualdade

$$\frac{R' + p_1}{R' + p'} = \frac{1}{n} \times \frac{p_1}{p'}.$$

Ao eliminar o denominador e dividir ambos os membros da equação por  $R'p_1p'$  chega-se a

$$\frac{-n}{p_1} + \frac{1}{p'} = \frac{n-1}{R'}. \quad (\text{A.7})$$

Assuma que a FIGURA A.3 representa uma lente espessa com espessura  $OO'$ . O objetivo é encontrar  $P'$ , o ponto imagem conjugado do ponto luminoso  $P$ . A equação (A.4) determina a primeira refração. Já a equação (A.7) calcula a segunda refração. Substituindo  $p'_1$  na equação (A.4) por

$p_1 + e,$ 

onde  $e$  é a espessura da lente, obtém-se a equação

$$\begin{cases} \frac{1}{p} + \frac{n}{p_1 + e} = \frac{n-1}{R} \\ \frac{1}{p'} - \frac{n}{p_1} = \frac{n-1}{R'} \end{cases} \quad (\text{A.8})$$

conhecida como equação dos pontos conjugados.

## Anexo B

```

/*-----*/
/* Projeto: bitracer */
/* Arquivo: bitracer.c */
/* Função : Implementar as funções para execução do ray tracing. */
/*-----*/

/*-----*/
/* Função: bitracer1() */
/*-----*/
int bitracer1() {
    int j, i, k, cont, cond1, cond2, cond3, objetos_processados;
    double n, dist_imagem, dist_focal, dist_luz, dist_imagem1, espessura, a, cos_esp, cond, fator, rc, di,
           mod_Lij, mod_Lkj, dk, cos_angi, cos_angk, cos_angik, ang_i, ang_k, ang_ik, ang_aux1, ang_ref,
           ang_aux2, d, TR, aux;
    PONTO3D Lij, Lji, Lkj, Ljk;
    TIPO_LUZ_SECUNDARIA * ult, * ptr;
    TIPO_BUFFER_LUZ x;
    TIPO_OBSTACULO * lista_sombras, * ults, * ptrs;

    secundarias = NULL;
    for (j = 0; j < nfontes; j++) {
        /* Gerar buffer_luz para a fonte j. */
        inicializacao_buffer_luz();
        for (i = 0; i < nobjetos; i++) {
            Lij = vetor_dif(fontes[j].posicao, objetos[i].obj.esfera->centro);
            mod_Lij = modulo_vetor3D(Lij);
            x.distancia = mod_Lij - objetos[i].obj.esfera->raio;
            x.indice = i;
            insere_buffer_luz(x, i);
        }
        /* Percorrer buffer_luz[] processando as esferas transparentes */
        i = 0;
        for (cont = 0; cont < num_objetos_transparentes; cont++) {
            if (cont == 0)
                while (objetos[buffer_luz[i].indice].coeficientes_face.kt == 0.0) {
                    i += 1;
                }
            else {
                i += 1;
                while (objetos[buffer_luz[i].indice].coeficientes_face.kt == 0.0)
                    i += 1;
            }
            /* Determinar dist_luz e dist_focal */
            dist_luz = buffer_luz[i].distancia;
            espessura = 2*objetos[buffer_luz[i].indice].obj.esfera->raio;
            n = objetos[buffer_luz[i].indice].coeficientes_face.n;
            a = n - 1;
            if (a != 0) {
                dist_imagem1 = -espessura + (n*objetos[buffer_luz[i].indice].obj.esfera->raio)/a;
                dist_focal = (objetos[buffer_luz[i].indice].obj.esfera->raio*dist_imagem1) / (dist_imagem1*a +
                    n*objetos[buffer_luz[i].indice].obj.esfera->raio);
                /* Determinar co-seno do angulo de espalhamento e dist_imagem */
                cos_angi = pow(mod_Lij,2) - pow(objetos[buffer_luz[i].indice].obj.esfera->raio,2);
                cos_angi = sqrt(fabs(cos_angi));
                cos_angi /= mod_Lij;
                ang_i = acos(cos_angi);
                ang_ref = asin (1/n); /*  $\delta = \text{sen}^{-1} (1/n)$  */
                ang_aux1 = 2*ang_ref + ang_i - PI/2; /*  $\phi = 2*\delta + \alpha - 90$  */
            }
        }
    }
}

```

```

d = objetos[buffer_luz[i].indice].obj.esfera->raio * sin(ang_aux1); /*d = Re*sen(φ)*/
cond = dist_luz - dist_focal;
cond = fabs(cond);
if (cond <= ZERO) { /*fonte secundária cilíndrica*/
    cos_esp = 1; /*ang_espalhamento = 0 grau*/
    dist_imagem = 0.0;
    rc = d; }
else if (dist_luz > dist_focal) { /*fonte secundária real*/
    Lij = vetor_dif(fontes[j].posicao, objetos[buffer_luz[i].indice].obj.esfera->centro);
    mod_Lij = modulo_vetor3D(Lij);
    di = sqrt(fabs(pow(mod_Lij,2) - pow(objetos[buffer_luz[i].indice].obj.esfera->raio,2)));
    if ((dist_luz*a - objetos[buffer_luz[i].indice].obj.esfera->raio) == 0)
        continue;
    dist_imagem1 = -espessura + (n*dist_luz*objetos[buffer_luz[i].indice].obj.esfera->raio)/
        (dist_luz*a - objetos[buffer_luz[i].indice].obj.esfera->raio);
    dist_imagem = dist_imagem1*objetos[buffer_luz[i].indice].obj.esfera->raio/
        (dist_imagem1*a + n*objetos[buffer_luz[i].indice].obj.esfera->raio);
    cos_esp = 1 / sqrt(1+pow(d/(dist_imagem + objetos[buffer_luz[i].indice].obj.esfera->raio -
        objetos[buffer_luz[i].indice].obj.esfera->raio * cos(ang_aux1)),2)); }
else { /*fonte secundária virtual*/
    ang_aux2 = 3*PI/2 - ang_i - 2*ang_ref;
    aux = fabs(i - objetos[buffer_luz[i].indice].obj.esfera->raio);
    cos_esp = 1 / sqrt(1 + pow((objetos[buffer_luz[i].indice].obj.esfera->raio * sin(ang_aux2)) /
        (aux - objetos[buffer_luz[i].indice].obj.esfera->raio * cos(ang_aux2)),2)); }
/*Teste para identificar obstáculos*/
cond1 = 1; /*esta atribuição é feita para o caso de i==0*/
cond2 = 0; /*esta atribuição é feita para o caso de i==0*/
cond3 = 0; /*esta atribuição é feita para o caso de i==0*/
objetos_processados = 0;
lista_sombras = NULL;
for (k = 0; k < i; k++) {
    objetos_processados++;
    cond2 = 0;
    Lkj = vetor_dif(fontes[j].posicao, objetos[buffer_luz[k].indice].obj.esfera->centro);
    mod_Lkj = modulo_vetor3D(Lkj);
    dk = sqrt(fabs(pow(mod_Lkj,2) - pow(objetos[buffer_luz[k].indice].obj.esfera->raio,2)));
    cond1 = dk < di;
    if (cond1) { /*esfera k satisfaz 1º teste de visibilidade*/
        Ljk = vetor_dif(objetos[buffer_luz[k].indice].obj.esfera->centro, fontes[j].posicao);
        Lji = vetor_dif(objetos[buffer_luz[i].indice].obj.esfera->centro, fontes[j].posicao);
        cos_angk = pow(mod_Lkj,2) - pow(objetos[buffer_luz[k].indice].obj.esfera->raio,2);
        cos_angk = sqrt(fabs(cos_angk));
        cos_angk /= mod_Lkj;
        ang_k = acos(cos_angk);
        normaliza (&Ljk);
        normaliza (&Lji);
        cos_angik = produto_escalar(Ljk, Lji);
        ang_ik = acos(cos_angik);
        /*teste*/
        /*Se (ang_i + ang_k > ang_ik) então esfera k é obstáculo */
        cond2 = (ang_i + ang_k) >= ang_ik;
        if (!cond2) /*teste 2 falhou => pegar próxima esfera k*/
            continue;
        else { /*esfera k é obstáculo => verificar se esfera k obstrui totalmente esfera i*/
            /*Se (abs(ang_k - ang_i) >= ang_ik) então esfera k obstrui total*/
            cond3 = fabs(ang_k - ang_i) >= ang_ik;
            if (fabs(ang_i - ang_ik) <= ZERO) /*ang_i == ang_ik*/
                cond3 = fabs(ang_k) - fabs(ang_i) >= ZERO;
            if (cond3) /*k obstrui totalmente i => não criar a fonte 2a. Lji*/

```

```

    break;
else /*insere k em lista_sombras*/
    ptrs = (TIPO_OBSTACULO *) malloc(sizeof(TIPO_OBSTACULO));
    ptrs->indice = buffer_luz[k].indice;
    ptrs->prox = NULL;
    if (lista_sombras == NULL) {
        lista_sombras = ptrs;
        ults = ptrs; }
    else {
        ults->prox = ptrs;
        ults = ptrs; } } }
else /*teste 1 falhou pela primeira vez => nenhuma esfera seguinte pode ser obstáculo*/
    break; }
if (cond3) /*Próxima esfera transparente*/
    continue;
if (!(cond1)||((cond1)&&(objetos_processados == nobjetos -1))) { /*criar fonte secundária*/
    ptr = (TIPO_LUZ_SECUNDARIA *) malloc(sizeof(TIPO_LUZ_SECUNDARIA));
    ptr->prox = NULL;
    if (secundarias == NULL) {
        secundarias = ptr;
        ult = ptr;}
    else {
        ult->prox = ptr;
        ult = ptr; }
    fator = -(dist_imagem + dist_luz + espessura)/mod_Lij;
    ptr->posicao.x = fontes[j].posicao.x + fator*Lij.x;
    ptr->posicao.y = fontes[j].posicao.y + fator*Lij.y;
    ptr->posicao.z = fontes[j].posicao.z + fator*Lij.z;
    ptr->cos_espalhamento = cos_esp;
    ptr->objeto_origem = buffer_luz[i].indice;
    ptr->fonte_origem = j;
    ptr->lista_sombras = lista_sombras;
    TR = sqrt(2 * pow(objetos[buffer_luz[i].indice].obj.esfera->raio, 2) - 2 *
        pow(objetos[buffer_luz[i].indice].obj.esfera->raio, 2) * cos(PI - 2 * ang_ref));
    ptr->intensidade = 1 / exp((1 - objetos[buffer_luz[i].indice].coeficientes_face.kt) * ((espessura +
        TR) / 2)); }
if ((cond1))
    for (k = i+1; k < nobjetos; k++) {
        cond1 = cond2 = 0;
        objetos_processados++;
        Lkj = vetor_dif(fontes[j].posicao, objetos[buffer_luz[k].indice].obj.esfera->centro);
        mod_Lkj = modulo_vetor3D(Lkj);
        dk = mod_Lkj - objetos[buffer_luz[k].indice].obj.esfera->raio;
        cond1 = dk < di;
        if (cond1) { /*k satisfaz 1o teste de visibilidade*/
            Ljk = vetor_dif(objetos[buffer_luz[k].indice].obj.esfera->centro, fontes[j].posicao);
            Lji = vetor_dif(objetos[buffer_luz[i].indice].obj.esfera->centro, fontes[j].posicao);
            cos_angk = pow(mod_Lkj, 2) - pow(objetos[buffer_luz[k].indice].obj.esfera->raio, 2);
            cos_angk = sqrt(fabs(cos_angk));
            cos_angk /= mod_Lkj;
            ang_k = acos(cos_angk);
            cos_angik = produto_escalar(Ljk, Lji)/(mod_Lkj*mod_Lij);
            ang_ik = acos(cos_angik);
            cond2 = (ang_i + ang_k) >= ang_ik;
            if (!cond2) /*teste 2 falhou => pegar próxima esfera k*/
                continue;
            else /*esfera k é obstáculo => verificar se esfera k obstrui totalmente esfera i*/
                cond3 = fabs(ang_k - ang_i) >= ang_ik;
                if (fabs(ang_i - ang_ik) <= ZERO) /*ang_i == ang_ik*/

```

```

    cond3 = fabs(ang_k - ang_i) >= ZERO;
    if (cond3) /*k obstrui totalmente i => não criar a fonte 2a. Lji*/
        break;
    else /*insere k em lista_sombras*/
        ptrs = (TIPO_OBSTACULO *) malloc(sizeof(TIPO_OBSTACULO));
        ptrs->indice = buffer_luz[k].indice;
        ptrs->prox = NULL;
        if (lista_sombras == NULL) {
            lista_sombras = ptrs;
            ults = ptrs; }
        else {
            ults->prox = ptrs;
            ults = ptrs; } } }
    else /*teste l falhou pela 1a. vez*/
        break; }
    if (cond3) /*Próxima esfera transparente*/
        continue;
    if (!cond1) /*criar fonte secundária*/
        ptr = (TIPO_LUZ_SECUNDARIA *) malloc(sizeof(TIPO_LUZ_SECUNDARIA));
        ptr->prox = NULL;
        if (secundarias == NULL) {
            secundarias = ptr;
            ult = ptr; }
        else {
            ult->prox = ptr;
            ult = ptr; }
        normaliza (&Lij);
        fator = -(dist_imagem + dist_luz + espessura);
        ptr->posicao.x = fontes[j].posicao.x + fator*Lij.x;
        ptr->posicao.y = fontes[j].posicao.y + fator*Lij.y;
        ptr->posicao.z = fontes[j].posicao.z + fator*Lij.z;
        ptr->cos_espalhamento = cos_esp;
        ptr->objeto_origem = buffer_luz[i].indice;
        ptr->fonte_origem = j;
        ptr->lista_sombras = lista_sombras;
        TR = sqrt(2 * pow(objetos[buffer_luz[i].indice].obj.esfera->raio, 2) - 2 *
            pow(objetos[buffer_luz[i].indice].obj.esfera->raio, 2) * cos(PI - 2 * ang_ref));
        ptr->intensidade = 1 / exp((1 - objetos[buffer_luz[i].indice].coeficientes_face.kt) * ((espessura
            + TR) / 2)); } } } } }

/*-----*/
/* Função: bitracer2() */
/*-----*/
void bitracer2() {
    int linha, pixel;
    PONTO3D raio, vet_visada;
    double dx_por_pixel, dy_por_pixel;
    PONTO3D vi, vj, raio_inicial;

    vet_visada = vetor_dif(camera.alvo, camera.observador);
    normaliza(&vet_visada);
    normaliza(&camera.vet_up);
    vi = produto_vetorial (vet_visada, camera.vet_up);
    vj = produto_vetorial (vi, vet_visada);
    dx_por_pixel = janela.dx/tela.dx;
    dy_por_pixel = janela.dy/tela.dy;
    vi.x *= dx_por_pixel;
    vi.y *= dx_por_pixel;
    vi.z *= dx_por_pixel;

```

```

vj.x *= dy_por_pixel;
vj.y *= dy_por_pixel;
vj.z *= dy_por_pixel;
raio_inicial.x = camera.alvo.x - camera.observador.x;
raio_inicial.y = camera.alvo.y - camera.observador.y;
raio_inicial.z = camera.alvo.z - camera.observador.z;
raio_inicial.x += tela.dy/2 * vj.x - tela.dx/2 * vi.x;
raio_inicial.y += tela.dy/2 * vj.y - tela.dx/2 * vi.y;
raio_inicial.z += tela.dy/2 * vj.z - tela.dx/2 * vi.z;
printf("Tela: %d x %d\n", tela.dx, tela.dy);
for (linha = 0; linha < tela.dy; linha++) {
    for (pixel = 0; pixel < tela.dx; pixel++) {
        raio.x = raio_inicial.x + pixel*vi.x - linha*vj.x;
        raio.y = raio_inicial.y + pixel*vi.y - linha*vj.y;
        raio.z = raio_inicial.z + pixel*vi.z - linha*vj.z;
        normaliza(&raio);
        buffer_imagem[pixel] = RTtrace(-1, camera.observador, raio, 1); }
    grava_linha(formato); /*gravacao no arquivo de saida da linha y*/
    printf("Linha %d finalizada\n", linha); } }

/*-----*/
/* Função: RTshade() */
/*-----*/
COR RTshade (obj_intersec, incidente, interseccao, normal, profundidade)
int obj_intersec;
PONTO3D incidente, interseccao, normal;
int profundidade; {

    PONTO3D rRaio, tRaio, V, N, L, R, S, P1, T1, T2, P2;
    COR cor, rCor, tCor, cor_amb, cor_dif, cor_esp;
    OBJETO objeto;
    SUPERFICIE superficie;
    int i, j, refracao, intersec_objeto, intersec_sombra, fonte_secundaria_real;
    double prodescalar_NL, prodescalar_RV, aux, potencia, modulo, kn, kf, mod1, mod2, fator, s, mod_S,
        distancia, k, prodescalar_NS, prodescalar_NV, d1, d2;
    TIPO_LUZ_SECUNDARIA * ptr;
    TIPO_OBSTACULO * sombra;

    cor.r = cor.g = cor.b = 0.0;
    objeto = objetos[obj_intersec];
    if (objeto.tipo == CONST_ESFERA)
        superficie = objeto.coeficientes_face;
    if (objeto.tipo == CONST_POLIEDRO)
        superficie = objeto.coeficientes_face;
    /*inicializacao das componentes difusa e especular*/
    cor_dif.r = cor_dif.g = cor_dif.b = 0.0;
    cor_esp.r = cor_esp.g = cor_esp.b = 0.0;
    /*pgs 365 a 367 de Rogers - 1985*/
    V.x = -1 * incidente.x;
    V.y = -1 * incidente.y;
    V.z = -1 * incidente.z;
    N.x = normal.x;
    N.y = normal.y;
    N.z = normal.z;
    /* calculo do raio refletido */
    prodescalar_NV = produto_escalar (V, N);
    if ( prodescalar_NV > 0.0) {
        if (superficie.n != 0.0)
            kn = superficie.n; }
    else

```

```

if (superficie.n != 0.0) {
    kn = 1 / superficie.n;
    N.x *= -1;
    N.y *= -1;
    N.z *= -1; }
k = 2*prodescalar_NV;
rRaio.x = k*N.x + incidente.x;
rRaio.y = k*N.y + incidente.y;
rRaio.z = k*N.z + incidente.z;
normaliza (&rRaio);
/*calculo do raio refratado - pag 367 de Rogers 1985*/
if (superficie.kt > 0.0) {
    refracao = 1;
    if (superficie.n != 0.0) {
        kf = 1 -kn*kn*(1-prodescalar_NV*prodescalar_NV);
        if (kf < 0.0) { /*reflexao interna*/
            tRaio.x = rRaio.x;
            tRaio.y = rRaio.y;
            tRaio.z = rRaio.z; }
        else {
            tRaio.x = (kf*prodescalar_NV - sqrt(kf))*N.x - kf*V.x;
            tRaio.y = (kf*prodescalar_NV - sqrt(kf))*N.y - kf*V.y;
            tRaio.z = (kf*prodescalar_NV - sqrt(kf))*N.z - kf*V.z;
            normaliza(&tRaio); } } }
/*calculo das componentes difusa e especular*/
for (i = 0; i < nfontes; i++) { /*fontes primarias*/
    L.x = fontes[i].posicao.x - interseccao.x;
    L.y = fontes[i].posicao.y - interseccao.y;
    L.z = fontes[i].posicao.z - interseccao.z;
    normaliza(&L);
    prodescalar_NL = produto_escalar(N, L);
    if (prodescalar_NL > 0.0) {
        /*calculo do produto escalar R.V: pgs 730 a 731 do Foley-1993 */
        aux = 2 * prodescalar_NL;
        R.x = aux*N.x - L.x;
        R.y = aux*N.y - L.y;
        R.z = aux*N.z - L.z;
        prodescalar_RV = produto_escalar(R, V);
        if (prodescalar_RV <= 0.0)
            potencia = 0.0;
        else
            potencia = pow(prodescalar_RV, superficie.distrib_espec_Phong);
        s = 1;
        for (j = 0; j < nobjetos; j++)
            if (calcula_interseccao (interseccao, L, j) > 0.0)
                if (objetos[j].coeficientes_face.kt == 0) {
                    s = 0.0;
                    break; }
                else
                    s *= objetos[j].coeficientes_face.kt;
        cor_dif.r += s*fontes[i].intensidade * fontes[i].cor.r * superficie.cor.r * prodescalar_NL;
        cor_dif.g += s*fontes[i].intensidade * fontes[i].cor.g * superficie.cor.g * prodescalar_NL;
        cor_dif.b += s*fontes[i].intensidade * fontes[i].cor.b * superficie.cor.b * prodescalar_NL;
        cor_esp.r += s*fontes[i].intensidade * fontes[i].cor.r * potencia;
        cor_esp.g += s*fontes[i].intensidade * fontes[i].cor.g * potencia;
        cor_esp.b += s*fontes[i].intensidade * fontes[i].cor.b * potencia; } }
ptr = secundarias; /*fontes secundarias*/
while ( ptr != NULL) {
    if (pertence(interseccao, ptr, &S)) {
        fator = 1;

```

```

mod_S = modulo_vetor3D(S);
distancia = MAXDOUBLE;
normaliza(&S);
if (produto_escalar (vetor_dif(ptr->posicao, objetos[ptr->objeto_origem].obj.esfera->centro), S) >=
    0.0)
    fonte_secundaria_real = 0;
else
    fonte_secundaria_real = 1;
for (i = 0; i < nobjetos; i++) {
    distancia = calcula_interseccao (interseccao, S, i);
    if ((distancia < MAXDOUBLE) && (distancia < mod_S) && (distancia > 1E-5))
        if ((i != ptr->objeto_origem)||((i == ptr->objeto_origem)&&(!fonte_secundaria_real)))
            fator *= objetos[i].coeficientes_face.kt;
        if (!fator)
            break; }
if (fator)
    if (ptr->lista_sombras == NULL) { /*plena*/
        prodescalar_NS = produto_escalar(N, S);
    if (prodescalar_NS > 0) {
        aux = 2 * prodescalar_NS;
        R.x = aux*N.x - S.x;
        R.y = aux*N.y - S.y;
        R.z = aux*N.z - S.z;
        prodescalar_RV = produto_escalar(R, V);
        if (prodescalar_RV <= ZERO)
            potencia = 0.0;
        else {
            potencia = pow(prodescalar_RV, superficie.distrib_espec_Phong);
            conta_auxiliar++; }
        cor_dif.r += fator*(ptr->intensidade) * (fontes[ptr->fonte_origem].intensidade) *
            (fontes[ptr->fonte_origem].cor.r) * (superficie.cor.r) * prodescalar_NS;
        cor_dif.g += fator*(ptr->intensidade) * (fontes[ptr->fonte_origem].intensidade) *
            (fontes[ptr->fonte_origem].cor.g) * (superficie.cor.g) * prodescalar_NS;
        cor_dif.b += fator*(ptr->intensidade) * (fontes[ptr->fonte_origem].intensidade) *
            (fontes[ptr->fonte_origem].cor.b) * (superficie.cor.r) * prodescalar_NS;
        cor_esp.r += fator*(ptr->intensidade) * (fontes[ptr->fonte_origem].intensidade) *
            (fontes[ptr->fonte_origem].cor.r) * potencia;
        cor_esp.g += fator*(ptr->intensidade) * (fontes[ptr->fonte_origem].intensidade) *
            (fontes[ptr->fonte_origem].cor.g) * potencia;
        cor_esp.b += fator*(ptr->intensidade) * (fontes[ptr->fonte_origem].intensidade) *
            (fontes[ptr->fonte_origem].cor.b) * potencia; } }
    else { /*nao plena*/
        sombra = ptr->lista_sombras;
        if (sombra != NULL) {
            /*calcula P1: interseccao de S com ptr->objeto_origem*/
            normaliza(&S);
            d1 = calcula_interseccao(interseccao, S, ptr->objeto_origem);
            P1.x = interseccao.x + d1*S.x;
            P1.y = interseccao.y + d1*S.y;
            P1.z = interseccao.z + d1*S.z;
            /*calcula T1: raio refratado dentro de ptr->objeto_origem*/
            N = vetor_dif(P1, objetos[ptr->objeto_origem].obj.esfera->centro);
            V.x = -1 * S.x;
            V.y = -1 * S.y;
            V.z = -1 * S.z;
            prodescalar_NV = produto_escalar(N, V);
            if (prodescalar_NV > 0.0)
                kn = objetos[ptr->objeto_origem].coeficientes_face.n;
            else

```

```

    kn = 1/objetos[ptr->objeto_origem].coeficientes_face.n;
    kf = 1 -kn*kn*(1-prodescalar_NV*prodescalar_NV);
    if (kf < 0.0) { /*reflexao interna*/
        k = -2*produto_escalar(S, N);
        T1.x = k*N.x + S.x;
        T1.y = k*N.y + S.y;
        T1.z = k*N.z + S.z; }
    else {
        T1.x = (kf*prodescalar_NV - sqrt(kf))*N.x - kf*V.x;
        T1.y = (kf*prodescalar_NV - sqrt(kf))*N.y - kf*V.y;
        T1.z = (kf*prodescalar_NV - sqrt(kf))*N.z - kf*V.z;
        normaliza(&T1); }
    /*calcula P2:interseccao de T1 com objeto_origem*/
    d2 = calcula_interseccao(P1, T1, ptr->objeto_origem);
    P2.x = P1.x + d2*T1.x;
    P2.y = P1.y + d2*T1.y;
    P2.z = P1.z + d2*T1.z;
    /*calcula T2: refratado a partir de P2 para o meio*/
    N = vetor_dif(P2, objetos[ptr->objeto_origem].obj.csfera->centro);
    V.x = -1 * T1.x;
    V.y = -1 * T1.y;
    V.z = -1 * T1.z;
    kn = 1/objetos[ptr->objeto_origem].coeficientes_face.n;
    kf = 1 -kn*kn*(1-prodescalar_NV*prodescalar_NV);
    if (kf < 0.0) { /*reflexao interna*/
        k = -2*produto_escalar(T1, N);
        T2.x = k*N.x + T1.x;
        T2.y = k*N.y + T1.y;
        T2.z = k*N.z + T1.z; }
    else {
        T2.x = (kf*prodescalar_NV - sqrt(kf))*N.x - kf*V.x;
        T2.y = (kf*prodescalar_NV - sqrt(kf))*N.y - kf*V.y;
        T2.z = (kf*prodescalar_NV - sqrt(kf))*N.z - kf*V.z;
        normaliza(&T2); }
    /*verifica se T2 intersecciona alguma sombra*/
    intersec_sombra = 0;
    while ((sombra != NULL) &&(!intersec_sombra)) {
        intersec_sombra = (calcula_interseccao (P2, T2, sombra->indice) > 0.0);
        sombra = sombra->prox; }
    if (!intersec_sombra) {
        V.x = -1 * incidente.x;
        V.y = -1 * incidente.y;
        V.z = -1 * incidente.z;
        N.x = normal.x;
        N.y = normal.y;
        N.z = normal.z;
        prodescalar_NS = produto_escalar(N, S);
        if (prodescalar_NS > 0) {
            conta_auxiliar++;
            aux = 2 * prodescalar_NS;
            R.x = aux*N.x - S.x;
            R.y = aux*N.y - S.y;
            R.z = aux*N.z - S.z;
            prodescalar_RV = produto_escalar(R, V);
            if (prodescalar_RV <= ZERO)
                potencia = 0.0;
            else
                potencia = pow(prodescalar_RV, superficie.distrib_espec_Phong);
            cor_dif.r += fator*(ptr->intensidade) * (fontes[ptr->fonte_origem].intensidade) *
                (fontes[ptr->fonte_origem].cor.r) * (superficie.cor.r) * prodescalar_NS;

```

```

cor_dif.g += fator * (ptr->intensidade) * (fontes[ptr->fonte_origem].intensidade) *
    fontes[ptr->fonte_origem].cor.g) * (superficie.cor.g) * prodescalar_NS;
cor_dif.b += fator * (ptr->intensidade) * (fontes[ptr->fonte_origem].intensidade) *
    (fontes[ptr->fonte_origem].cor.b) * (superficie.cor.r) * prodescalar_NS;
cor_esp.r += fator * (ptr->intensidade) * (fontes[ptr->fonte_origem].intensidade) *
    (fontes[ptr->fonte_origem].cor.r) * potencia;
cor_esp.g += fator * (ptr->intensidade) * (fontes[ptr->fonte_origem].intensidade) *
    (fontes[ptr->fonte_origem].cor.g) * potencia;
cor_esp.b += fator * (ptr->intensidade) * (fontes[ptr->fonte_origem].intensidade) *
    (fontes[ptr->fonte_origem].cor.b) * potencia; } } } } }

ptr = ptr->prox; }
/* reflexao */
rCor.r = rCor.g = rCor.b = 0.0;
if (superficie.ks > 0.0) { /*ver eq 5-16 pag 366 de Rogers-1985*/
    rCor = RTtrace (obj_intersec, interseccao, rRaio, profundidade + 1); }
/* refracao */
tCor.r = tCor.g = tCor.b = 0.0;
if ((superficie.kt > 0.0)&&(refracao == 1)) {
    tCor = RTtrace_refratado(obj_intersec, interseccao, tRaio, profundidade + 1); }
/* termo ambiente */
cor_amb.r = cor_ambiente.r*superficie.cor.r;
cor_amb.g = cor_ambiente.g*superficie.cor.g;
cor_amb.b = cor_ambiente.b*superficie.cor.b;
/* calculo da cor final */
cor.r += superficie.ka * cor_amb.r + superficie.kd * cor_dif.r + superficie.ks * cor_esp.r + superficie.ks
    * rCor.r + superficie.kt * tCor.r;
cor.g += superficie.ka * cor_amb.g +superficie.kd * cor_dif.g + superficie.ks * cor_esp.g +
    superficie.ks * rCor.g + superficie.kt * tCor.g;
cor.b += superficie.ka * cor_amb.b + superficie.kd * cor_dif.b + superficie.ks * cor_esp.b +
    superficie.ks * rCor.b + superficie.kt * tCor.b;
return (cor); }

```

## Bibliografia

- [AMA 84] AMANATIDES, J. Ray Tracing with Cones. **Computer Graphics**, New York, v.18, n.3, p.129-135, July 1984. Trabalho apresentado na SIGGRAPH, 11., 1984, Minneapolis, US.
- [AMA 87] AMANATIDES, J.; WOO, A. A Fast Voxel Traversal Algorithm for Ray Tracing. In: EUROPEAN COMPUTER GRAPHICS CONFERENCE AND EXHIBITION, EUROGRAPHICS, 1987. **Proceedings...** [S.l.:s.n.], 1987. p.3-10.
- [APP 68] APPLE, A. Some Techniques for Shading Machine Rendering of Solids. **Spring Joint Computer Conference**, [S.l.], v.32, p.37-45, 1968.
- [ARV 86] ARVO, J. Backward Ray Tracing. **Computer Graphics**, New York, v.20, n.4, July 1986. Trabalho apresentado na SIGGRAPH, 11., 1986, Dallas, US.
- [ARV 87] ARVO, James; KIRK, David. Fast Ray Tracing by Ray Classification. **Computer Graphics**, New York, v.21, n.4, p.55-64, 1987.
- [ARV 91] ARVO, J. **Graphics Gems II**. San Diego: Academic Press, 1991. 644 p.
- [ASS 96] ASSIS, Gilda A. Aplicação da Técnica de Ray Tracing Bidirecional à Síntese de Objetos Transparentes. In: SEMANA ACADÊMICA DO CPGCC, 1., 1996, Porto Alegre. **Anais...** Porto Alegre:CPGCC da UFRGS, 1996. p.107-110.
- [BOU 85] BOUVILLE, C. Bounding Ellipsoids for Ray-Fractal Intersection. **Computer Graphics**, New York, v.19, n.3, p.45-52, July 1985. Trabalho apresentado na SIGGRAPH, 11., 1985, San Francisco, US.
- [BUI 75] BUI-TUONG, Phong. Illumination for Computer Generated Pictures. **Communications of the ACM**, New York, v.18, n.6, p.311-317, 1975.
- [COO 84] COOK, R. L.; PORTER, T.; CARPENTER, L. Distributed Ray Tracing. **Computer Graphics**, New York, v.18, n.3, p.137-145, July 1984. Trabalho apresentado na SIGGRAPH, 11., 1984, Minneapolis, UC July 1986.
- [COO 86] COOK, R. L. Stochastic Sampling in Computer Graphics. **Communications of the ACM Transactions on Graphics**, New York, v.5, n.1, p.51-72, 1986.
- [CUN 88] CUNHA, Rudnei D. **Algoritmos para a Síntese de Imagens através do Método "Ray-Tracing"**. Porto Alegre: CPGCC-UFRGS, 1988. Dissertação de Mestrado.
- [DIA 94] DIAS, M. L. Ray Tracing Interference Color: Visualizing Newton's Rings. **IEEE Computer Graphics and Applications**, Los Alamitos, v.14, n.3, p.17-20, 1994.
- [DIP 84] DIPPÉ, M.; SWENSEN, J. An Adaptive Subdivision Algorithm and Parallel Architecture for Realistic Image Synthesis. **Computer Graphics**, New York, v.18, n.3, p. 149-158, July 1984. Trabalho apresentado na SIGGRAPH, 11., 1984, Minneapolis, US.

- [DIP 85] DIPPÉ, M.; WOLD, E. H. Antialiasing through Stochastic Sampling. **Computer Graphics**, New York, v.19, n.3, p.69-78, July 1985. Trabalho apresentado na SIGGRAPH, 11., 1985, San Francisco, US.
- [FOL 90] FOLEY, James D. et al. **Computer Graphics: Principles and Practice**. 2.ed. New York: Addison Wesley, 1990, 1174p.
- [FUJ 85] FUJIMURA, K.; KUNII, T. L. A Hierarchical Space Indexing Method. In: **COMPUTER GRAPHICS TOKYO CONFERENCE, 1985. Proceedings...**, [S.l.i.s.n.], 1985. p.21-34.
- [GAR 93] GARGANTINI, I.; ATKINSON, H. H. Ray Tracing an Octree: Numerical Evaluation of the First Intersection. **Computer Graphics Forum**, Amsterdam, v.12, n.4, p.199-210, 1993.
- [GER 92] GERVAUTZ, Michael. Consistent Schemes for Addressing Surfaces when Ray Tracing Transparent CSG Objects. **Computer Graphics Forum**, Amsterdam, v.11, n.4, p.203-211, 1992.
- [GLA 84] GLASSNER, A. S. Space Subdivision for Fast Ray Tracing. **IEEE Computer Graphics & Applications**, Los Alamitos, v.4, n.10, p.15-22, 1984.
- [GOL 87] GOLDSMITH, J.; SALMON, J. Automatic Creation of Objects Hierarchies for Ray Tracing. **IEEE Computer Graphics & Applications**, Los Alamitos, v.7, n.5, p.14-20, 1987.
- [GOM 90] GOMES, Jonas; VELHO, Luís C. Conceitos Básicos de Computação Gráfica. In: **ESCOLA DE COMPUTAÇÃO, 7.**, 1990. **Anais...** São Paulo: IME-USP, 1990, 311p.
- [GRE 79] GREENBERG, D.; KAY, D. S. Transparency for Computer Synthesized Images. **Computer Graphics**, New York, v.13, n.2, p.158-164, August 1979. Trabalho apresentado na SIGGRAPH, 11., 1979, Chicago, US.
- [HAI 89] HAINES, E. A.; GREENBERG, D. P. Essential Ray Tracing Algorithms. In: GLASSNER, A. S. (Ed). **Introduction to Ray Tracing**. Londres: Academic Press, 1989. p.33-77.
- [HAL 83] HALL, R. A.; GREENBERG, D. P. A Testbed for Realistic Image Synthesis. **IEEE Computer Graphics & Applications**, Los Alamitos, v.3, n.8, p.10-20, 1983.
- [HEA 90] HEAL, B. W. Node Partitioning in an Octree Display Pipeline. **Computer Graphics Forum**, Amsterdam, v.9, n.3, p.205-211, 1990.
- [HEC 84] HECKBERT, P. S.; HANRAHAN, P. Beam Tracing Polygonal Objects. **Computer Graphics**. New York, v.18, n.3, p.119-127, July 1984. Trabalho apresentado na SIGGRAPH, 11., 1984, Minneapolis, US.
- [JEN 76] JENKINS, F. A.; WHITE, H. E. **Fundamentals of Optics**. [S.l.]:McGraw-Hill Inc., 1976.
- [KAJ 85] KAJIYA, J. T. Anisotropic Reflection Models. **Computer Graphics**, New York, v.19, n.3, p.15-21, July 1985. Trabalho apresentado na SIGGRAPH, 11., 1985, San Francisco, US.
- [KAY 86] KAY, T. L.; KAJIYA, J. T. Ray Tracing Complex Scenes. **Computer Graphics**, New York, v.20, n.4, p.269-278, July 1986. Trabalho apresentado na SIGGRAPH, 11., 1986, Dallas, US.

- [KIM 92] KIM, H. Y. Rastreamento Bidirecional: Um Algoritmo para Gerar as Sombras dos Objetos Transparentes. In: SEMINÁRIO INTEGRADO DE SOFTWARE E HARDWARE, 19., 1992, Rio de Janeiro. **Anais...** Rio de Janeiro:SBC, 1992. p.188-199.
- [KOK 92] KOK, Arjan J. F.; JANSEN, Frederik W. Adaptive Sampling of Area Light Sources in Ray Tracing Including Diffuse Interreflection. **Computer Graphics Forum**, Amsterdam, v.11, n.3, p.289-298, Sept. 1992. Trabalho apresentado na EUROGRAPHICS, 13., 1992, Cambridge.
- [KOR 83] KOREIN, J. U.; BADLER, N. I. Temporal Anti-Aliasing in Computer Generated Animation. **Computer Graphics**, New York, v.17, n.3, p.377-388, July 1983. Trabalho apresentado na SIGGRAPH, 11., 1983, Detroit, US.
- [KRU 92] KRUG, Ricardo C. **Um Estudo sobre Transparência**: trabalho individual. Porto Alegre: CPGCC-UFRGS, 1992. (TI-265).
- [KRU 94] KRUG, Ricardo C. **Aplicação da Refração Luminosa em Computação Gráfica**. Porto Alegre: CPGCC-UFRGS, 1994. Dissertação de Mestrado.
- [LAF 94] LAFORTUNE, Eric P.; WILLEMS, Yves D. A Theoretical Framework for Physically Based Rendering. **Computer Graphics Forum**, Amsterdam, v.13, n.2, p.97-107, 1994.
- [LEE 85] LEE, M. E.; REDNER, R. A.; USELTON, S. P. Statistically Optimized Sampling for Distributed Ray Tracing. **Computer Graphics**, New York, v.19, n.3, p.61-65, July 1985. Trabalho apresentado na SIGGRAPH, 11., 1985, San Francisco, US.
- [LEV 81] LEVINE, M. W.; SHEFNER, J. M. Fundamentals of Sensation and Perception. In: CASSEL, G. H. et al. **The Eye**. [S.l.]: Addison-Wesley, 1981. p.65-83.
- [McN 92] McNEILL, M. D. J. et al. Performance of Space Subdivision Techniques in Ray Tracing. **Computer Graphics Forum**, Amsterdam, v.11, n.4, p.213-220, 1992.
- [MEY 90] MEYER, U. Hemi-Cube Ray-Tracing: A Method for Generating Soft Shadows. In: EUROPEAN COMPUTER GRAPHICS CONFERENCE AND EXHIBITION, EUROGRAPHICS, 1990. **Proceedings...** [S.l.:s.n.], 1990. p.365-375.
- [MIT 87] MITCHELL, D. P. Generating Antialiased Images at Low Sampling Densities. **Computer Graphics**, New York, v.21, n.4, p.65-72, July 1987. Trabalho apresentado na SIGGRAPH, 11., 1987, Anaheim, US.
- [NAS 92] NASCIMENTO, Marcos E. **Estudo de Algoritmos para Geração de Sombras Projetadas na Síntese de Imagens Foto-realísticas**. Porto Alegre: CPGCC-UFRGS, 1992. Dissertação de Mestrado.
- [NEW 79] NEWMAN, William M.; SPROULL, Robert F. **Principles of Interactive Computer Graphics**. 2.ed. New York: McGraw-Hill, 1979. p. 157-164.
- [NOA 92] NOAL, Eronita A. C. **Um Estudo Sistemático dos Principais Algoritmos para Remoção de Superfícies Ocultas**:trabalho individual. Porto Alegre: CPGCC-UFRGS, 1992. (TI-283).
- [PAD 55] PADGHAM, C. A. **Optics and Sound**. London: Sir Isaac Pitman & Sons, 1955.

- [POT 82] POTMESIL, M.; CHAKRAVARTY, I. Synthetic Image Generation with a Lens and Aperture Camera Model. **Communications of the ACM Transations on Graphics**, [S.l.], v.1, n.2, p.85-108, 1982.
- [POT 83] POTMESIL, M.; CHAKRAVARTY, I. Modeling Motion Blur in Computer-Generated Images. **Computer Graphics**, New York, v.17, n.3, p. 389-399, July 1983. Trabalho apresentado na SIGGPRAPH, 11. 1983, Detroit, US.
- [RAM 84] RAMALHO, Francisco et al. **Os Fundamentos da Física 2**. São Paulo: Moderna, 1984.
- [RES 96] RESNICK, R.; HALLIDAY, D.; WALKER, J. **Fundamentos de Física 4 Ótica e Física Moderna**. [S.l.]: LTC - Livros Técnicos e Científicos, 1996.
- [ROG 85] ROGERS, David F. **Procedural Elements for Computers Graphics**. New York: McGraw-Hill, 1985.
- [RUB 80] RUBIN, S. M.; WHITTED, T. A 3-Dimensional Representation for Fast Rendering of Complex Scenes. **Computer Graphics**, New York, v.14, n.3, p. 110-116, July 1980. Trabalho apresentado na SIGGPRAPH, 11. 1980, Washington, US.
- [SAM 89] SAMET, H. Implementing Ray Tracing with Octrees and Neighbor Finding. **Computers & Graphics**, [S.l.], v.13, n.4, p.445-460, 1989.
- [SHI 87] SHINYA, M.; TAKAHASHI, T.; NAITO, S. Principles and Applications of Pencil Tracing. **Computer Graphics**, New York, v.21, n.4, p. 45-54, July 1987. Trabalho apresentado na SIGGPRAPH, 11. 1987, Anaheim, US.
- [SNY 87] SNYDER, J. M.; BARR, A. H. Ray Tracing Complex Models Containing Surface Tessellations. **Computer Graphics**, New York, v.21, n.4, p. 119-128, July 1987. Trabalho apresentado na SIGGPRAPH, 11. 1987, Anaheim, US.
- [SUN 92] SUNG, Kelvin. Area Sampling Buffer: Tracing Rays with Z-Buffer Hardware. **Computer Graphics Forum**, Amsterdam, v.11, n.3, p.299-310, Sept. 1992. Trabalho apresentado na EUROGRAPHICS, 13., 1992, Cambridge.
- [THO 86] THOMAS, S. W. Dispersive refraction in ray tracing. **The Visual Computer**, [S.l.], v.2, n.1, p.3-8, 1986.
- [TIP 84] TIPLER, Paul A. **Física**. Rio de Janeiro: Guanabara Dois, 1984. v.2.
- [TOT 85] TOTTH, D. L. On Ray Tracing Parametric Surfaces. **Computer Graphics**, New York, v.19, n.3, p.171-179, July 1985. Trabalho apresentado na SIGGRAPH, 11., 1985, San Francisco, US.
- [URB 92] URBAN, B.; HABERSTROH, T. Ray Tracing and Graphics Standards. **Computer Graphics Forum**, [S.l.], v.2, n.1, p.55-60, 1992.
- [WAT 90] WATT, Mark. Light-Water Interaction using Backward Beam Tracing. **Computer Graphics**, New York, v.24, n.4, p.377-385, August 1990. Trabalho apresentado na SIGGRAPH, 11., 1990, Dallas, US.
- [WEG 84] WEGHORST, H.; HOOPER, G.; GREENBERG, D. P. Improved Computational Methods for Ray Tracing. **Communications of the ACM Transations on Graphics**, [S.l.], v.3, n.1, p.52-69, January 1984.

- [WHI 80] WHITTED, T. An Improved Illumination Model for Shaded Display. **Communications of the ACM Transactions on Graphics**, [S.l.], v.23, n.6, p.343-349, June 1980.
- [WOL 90] WOLF, L.; KURLANDER, D. Ray Tracing with Polarization Parameters. **IEEE Computer Graphics and Applications**, Los Alamitos, v.10, n.6, p.44-55, 1990.

Informática



UFRGS

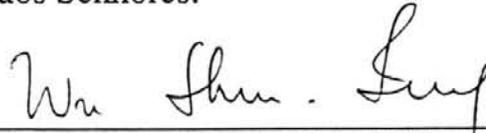
CURSO DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

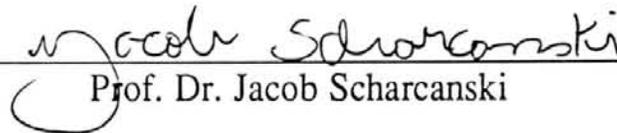
*"Aplicação da Técnica de Ray Tracing Bidirecional à Síntese de Objetos  
Transparentes"*

por

Gilda Aparecida de Assis

Dissertação apresentada aos Senhores:

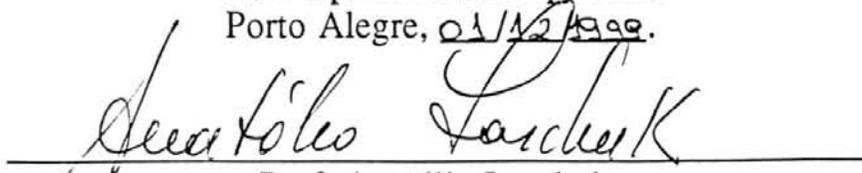
  
\_\_\_\_\_  
Profa. Dra. Wu Shin Ting (UNICAMP)

  
\_\_\_\_\_  
Prof. Dr. Jacob Scharcanski

  
\_\_\_\_\_  
Profa. Dra. Carla Maria Dal Sasso Freitas

Vista e permitida a impressão.

Porto Alegre, 01/12/1999.

  
\_\_\_\_\_  
Prof. Anatólio Laschuk,  
Orientador.

  
\_\_\_\_\_

*Profa. Carla Maria Dal Sasso Freitas*  
Coordenadora do Programa de Pós-Graduação  
em Computação - PPGC  
Instituto de Informática - UFRGS