

impa



INSTITUTO NACIONAL DE MATEMÁTICA PURA E APLICADA

Esquema híbrido para amostragem de mapas de iluminação em renderizações foto-realistas

Aldo René Zang

Orientador : Luiz Carlos Pacheco Rodrigues Velho

Co-orientador: Paulo Cezar Pinto Carvalho

Rio de Janeiro

Julho de 2009

Aldo René Zang

Esquema híbrido para amostragem de mapas de iluminação em renderizações foto-realistas

Dissertação de Mestrado

Dissertação apresentada como requisito
para obtenção do grau de Mestre em
**Matemática Aplicada - opção:
Computação Gráfica**

Orientador:

Luiz Carlos Pacheco Rodrigues Velho

Co-orientador:

Paulo Cezar Pinto Carvalho

IMPA - INSTITUTO NACIONAL DE MATEMÁTICA PURA E APLICADA

Rio de Janeiro

1 de julho de 2009

Aos meus pais Mario e Lucía, e minha esposa Roxane.

Aquí me pongo a cantar
Al compás de la vigüela;
Que el hombre que lo desvela
Una pena extraordinaria,
Como la ave solitaria
Con el cantar se consuela.

Pido a los Santos del Cielo
Que ayuden mi pensamiento;
Les pido en este momento
Que voy a cantar mi historia,
Me refresquen la memoria
Y aclaren mi entendimiento.

Vengan, santos milagrosos,
Vengan todos en mi ayuda,
Que la lengua se me añuda
Y se me turba la vista.
Pido a mi Dios que me asista
En una ocasión tan ruda.

Yo he visto muchos cantores
Con famas bien obtenidas,
Y que después de adquiridas
No las quieren sustentar:
Parece que sin largar
Se cansaron en partidas.

Cantando me he de morir,
Cantando me han de enterrar,
Y cantando he de llegar
Al pie del Eterno Padre.
Dende el vientre de mi madre
Vine a este mundo a cantar.

Que no se trabe mi lengua
Ni me falte la palabra
El cantar mi gloria labra
Y poniéndome a cantar,
Cantando me han de encontrar
Aunque la tierra se abra.

Martín Fierro

Agradecimentos

Em primeiro lugar gostaria de agradecer a minha esposa e companheira Roxane, pois sem sua paciência e força não estaria escrevendo estas linhas. Aos meus pais que sempre me ajudaram a alcançar meus objetivos. A eles dedico este trabalho.

Obrigado Sérgio por toda a ajuda com toda a parte de suporte informático, que não foi pouca. Agradeço especialmente ao meu orientador Luiz Velho pelas ótimas idéias e motivadoras conversas. Aos professores Luiz Henrique e Paulo Cezar pelas ótimas aulas ministradas durante os cursos que tive com eles. Aos amigos que sempre me deram força nos momentos difíceis, especialmente ao Jefferson, meu caro. Obrigado Ives, Leonardo, Anna, Emilio, Dalia, Thiago, Dimas e todos os colaboradores do Visgraf.

Resumo

Atualmente a renderização foto-realista de cenas sintéticas, com iluminação proveniente de mapas de iluminação extraídos de ambientes reais, é um ponto muito estudado no processo de síntese de imagens, e também com grande demanda de soluções eficientes, uma vez que técnicas de incorporação de objetos sintéticos, em filmes e comerciais, são muito procuradas.

Discutiremos as abordagens tradicionais para o problema de iluminação direta. Finalmente apresentamos uma nova abordagem para o problema, um *Esquema Híbrido*, que extrai o melhor das duas principais abordagens existentes: aproximação dos mapas de iluminação com luzes direcionais e amostragem estocástica dos mapas. O *Esquema Híbrido* proposto pode ser utilizado tanto com métodos de iluminação direta quanto com algoritmos de iluminação global. Nossa proposta vem complementar e consolidar as técnicas tradicionais, pois permite utilizá-las separadamente ou combiná-las para tratar o problema de amostragem das fontes luminosas durante a renderização com iluminação real.

Palavras-chave: Gráficos Tridimensionais e Realismo, Amostragem, Renderização, Mapeamento de Ambientes, Amostragem por Importância, Amostragem Determinística.

Abstract

Currently, photorealistic rendering of synthetic 3D scenes using *illumination maps* captured from real environments is an important topic in image synthesis. There is also a great demand for efficient solutions, these techniques are used for incorporation of synthetic objects in films and commercial.

In this thesis we introduce a new approach to the problem of direct illumination in physically based rendering of 3D scenes using *illumination maps* captured from real environments. We developed a system that takes advantage of the best features of the current solutions to the problem: namely, the approximation of illumination maps through directional lights; and stochastic sampling of the light maps. Our framework is flexible and can be used with most rendering programs.

The proposed *Hybrid Scheme* can be used with both direct illumination methods and algorithms for global illumination. Our proposal complements and consolidates the traditional techniques, because they can be used either separately or combined to address the problem of sampling light sources during rendering with real lighting.

Keywords: Three-dimensional Graphics and Realism, Sampling, Rendering, Environment Mapping, Importance Sampling, Deterministic Sampling.

Sumário

Lista de Figuras	p. xiv
Lista de Tabelas	p. xix
Lista de Abreviaturas	p. xx
I Introdução ao problema	1
1 O Problema	p. 2
1.1 Introdução	p. 2
1.2 O problema de iluminação direta	p. 2
1.3 Estrutura da dissertação	p. 5
2 Contexto histórico	p. 6
2.1 Trabalhos relacionados	p. 6
II Teoria de síntese de imagens	15
3 Integração de Monte Carlo	p. 16
3.1 Conceitos de probabilidade	p. 16
3.1.1 Variáveis aleatórias discretas	p. 17
3.1.2 Variáveis aleatórias contínuas	p. 19
3.2 Estimador de Monte Carlo	p. 21
3.3 Amostragem de variáveis aleatórias	p. 23

3.3.1	Método de inversão	p. 23
3.3.2	Funções constantes por partes 1D	p. 25
3.4	Transformações entre distribuições	p. 28
3.4.1	Transformação em múltiplas dimensões	p. 30
3.5	Amostragem 2D com transformações multidimensionais	p. 30
3.6	Roleta-russa	p. 31
4	Simulação foto-realista	p. 33
4.1	Renderização foto-realista	p. 33
4.1.1	Câmeras	p. 34
4.1.2	Intersecções raio-objeto	p. 35
4.1.3	Distribuição de luz	p. 36
4.1.4	Visibilidade	p. 37
4.1.5	Espalhamento sobre superfícies	p. 37
4.1.6	Traçado de raios recursivo	p. 38
4.1.7	Propagação dos raios	p. 40
4.2	Transporte de luz	p. 40
4.2.1	Iluminação direta	p. 41
4.2.1.1	Estimando a integral de iluminação direta	p. 43
4.2.2	Equação de transporte de luz	p. 45
4.2.2.1	Obtendo a equação	p. 45
4.2.2.2	A forma de superfície da LTE	p. 47
4.2.2.3	Integral sobre caminhos	p. 49
4.2.2.4	Distribuições delta no integrando	p. 51
4.2.2.5	Particionamento do integrando	p. 52
4.2.2.6	A equação de medição e a importância	p. 53
4.2.3	Path Tracing	p. 55

4.2.3.1	Visão geral	p. 55
4.2.3.2	Amostragem de caminhos	p. 56
4.2.3.3	Construção incremental do caminho	p. 58
4.3	Mapeamento de fótons	p. 59
4.3.1	Construção do mapa de fótons	p. 60
4.3.1.1	Potência do fóton	p. 60
4.3.2	Estimando a radiância com o mapa de fótons	p. 61
4.3.3	Visão direta do mapa de fótons e reunião final (gather)	p. 62
	Visão direta do mapa de fótons	p. 62
	Reunião final e o mapa cáustico	p. 62
5	Mapas de Iluminação	p. 64
5.1	Imagens <i>HDR</i> versus <i>LDR</i>	p. 64
5.2	Mapas de iluminação	p. 66
5.2.1	Mapeamento Latitude-Longitude	p. 67
5.2.2	Deformação de área e representação do mapa de iluminação	p. 69
6	Amostragem de mapas de iluminação	p. 73
6.1	Amostragem por Importância	p. 73
6.1.1	Esquema prático de amostragem por importância com técnicas unidimensionais	p. 74
	Passo 1: Definir a função $p(u, v)$	p. 74
	Passo 2: Amostragem 2D por inversão	p. 76
	Passo 3: Cálculo da PDF	p. 80
6.2	Amostragem por Algoritmo de Corte Mediano	p. 81
6.2.1	O algoritmo	p. 81
6.3	Amostragem hierárquica por importância	p. 83
6.3.1	Amostragem hierárquica baseada no mosaico de Penrose	p. 84

6.3.2	Relaxação com dados tabelados	p. 87
6.3.3	Sistema numérico de Fibonacci	p. 88

III Nossa proposta 90

7 Abordagem combinada p. 91

7.1	Diagramas de Voronoi	p. 91
7.1.1	Diagramas de Voronoi na esfera unitária \mathbb{S}^2	p. 91
7.2	Metodologias tradicionais de renderização	p. 94
7.2.1	Geração de luzes direcionais	p. 94
7.2.1.1	Vantagens e desvantagens	p. 96
7.2.2	Amostragem direta	p. 96
7.2.2.1	Vantagens e desvantagens	p. 97
7.3	Geração de mapas de pré-amostragem	p. 97
7.4	Esquema híbrido para renderização	p. 99
7.4.1	Separação do mapa de iluminação	p. 100
7.4.2	Estratégias para determinação dos estratos e limiar de separação <i>pp.</i>	104
7.4.2.1	Estimativa por quantidade de luzes desejadas e ângulo de separação mínima	p. 104
7.4.2.2	Estimativa por taxa de variação máxima	p. 106
7.4.3	Clusterização de amostras	p. 115
7.4.3.1	Frequência $F_i = 1$	p. 115
7.4.3.2	Frequência $F_i = \int_{Vor(c_i)} L(x) dx$	p. 116
7.4.3.3	Frequência $F_i = \frac{\int_{Vor(c_i)} L(x) dx}{\overline{Área} Vor(c_i)}$	p. 116
7.4.3.4	Critérios de parada	p. 118

8 Resultados p. 119

8.1	Medição da convergência dos algoritmos de renderização	p. 119
8.1.1	Diferença de cores	p. 119
8.1.2	Diferença de imagens	p. 120
8.2	Resultados	p. 120
8.2.1	Resultados para Iluminação Direta (Direct Lighting)	p. 121
8.2.1.1	Interpretação das tabelas de resultados para iluminação direta	p. 122
8.2.2	Resultados para iluminação global com Traçado de Caminhos (Path Tracing)	p. 148
8.2.2.1	Interpretação das tabelas de resultados para Traçado de Caminhos	p. 148
8.2.3	Resultados para iluminação global com Mapeamento de Fótons (Photon Mapping)	p. 158
8.2.3.1	Interpretação das tabelas de resultados para Mapea- mento de Fótons	p. 158
8.3	Conclusões	p. 169
8.4	Trabalhos futuros	p. 170

IV Apêndice 171

Referências Bibliográficas	p. 176
-----------------------------------	--------

Lista de Figuras

2.1	Traçado de raios e traçado de caminhos	p. 7
2.2	Mapeamento de reflexão	p. 8
2.3	Mapeamento de ambiente em filmes	p. 9
2.4	Mapas de ambiente pré-filtrados	p. 10
2.5	Iluminação baseada em imagens	p. 12
2.6	Métodos de amostragem	p. 13
3.1	Densidade de probabilidade contínua	p. 20
3.2	Método de Inversão	p. 24
3.3	Inversão de funções constantes por partes 1D	p. 25
4.1	Câmera de furo	p. 34
4.2	Câmera virtual	p. 35
4.3	Distribuição de luz	p. 37
4.4	Espalhamento sobre superfícies	p. 38
4.5	Traçado de raios recursivo	p. 39
4.6	Distribuições da <i>BSDF</i> e fontes de luz	p. 44
4.7	Radiância ao longo de um raio no vácuo	p. 47
4.8	<i>LTE</i> em superfícies	p. 48
4.9	Integral sobre caminhos	p. 50
4.10	Amostragem de caminhos	p. 57
4.11	Potência do fóton	p. 60
4.12	Estimativa de radiância baseada no mapa de fótons	p. 61
4.13	Exemplo de mapeamento de fótons	p. 63

5.1	<i>HDR</i> vs. <i>LDR</i>	p. 65
5.2	Transformações entre Coordenadas	p. 68
5.3	Diferencial de área na esfera	p. 70
5.4	Amostragem de mapas com correção de área	p. 71
6.1	Mapa Galileo (100×50)	p. 75
6.2	Mapeamento da deformação de área	p. 76
6.3	Distribuição conjunta $p(u, v)$	p. 77
6.4	Probabilidade marginal $p_u(u)$	p. 77
6.5	Função de densidade acumulada de $p_u(u)$	p. 78
6.6	Probabilidade condicional $p_v(v \bar{u})$	p. 78
6.7	Função de densidade acumulada de $p_v(v \bar{u})$	p. 79
6.8	Representação logarítmica da densidade de probabilidade	p. 82
6.9	Amostragem por Corte Mediano	p. 83
6.10	Mosaicos de Penrose	p. 84
6.11	Regras de produção do mosaico de Penrose modificado	p. 85
6.12	Subdivisões do mosaico de Penrose modificado	p. 86
6.13	Amostragem de mapas com mosaicos de Penrose	p. 89
7.1	Diagrama de Voronoi sobre a esfera	p. 92
7.2	Geração de luzes direcionais	p. 95
7.3	Amostragem por importância	p. 97
7.4	Mapa de pré-amostragem	p. 98
7.5	Estratificação do mapa <i>Kitchen</i>	p. 103
7.6	Estratificação do mapa <i>Kitchen</i>	p. 104
7.7	Calota esférica	p. 105
7.8	Construção da função $\zeta(x)$	p. 107
7.9	Função de luminância do mapa <i>Galileo</i>	p. 109

7.10	Integral de luminância e função $\zeta(x)$ do mapa <i>Galileo</i>	p. 109
7.11	Estratificação do mapa <i>Galileo</i> com limiar de variação máxima	p. 110
7.12	Função de luminância do mapa <i>Grace</i>	p. 111
7.13	Integral de luminância e função $\zeta(x)$ do mapa <i>Grace</i>	p. 111
7.14	Estratificação do mapa <i>Grace</i> com limiar de variação máxima	p. 112
7.15	Função de luminância do mapa <i>Rnl</i>	p. 113
7.16	Integral de luminância e função $\zeta(x)$ do mapa <i>Rnl</i>	p. 113
7.17	Estratificação do mapa <i>Rnl</i> com limiar de variação máxima	p. 114
7.18	Clusterização de amostras	p. 116
7.19	Exemplos de clusterização	p. 117
8.1	Mapeamento de tons do modelo <i>Killeroo</i>	p. 122
8.2	Erro quadrado médio euclideano	p. 123
8.3	Referência do modelo <i>Killeroo</i>	p. 124
8.4	Resultados, modelo <i>Killeroo</i>	p. 126
8.5	Resultados, modelo <i>Killeroo</i>	p. 127
8.6	Resultados, modelo <i>Killeroo</i>	p. 128
8.7	Resultados, modelo <i>Killeroo</i>	p. 129
8.8	Resultados, modelo <i>Killeroo</i>	p. 130
8.9	Resultados, modelo <i>Killeroo</i>	p. 131
8.10	Resultados, modelo <i>Killeroo</i>	p. 132
8.11	Resultados, modelo <i>Killeroo</i>	p. 133
8.12	Erro quadrado médio euclideano	p. 135
8.13	Erro quadrado médio euclideano	p. 135
8.14	Imagens de referência, modelo <i>Buddha</i>	p. 136
8.15	Imagens de referência, modelo <i>Buddha</i>	p. 137
8.16	Resultados, modelo <i>Buddha</i>	p. 138

8.17 Resultados, modelo <i>Buddha</i>	p. 139
8.18 Resultados, modelo <i>Buddha</i>	p. 140
8.19 Comparação de resultados	p. 141
8.20 Comparação de resultados	p. 142
8.21 Resultados, modelo <i>Buddha</i>	p. 143
8.22 Resultados, modelo <i>Buddha</i>	p. 144
8.23 Resultados, modelo <i>Buddha</i>	p. 145
8.24 Comparação de resultados	p. 146
8.25 Comparação de resultados	p. 147
8.26 Integradores de Traçado de Caminhos	p. 148
8.27 Resultados de Traçado de Caminhos	p. 150
8.28 Resultados de Traçado de Caminhos	p. 151
8.29 Comparação de resultados com Traçado de Caminhos	p. 152
8.30 Comparação de resultados com Traçado de Caminhos	p. 153
8.31 Resultados de Traçado de Caminhos	p. 154
8.32 Resultados de Traçado de Caminhos	p. 155
8.33 Comparação de resultados com Traçado de Caminhos	p. 156
8.34 Comparação de resultados com Traçado de Caminhos	p. 157
8.35 Resultados com Mapeamento de Fótons	p. 160
8.36 Resultados com Mapeamento de Fótons	p. 161
8.37 Resultados com Mapeamento de Fótons	p. 162
8.38 Resultados com Mapeamento de Fótons	p. 163
8.39 Resultados com Mapeamento de Fótons	p. 164
8.40 Resultados com Mapeamento de Fótons	p. 165
8.41 Resultados com Mapeamento de Fótons	p. 166
8.42 Resultados com Mapeamento de Fótons	p. 167

8.43 Resultados com Mapeamento de Fótons	p. 168
8.44 Mapa <i>Galileo</i>	p. 172
8.45 Mapa <i>Grace</i>	p. 173
8.46 Mapa <i>Rnl</i>	p. 174
8.47 Mapa <i>Kitchen</i>	p. 175

Lista de Tabelas

5.1	Iluminação de ambientes comuns	p. 65
7.1	Mapas de iluminação	p. 108
8.1	Códigos e parâmetros do Sistema Híbrido	p. 121
8.2	Iluminação direta do modelo <i>Killeroo</i> com o mapa <i>Galileo</i>	p. 125
8.3	Iluminação direta do modelo <i>Buddha</i> com o mapa <i>Grace</i>	p. 134
8.4	Iluminação direta do modelo <i>Buddha</i> com o mapa <i>Rnl</i>	p. 134
8.5	Traçado de Caminhos do modelo <i>2 esferas</i> com o mapa <i>Galileo</i>	p. 149
8.6	Traçado de Caminhos do modelo <i>Killeroo</i> com o mapa <i>Galileo</i>	p. 154
8.7	Mapeamento de Fótons do modelo <i>Sponza 1</i> com o mapa <i>Galileo</i>	p. 159
8.8	Mapeamento de Fótons do modelo <i>Sponza 2</i> com o mapa <i>Galileo</i>	p. 162
8.9	Mapeamento de Fótons do modelo <i>3 esferas</i> com o mapa <i>Galileo</i>	p. 167

Lista de Abreviaturas

BRDF	<i>(Bidirectional Reflectance Distribution Function) - Função de Distribuição de Refletância Bidirecional</i>
BSDF	<i>(Bidirectional Scattering Distribution Function) - Função de Distribuição de Espalhamento Bidirecional</i>
BTDF	<i>(Bidirectional Transmittance Distribution Function) - Função de Distribuição de Transmitância Bidirecional</i>
CDF	<i>(Cumulative Distribution Function) - Função de Distribuição Acumulada</i>
CRT	<i>(Cathode Ray Tube) - Tubo de Raios Catódicos</i>
HDR	<i>(High Dynamic Range) - Alta Faixa Dinâmica</i>
IBL	<i>(Image Based Lighting) - Iluminação Baseada em Imagens</i>
LDR	<i>(Low Dynamic Range) - Baixa Faixa Dinâmica</i>
LTE	<i>(Light Transport Equation) - Equação de Transporte de Luz</i>
MIS	<i>(Multiple Importance Sampling) - Amostragem Múltipla por Importância</i>
MSE	<i>(Mean Squared Error) - Erro Médio Quadrático</i>
PBRT	<i>(Physically Based Ray Tracing) - Traçado de Raios Baseado na Física</i>
PDF	<i>(Probability Density Function) - Função de Densidade de Probabilidade</i>

Parte I

Introdução ao problema

1 *O Problema*

1.1 Introdução

A renderização de cenas complexas, com iluminação indireta, iluminação de ambientes reais, e outros tipos de fontes de luz representa um problema desafiador e de grande interesse da comunidade de computação gráfica. É verdade que um resultado foto-realista depende de vários fatores além da iluminação, como a modelagem dos objetos e propriedades dos materiais, mas tudo isto não pode ser apreciado se não tivermos um método de visualização efetivo e uma iluminação realista.

Para renderizar objetos sintéticos em cenas reais é preciso dispor da iluminação do ambiente real onde será feita a inserção, que pode ser capturada como uma imagem esférica de alta variação dinâmica (HDR - High Dynamic Range). Esta técnica de captura de iluminação, introduzida por Paul Debevec [DEB98], está presente em muitos programas comerciais e foi utilizada em vários filmes nesta década.

1.2 O problema de iluminação direta com mapas de iluminação

A equação que descreve a radiância emanando desde um ponto de uma superfície em uma determinada direção é chamada de *equação de transporte de luz*. Ela foi simultaneamente apresentada à comunidade de computação gráfica por Kajiya [KAJ86] e Immel et al. [ICG86], em 1986. A *equação de transporte* relaciona a radiância emanante de um ponto com as características emissivas do ponto, e a radiância incidente no ponto com as propriedades de espalhamento da superfície, chamada de *BSDF* (função de espalhamento bidirecional). A equação 1.1 é uma das formas em que a equação de transporte pode ser apresentada:

$$L_o(p, \omega_o) = L_e(p, \omega_o) + \int_{\mathbb{S}^2} f(p, \omega_o, \omega_i) L_i(p, \omega_i) |\cos \theta_i| d\omega_i, \quad (1.1)$$

onde L_o é a radiância que chega ao ponto p na direção ω_o ; $L_i(p, \omega_i)$ é a radiância incidente em p na direção ω_i ; $f(p, \omega_o, \omega_i)$ é a BSDF ; e θ_i é o ângulo de incidência [PH04b].

Ao estudar a equação de transporte geralmente a integral é separada em somas integrais obtendo desta forma sub-problemas, cada um dos quais pode ser resolvido com métodos especialmente eficientes para cada caso. Uma primeira separação que vamos considerar é a de separar a integral da direita na equação 1.1 em uma soma de duas integrais: uma delas somando a iluminação incidente no ponto p proveniente diretamente das fontes luminosas, e outra calculando as contribuições indiretas tais como reflexão dos objetos da cena e outros tipos de propagação indireta.

$$L_o(p, \omega_o) = L_e(p, \omega_o) + \int_{\mathbb{S}^2} f(p, \omega_o, \omega_i) L_{direta}(p, \omega_i) |\cos \theta_i| d\omega_i + \int_{\mathbb{S}^2} f(p, \omega_o, \omega_i) L_{indireta}(p, \omega_i) |\cos \theta_i| d\omega_i$$

O foco desta dissertação é estudar o problema de calcular a integral de iluminação direta em cenas onde a iluminação é baseada em imagens extraídas de ambientes reais. As imagens utilizadas para realizar a iluminação da cena são chamadas de mapas de iluminação e contém informação sobre a radiância incidente num ponto p do mundo real a partir de todas as direções. Dada a impossibilidade computacional de calcular a integral

$$\int_{\mathbb{S}^2} f(p, \omega_o, \omega_i) L_{direta}(p, \omega_i) |\cos \theta_i| d\omega_i$$

de maneira exata, é preciso recorrer a discretizações e técnicas de aproximação para obter uma estimativa aceitável da mesma. Uma primeira opção é quebrar a integral num somatório sobre as fontes de luz da cena

$$\sum_{j=1}^{luzes} \int_{\mathbb{S}^2} f(p, \omega_o, \omega_i) L_{d(j)}(p, \omega_i) |\cos \theta_i| d\omega_i,$$

onde $L_{d(j)}$ denota a radiância incidente em p proveniente da j -ésima fonte de luz e

$$L_d(p, \omega_i) = \sum_j L_{d(j)}(p, \omega_i).$$

Como num ambiente real temos iluminação proveniente de todas as direções, considerando um mapa de ambiente tradicional de 1024×512 pixels, teríamos 1024×512 somas se considerarmos cada pixel como uma fonte luminosa, o que ainda torna a solução computacionalmente inviável. Para reduzir este somatório, entram em cena os métodos de integração probabilísticos com estimadores de Monte Carlo e técnicas de redução da quantidade de luzes. No caso de estimadores de Monte Carlo, escolhem-se N direções ω_i com algum método probabilístico, obtendo a estimativa

$$\frac{1}{N} \sum_{j=1}^N \frac{f(p, \omega_o, \omega_j) L_d(p, \omega_j) |\cos \theta_j|}{p(\omega_j)}.$$

Outra possibilidade é escolher um número limitado de luzes importantes e realizar a renderização utilizando exclusivamente este conjunto de luzes. Nosso propósito é estudar as principais técnicas desenvolvidas para realizar a amostragem dos mapas de iluminação, a fim de determinar conjuntos de luzes que permitam uma aproximação realista da iluminação da cena, e estudar técnicas de amostragem para obter as direções a serem utilizadas pelo integrador de Monte Carlo. Propomos também uma abordagem híbrida, que pretende extrair as vantagens dos métodos de integração de Monte Carlo, introduzindo a utilização de pré-amostragem e a geração de luzes direcionais a partir dos mapas de iluminação. A maioria dos softwares de renderização não permite muita escolha de modelagem da iluminação real. Em alguns softwares pode-se trabalhar com o próprio mapa, o qual é utilizado pelas rotinas de amostragem de Monte Carlo, como no caso do *PBRT* [PH04b], ou em outros casos é preciso realizar algum pré-processamento e transformar o mapa numa coleção de luzes pontuais ou direcionais, como é feito no *Maya*.

Nosso objetivo é apresentar um sistema de pré-processamento multifuncional que permita trabalhar com Mapas de Iluminação de variadas formas: obtendo uma coleção de luzes direcionais mediante métodos de amostragem como Corte Mediano [DEB05]; ou amostragem por importância baseado em mosaicos de Penrose [ODJ04]; ou selecionando um número grande de amostras para serem passadas às rotinas de Monte Carlo de renderizadores como *PBRT* e *POV-Ray* entre outros. Também trataremos os Mapas de Iluminação de maneira intermediária, estratificando-os, selecionando regiões das quais extrairemos luzes direcionais e outras que manteremos para passá-las às rotinas de Monte Carlo. Um sistema desta índole é muito útil, pois permite ter um controle fino dos resultados nas renderizações. Permite adaptar um Mapa de Iluminação para a maioria dos softwares de renderização, ou pela utilização do mapa como um todo, ou por partes (luzes

e um conjunto de amostras).

1.3 Estrutura da dissertação

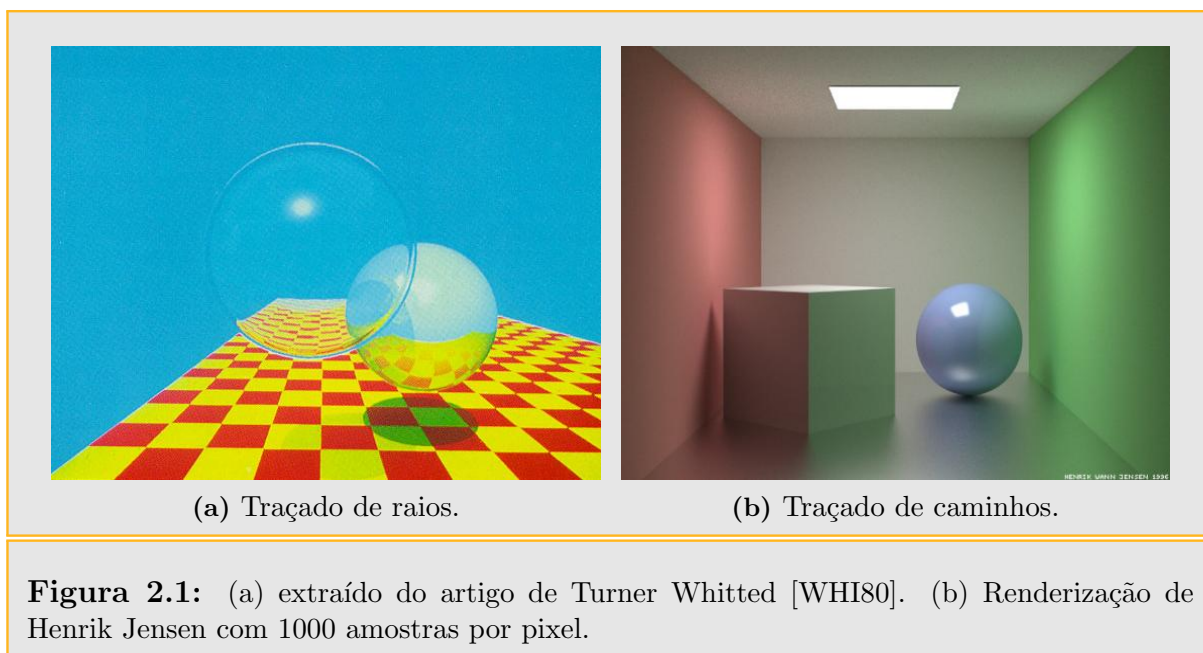
Esta dissertação consta de 8 capítulos distribuídos em três partes. A Parte I consta dos capítulos 1 e 2, onde realizamos a introdução do problema a ser tratado e a contextualização histórica do mesmo. A Parte II, formada pelos capítulos 3, 4, 5 e 6, descreve os aspectos teóricos necessários para o tratamento do problema. No capítulo 3, descrevemos os fundamentos básicos de probabilidade e integração de Monte Carlo necessários; já no capítulo 4, falamos sobre renderização foto-realista e o algoritmo de traçado de raios, da teoria de transporte de luz, abordando a iluminação direta e a equação de transporte de luz, e apresentando alguns algoritmos comuns de traçado de raios. No capítulo 5, tratamos da teoria de mapas de iluminação, as diferenças entre imagens *LDR* e *HDR*, e as formas de representação dos mapas. No capítulo 6 são descritas técnicas de amostragens para mapas de iluminação (Algoritmo de Corte Mediano [DEB05]; Amostragem hierárquica por importância com mosaicos de Penrose [ODJ04]; Amostragem por importância [PH04a]). Finalmente, a Parte III, composta pelos capítulos 7 e 8, trata sobre o problema proposto nesta dissertação e os resultados obtidos como fruto da nossa pesquisa. No capítulo 7 apresentaremos nossa abordagem para o problema de renderização utilizando mapas de iluminação e técnicas de amostragem. No capítulo 8 mostramos os resultados obtidos com nossa abordagem e as comparamos com outras técnicas utilizadas para renderização com iluminação baseada em imagens, as conclusões e os projetos futuros para esta pesquisa. Além dos capítulos citados, incluímos também uma seção de apêndice IV, onde colocamos material complementar. Este material contém imagens dos mapas de iluminação que utilizamos durante a dissertação.

2 *Contexto histórico*

2.1 Trabalhos relacionados

Em computação gráfica, o *traçado de raios* é uma técnica para geração de imagens mediante o traçado do caminho dos raios luminosos a través dos pixels no plano da imagem. Esta técnica foi inicialmente proposta por Appel em 1968 [APP68]. Na versão de Appel a idéia era lançar raios desde o olho, um por pixel, e encontrar o objeto mais próximo que bloqueia esse raio. Usando as propriedades do material do objeto é possível calcular a cor e uma intensidade luminosa. Em 1980 Turner Whitted [WHI80] implementou uma versão mais completa sobre a idéia original de Appel, continuando o traçado do raio, ou seja depois que o raio atinge uma superfície ele continua seu caminho, que é desviado, em seguida ele atinge outras superfícies e assim por diante. Raios refratados que viajam através de material transparente trabalham de forma similar. Também a adição de sombra ficou possível, pois toda vez a o raio atinge a superfície um “raio de sombra” é traçado para saber se aquele ponto da superfície é iluminado por alguma fonte de luz. Whitted produziu inúmeras imagens que impressionaram na época pela qualidade e desde então a técnica de Traçado de Raios desenvolveu-se significativamente (veja um exemplo do artigo de 1980 na figura 2.1a).

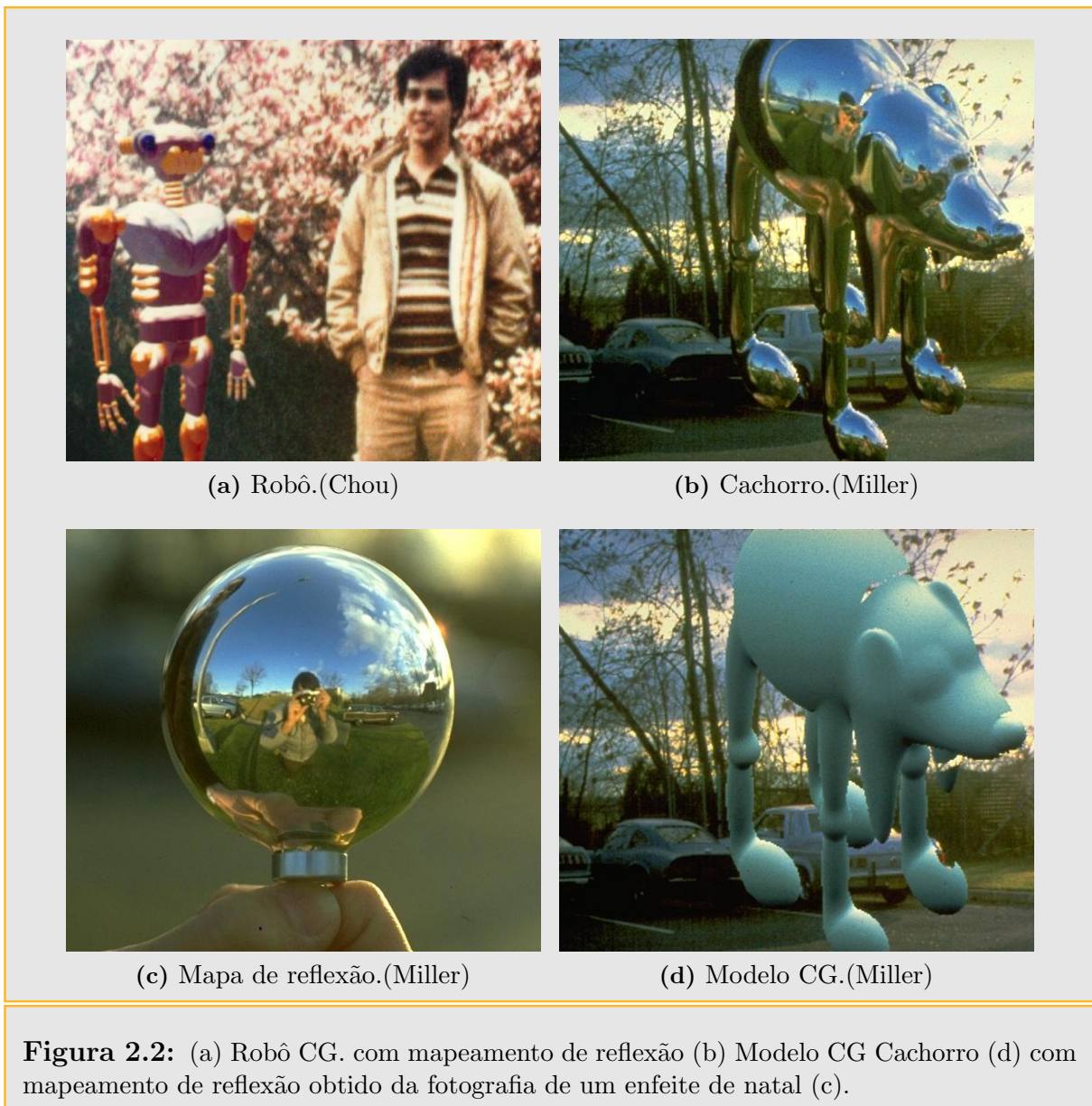
Um trabalho seminal no campo da simulação fisicamente correta da iluminação, intitulado “*The Rendering Equation*” [KAJ86], foi apresentado no *SIGGRAPH’86* por Jim Kajiya, e generalizou as idéias de transporte de luz para qualquer tipo de geometria e qualquer tipo de reflectância das superfícies. A equação principal do artigo de Kajiya expressa que a iluminação emanante de uma superfície em cada direção é uma convulsão da função da luz que chegando na superfície desde todas as direções, com a função que descreve como a superfície reflete a luz. Esta última função, chamada de função distribuição reflectância bidirecional (*BRDF*), é constante para superfícies difusas, mas varia de acordo com as direções de incidência e reflexão da luz para superfícies com brilho.



Kajiya descreveu um processo para renderizar as imagens de acordo com esta equação utilizando uma técnica numérica aleatória conhecida como traçado de caminhos (*Path Tracing*). Assim como a técnica de traçado de raios (Whitted, 1980 [WHI80]), o traçado de caminhos gera imagens traçando raios desde uma câmera para as superfícies na cena e, em seguida, traça raios a partir das superfícies para determinar a iluminação incidente sobre as superfícies. No traçado de caminhos, os raios são traçados não somente na direção das fontes de luz, mas também aleatoriamente em todas as direções a fim de contabilizar a iluminação indireta proveniente do resto da cena. A cena da figura 2.1b foi renderizada utilizando *traçado de caminhos* (path tracing) e mostra de forma realista as interações da luz com os objetos da cena.

O realismo em computação gráfica avançou muito com técnicas que podem capturar iluminação a partir do mundo real e usá-la para criar iluminação em cenas geradas por computador. Se considerarmos um ponto em particular em uma cena, a luz nesse ponto pode ser descrita como o conjunto de todas as cores e intensidades de luz chegando ao ponto desde todas as direções. Uma forma relativamente simples de capturar esta função para um ponto de localização do mundo real é obtendo uma imagem de uma esfera espelhada, chamada de imagem omnidirecional, que reflete luz proveniente de todo o ambiente em direção à câmera. Existem também outras técnicas para capturar imagens omnidirecionais que incluem a utilização de lentes olho de peixe, mosaico de panoramas, e câmeras panorâmicas.

A primeira e mais simples forma de iluminação a partir de imagens, obtida a partir



de uma esfera espelhada, é conhecida como mapeamento de ambiente ou também como mapeamento de reflexão. Nesta técnica, a imagem recebe diretamente uma distorção e é aplicada à superfície do objeto sintético. A aplicação da técnica utilizando fotografias de um cenário real foi realizada independentemente por Gene Miller [MH84] e Mike Chou [WIL83]. Miller utilizou uma bola brilhante utilizada para ornamentar árvores de natal para fotografar o ambiente real e extrair o mapa de reflexão, que utilizou para incluir um balão com formato de cachorro num estacionamento (figura 2.2b). A imagem da figura 2.2a feita por Chou aparece no artigo de Williams 1983 [WIL83], que introduz o *MIP-mapping*, um esquema de pré-filtragem para eliminar aliasing nos algoritmos de mapeamento de textura. O *MIP-mapping* também é utilizado no *PBRT*, software que adotamos para este trabalho.

Pouco tempo depois, a técnica já foi utilizada na indústria do entretenimento para simular as reflexões sobre objetos prateados, como a espaçonave no filme *Vôo do Navegador* (figura 2.3a), de Randal Kleiser em 1986 e, a mais famosa foi o robô metálico *T1000* do filme *Terminator 2* dirigida por James Cameron em 1991 (figura 2.3b). Foi Ned Greene [GRE86] quem em 1986 publicou um artigo formalizando a técnica de mapeamento de reflexões, mostrando que os mapas de ambientes podem ser pré-filtrados e indexados com tabelas de somas de áreas a fim de obter boas aproximações e antialiasing. Na figura 2.4 é possível observar imagens feitas por Greene. Em todos estes exemplos a técnica não somente tem produzido reflexões realistas, mas também fez parecer que o objeto realmente forma parte do ambiente onde foi inserido. Este foi um avanço importante para o realismo nos efeitos visuais.

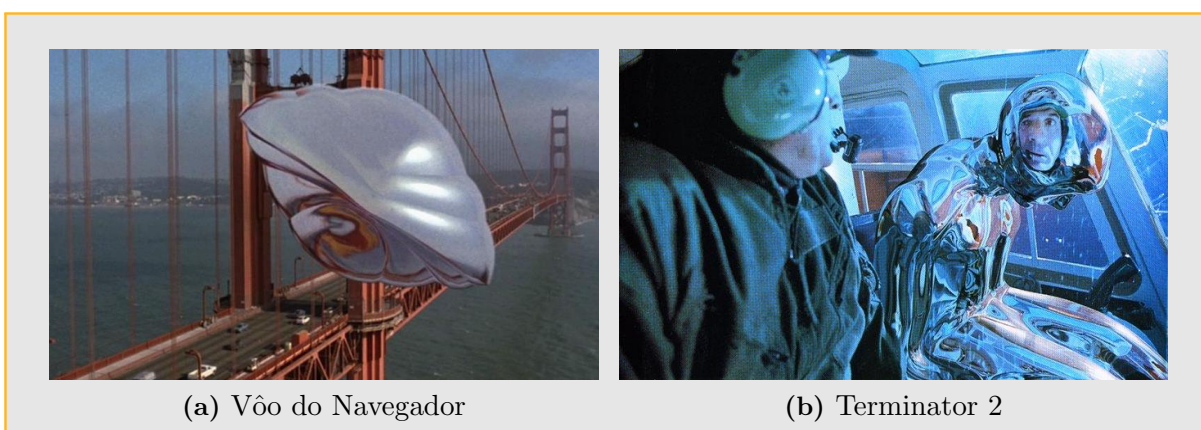
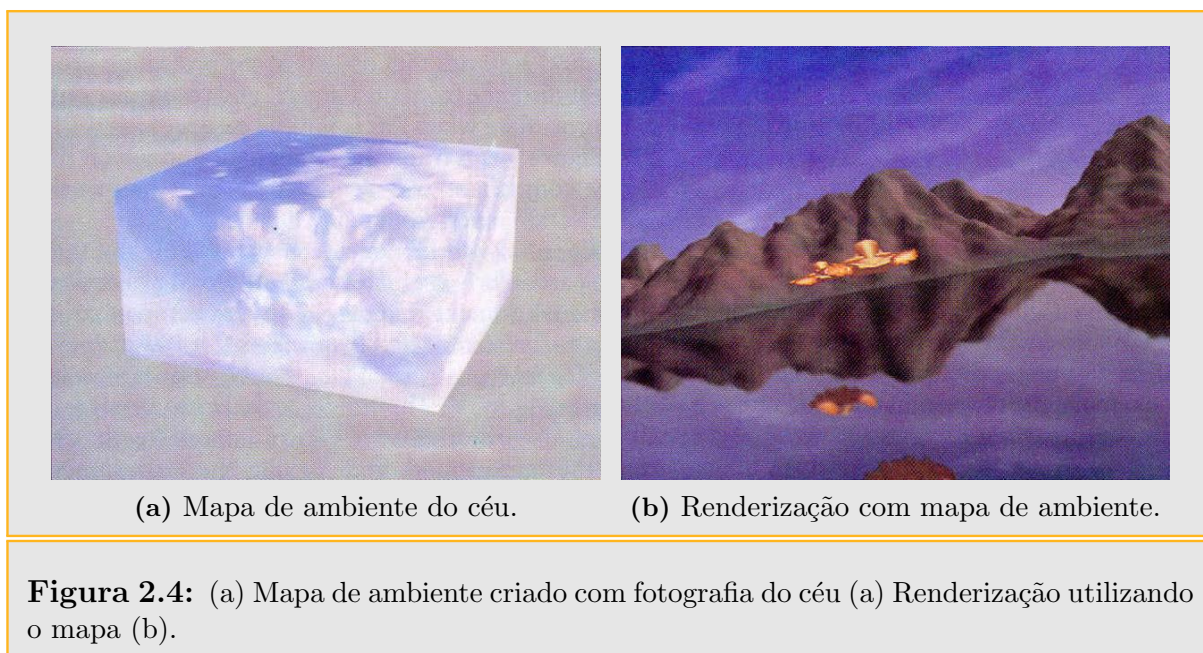


Figura 2.3: (a) Nave espacial com mapeamento de reflexão. (b) Mapeamento de reflexão no robô *T1000* no filme *Terminator 2*.

O mapeamento de ambientes tem produzido resultados convincentes para objetos brilhantes, mas foi necessário continuar inovando para conseguir resultados com objetos animados e sem o brilho metálico, tais como criaturas e seres humanos ou objetos diversos. Uma limitação do mapeamento de ambiente é que ele não pode reproduzir efeitos de sombreamento de um objeto nele próprio ou a reflexão da luz entre superfícies do objeto inserido artificialmente. A razão para esta limitação é que a iluminação ambiente é aplicada diretamente à superfície do objeto de acordo com a orientação da superfície, independentemente do grau de visibilidade de cada superfície no ambiente. Para pontos que estão sobre a superfície da envoltura convexa de um objeto, é possível fazer os cálculos corretamente. No entanto, para a maioria dos pontos sobre um objeto, sua aparência depende tanto das direções do ambiente em que são visíveis como a luz recebida de outros pontos do objeto.

Uma segunda limitação do mapeamento de ambiente tradicional é que uma única foto-



grafia digital de um ambiente dificilmente capta toda a gama de luz visível em uma cena. Por exemplo, em uma cena típica, as fontes de luz diretamente visíveis são centenas ou milhares de vezes mais brilhantes do que a iluminação indireta proveniente do resto da cena, e ambos os tipos de iluminação devem ser capturados para representar a iluminação de maneira precisa. Esta ampla gama dinâmica excede a gama dinâmica tanto das câmeras digitais quanto das câmeras de filmes. Assim, normalmente as fontes luminosas são “cortadas” pelo ponto de saturação do sensor de imagem, não permitindo registrar a sua verdadeira cor ou intensidade. Isto não é um problema grave para superfícies metálicas brilhantes, pois as reflexões também são cortadas na renderização da imagem final. No entanto, o problema aparece nas superfícies que espalham a iluminação incidente refletindo para a câmera somente uma parte da luz incidente, onde a captura incorreta da iluminação do ambiente pode produzir erros graves na visualização final.

Debevec e Malik [DM97] desenvolveram uma técnica para capturar a totalidade da gama dinâmica da luz em uma cena, incluindo até mesmo a luz proveniente diretamente das fontes luminosas. As fotografias são feitas através de uma série de diferentes configurações de exposição da câmera: imagens expostas com muito brilho gravam bem a iluminação indireta da cena, e as exposições baixas gravam adequadamente a iluminação direta a partir das fontes de luz, sem saturar o sensor. Depois, utilizando técnicas para obter a curva de resposta do sistema de imagem, as imagens de gama dinâmica limitada são reunidas numa única imagem de alta gama dinâmica representando toda a gama de iluminação para cada ponto da cena. Números de ponto flutuante são utilizados para representar o valor dos pixels nesta imagem de alta gama dinâmica chamada de imagem

HDR. Seguidamente a este trabalho de 1997, em 1998 Paul Debevec apresenta uma abordagem para iluminar objetos sintéticos com iluminação extraída do mundo real, técnica conhecida como iluminação baseada em imagens (*IBL*), que trata das restantes limitações do mapeamento de ambiente [DEB98]. O primeiro passo na técnica de *IBL* é mapear a imagem no interior de uma superfície, tal como uma esfera infinita, em torno do objeto, ao invés de mapear a imagem diretamente sobre a superfície do objeto como era feito no mapeamento de reflexões. Depois é utilizado um sistema de iluminação global (como o traçado de caminhos descrito por Kajiya [KAJ86]) para simular a iluminação incidente sobre a superfície do objeto. Assim, o algoritmo de iluminação global traça raios de cada ponto do objeto para a cena a fim de determinar a iluminação recebida ao longo desse raio. Alguns dos raios tem caminho livre entre o objeto e a superfície do ambiente de iluminação e, assim, a iluminação do ambiente atinge o objeto. Desta forma, a iluminação proveniente da parte visível do ambiente de iluminação pode ser contabilizada. Outros raios batem no objeto, bloqueando a luz que teria recebido do ambiente nessa direção em outro ponto posterior (auto-sombreamento). O sistema calcula raios adicionais, e a cor do objeto no ponto ocluído da superfície é calculado de forma semelhante, caso contrário, o algoritmo aproxima a luz que chega a esta direção como zero. O algoritmo soma as quantias de toda a luz que chega direta e indiretamente a partir do ambiente a cada ponto da superfície e utiliza esta soma como iluminação do ponto. Esta abordagem produz todos os efeitos da aparência real do objeto iluminado pela luz do ambiente, incluindo o auto-sombreamento, e pode ser aplicada a qualquer tipo material, tais como metal, plástico, vidro, entre outros. As primeiras demonstrações dos algoritmos *HDRI* e *IBL* em um computador foram feitas numa curta animação chamada *Renderização com Luz Natural*, apresentado no festival de animação por computador do *SIGGRAPH'98* (Figura 2.5a).

Os métodos de *HDRI* e *IBL*, além de suas técnicas e sistemas derivados, são hoje amplamente utilizados na indústria de efeitos visuais. Estas técnicas forneceram aos artistas de efeitos visuais novas ferramentas de iluminação e composição digital de atores, aviões, carros e criaturas, dando impressão que os mesmos realmente estão presentes durante a filmagem, e não que foram adicionados mais tarde através de computação gráfica. Exemplos de elementos iluminados desta forma são os mutantes nos filmes *X-Men* e *X-Men 2*, carros virtuais e atores dublês no filme *The Matrix Reloaded*.

As técnicas de *IBL* induziram novos trabalhos associados à utilização eficiente dos mapas de ambientes para recriar a iluminação real na cena sintética. Estes trabalhos focalizam em técnicas de amostragem dos mapas de ambiente para conseguir acelerar o processo de renderização. Entre os primeiros trabalhos importantes na área de amostra-

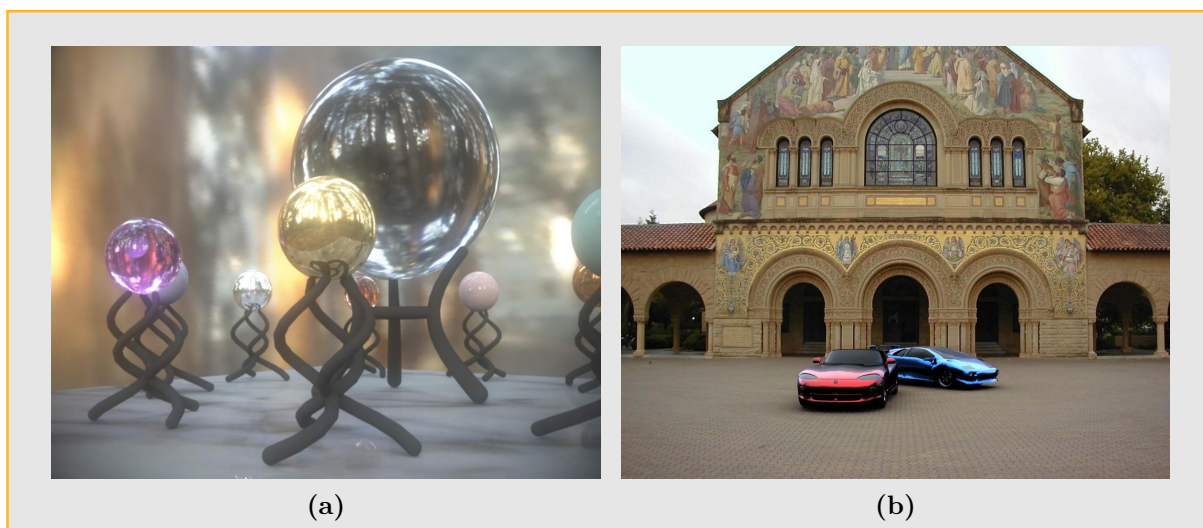


Figura 2.5: (a) Renderização com luz natural de Paul Debevec, imagem sintética iluminada com mapa de iluminação real (b) Renderização de Erick Lee, carros iluminados com mapa de iluminação real e incorporados no ambiente, [DEB98].

gem de mapas de iluminação HDR podemos mencionar o plugin *LightGen* do *HDRShop* [CD01], desenvolvido por Cohen e Debevec em 2001. O *LightGen* utiliza clusterização para gerar uma lista de fontes luminosas direcionais que aproximam o mapa de iluminação. O problema prático do plugin é o tempo excessivo para obter uma lista com uma quantidade razoável de luzes utilizando um mapa de iluminação de tamanho aceitável.

Um segundo trabalho sobre amostragem estruturada por importância, apresentado por Agarwal et al. [ARBJ03], apresenta uma técnica para realizar renderizações eficientes de cenas iluminadas com mapas de ambiente. Esta técnica otimiza a integração da iluminação distante sobre superfícies com BRDFs Lambertianas e semi-brilhantes, levando em conta também a oclusão de partes do mapa pela cena, as altas frequências da iluminação, e é significativamente mais rápido que os métodos alternativos baseados em amostragem de Monte Carlo. Na figura 2.6 é possível ver comparações entre alguns dos métodos de amostragem mencionados. Outro trabalho de 2003 sobre renderização utilizando mapas de iluminação HDR foi o de Kollig e Keller [KK03], onde apresentam um algoritmo para determinar regras de quadratura para realizar o cálculo da iluminação direta de cenas com predominância de objetos difusos, mediante a utilização de mapas de iluminação de alta faixa dinâmica. A técnica produz imagens com bons detalhes nas sombras, e é mais eficiente que as técnicas de integração de Monte Carlo.

Em 2004, Ostromoukhov et al. [ODJ04] introduzem um esquema de amostragem de mapas de iluminação utilizando subdivisões recursivas de maneira hierárquica segundo a função de importância dada pela luminância do mapa. O esquema de subdivisão utilizado

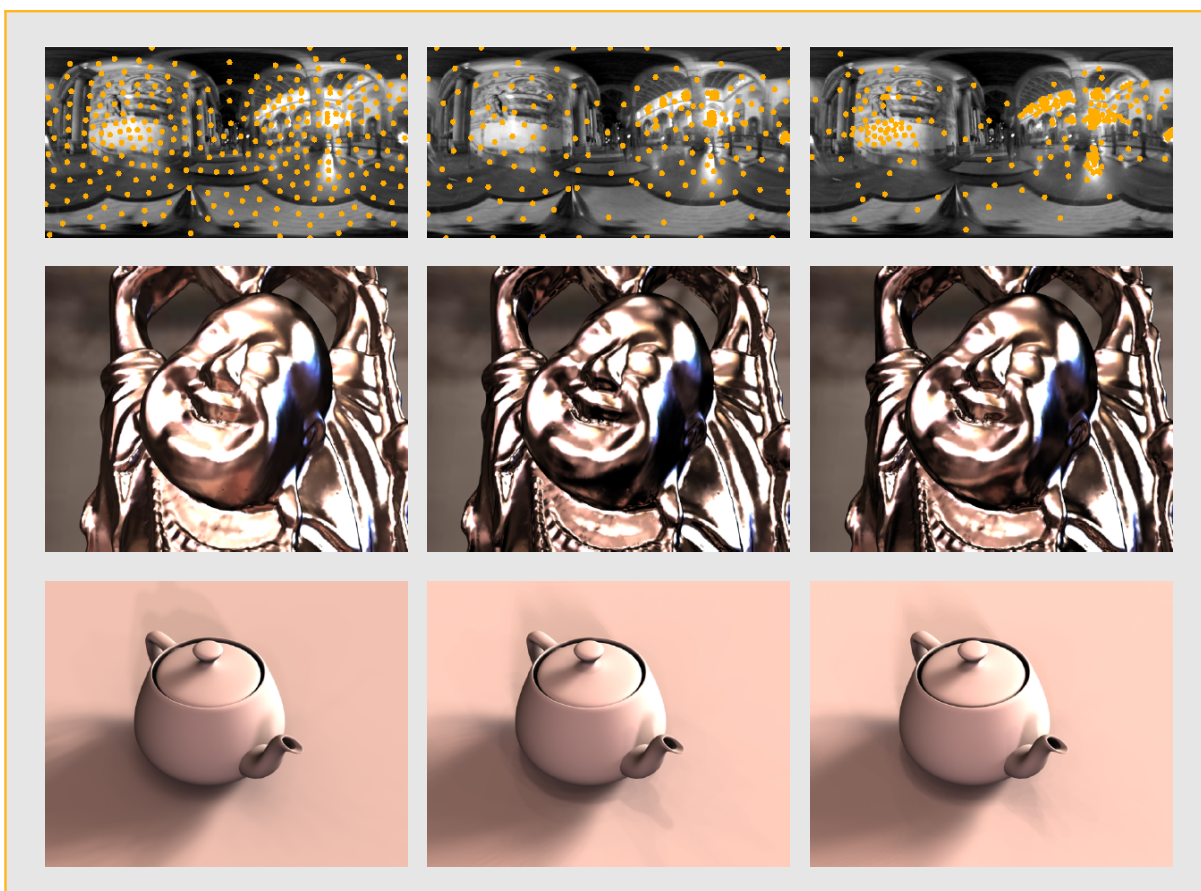


Figura 2.6: 300 amostras do mapa de iluminação Galileo (1024×512), e renderização do *buddha* e *teapot* com **Esquerda:** LightGen [CD01], **Centro:** Structured Importance Sampling [ARBJ03], **Direita:** Importance Sampling de Ostomoukhov [ODJ04].

é feito realizando uma partição da região de interesse em mosaicos de Penrose modificados que cobrem o plano de maneira aperiódica. Este esquema depende somente do mapa de iluminação para obter as amostras e é extremamente rápido.

Paul Debevec [DEB05], apresenta em 2005 um esquema de amostragem para mapas de iluminação que aproveita a idéia de corte mediano utilizada no algoritmo de quantização por corte mediano. O esquema utiliza uma tabela de soma de áreas para realizar a amostragem do mapa dividindo-o em regiões de igual luminância. O esquema é rápido, porém tem o inconveniente de que a quantidade de amostras deve ser uma potência de 2. Trabalhos mais recentes tentam amostrar não somente o mapa de iluminação, mas também o resultado do produto do mapa com a *BRDF* da cena, para minimizar a quantidade final de amostras. Um dos primeiros trabalhos a abordar este problema foi o trabalho de Burke et al. [BGH05], que introduz a noção de amostragem bidirecional por importância, onde as amostras são tomadas da distribuição do produto da iluminação pela refletância da superfície. Este esquema requer maior complexidade no cálculo de amostras, porém

reduz substancialmente a quantidade final de amostras necessárias para obter bons resultados. Outros trabalhos seguindo esta linha de pesquisa são os de Cline et al. [CETC06] e Mei et al. [MJH06], ambos de 2006. As principais diferenças entre estes trabalhos está na metodologia adotada para realizar a amostragem dos mapas de iluminação. Os trabalhos dos últimos anos apontam para novas direções, tais como realizar amostragem de mapas dinâmicos, para incluir iluminação variável numa cena que pode ser estática ou não. Um trabalho interessante nesta direção foi apresentado por Ghosh et al. [GDH06], que apresenta um algoritmo de amostragem sequencial para mapas de iluminação dinâmicos. O algoritmo amostra eficientemente a partir do produto da iluminação e a *BRDF*, enquanto explora também a coerência temporal do mapa.

Parte II

Teoria de síntese de imagens

3 *Integração de Monte Carlo*

Em algoritmos de síntese de imagens, tais como traçado de raios, é preciso calcular a quantidade de energia que chega à câmera ao longo de um raio. Para isso é necessário calcular o valor de integrais de iluminação que geralmente não tem solução analítica ou é inviável computacionalmente. Nestes casos é preciso recorrer a métodos numéricos para calcular eficientemente estas integrais.

Os métodos de integração numérica de Monte Carlo são uma solução para estes tipos de problemas numéricos. Eles utilizam aleatoriedade para avaliar integrais com raio de convergência que é independente da dimensão do integrando. A seguir veremos alguns conceitos importantes de probabilidade.

3.1 Conceitos de probabilidade

Uma variável aleatória pode ser considerada como: o resultado numérico de operar um mecanismo não-determinístico, ou o resultado de fazer uma experiência não-determinística, a fim de gerar resultados aleatórios.

Matematicamente, uma variável aleatória é definida como uma função mensurável de um espaço probabilidade para um espaço mensurável (σ -álgebra). Este espaço mensurável é o espaço de possíveis valores da variável, e é normalmente tomado como a σ -álgebra de Borel baseada na topologia usual dos números reais.

Variáveis aleatórias são tomadas em algum domínio, que pode ser discreto (por exemplo, um número fixo de possibilidades) ou contínuo (por exemplo, o conjunto dos números reais). Aplicando a função f à variável aleatória X , resulta uma nova variável aleatória $Y = f(X)$.

3.1.1 Variáveis aleatórias discretas

Uma variável aleatória ξ é chamada discreta se ela pode tomar qualquer valor entre um conjunto de valores discretos x_1, x_2, \dots, x_n . Uma variável aleatória discreta é definida por uma tabela

$$\xi \sim \begin{pmatrix} x_1 & x_2 & \cdots & x_n \\ p_1 & p_2 & \cdots & p_n \end{pmatrix} \quad (3.1)$$

onde: x_1, x_2, \dots, x_n são os possíveis valores de ξ ; e p_1, p_2, \dots, p_n as correspondentes probabilidades. A probabilidade de que a variável aleatória ξ seja igual a x_i (denotada $P\xi = x_i$) é igual a p_i :

$$P\xi = x_i = p_i$$

A tabela 3.1 é chamada de distribuição da variável aleatória ξ . Os valores x_1, x_2, \dots, x_n podem ser arbitrários. No entanto, as probabilidades p_1, p_2, \dots, p_n devem satisfazer as seguintes condições:

1. Todos os p_i são positivos:

$$p_i > 0 \quad (3.2)$$

2. A soma de todos os p_i deve ser igual a 1:

$$p_1 + p_2 + \cdots + p_n = 1 \quad (3.3)$$

O número

$$E\xi = \sum_{i=1}^n x_i p_i \quad (3.4)$$

é chamado de esperança matemática, ou valor esperado da variável aleatória ξ .

Para elucidar o significado físico desse valor, reescrevemos a fórmula da seguinte forma:

$$E\xi = \frac{\sum_{i=1}^n x_i p_i}{\sum_{i=1}^n p_i}.$$

Desta relação, vemos que $E\xi$ é o valor médio da variável ξ , na qual os valores mais prováveis estão incluídos com maiores pesos.

Apresentamos agora as propriedades do valor esperado. Se c é um número arbitrário não-aleatório, então:

$$E(\xi + c) = E\xi + c \quad (3.5)$$

e

$$E(c\xi) = cE\xi. \quad (3.6)$$

Se ξ e η são duas variáveis aleatórias, então:

$$E(\xi + \eta) = E\xi + E\eta \quad (3.7)$$

O número

$$V\xi = E([\xi - E\xi]^2) \quad (3.8)$$

é chamado de variância da variável aleatória ξ . Assim a variância é o valor esperado do desvio quadrado da variável aleatória ξ do seu valor médio $E\xi$. Obviamente, $V\xi$ é sempre maior que zero.

A equação 3.8 da variância pode ser transformada utilizando as equações 3.5, 3.6 e 3.7:

$$\begin{aligned} V\xi &= E(\xi^2 - 2\xi \cdot E\xi + [E\xi]^2) \\ &= E(\xi^2) - 2E\xi \cdot E\xi + [E\xi]^2, \end{aligned}$$

de onde segue que:

$$V\xi = E(\xi^2) - (E\xi)^2. \quad (3.9)$$

Geralmente o cálculo da variância pela equação 3.9 é mais simples que pela equação 3.8. A variância tem as seguintes propriedades: se c é um número arbitrário não-aleatório, então

$$V(\xi + c) = V\xi \quad (3.10)$$

e

$$V(c\xi) = c^2V\xi. \quad (3.11)$$

O conceito de independência de variáveis aleatórias joga um papel importante na teoria de probabilidade. Dadas duas variáveis aleatórias ξ e η , se a distribuição de ξ não muda quando conhecemos o valor que η assume, então é natural considerar ξ independente de η .

As seguintes relações valem para variáveis aleatórias ξ e η independentes:

$$E(\xi \cdot \eta) = E\xi \cdot E\eta, \quad (3.12)$$

e

$$V(\xi + \eta) = V\xi + V\eta. \quad (3.13)$$

3.1.2 Variáveis aleatórias contínuas

Em computação gráfica, geralmente são mais usuais as variáveis aleatórias discretas do que variáveis aleatórias contínuas, as quais tomam seus valores em domínios contínuos como por exemplo os números reais ou as direções sobre a esfera unitária.

Dizemos que uma variável aleatória é contínua se ela pode assumir qualquer valor num certo intervalo (a, b) .

Uma variável aleatória contínua ξ é definida especificando um intervalo contendo todos seus possíveis valores, e uma função $\rho(x)$ que é chamada de densidade de probabilidade da variável aleatória ξ (ou distribuição de densidade de ξ).

O significado físico de $\rho(x)$ é o seguinte: seja (a', b') um intervalo arbitrário contido em (a, b) (isto é, $a \leq a', b' \leq b$). Então a probabilidade de que ξ caia no intervalo (a', b') é igual à integral

$$Pa' < \xi < b' = \int_{a'}^{b'} \rho(x) dx. \quad (3.14)$$

Esta integral 3.14 é igual à área sombreada na Figura 3.1.

O conjunto de valores de ξ pode ser qualquer intervalo. Os casos $a = -\infty$ e/ou $b = \infty$ também são possíveis. Contudo, a densidade $\rho(x)$ deve satisfazer duas condições análogas às condições 3.2 e 3.3 para variáveis discretas:

1. A densidade $\rho(x)$ é positiva em (a, b) :

$$p(x) > 0 \quad (3.15)$$

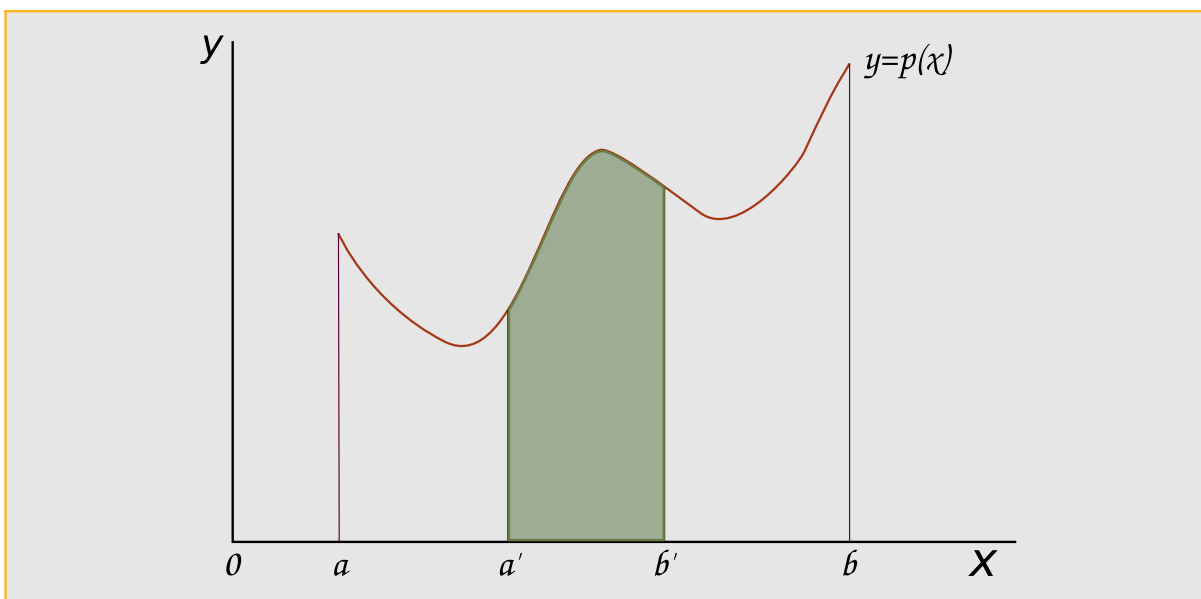


Figura 3.1: PDF $p(x)$ para uma variável aleatória não negativa, ξ . A avaliação de $p(x)$ não tem, por si própria, significado em termos de probabilidades para valores específicos de ξ . Probabilidades são obtidas pela integração de porções de $p(x)$.

2. A integral da densidade $\rho(x)$ sobre todo o intervalo (a, b) é igual a um:

$$\int_a^b \rho(x) dx = 1. \quad (3.16)$$

O número

$$E\xi = \int_a^b x\rho(x) dx \quad (3.17)$$

é chamado de valor esperado da variável aleatória contínua ξ .

O valor esperado tem o mesmo significado que no caso de variáveis aleatórias discretas. De fato, desde

$$E\xi = \frac{\int_a^b x\rho(x) dx}{\int_a^b \rho(x) dx}$$

pode ser visto que é o valor médio de ξ : cada valor de x do intervalo (a, b) entra na integral com sua ponderação $\rho(x) dx$.

Todas as equações desde 3.5 até 3.13 são validadas também para variáveis aleatórias contínuas. Isto inclui a definição de variância 3.8, a equação 3.9, e todas as propriedades de $E\xi$ e $V\xi$.

Vamos falar de mais uma fórmula para a expectativa para a função de ξ . Considere a variável aleatória ξ com densidade de probabilidade $\rho(x)$. Considere uma função contínua

arbitraria $f(x)$ e defina uma variável aleatória $\eta = f(x)$. Pode-se provar que

$$Ef(\xi) = \int_a^b f(x)\rho(x)dx, \quad (3.18)$$

porém cabe destacar que em geral, $Ef(\xi) \neq f(E\xi)$.

Finalmente, para uma variável aleatória contínua ξ e um valor arbitrário x

$$P\xi = x = 0.$$

Portanto, a probabilidade da igualdade $\xi = x$ é fisicamente sem sentido. São fisicamente significativas probabilidades de cair num intervalo pequeno:

$$Px \leq \xi < x + dx = p(x)dx. \quad (3.19)$$

3.2 Estimador de Monte Carlo

Podemos definir o estimador de Monte Carlo aquele que aproxima o valor de uma integral arbitrária. Esta é a base para os algoritmos de transporte de luz descritos neste trabalho. Num primeiro momento, suponha que desejamos calcular uma integral unidimensional $\int_a^b f(x)dx$. Dado um fornecimento de variáveis aleatórias uniformes $X_i \in [a, b]$, o estimador de Monte Carlo nos diz que o valor esperado do estimador é

$$F_N = \frac{b-a}{N} \sum_{i=1}^N f(X_i),$$

$E[F_N]$, é de fato igual à integral. Primeiro, notemos que a *PDF* (função de densidade de probabilidade) $p(x)$ correspondente à variável aleatória X_i deve ser igual a $1/(b-a)$, pois p deve ser constante, e sua integral sobre o domínio $[a, b]$ deve ser igual a 1. Temos:

$$\begin{aligned}
E[F_N] &= E \left[\frac{b-a}{N} \sum_{i=1}^N f(X_i) \right] \\
&= \frac{b-a}{N} \sum_{i=1}^N E[f(X_i)] \\
&= \frac{b-a}{N} \sum_{i=1}^N \int_a^b f(x)p(x)dx \\
&= \frac{1}{N} \sum_{i=1}^N \int_a^b f(x)dx \\
&= \int_a^b f(x)dx.
\end{aligned}$$

A restrição para variáveis aleatórias uniformes pode ser deixada fazendo uma generalização. Isto é um aspecto importante, pois a escolha certa da *PDF* de onde provem as variáveis, é uma técnica importante para a redução de variância no método de Monte Carlo. Se as variáveis aleatórias X_i são tomadas de uma *PDF* arbitrária $p(x)$, então o estimador

$$F_N = \frac{1}{N} \sum_{i=1}^N \frac{f(X_i)}{p(X_i)} \quad (3.20)$$

pode ser usado para estimar a integral anterior. Vemos que a única restrição para $p(x)$ é que deve ser não-nulo para todo x onde, $|f(x)| > 0$. O valor esperado deste estimador é a integral de f :

$$\begin{aligned}
E[F_N] &= E \left[\frac{1}{N} \sum_{i=1}^N \frac{f(X_i)}{p(X_i)} \right] \\
&= \frac{1}{N} \sum_{i=1}^N \int_a^b \frac{f(x)}{p(x)} p(x)dx \\
&= \frac{1}{N} \sum_{i=1}^N \int_a^b f(x)dx \\
&= \int_a^b f(x)dx.
\end{aligned}$$

É possível estender este estimador para múltiplas dimensões ou domínios complexos. Tomam-se N amostras X_i de uma *PDF* multidimensional, e o estimador é aplicado como anteriormente. Por exemplo, considerando a integral tridimensional

$$\int_{x_0}^{x_1} \int_{y_0}^{y_1} \int_{z_0}^{z_1} f(x, y, z) dx dy dz.$$

Se a amostra $X_i = (x_i, y_i, z_i)$ e tomada uniformemente do paralelepípedo formado pelos vértices (x_0, y_0, z_0) e (x_1, y_1, z_1) , a *PDF* é a constante

$$\frac{1}{(x_1 - x_0)} \frac{1}{(y_1 - y_0)} \frac{1}{(z_1 - z_0)},$$

e o estimador é

$$\frac{(x_1 - x_0)(y_1 - y_0)(z_1 - z_0)}{N} \sum_i f(X_i).$$

O número de amostras N pode ser escolhido arbitrariamente, independentemente da dimensão do integrando. Esta é outra vantagem importante do método de Monte Carlo sobre as técnicas de quadratura determinista. O número de amostras tomadas em Monte Carlo é independente da dimensão da integral, enquanto que nas técnicas de quadratura, o número de amostras exigidas cresce de forma exponencial com a dimensão.

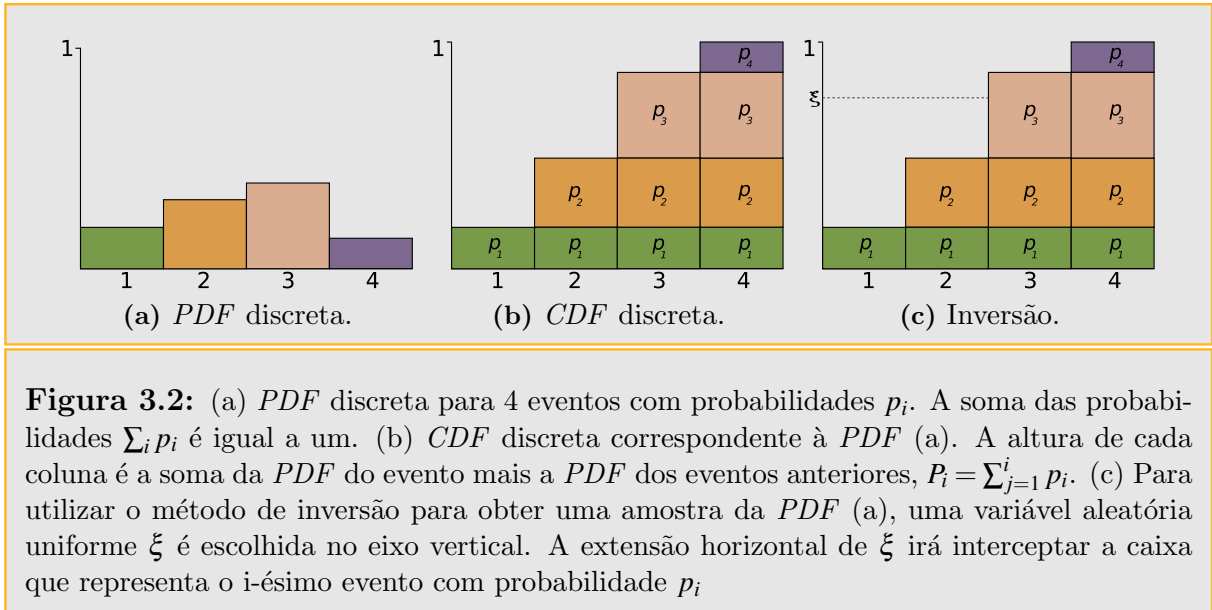
3.3 Amostragem de variáveis aleatórias

A fim de poder calcular o estimador de Monte Carlo descrito anteriormente, é preciso tomar amostras a partir da função de probabilidade. Nesta seção concentraremos no *método de inversão* aplicado a funções unidimensionais constantes por partes, uma vez que será suficiente para os propósitos desta dissertação. Para um tratamento mais detalhado dos métodos disponíveis e mais exemplos, o leitor pode remitir-se a [PH04b].

3.3.1 Método de inversão

O método de inversão utiliza uma ou mais variáveis aleatórias uniformes e as mapeia em variáveis aleatórias na distribuição desejada. Para explicar melhor como esse processo funciona de um modo geral, vamos começar com um exemplo discreto e simples. Suponhamos que temos um processo com quatro resultados possíveis. As probabilidades de cada um dos quatro resultados são p_1 , p_2 , p_3 e p_4 , respectivamente, e com a exigência de que $\sum_{i=1}^4 p_i = 1$. A *PDF* correspondente é mostrada na figura 3.2a.

A fim de obter uma amostra desta distribuição, primeiro encontramos a *CDF* $P(x)$. No caso contínuo, P é a integral indefinida de p . No caso discreto, podemos construir a *CDF* diretamente empilhando as grades uma em cima das outras, começando pela esquerda. A idéia está exemplificada na figura 3.2b. Observe que a altura total no lado direito deve ser igual a um, pois é necessário que a soma de todas as probabilidades seja um.



Para obter uma amostra da distribuição, tomamos um número aleatório uniforme ξ e o utilizamos para selecionar um dos possíveis resultados usando a *CDF*, fazendo-o de forma tal a escolher um resultado particular com probabilidade igual à sua própria probabilidade. Esta idéia é ilustrada na figura 3.2c, onde as probabilidades dos eventos são projectadas sobre o eixo vertical e uma variável aleatória ξ escolhe entre elas. Deve ficar claro que esta seleciona a partir da distribuição certa — a probabilidade da amostra uniforme atingindo uma determinada barra é exatamente igual à altura dessa barra. No intuito de generalizar essa técnica para distribuições contínuas, devemos considerar o que acontece quando o número de abordagens discretas tende a infinito. A *PDF* da figura 3.2a torna-se uma curva suave, e a *CDF* da figura 3.2b torna-se o sua integral. O processo de projeção descrito no parágrafo anterior ainda é o mesmo, embora, se as funções são contínuas, a projeção tem uma interpretação matemática conveniente — representado invertendo a *CDF* e avaliando o inverso em ξ . Esta técnica é chamada de *método de inversão*.

Podemos obter uma amostra X_i a partir de uma *PDF* arbitraria $p(x)$ seguindo os seguintes passos:

1. Calcular a *CDF* $P(x) = \int_0^x p(x') dx'$.
2. Calcular a inversa $P^{-1}(x)$.
3. Obter um número aleatório uniforme ξ .
4. Calcular $X_i = P^{-1}(\xi)$.

3.3.2 Funções constantes por partes 1D

Mostraremos agora como amostrar a partir de uma função unidimensional constante por partes pelo método de inversão. Para isto vamos considerar sem perda de generalidade que o domínio da função é o intervalo $[0, 1]$.

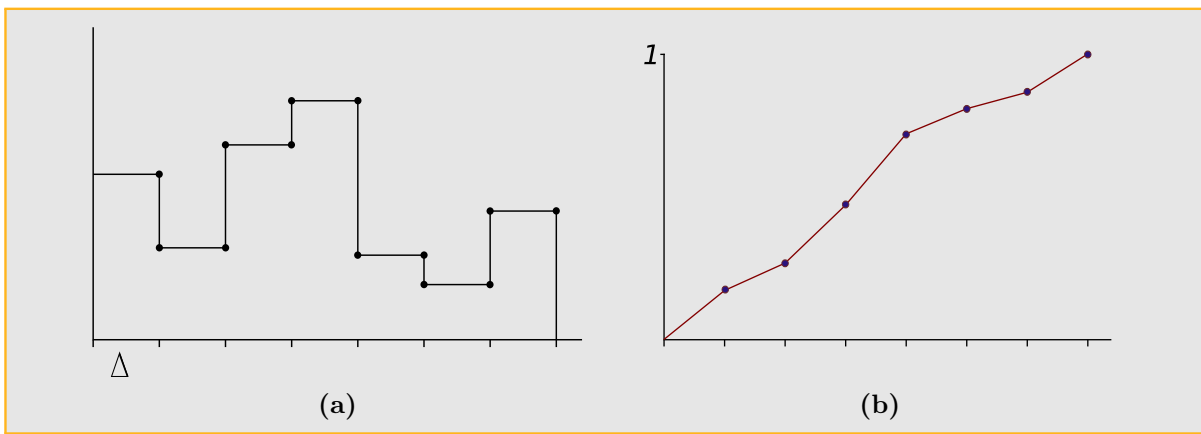


Figura 3.3: (a) Função de Densidade de Probabilidade de uma função 1D constante por partes. (b) Função de Densidade Acumulada da *PDF* (a).

Vamos supor que o domínio da função está dividido em N partes iguais de tamanho $\Delta = 1/N$. Testas regiões começam e acabam nos pontos $x_i = i\Delta$, onde i varia de 0 a N . Em cada região o valor da função $f(x)$ é constante (figura 3.3). O valor de $f(x)$ é

$$f(x) = \begin{cases} v_0 & : x_0 \leq x < x_1 \\ v_1 & : x_1 \leq x < x_2 \\ \dots & : \dots \end{cases}$$

A integral $\int f(x)dx$ é

$$c = \int_0^1 f(x)dx = \sum_{i=0}^{N-1} \Delta v_i = \sum_{i=0}^{N-1} \frac{v_i}{N}, \quad (3.21)$$

Desta forma é fácil construir a *PDF* $p(x)$ para $f(x)$ como sendo $f(x)/c$. Aplicando diretamente a fórmula relevante, a *CDF* $P(x)$ é uma função linear por partes definida nos pontos X_i por

$$\begin{aligned}
P(x_0) &= 0 \\
P(x_1) &= \int_{x_0}^{x_1} p(x) dx = \frac{v_0}{Nc} = P(x_0) + \frac{v_0}{Nc} \\
P(x_2) &= \int_{x_0}^{x_2} p(x) dx = \int_{x_0}^{x_1} p(x) dx + \int_{x_1}^{x_2} p(x) dx = P(x_1) + \frac{v_1}{Nc} \\
P(x_i) &= P(x_{i-1}) + \frac{v_{i-1}}{Nc}.
\end{aligned}$$

Entre dois pontos x_i e x_{i+1} , a *CDF* cresce linearmente com pendente v_i/c . Devemos lembrar que para amostrar $f(x)$ precisamos inverter a *CDF* para encontrar o valor x tal que

$$\xi = \int_0^x p(x') dx' = P(x).$$

Como a *CDF* é monótona crescente, o valor de x deve estar entre x_i e x_{i+1} tal que $P(x_i) \leq \xi \leq P(x_{i+1})$.

Para fazer esta amostragem eficiente, primeiro forneceremos uma função que toma o conjunto de valores v_i de $f(x)$ e calcula os valores da *CDF* em x_i . A função retornará também a integral de $f(x)$ numa variável c fornecida pelo usuário. Na continuação apresentamos trechos da implementação correspondente ao *PBRT* para este tipo de amostragem:

```

<Definições da função MC > ≡
void ComputeStep1dCDF(float *f, int nSteps, float *c,
float *cdf){
    <Cálculo da integral da função passo em x_i >
    <Transformação da integral da função passo numa cdf >
}

```

Esta função calcula primeiro a integral de $f(x)$ mediante a equação 3.21. O resultado é armazenado no array `cdf`. A função de chamada deve alocar `nSteps+1` lugares no array `cdf`, dado que se $f(x)$ tem N intervalos de valores, devemos armazenar o valor da *CDF* em cada um dos $N+1$ valores de x_i .

⟨Cálculo da integral da função passo em x_i ⟩ ≡

```
int i;
cdf[0] = 0.;
for (i = 1; i < nSteps+1; ++i)
    cdf[i] = cdf[i-1] + f[i-1] / nSteps;
```

Agora que o valor da integral foi calculado sobre todo o intervalo $[0, 1]$ e armazenado em `cdf[nSteps]`, a *CDF* pode ser normalizada dividindo-a por este valor:

⟨Transformação da integral da função passo numa cdf⟩ ≡

```
*c = cdf[nSteps];
for (i = 1; i < nSteps+1; ++i)
    cdf[i] /= *c;
```

A amostragem da função pela *CDF* é feita pelo método `SampleStep1d()`:

⟨Definições da função MC⟩ ≡

```
float SampleStep1d(float *f, float *cdf, float c,
    int nSteps. float u, float *pdf){
    ⟨Encontrar segmentos em torno cdf⟩
    ⟨Retorno compensado ao longo do segmento atual da cdf⟩
}
```

Primeiro, é preciso encontrar o par de valores de *CDF* entre os quais está ξ . Como o array `cdf` é monotonamente crescente, é possível utilizar o método de busca binária `lower_bound()` da biblioteca estandar do C++, que utiliza como argumentos um ponteiro para o começo do array, um ponteiro uma posição a frente do final do array e o valor a ser achado. O ponteiro devolvido por `lower_bound()` pode ser transformado num inteiro `offset`, que indica a posição no array.

```

⟨Encontrar segmentos em torno cdf⟩ ≡
    float *ptr = std::lower_bound(cdf, cdf+nStep+1, u);
    int offset = (int) (ptr-cdf-1);

```

Conhecido o par de valores da *CDF* entre os quais está ξ , é possível calcular x . Primeiramente, determina-se a distância relativa de ξ para $\text{cdf}[\text{offset}]$ e $\text{cdf}[\text{offset}+1]$. Como a *CDF* é linear, x mantém esta distância relativa entre x_i e x_{i+1} (figura 3.3). A *PDF* para esta amostra $p(x)$ é calculada facilmente pois temos o valor da integral de $f(x)$ armazenado em c , assim

$$p(x) = \frac{f(x)}{c}.$$

```

⟨Retorno compensado ao longo do segmento atual da cdf⟩ ≡
    u = (u - cdf[offset]) / (cdf[offset+1] - cdf[offset]);
    *pdf = f[offset] / c;
    return (offset + u) / nSteps;

```

3.4 Transformações entre distribuições

Durante a descrição do método de inversão, introduzimos uma técnica que gera amostras de acordo com uma distribuição transformando variáveis aleatórias uniformes canônicas de uma determinada maneira. Aqui, vamos investigar o problema mais geral de quais são os resultados da distribuição, quando transformar amostras de uma distribuição arbitrária para algum outro tipo de distribuição com uma função f .

Suponha que tenhamos variáveis aleatórias X_i que são tomadas de alguma *PDF* $p_x(x)$. Se computarmos $Y_i = y(x_i)$, é importante também calcular a distribuição para a nova variável aleatória Y_i . O estudo deste tipo de transformação é importante porque nos fornecerá uma idéia de como obter amostras a partir de funções de distribuição multidimensionais.

A função $y(x)$ deve ser uma transformação injetiva, pois no caso de ter vários valores de x mapeados para um valor y , será impossível descrever precisamente a densidade de probabilidade para um valor de y dado. Como frequência do fato da injetividade de y , temos que a derivada de y é estritamente positiva ou estritamente negativa, portanto

$$\Pr Y \leq y(x) = \Pr X \leq x,$$

e, por conseguinte

$$P_y(y) = P_y(y(x)) = P_x(x).$$

Esta relação entre *CDFs* leva à relação entre as *PDFs*. Se assumirmos que a derivada de y é maior que zero, diferenciando obtemos:

$$p_y(y) \frac{dy}{dx} = p_x(x),$$

e assim

$$p_y(y) = \left(\frac{dy}{dx} \right)^{-1} p_x(x).$$

Em geral, a derivada de y é sempre estritamente positiva ou estritamente negativa, portanto a relação entre as densidades é:

$$p_y(y) = \left| \frac{dy}{dx} \right|^{-1} p_x(x).$$

Vejamos agora como utilizar isto: seja $p_x(x) = 2x$ no domínio $[0, 1]$; e seja $Y = \sin X$. Vejamos quem é a *PDF* da variável aleatória Y . Como $dy/dx = \cos x$, temos:

$$p_y(y) = \frac{p_x(x)}{|\cos x|} = \frac{2x}{\cos x} = \frac{2 \sin^{-1} y}{\sqrt{1-y^2}}.$$

O procedimento apresentado não tem muita utilidade prática, pois em geral pretendemos amostrar a partir de uma *PDF* dada sem ter a transformação. Por exemplo, temos uma variável aleatória X amostra de alguma distribuição $p_x(x)$, e precisamos calcular Y de alguma distribuição $p_y(y)$. Qual é a transformação a ser utilizada? Precisamos que as *CDFs* sejam iguais, isto é, $P_y(y) = P_x(x)$, de onde temos:

$$y(x) = P_y^{-1}(P_x(x)).$$

Basicamente isto é a generalização do método de inversão, desde que X seja uniformemente distribuída sobre $[0, 1]$, então $P_x(x) = x$.

3.4.1 Transformação em múltiplas dimensões

No caso geral n -dimensional, uma derivação semelhante ao caso unidimensional dá uma relação entre densidades diferentes. Supondo que tenhamos uma variável aleatória n -dimensional X com função densidade $p_x(x)$; seja $Y = T(X)$, onde T é uma bijeção. Neste caso, as densidades estão relacionados da seguinte forma:

$$p_y(y) = p_y(T(x)) = \frac{p_x(x)}{|J_T(x)|},$$

onde: $|J_T|$ é o valor absoluto do determinante do Jacobiano da matriz T , que é

$$\begin{pmatrix} \partial T_1/\partial x_1 & \cdots & \partial T_1/\partial x_n \\ \vdots & \ddots & \vdots \\ \partial T_n/\partial x_1 & \cdots & \partial T_n/\partial x_n \end{pmatrix},$$

onde: T_i é definido por $T(x) = (T_1(x), \dots, T_n(x))$.

3.5 Amostragem 2D com transformações multidimensionais

Considere uma função de densidade conjunta 2D $p(x, y)$ a partir do qual tiramos amostras. Às vezes densidades multidimensionais são separáveis e podem ser expressas como produtos de densidades unidimensionais. Por exemplo:

$$p(x, y) = p_x(x) \cdot p_y(y)$$

para alguma p_x e p_y . Neste caso, variáveis aleatórias (X, Y) podem ser encontradas amostrando independentemente X a partir de p_x e Y a partir de p_y . Dado que muitas funções de densidade não são separáveis, vamos ver como amostrar distribuições multidimensionais em geral. Dada uma função de densidade 2D, a *Função de Densidade Marginal* $p(x)$ obtém-se integrando uma das dimensões:

$$p(x) = \int p(x, y) dy.$$

Podemos pensar nisto como a função de densidade da variável X . Mais precisamente, é a densidade média para um determinado x sobre todos os valores possíveis para y .

A *Função de Densidade Condicional* $p(y|x)$ é a função de densidade para y para uma escolha particular de x :

$$p(y|x) = \frac{p(x,y)}{p(x)}.$$

Desta forma, a idéia para amostragem 2D de distribuições conjuntas se resume a calcular a densidade marginal para isolar uma variável e obter amostras desta densidade utilizando técnicas unidimensionais usuais. Uma vez extraída a amostra, calcula-se a função de densidade condicional para o valor dado, e extraímos uma amostra desta distribuição com as técnicas unidimensionais.

3.6 Roleta-russa

A Roleta-Russa é uma que pode melhorar a eficiência das estimativas de Monte Carlo, aumentando a probabilidade de que cada amostra terá uma contribuição significativa para o resultado. A Roleta-Russa aborda o problema das amostras que são custosas de avaliar, mas fazem uma pequena contribuição para o resultado final. A intenção é evitar o trabalho de avaliação destas amostras, enquanto estes ainda calculam uma estimativa correta.

A modo de exemplo, considere o problema de estimar a integral de iluminação direta descrita na seção 4.2.1, que dá a radiância refletida num ponto devido a iluminação direta das fontes de luz da cena, L_d :

$$L_o(p, \omega_o) = \int_{S^2} f_r(p, \omega_o, \omega_i) L_d(p, \omega_i) |\cos \theta_i| d\omega_i.$$

Assumindo que decidimos tomar $N = 2$ amostras de uma distribuição $p(\omega)$ para calcular o estimador

$$\frac{1}{2} \sum_{i=1}^2 \frac{f_r(p, \omega_o, \omega_i) L_d(p, \omega_i) |\cos \theta_i|}{p(\omega_i)}.$$

A maior parte do gasto computacional de cada um desses termos do somatório vem do traçado do raio de sombra a partir do ponto p , para ver se a fonte de luz é visível desde p .

Para todas as direções ω_i , onde o valor do integrando é próximo de zero, podemos ignorar o traçado dos raios de sombra, uma vez que o traçado não irá alterar o valor final

calculado. A Roleta-Russa torna possível também evitar o traçado dos raios quando o valor do integrando é muito baixo, mas não necessariamente zero, enquanto ainda calcula o valor correto, em média. Por exemplo, queremos evitar o traçado dos raios, onde $f_r(p, \omega_o, \omega_i)$ é pequena, ou quando ω_i está perto do horizonte, porque $|\cos \theta_i|$ é pequeno. Evidentemente, essas amostras não podem ser ignoradas completamente, pois senão o estimador não vai estimar consistentemente o resultado correto.

Para aplicar a Roleta-Russa, selecionamos alguma probabilidade de rescisão q . Este valor pode ser escolhido de muitas formas: por exemplo, poderia ter por base uma estimativa do valor do integrando para uma amostra particular escolhida, incrementado a mesma, o valor do integrando torna-se menor. Com probabilidade q , o integrando não é avaliado para a amostra particular e algum valor constante c é usado em seu lugar, geralmente $c = 0$. Com probabilidade $1 - q$, o integrando ainda é avaliado, mas é ponderado por um termo, $1/(1 - q)$, que efetivamente leva em conta todas as amostras que foram ignoradas:

$$F' = \begin{cases} \frac{F - qc}{1 - q} & \xi > q \\ c & \xi \leq q \end{cases}$$

Convenientemente, o valor esperado do estimador resultante é igual ao valor esperado do estimador original:

$$E[F'] = (1 - q) \left(\frac{E[F] - qc}{1 - q} \right) + qc = E[F].$$

A Roleta-Russa nunca reduz a variância. Na verdade, a menos que alguma $c = F$, a variância irá sempre aumentar. Todavia, pode-se melhorar a eficiência se as probabilidades são escolhidas de forma a que as amostras susceptíveis de fazer uma pequena contribuição para o resultado final são ignoradas.

Um inconveniente é que uma má escolha dos pesos para a Roleta-Russa pode aumentar substancialmente a variância. Considere aplicar Roleta-Russa a todos os raios da câmera com probabilidade de rescisão 0,99: teríamos apenas 1% de raios traçados desde a câmera, cada um deles com peso $1/0.01 = 100$. A imagem resultante seria correta desde um ponto de vista matemático, mas visualmente seria terrível: muitos pixels pretos com alguns poucos pixels muito brilhantes.

4 *Simulação foto-realista*

Neste capítulo descreveremos rapidamente a estrutura básica de um sistema de simulação foto-realista baseado no método de traçado de raios. Existem em circulação vários softwares de simulação foto-realista que utilizam traçado de raios, entre eles podemos citar o *POV-Ray*, *YafRay* e *PBRT*; sendo os dois primeiros bastante difundidos e aceitos pelos usuários ao redor do mundo todo. Porém, nesta dissertação vamos trabalhar com o sistema *PBRT* desenvolvido com fins educativos e que apresenta uma configuração de plugins especialmente útil para nosso trabalho, pois desta forma poderemos implementar plugins das nossas técnicas e implementá-las no *PBRT*.

4.1 **Renderização foto-realista e o algoritmo de traçado de raios**

O objetivo da renderização foto-realista é criar uma imagem de uma cena 3D que é indistinguível de uma fotografia da mesma cena desde o ponto de vista perceptual. Muitos sistemas de renderização foto-realista são baseados no algoritmo de traçado de raios. O *traçado de raios* é um algoritmo simples e poderoso de renderização, pois além de ser um algoritmo de visualização, pode resolver de forma efetiva e relativamente simples vários fenômenos luminosos como sombras, reflexão e refração. O método é baseado no princípio da ótica geométrica em que a energia luminosa propaga-se em forma retilínea num meio isotrópico. A energia que chega ao observador numa direção é calculada lançando um raio nesta direção. Como o olho é sensível à radiância, e esta é constante ao longo do raio, a energia luminosa que o raio lançado transporta é caracterizada pela radiância que sai da superfície interceptada pelo raio. O algoritmo de *traçado de raios* simula o processo de propagação da luz no sentido inverso em que ele ocorre. Isto é feito por motivos de eficiência, já que só os raios que atingem o observador serão lançados, embora esta forma de proceder dificulta a simulação de alguns mecanismos de transporte de energia. A idéia

de utilizar o algoritmo de *traçado de raios* para o cálculo da iluminação foi sugerida, por Whitted [WHI80], como uma extensão ao algoritmo de visualização de *ray-casting* utilizado para determinar as superfícies visíveis da cena. Na geração de imagens por *traçado de raios* a intensidade de cada ponto da imagem é calculada lançando-se um raio em direção ao ponto desde a posição da câmera. Na primeira interseção deste raio com os objetos da cena é calculada a intensidade luminosa utilizando algum modelo de iluminação local. A contribuição luminosa de cada fonte é computada se o ponto do objeto for visível, isto é, não estiver na sombra. Isto é determinado lançado-se um raio desde o ponto em questão em direção a cada fonte, e verificando se ele não intercepta outros objetos. Se a superfície for especular o raio refletido é seguido de modo de capturar a radiância vinda de outras superfícies. Se a superfície for translúcida procede-se de forma similar com o raio refratado. Se bem é certo que há diversas maneiras de escrever um sistema baseado em *traçado de raios*, em geral todos eles devem simular os seguintes objetos e fenômenos: câmeras, intersecção raio-objeto, distribuição de luzes, visibilidade, espalhamento sobre superfícies, *traçado de raios* recursivo e propagação do raio.

4.1.1 Câmeras

Basicamente as câmeras determinam como e desde onde a cena é vista. As câmeras geram os raios do ponto de vista da cena. A câmara de furo consiste numa caixa obscura com um pequeno orifício num dos lados, pelo qual a luz entra na caixa e exita um papel fotográfico colocado no fundo da caixa, figura 4.1.

Em geral as câmeras são mais complexas do que uma câmera de furo, mas elas são

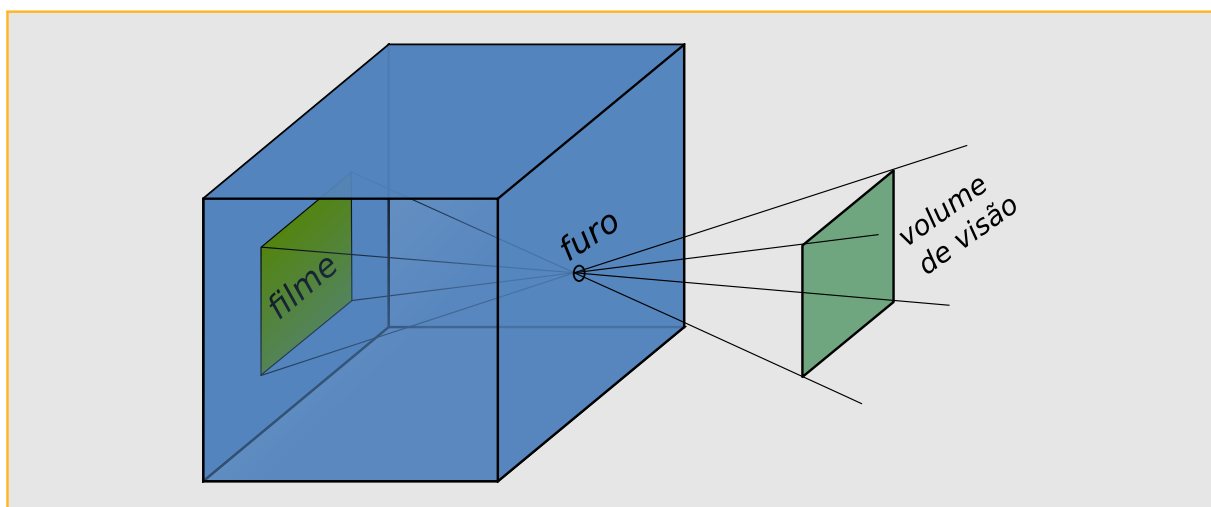


Figura 4.1: Câmera de furo.

convenientes como ponto de partida para a simulação. A função mais importante da câmera é definir a porção da cena que será gravada no filme. Na figura 4.1, vê-se que ligando o orifício com as bordas do filme determina uma pirâmide dupla, que se estende para a cena. Os objetos que não estão dentro desta pirâmide não pode ser impressas no filme. Dado que a câmeras modernas uma imagem mais complexa do que uma forma piramidal, é comum referir-se à região do espaço que pode potencialmente ser digitalizado para o filme como *volume de visualização*. Uma outra forma de pensar numa câmera de furo é colocando o plano do filme na frente do furo, como na figura 4.2. Ligando o furo com a borda do filme, obtém-se o volume de visualização. Este modelo não é prático no mundo real, mas é bem conveniente para simulações. Quando o plano da imagem é colocado na frente do furo, refere-se ao furo como olho ou ponto de vista. Uma pergunta freqüente é: em cada ponto da imagem qual é a cor que será armazenada? Considerando o modelo de câmera virtual, com o plano na frente do furo, basicamente temos interesse em saber quanta luz viaja desde o ponto da imagem até o olho.

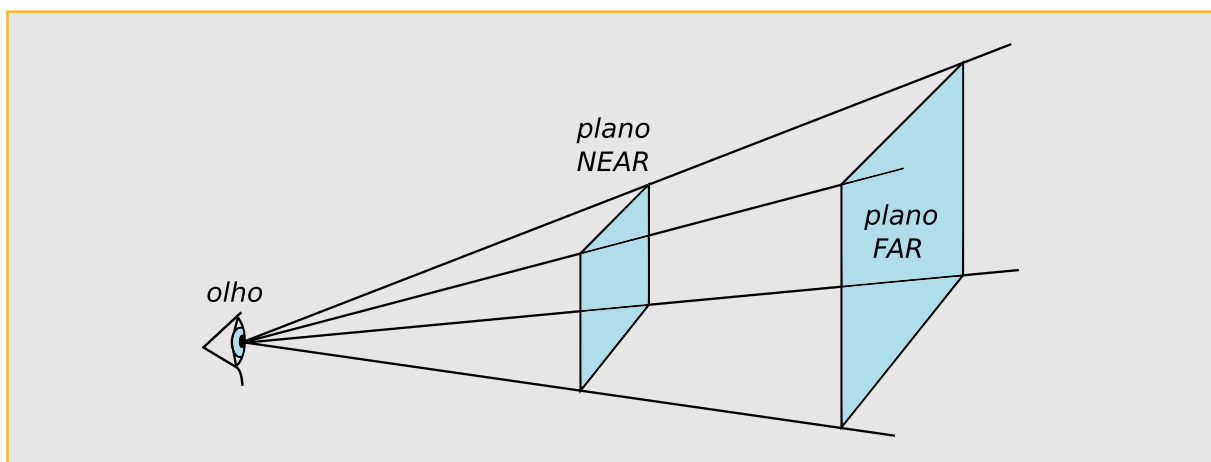


Figura 4.2: Câmera virtual: o plano da imagem está na frente do furo, que recebe o nome de *olho*.

4.1.2 Intersecções raio-objeto

Cada vez que a câmera gera um raio, a primeira tarefa do renderizador é determinar se o raio intersesta algum objeto da cena e qual é o primeiro objeto intersetado e onde a interseção acontece. Esta interseção é o ponto visível ao longo do raio que parte da câmera, e queremos simular a interação da luz com o objeto no ponto de interseção. Para achar a interseção, devemos testar a interseção do raio para todos os objetos da cena e selecionar o objeto que é intersetado primeiro. Dado um raio r , primeiro escrevemos o mesmo na forma paramétrica:

$$r(t) = o + t\mathbf{d},$$

onde o é a origem do raio (furo da câmera virtual), \mathbf{d} é o vetor direção, e $t \in [0, \infty)$ é o parâmetro. Podemos obter um ponto ao longo do raio especificando o parâmetro t .

4.1.3 Distribuição de luz

O estágio de interseção raio-objeto nos dá um ponto a ser colorido e algumas informações sobre a geometria neste ponto. Lembramos que o objetivo é encontrar a quantidade de luz saindo deste ponto na direção do olho (furo da câmera virtual). Para isso, precisamos de saber quanta luz está chegando a esse ponto. Isso envolve tanto aspectos geométricos quanto radiométricos da distribuição de luz na cena. Para cada fonte de luz simples (luzes pontuais), a distribuição geométrica da iluminação é obtida conhecendo a posição das luzes. No entanto, luzes pontuais não existem no mundo real, assim uma iluminação baseada na física é dada por fontes luminosas de área. Isto significa que a fonte de luz está associada a um objeto geométrico, que emite luz a partir de sua superfície.

Gostaríamos de saber a quantidade de energia luminosa que é depositada na área diferencial circundante ao ponto de interseção (figura 4.3). Vamos supor que a fonte de luz tem uma certa potência Φ , e que a fonte brilha igual em todas as direções. Isto significa que a quantidade total de energia em uma esfera unitária em torno da fonte de luz é $\Phi/(4\pi)$. Mas se tivermos em conta duas esferas, é claro que a energia num ponto esfera maior deve ser menor que a energia num ponto da esfera menor, porque a mesma quantidade total de energia é distribuída sobre uma área maior. Especificamente, a quantidade de energia num ponto de uma esfera de raio r é proporcional a $1/r^2$. Também pode-se provar que se uma pequena região dA é iluminada num ângulo θ , dado pelo vetor determinado entre o ponto da superfície e a fonte luminosa e a normal n à superfície dA , a quantidade de energia luminosa depositada em dA é proporcional a $\cos \theta$ (este fato é o responsável pelas quatro estações no planeta Terra). Colocando tudo isto, temos que a energia luminosa total dE depositada em dA é:

$$dE = \frac{\Phi \cos \theta}{4\pi r^2}.$$

As cenas com várias luzes podem ser tratadas facilmente por causa da iluminação ser linear: isto é, a contribuição de cada luz pode ser calculada separadamente e somar todas as contribuições no final para obter a contribuição total.

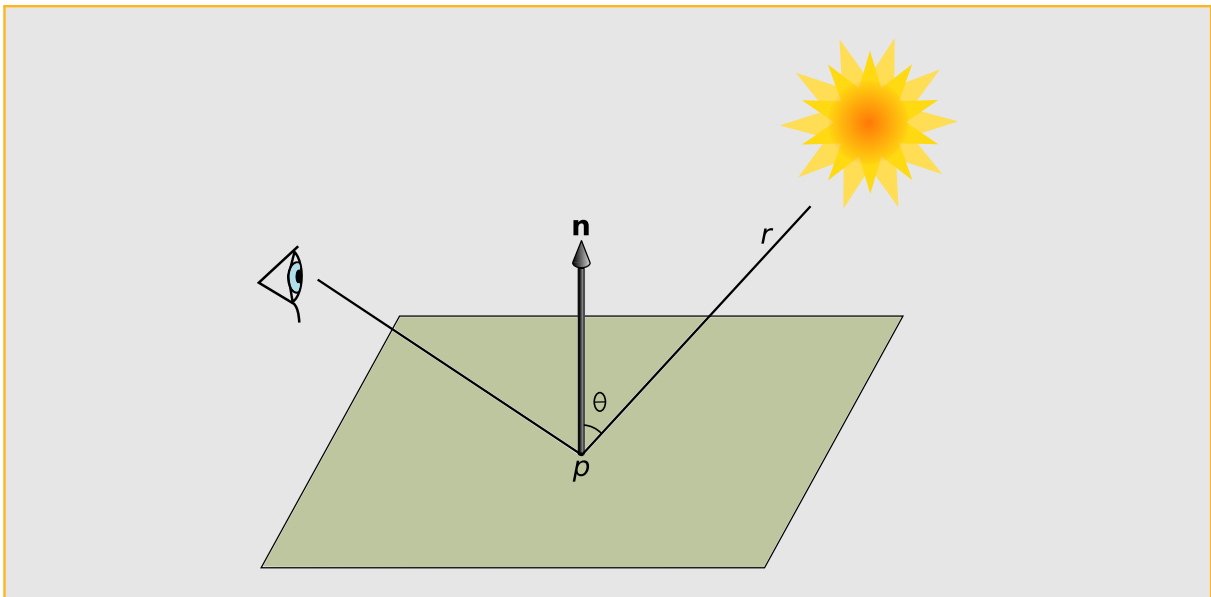


Figura 4.3: Construção geométrica para avaliar a energia luminosa num ponto proveniente desde uma fonte de luz. A distância desde o ponto até a fonte de luz é r .

4.1.4 Visibilidade

A distribuição de luz descrita acima ignora uma componente muito importante: as sombras. Cada luz contribui com a iluminação do ponto a ser sombreado apenas se o caminho do ponto à posição da luz não estiver obstruída. Afortunadamente num *traçador de raios* é fácil determinar se uma luz é visível ou não desde um ponto numa superfície. Simplesmente constrói-se um raio com origem no ponto na superfície e cuja direção aponta para a luz. Este raio é chamado de raio de sombra ou shadow ray. Se traçarmos este raio através do ambiente, podemos usar o valor paramétrico t encontrado pela rotina de intersecção raio-objeto para determinar a distância até o ponto de intersecção do raio de sombra. Se esta distância é maior que a distancia à fonte luminosa, ou se não foi encontrada nenhuma intersecção, o raio de sombra não é bloqueado e a fonte luminosa ilumina o ponto e portanto a contribuição desta luz é incluída no cálculo de iluminação do ponto.

4.1.5 Espalhamento sobre superfícies

Temos até agora duas informações vitais para o bom sombreado de um ponto: a sua localização e a iluminação incidente. Agora temos que determinar como a iluminação incidente está espalhada sobre a superfície. Especificamente, estamos interessados na quantidade de energia luminosa espalhada de volta ao longo do raio que originalmente

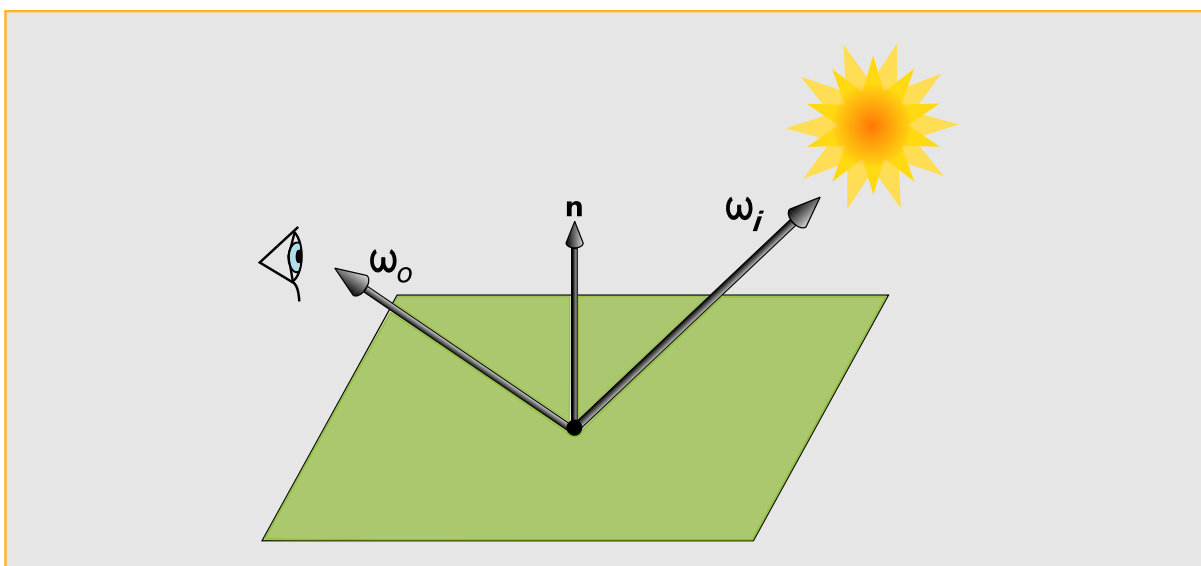


Figura 4.4: Geometria do espalhamento sobre superfícies. A luz chegando ao longo de uma direção ω_i interage com a superfície no ponto p e é espalhada para o olho ao longo da direção ω_o . A quantidade de luz direcionada para o olho é dada pelo produto da energia da luz incidente e a $BRDF$.

marcado para encontrar a intersecção ponto, uma vez que leva ao raio do olho.

Cada objeto na cena fornece um material, que é uma descrição de sua aparência propriedades em cada ponto na superfície. Esta descrição é dada pela ($BRDF$ - Bidirectional Reflectance Distribution Function). Esta função nos diz quanta energia é refletida a partir de uma direção de entrada ω_i numa determinada direção de saída ω_o . A $BRDF$ no ponto p será escrita como $f_r(p, \omega_o, \omega_i)$. Se temos L_i a luz incidente no ponto p , e a quantidade refletida é $r = f_r(p, \omega_o, \omega_i)$, a luz espalhada de volta para o olho é $L = r \times L_i$.

E fácil generalizar a noção de $BRDF$ para a luz transmitida ($BTDF$) ou para o espalhamento em geral para luz proveniente de qualquer dos lados da superfície. Uma tal função que descreve o espalhamento em geral é chamada em inglês de *Bidirectional Scattering Distribution Function* ($BSDF$).

4.1.6 Traçado de raios recursivo

O artigo original de Turner Whitted sobre traçado de raios enfatizou sua natureza recursiva. Por exemplo, se um raio a partir do olho (furo da câmera) atinge um objeto brilhante como um espelho, podemos refletir o raio sobre a normal da superfície no ponto de intersecção e chamar recursivamente a rotina do traçado de raios para encontrar a luz que chega ao ponto sobre o espelho, adicionando sua contribuição ao raio original

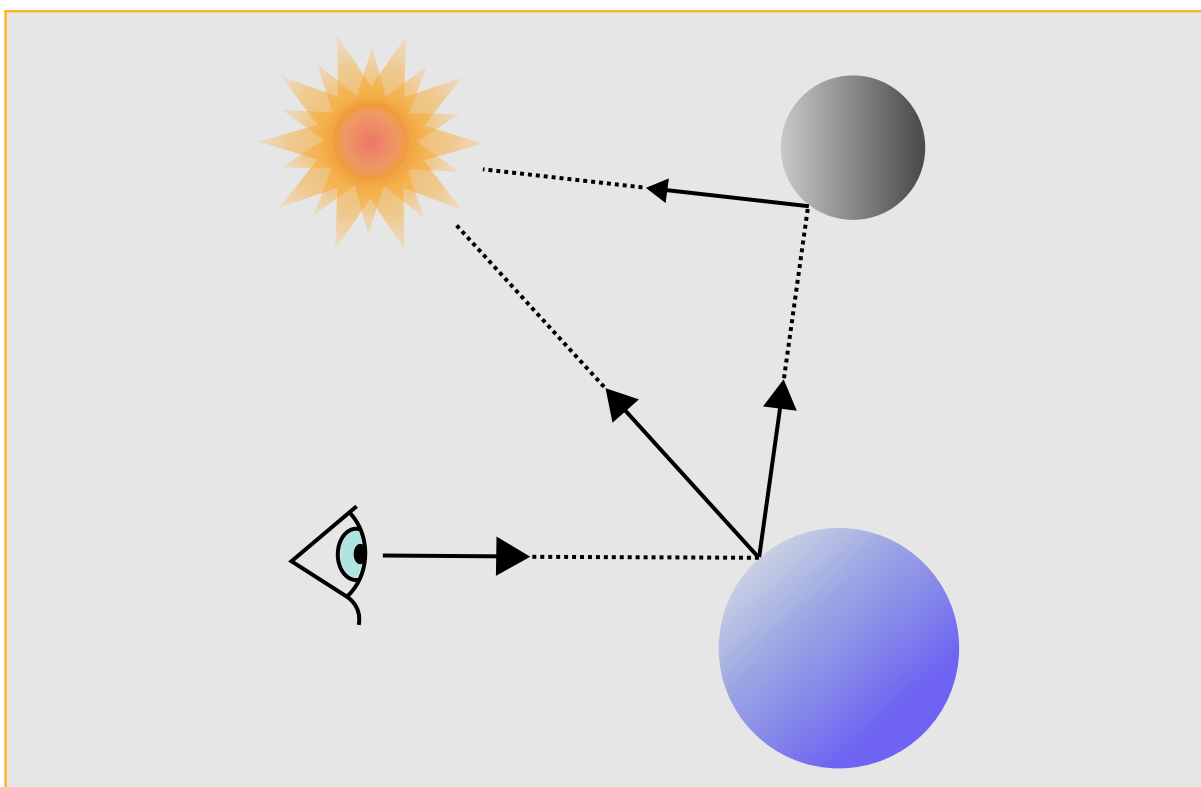


Figura 4.5: O traçado de raios recursivo associa uma árvore de raios para cada ponto da imagem.

proveniente da câmara. Esta mesma técnica pode ser usada para traçar raios transmitidos que interceptam objetos transparentes. Durante muito tempo, os exemplos mais comuns de traçado de raios eram espelhos e bolas de vidro porque estes tipos de efeitos são difíceis de capturar com outras técnicas renderização.

Em geral, a quantidade de luz que chega ao olho desde um ponto sobre um objeto é dada pela soma da luz emitida e luz refletida. Esta idéia é formalizada pela equação de transporte de luz, que afirma que a radiância saliente $L_o(p, \omega_o)$ a partir de um ponto p na direção ω_o é a radiância emitida nesse ponto nessa direção, $L_e(p, \omega_o)$, somada a radiância incidente de todas as direções da esfera S^2 em torno de p multiplicados pela $BSDF$ e um termo do cosseno:

$$L_o(p, \omega_o) = \int_{S^2} f(p, \omega_o, \omega_i) L_i(p, \omega_i) |\cos \theta_i| d\omega_i.$$

Mais adiante neste capítulo analisaremos esta equação com maiores detalhes. O algoritmo de Whitted aproxima esta integral ignorando a luz que chega desde muitas direções e avalia somente $L_i(p, \omega_i)$ para as direções onde tem fontes de luz e para as direções de reflexão perfeita e de refração, assim a integral é uma soma sobre uma quantidade relati-

vamente pequena de direções. O algoritmo de Whitted pode ser estendido de maneira a contemplar mais efeitos. Uma alternativa é traçar vários raios recursivos perto da direção de reflexão especular, e somando suas contribuições, obtém-se o efeito de uma reflexão não especular brilhante. Também pode-se escolher aleatoriamente direções de reflexão ω_i e somar o peso de sua contribuição dado pela função *BRDF* $f_r(p, \omega_o, \omega_i)$. Este procedimento possibilita a obtenção de imagens incrivelmente realistas, porém o algoritmo converge muito devagar para resultados satisfatórios.

4.1.7 Propagação dos raios

Até agora assumimos que os raios se propagam no vazio. Ao descrever a distribuição da luz a partir de uma fonte pontual, assumimos que a energia era distribuída igualmente sobre a esfera centrada na luz sem decrescer ao longo do caminho. A presença de um meio participativo, como fumaça, nevoa ou poeira pode invalidar esta suposição. Tem duas formas em que um meio participativo pode afetar a propagação da luz ao longo de um raio. Primeiro, o meio pode atenuar ou extinguir o raio, seja absorvendo-a ou desviando-a para outra direção. Este efeito pode ser adicionado computando a transmitância T entre a origem do raio e o ponto de interseção. A transmitância nos diz qual é o percentual de luz bloqueado ou desviado ao longo do raio. Um meio participativo pode ser adicionado à luz ao longo de um raio. Isto pode ser feito se o meio emite luz, ou se o meio espalha iluminação proveniente de outras direções (ambiente de poeira). A quantidade pode ser obtida calculando numericamente a equação volumétrica de transporte de luz, da mesma forma que se avalia a equação de transporte de luz para obter a quantidade de luz refletida por uma superfície.

4.2 Transporte de luz

A iluminação de uma cena é o resultado da interação global entre as fontes luminosas e as superfícies da cena. Neste capítulo vamos descrever a equação de iluminação ou equação de transporte de luz, a qual descreve os fenômenos de interação na cena. Como a equação de transporte somente pode ser resolvida completamente para cenas triviais, em geral é necessário aplicar métodos de integração numérica para aproximar a solução. Este ponto foi motivo de muita pesquisa na área de renderização, e atualmente existem diversos métodos para abordar o problema, entretanto neste capítulo apresentamos somente uma breve descrição teórica da equação de transporte, e alguns métodos de para resolver o

problema, baseados no método de integração de Monte Carlo, implementados no software *PBRT* utilizado como suporte nesta dissertação.

4.2.1 Iluminação direta

Antes de apresentar formalmente a equação de transporte de luz, apresentaremos um método *integrador* que somente leva em conta a iluminação direta, i.e., calcula as contribuições das fontes luminosas emissoras e ignora iluminação indireta de objetos não emissores, como por exemplo os reflexos de uma fonte luminosa sobre uma superfície. A implementação sugerida no *PBRT* [PH04b] apresenta duas estratégias para o cálculo da iluminação direta. Cada um dos métodos calcula uma estimativa imparcial da radiância partindo de um ponto numa direção dada. A primeira estratégia faz um laço sobre todas as luzes e toma as amostras indicadas para cada uma delas e soma o resultado. A segunda toma uma única amostra de uma única luz, escolhida aleatoriamente.

Segundo o tipo de cena que for renderizada, algum destes enfoques é mais apropriado. Por exemplo, no caso de ser preciso várias amostras por pixel (ex., cenas com profundidade de campo), é melhor tomar uma amostra de luz, pois no final teremos suficientes amostras para um bom resultado. No caso de tomar poucas amostras por pixel, é melhor amostrar todas as luzes.

O número e o tipo de amostras solicitadas pelo *integrador* dependerá da estratégia utilizada:

```
⟨Método Iluminação Direta⟩ ≡  
void RequistarAmostras(Sample *amostra, const Scene *cena) {  
    if ( estrategia == AMOSTRAR_TUDO ) {  
        ⟨ Aloca espaço e requisita amostras para todas as luzes ⟩  
    }  
    else {  
        ⟨ Aloca espaço e requisita amostras para uma luz ⟩  
    }  
}
```

No caso de ter que amostrar todas as fontes de luz, o integrador do *PBRT* trabalha utilizando um modelo bidimensional para selecionar os pontos nas fontes de luz e outro para selecionar direções da *BSDF*; logo ambas as abordagens são combinadas com amostragem por importância múltipla.

A equação de espalhamento é a peça fundamental na renderização, ela descreve como uma distribuição de luz incidente num ponto é transformada numa distribuição de saída, baseado nas propriedades de espalhamento da superfície.¹

$$L_o(p, \omega_o) = \int_{S^2} f(p, \omega_o, \omega_i) L_i(p, \omega_i) |\cos \theta_i| d\omega_i. \quad (4.1)$$

A equação de espalhamento 4.1 diz que a radiância $L_o(p, \omega_o)$ de um ponto p numa superfície na direção ω_o é a soma da radiância emitida pela superfície no ponto mais a integral da radiância entrante sobre a esfera, multiplicado por *BSDF* para cada direção e o termo do cosseno do ângulo entre a normal à superfície em p e a direção ω_i determinada pela fonte de luz e o ponto p . No *integrador DirectLighting* do *PBRT*, o interesse está apenas na radiância incidente diretamente das fontes luminosas, denotado por $L_d(p, \omega)$:

$$L_o(p, \omega_o) = L_e(p, \omega_o) + \int_{S^2} f(p, \omega_o, \omega_i) L_d(p, \omega_i) |\cos \theta_i| d\omega_i. \quad (4.2)$$

O valor de $L_e(p, \omega_o)$ é facilmente calculado chamando a rotina `intersection:Le()` no ponto de interseção, que retorna a radiância emitida pelo ponto p , se ele for uma fonte luminosa; e zero se p não for emissor. Para estimar a integral sobre a esfera, aplica-se integração de Monte Carlo.

Consideremos a seguinte parte da equação de iluminação direta:

$$\int_{S^2} f(p, \omega_o, \omega_i) L_i(p, \omega_i) |\cos \theta_i| d\omega_i.$$

A integral pode ser quebrada num somatório sobre as luzes da cena:

$$\sum_{j=1}^{\text{luzes}} \int_{S^2} f(p, \omega_o, \omega_i) L_{d(j)}(p, \omega_i) |\cos \theta_i| d\omega_i,$$

onde $L_{d(j)}$ denota a radiância incidente em p proveniente da j -ésima fonte de luz e

¹Para maiores detalhes sobre a equação de espalhamento, veja [PH04b] (Capítulo 5).

$$L_d(p, \omega_i) = \sum_j L_{d(j)}(p, \omega_i).$$

Uma forma de fazer o cálculo é estimar cada termo do somatório individualmente e somar os resultados no final. Esta idéia básica é implementada no integrador do *PBRT* na hora de amostrar todas as luzes, mediante a rotina global `UniformSampleAllLights()`, que utiliza a rotina `EstimateDirect()`. O método `EstimateDirect()`, é utilizado para calcular uma estimativa de Monte Carlo do efeito de uma amostra de uma luz. No final, o método `UniformSampleAllLights()` faz uma média entre todos os valores das estimativas dadas por `EstimateDirect()`.

Numa cena com grande quantidade de luzes, não é conveniente computar sempre a iluminação direta de todas as luzes em cada ponto. O método de Monte Carlo nos fornece de uma forma de fazer isto de uma maneira que calcula o resultado correto na média. Consideremos como exemplo calcular o valor esperado da soma de duas funções $E[f(x) + g(x)]$. Se avaliarmos aleatoriamente somente $f(x)$ ou $g(x)$ e multiplicamos por 2, então o valor esperado do resultado continuará sendo $f(x) + g(x)$. De fato, isto pode ser generalizado para somas de N termos². Aqui a iluminação direta será estimada para uma só amostra de luz escolhida aleatoriamente e o resultado será multiplicado pelo número de luzes para compensar.

4.2.1.1 Estimando a integral de iluminação direta

Tendo escolhido uma luz em particular para estimar a iluminação direta dada por ela, devemos estimar a integral

$$\int_{S^2} f(p, \omega_o, \omega_i) L_{d(j)}(p, \omega_i) |\cos \theta_i| d\omega_i,$$

para esta luz. Para calcular a estimativa, precisamos escolher uma ou mais direções ω_i e aplicar a estimativa de Monte Carlo

$$\frac{1}{N} \sum_{j=1}^N \frac{f(p, \omega_o, \omega_j) L_d(p, \omega_j) |\cos \theta_j|}{p(\omega_j)}.$$

Para reduzir a variância utiliza-se amostragem por importância para escolher as dire-

²Isto é uma aplicação de probabilidade condicional; veja [ROS03] (p. 102)

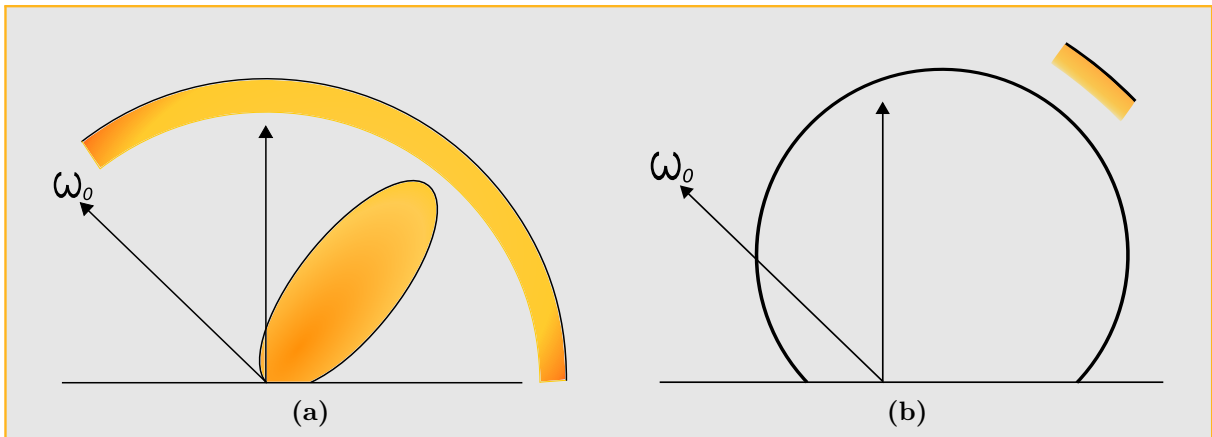


Figura 4.6: Dependendo da distribuição da *BSDF* e da fonte de luz como uma função da direção a partir do ponto a ser iluminado, uma ou outra distribuição pode ser muito mais eficaz para realizar amostragem por importância.

ções ω_j . Devido a que tanto o termos da *BSDF* e da radiância direta são individualmente complexos, torna-se difícil encontrar distribuições de amostragem que aproximem bem o produto de ambos os termos. Para salvar esta dificuldade, utiliza-se a distribuição de amostragem *BSDF* para algumas amostras e as luzes para as restantes. Dependendo das características de cada um deles, um destes métodos de amostragem é mais efetivo do que o outro. Por isso, utiliza-se amostragem múltipla por importância para melhorar os resultados.

Na figura 4.6 são apresentados dois casos, que exemplificam onde os métodos se superam mutuamente. Na figura 4.6a a *BSDF* é muito especular e a fonte de luz é relativamente grande e espalhada. Amostrar a *BSDF* é mais efetivo para achar direções onde o valor do integrando é maior, enquanto que amostrar a fonte de luz é menos efetivo, uma vez que várias amostras não vão contribuir quase nada com a integral, pois a *BSDF* é pequena para a maioria das direções de fontes luminosas. Quando a luz é amostrada num ponto de uma região brilhante da *BSDF*, haverá pico na imagem, pois a luz devolverá um valor de *PDF* baixo, enquanto que o valor do integrando é relativamente grande. Como resultado, a variância é grande por causa de que a distribuição da amostragem não coincide muito bem com a distribuição real dos valores das funções.

Por outro lado, às vezes amostrar a luz é a estratégia certa. Na figura 4.6b a *BSDF* é não nula sobre muitas direções, e a luz é relativamente pequena. É mais efetivo neste caso amostrar pontos na fonte de luz para computar ω_i , desde que amostrando a *BSDF* terá maiores problemas para se achar direções onde a radiância proveniente da luz é não nula. Igualmente ao primeiro caso encontra-se variância sustancial, desde que a distribuição das

amostras não corresponde muito bem com a função geral.

Aplicando amostragem múltipla por importância, não somente é possível utilizar ambos os métodos descritos, mas fazer isto de forma a eliminar a variância extrema destas situações, dado que os termos ponderados do *MIS* (amostragem múltipla por importância) reduzem a variância substancialmente.³

4.2.2 Equação de transporte de luz

A *equação de transporte de luz (LTE)* descreve o equilíbrio da distribuição da radiação na cena. Ela dá a radiação total refletida num ponto de uma superfície em termos da emissão da superfície, sua *BSDF*, e a distribuição da iluminação incidente que chega no ponto. A tarefa principal dos objetos *Integrator* no software *PBRT* é calcular numericamente uma solução da *LTE* para determinar a radiação incidente sobre a câmera. Examinaremos somente o caso em que não há meios participativos na cena.

O que dificulta o cálculo da *LTE* é o fato de que a radiação incidente num ponto vê-se afetada pela geometria e as propriedades de dispersão dos objetos da cena. Por exemplo, uma luz brilhante de cor azul sobre um objeto pode causar um tom azulado sobre os objetos próximos na cena. Os algoritmos que dão conta desta complexidade são conhecidos como algoritmos de *iluminação global*, para diferenciá-los dos algoritmos de iluminação local, que somente podem utilizar a informação das propriedades locais da superfície nos cálculos de sombreamento.

Nesta seção, vamos derivar a equação de transporte de luz *LTE* e descrever alguns critérios sobre a manipulação da equação para que esta seja mais fácil de resolver numericamente. Em seguida, descreveremos duas generalizações da *LTE* que deixaram mais claras suas principais propriedades e vão servir de base para alguns dos integradores implementados no *PBRT*, apresentado mais adiante nesta seção.

4.2.2.1 Obtendo a equação

A *equação de transporte de luz* depende das seguintes suposições básicas feitas ao utilizar a radiometria⁴ para descrever a luz: os efeitos das ondas ópticas não são importantes e a distribuição da radiação na cena está em equilíbrio. Para calcular a radiação que

³Informação detalhada sobre a implementação do integrador *DirectLighting* pode ser encontrada em [PH04b] (capítulo 16)

⁴Veja [PH04b] (capítulo 5), para uma descrição detalhada dos conceitos radiométricos e as implementações feitas no *PBRT*

chega ao filme ao longo de um raio da câmera; a radiância saindo do ponto de interseção p , do raio da câmera, na direção ω_o desde p para a amostra no filme, será denotada por $L_o(p, \omega_o)$.

O princípio fundamental subjacente à *LTE* é o *balanço de energia*. Qualquer mudança na energia tem que ser associado a algum processo, e deve-se fazer um seguimento de toda a energia. Como estamos supondo que a iluminação é um processo linear, a diferença entre a quantidade de energia que entra e a energia que sai do sistema, deve ser igual à diferença entre a energia emitida e a energia absorvida. Esta idéia tem vários níveis de escala. Num nível macro temos a conservação da potência:

$$\Phi_o - \Phi_i = \Phi_e - \Phi_a.$$

A diferença entre a potência saindo de um objeto, Φ_o , e a potência entrando no mesmo, Φ_i , é igual à diferença entre a potência emitida e a potência absorvida, $\Phi_e - \Phi_a$.

Para satisfazer o balanço de energia numa superfície, a radiância partindo da superfície deve ser igual à radiância emitida, somada a fração da radiância incidente que é dispersada pela superfície. A radiância emitida é dada por L_e , e a dispersão da radiância é dada pela equação de dispersão, obtendo:

$$L_o(p, \omega_o) = L_e(p, \omega_o) + \int_{S^2} f(p, \omega_o, \omega_i) L_i(p, \omega_i) |\cos \theta_i| d\omega_i.$$

Como não estamos considerando meios participativos, a radiância é constante ao longo dos raios através da cena. Portanto, podemos relacionar a radiância incidente em p com a radiância partindo desde outro ponto p' , tal como na figura 4.7. Se definirmos a *função ray-casting* $t(p, \omega)$ como uma função que calcula o primeiro ponto p' da superfície, que é intersetada por um raio proveniente de p na direção ω , podemos escrever a radiância incidente no ponto p em termos da radiância partindo em p' :

$$L_i(p, \omega) = L_o(t(p, \omega), -\omega).$$

No caso em que a cena não seja fechada, vamos definir a *função ray-casting* para devolver um valor especial Λ , se o raio (p, ω) não interseta nenhum objeto da cena, de tal forma que $L_o(\Lambda, \omega)$ é sempre zero. Tirando os sub-índices de L_o a fim de abreviar, a

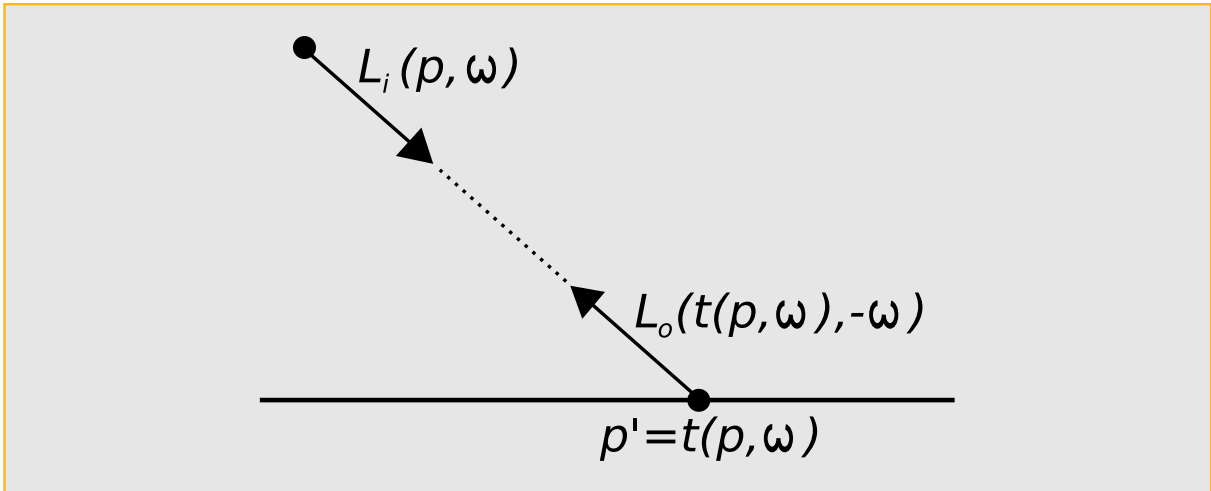


Figura 4.7: Radiância constante ao longo de um raio no vácuo. Para calcular a radiância incidente ao longo de um raio desde um ponto p na direção ω , pode-se achar a primeira superfície que o raio intersesta e calcular a radiância escapando na direção $-\omega$. O operador $t(p, \omega)$ retorna o ponto p' na primeira superfície que o raio (p, ω) intersesta.

relação anterior permite escrever a *LTE* como:

$$L(p, \omega_o) = L_e(p, \omega_o) + \int_{S^2} f(p, \omega_o, \omega_i) L(t(p, \omega_i), -\omega_i) |\cos\theta_i| d\omega_i. \quad (4.3)$$

O ponto chave na representação acima é que somente há uma quantidade de interesse, a radiância que sai dos pontos das superfícies. Claro que, ainda aparece em ambos os lados da equação, de modo que a tarefa não é fácil. É importante ter em conta que chegamos nesta equação mediante a aplicação do balanço da energia na cena.

4.2.2.2 A forma de superfície da LTE

Uma das razões que fazem da forma da *LTE* na equação 4.3 ser complexa é que a relação entre os objetos geométricos da cena está implícita no operador de traçado de raio $t(p, \omega)$. Fazendo com que o operador de traçado de raios trabalhe de forma explícita no integrando, podemos conseguir alguma idéia sobre a estrutura da equação. Para isto, vamos reescrever a equação 4.3 como uma integral de área em lugar de uma integral sobre direções sobre a esfera.

Em primeiro lugar, definimos a radiância partindo desde um ponto p' para um ponto p como:

$$L(p' \rightarrow p) = L(p', \omega),$$

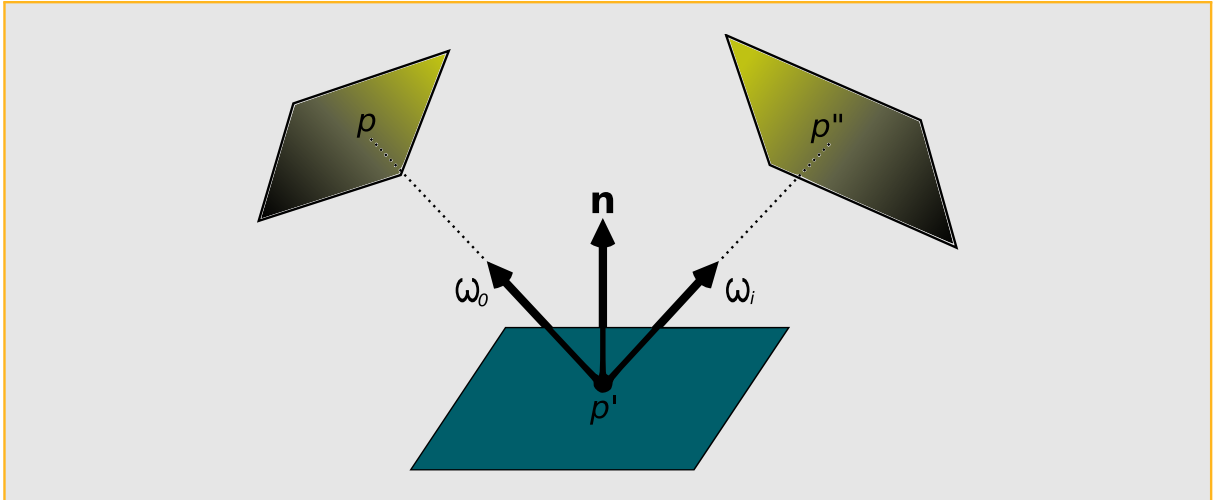


Figura 4.8: A forma da equação de transporte para três pontos converte a integral para ser calculada sobre o domínio dos pontos nas superfícies da cena, em lugar das direções sobre a esfera. É a transformação base para derivar a forma integral de caminhos da equação de transporte de luz.

se p' e p forem mutuamente visíveis e $\omega = \widehat{p - p'}$. Também podemos escrever a $BPDF$ no ponto p' como:

$$f(p'' \rightarrow p' \rightarrow p) = f(p', \omega_o, \omega_i),$$

onde $\omega_i = \widehat{p'' - p'}$ e $\omega_o = \widehat{p - p'}$ (figura 4.8).

Reescrevendo os termos na LTE desta forma não é o bastante. Também temos que multiplicar pelo Jacobiano que relaciona ângulo sólido com área para passar a LTE de uma integral de direções para uma integral de área, i.e, fazer a mudança de variável. Vale lembrar que o mesmo é $|\cos \theta'|/r^2$.

Vamos combinar esta mudança de variáveis, o termo original $|\cos \theta|$ da LTE , e uma função binária de visibilidade V ($V = 1$ se dois pontos são mutuamente visíveis, e $V = 0$ caso contrário), num único termo $G(p \leftrightarrow p')$:

$$G(p \leftrightarrow p') = V(p \leftrightarrow p') \frac{|\cos \theta| |\cos \theta'|}{\|p - p'\|^2}.$$

Substituindo isto na equação de transporte e convertendo-a numa integral de área, temos:

$$L(p' \rightarrow p) = L_e(p' \rightarrow p) + \int_A f(p'' \rightarrow p' \rightarrow p)L(p'' \rightarrow p')G(p'' \leftrightarrow p')dA(p''), \quad (4.4)$$

onde: A é superfície total da cena.

Mesmo que as equações 4.3 e 4.4 sejam equivalentes, elas representam duas maneiras diferentes de aproximar o transporte da luz. Para avaliar a equação 4.3 com Monte Carlo, devemos amostrar direções a partir de uma distribuição de direções sobre a esfera e raios emitidos para avaliar o integrando. Na equação 4.4, por outro lado, escolhe-se um número de pontos nas superfícies, em função da distribuição de área, e calcula-se o acoplamento entre os pontos para avaliar o integrando. Os raios são traçados para avaliar o termo de visibilidade $V(p \leftrightarrow p')$.

4.2.2.3 Integral sobre caminhos

Para passar de uma integral de área para uma forma mais flexível da *LTE*, uma soma sobre caminhos de transporte de luz de diferentes longitudes, é possível começar a expandir a equação de transporte de luz de três pontos, fazendo a substituição sucessiva do lado direito da equação no termo $L(p'' \rightarrow p')$ dentro da integral. Em seguida, apresentamos os primeiros termos que dão a radiância incidente no ponto p_0 desde o ponto p_1 , onde p_1 é o primeiro ponto sobre uma superfície ao longo do raio partindo de p_0 na direção $p_1 - p_0$:

$$\begin{aligned} L(p_1 \rightarrow p_0) = & L_e(p_1 \rightarrow p_0) + \\ & \int_A L_e(p_2 \rightarrow p_1) f(p_2 \rightarrow p_1 \rightarrow p_0) G(p_2 \leftrightarrow p_1) dA(p_2) + \\ & \int_A \int_A L_e(p_3 \rightarrow p_2) f(p_3 \rightarrow p_2 \rightarrow p_1) G(p_3 \leftrightarrow p_2) \\ & f(p_2 \rightarrow p_1 \rightarrow p_0) G(p_2 \leftrightarrow p_1) dA(p_3) dA(p_2) + \dots \end{aligned}$$

Cada termo do lado direito da equação representa um caminho de comprimento crescente. Por exemplo, o terceiro termo é ilustrado na figura 4.9. Este caminho tem 4 vértices, conetados por três segmentos. A contribuição total de todos os caminhos de longitude 4 (i.e., um vértice no olho (câmera), dois vértices em superfícies da cena, e um vértice numa fonte de luz) é dado pelo terceiro termo. Aqui, os dois primeiros vértices do caminho, p_0 e p_1 , são predeterminados pelo raio da câmera e o ponto que este intersecta na cena, mas p_2 e p_3 podem variar sobre todos os pontos das superfícies da cena. A integral sobre todos os pontos p_2 e p_3 dá a contribuição total de radiância dos caminhos de longitude 4.

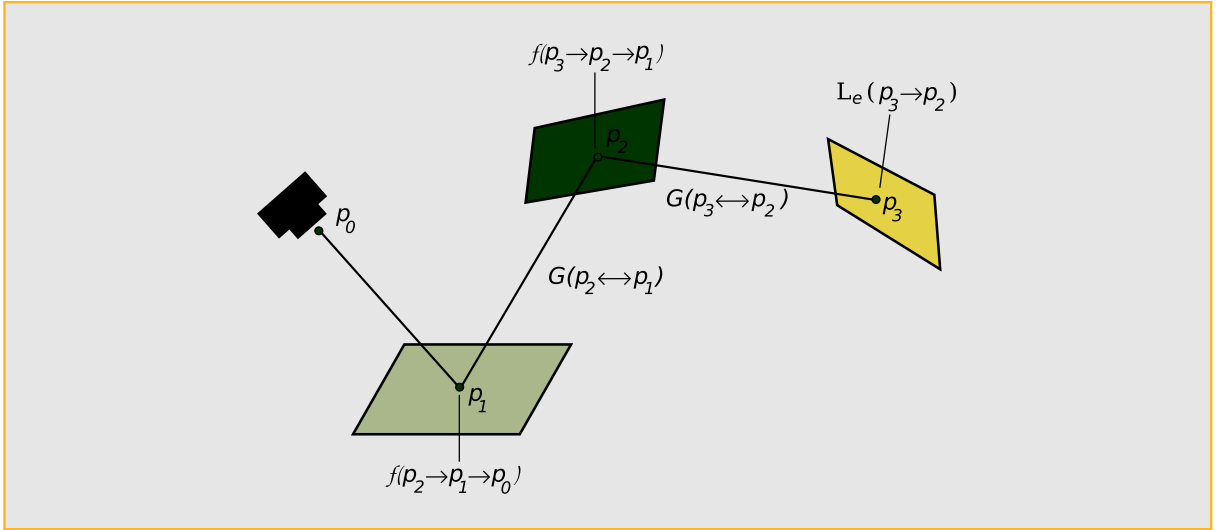


Figura 4.9: A integral sobre todos os pontos p_2 e p_3 em superfícies na cena dadas pela equação de transporte de luz fornece a contribuição total de radiância, partindo de p_1 na direção de p_0 , dada por dois segmentos de caminho. As componentes do produto do integrando são: a radiância emitida pela fonte luminosa L_e , os termos geométricos entre vértices G , e o espalhamento das *BSDFs*, f .

Este somatório infinito pode ser escrito da seguinte maneira:

$$L(p_1 \rightarrow p_0) = \sum_{i=1}^{\infty} P(\bar{p}_i). \quad (4.5)$$

$P(\bar{p}_i)$ representa a quantidade de radiância espalhada ao longo de um caminho \bar{p}_i com $i + 1$ vértices,

$$\bar{p}_i = p_0, p_1, \dots, p_i,$$

onde p_0 está no plano da imagem e p_i está sobre a fonte de luz, e

$$P(\bar{p}_i) = \underbrace{\int_A \int_A \cdots \int_A}_{i-1} L_e(p_i \rightarrow p_{i-1}) \left(\prod_{j=1}^{i-1} f(p_{j+1} \rightarrow p_j \rightarrow p_{j-1}) G(p_{j+1} \leftrightarrow p_j) \right) dA(p_2) \cdots dA(p_i).$$

Antes de continuar, vamos definir outro termo de utilidade. O produto entre a *BSDF* de um caminho e os termos da geometria é chamado de *rendimento* do caminho, e descreve a porção de radiância da fonte de luz que chega até a câmera depois de sofrer difusão em todos os vértices ao longo do caminho. Vamos denotar este termo como:

$$T(\bar{p}_i) = \prod_{j=1}^{i-1} f(p_{j+1} \rightarrow p_j \rightarrow p_{j-1}) G(p_{j+1} \leftrightarrow p_j),$$

obtendo assim:

$$P(\bar{p}_i) = \underbrace{\int_A \int_A \cdots \int_A}_{i-1} L_e(p_i \rightarrow p_{i-1}) T(\bar{p}_i) dA(p_2) \cdots dA(p_i).$$

Dada a equação 4.5 e um determinado comprimento i , tudo o que precisamos fazer para calcular uma estimativa de Monte Carlo da radiância, que chega ao ponto p_0 através dos caminhos de comprimento i , é amostrar um conjunto de vértices com uma densidade de amostragem apropriada na cena para gerar um caminho, e estimar $P(\bar{p}_i)$ usando os vértices amostrados. Podemos gerar os vértices iniciando um percurso a partir da câmara, a partir da luz, a partir de ambas as extremidades, ou a partir de um ponto no meio. Isto é um detalhe que apenas afeta a forma como os pesos para as estimativas de Monte Carlo são calculadas.

4.2.2.4 Distribuições delta no integrando

Funções Delta podem estar presentes nos termos de $P(\bar{p}_i)$, devido a que tanto componentes da $BSDF$ como certos tipos de fontes luminosas são descritos com distribuições delta (por exemplo, luzes pontuais e luzes direcionais). Se presentes, essas distribuições precisam ser tratados de forma explícita pelo algoritmo de transporte de luz. Por exemplo, é impossível escolher aleatoriamente uma direção de saída a partir de um ponto sobre uma superfície tal que interceptam uma fonte luminosa pontual; portanto é necessário escolher explicitamente a direção a partir do ponto para a fonte luminosa, a fim de incluir a sua contribuição. (O mesmo vale para a escolha de amostras de $BSDFs$ com componentes delta.) Enquanto a manipulação deste caso introduz alguma complexidade nos integradores, a mesma também reduz a dimensão da integral a ser avaliada, transformando algumas partes dela num simples somatório.

Considere, por exemplo, o termo iluminação direta, $P(\bar{p}_2)$, numa cena com uma única fonte luminosa pontual no ponto p_{luz} descrito por uma distribuição delta:

$$\begin{aligned}
P(\bar{p}_2) &= \int_A L_e(p_2 \rightarrow p_1) f(p_2 \rightarrow p_1 \rightarrow p_0) G(p_2 \leftrightarrow p_1) dA(p_2) \\
&= \frac{\delta(p_{luz} - p_2) L_e(p_{luz} \rightarrow p_1)}{p(p_{luz})} f(p_2 \rightarrow p_1 \rightarrow p_0) G(p_2 \leftrightarrow p_1)
\end{aligned}$$

Em outras palavras, p_2 deve ser a posição da luz na cena, a distribuição delta no numerador e cancelada devido a uma distribuição delta implícita⁵ em p_{luz} , e ficamos com termos que podem ser avaliados diretamente, sem a necessidade de Monte Carlo. Uma situação análoga vale para *BSDF* com distribuições delta no *rendimento* do caminho $T(\bar{p}_i)$; cada uma elimina uma integral de área da estimativa a ser calculada.

4.2.2.5 Particionamento do integrando

Muitos algoritmos de renderização desenvolvidos são particularmente bons na solução da *LTE* baixo certas condições, mas não funcionam bem para outras. Por exemplo, o integrador *Whitted* somente trata a reflexão especular de *BSDFs* delta e não multiplica a luz dispersada por *BSDFs* difusas e brilhantes, e a técnica de *Cache de Irradiância* ocupa-se da dispersão de superfícies difusas, mas introduz importantes erros se utilizada para reflexão especular o brilhante.

Como o objetivo é obter algoritmos de transporte de luz corretos, capazes de calcular todas formas possíveis de dispersão sem omitir contribuições nem fazer contagens duplas, é importante observar cuidadosamente que partes da *LTE* um método em particular consegue calcular. Uma boa forma de tratar este problema é particionando a *LTE* de variadas formas. Por exemplo, poderia-se ampliar o somatório dos caminhos para:

$$L(p_1 \rightarrow p_0) = P(\bar{p}_1) + P(\bar{p}_2) + \sum_{i=3}^{\infty} P(\bar{p}_i),$$

onde o primeiro termo é obtido trivialmente calculando a radiância emitida em p_1 ; o segundo termo é obtido aplicando uma técnica de iluminação direta; mas os termos restantes são tratados com um enfoque mais rápido, porém menos preciso. Se a contribuição destes termos adicionais à radiância total refletida é relativamente pequena, este enfoque é válido. Um detalhe importante a ter em conta é ignorar $P(\bar{p}_1)$ e $P(\bar{p}_2)$ no algoritmo que calcula $P(\bar{p}_3)$ e os termos subsequentes.

⁵Uma situação semelhante acontece nas reflexões especulares. Para mais detalhes sobre situações de cancelamento de deltas de Dirac, veja [PH04b] (seção 15.5.4).

Também é útil particionar os termos individuais $P(\bar{p}_i)$. Por exemplo, dividir o termo de emissão segundo emissões de pequenas fontes de luz, $L_{e,s}$, e a emissão de grandes fontes de luz, $L_{e,l}$, obtendo duas integrais para calcular:

$$\begin{aligned} P(\bar{p}_i) &= \int_{A^{i-1}} (L_{e,s}(p_i \rightarrow p_{i-1}) + L_{e,l}(p_i \rightarrow p_{i-1})) T(\bar{p}_i) dA(p_2) \cdots dA(p_i) \\ &= \int_{A^i} L_{e,s}(p_i \rightarrow p_{i-1}) T(\bar{p}_i) dA(p_2) \cdots dA(p_i) + \\ &\quad \int_{A^i} L_{e,l}(p_i \rightarrow p_{i-1}) T(\bar{p}_i) dA(p_2) \cdots dA(p_i). \end{aligned}$$

As duas integrais podem ser calculadas independentemente, possivelmente utilizando algoritmos diferentes, ou diferentes número de amostras, selecionadas da melhor maneira para cada caso. Enquanto que a estimativa da integral de $L_{e,s}$ ignora qualquer emissão proveniente de fontes de luz grande, a estimativa da integral $L_{e,l}$ ignora as emissões provenientes de fontes luminosas pequenas.

Por último, os termos da *BSDF* também podem ser particionados. Por exemplo, se f_Δ denota componentes da *BSDF* descritas por distribuições delta, e $f_{-\Delta}$ denota as componentes restantes,

$$\begin{aligned} P(\bar{p}_i) &= \int_{A^{i-1}} L_e(p_i \rightarrow p_{i-1}) \prod_{j=1}^{i-1} (f_\Delta(p_{j+1} \rightarrow p_j \rightarrow p_{j-1}) + \\ &\quad f_{-\Delta}(p_{j+1} \rightarrow p_j \rightarrow p_{j-1})) G(p_{j+1} \leftrightarrow p_j) dA(p_2) \cdots dA(p_i). \end{aligned}$$

Como há $i - 1$ termos da *BSDF* no produto, é importante ter cuidado de não contabilizar somente termos com componentes f_Δ ou componentes $f_{-\Delta}$; todos os termos do tipo $f_\Delta f_{-\Delta} f_{-\Delta}$ também devem ser contabilizados se for utilizado um sistema de particionamento como este.

4.2.2.6 A equação de medição e a importância

A *equação de medição* descreve o valor de uma medição abstrata obtida pela integração ao longo de um conjunto de raios transmitindo radiância. Por exemplo, ao calcular o valor de um pixel da imagem, queremos integrar sobre raios que começam na vizinhança do pixel, com contribuição ponderada pelo filtro de reconstrução da imagem. Ignorando profundidade de campo por enquanto (de modo que cada ponto do plano do filme corresponde a uma única direção saindo da câmera), podemos escrever o valor do pixel como uma integral sobre pontos no plano do filme de uma função de ponderação vezes a radi-

ância incidente ao longo dos correspondentes raios da câmera:

$$\begin{aligned} I_j &= \int_{A_{filme}} \int_{S^2} W_e(p_{filme}, \boldsymbol{\omega}) L_i(p_{filme}, \boldsymbol{\omega}) |\cos \theta| dA(p_{filme}) d\boldsymbol{\omega} \\ &= \int_{A_{filme}} \int_A W_e(p_0 \rightarrow p_1) L(p_1 \rightarrow p_0) G(p_0 \leftrightarrow p_1) dA(p_0) dA(p_1), \end{aligned}$$

onde I_j é a medição para o j -ésimo pixel, e p_0 é um ponto sobre o filme. Nesta configuração, o termo $W_e(p_0 \rightarrow p_1)$ é o produto da função filtro em torno do pixel, f_j , com uma função delta que seleciona a direção apropriada do raio da câmera da amostra de p_0 , $\boldsymbol{\omega}_{c\grave{a}mera}(p_0)$:

$$W_e(p_0 \rightarrow p_1) = f_j(p_0) \boldsymbol{\delta}(t(p_0, \boldsymbol{\omega}_{c\grave{a}mera}(p_0)) - p_1).$$

Esta formulação pode parecer inutilmente complexa, mas leva a uma perspectiva importante. Se expandirmos o termo $P(\bar{p}_i)$ do somatório da *LTE*, temos:

$$\begin{aligned} I_j &= \int_{A_{filme}} \int_A W_e(p_0 \rightarrow p_1) L(p_1 \rightarrow p_0) G(p_0 \leftrightarrow p_1) dA(p_0) dA(p_1), \\ &= \sum_i \int_A \int_A W_e(p_0 \rightarrow p_1) P(\bar{p}_i) G(p_0 \leftrightarrow p_1) dA(p_0) dA(p_1) \\ &= \sum_i \int_A \cdots \int_A W_e(p_0 \rightarrow p_1) T(\bar{p}_i) L_e(p_{i+1} \rightarrow p_i) G(p_0 \leftrightarrow p_1) dA(p_0) \cdots dA(p_i). \end{aligned}$$

Veio à tona uma simetria entre a radiância emitida a partir de fontes de luz (L_e) e a contribuição da amostra sobre o filme para a medição do pixel (W_e). As implicações desta simetria são importantes, pois ela diz que podemos pensar na renderização de duas maneiras diferentes: a luz pode ser emitida a partir de fontes de luz, vai percorrendo a cena, e chega a um sensor onde W_e descreve a sua contribuição para a medição. Alternativamente, podemos pensar numa quantidade, emitida a partir do sensor, percorrendo a cena, e fazendo uma contribuição quando ela atinge uma fonte de luz.

O valor descrito pelo termo $W_e(p_0 \rightarrow p_1)$ é conhecido como a *importância* para o raio entre p_0 e p_1 na cena. Quando a equação de medição é utilizada para calcular as medidas do pixel. A importância costuma ser parcial ou totalmente descrita pelas distribuições delta. Muitos outros tipos de medição, além da formação de imagens podem ser descritas construindo adequadamente funções de importância, e, portanto, os formalismos descritos podem ser utilizados para mostrar como a integral sobre caminhos, descrita pela equação de medida é a integral que deve ser estimada para computá-la.

4.2.3 Path Tracing

Agora que derivamos a forma da integral de caminhos da equação de transporte de luz, vamos mostrar como ela pode ser utilizada para determinar o algoritmo *path-tracing* de transporte de luz.

O *traçado de caminhos* (path-tracing) foi o primeiro algoritmo imparcial de Monte Carlo geral proposto para transporte de luz utilizado em computação gráfica. Kajiya introduziu-o no mesmo artigo que descreveu pela primeira vez a equação de transporte de luz (1986) [KAJ86]. O *traçado de caminhos* (path-tracing) gera incrementalmente caminhos de eventos de dispersão começando na câmera e terminando nas fontes de luz da cena. Uma maneira de pensar isto, é como uma extensão do métodos de *Whitted* para incluir distribuições delta, *BSDFs* não-delta e fontes de luz, em lugar de somente contabilizar termos delta.

Embora seja um pouco mais fácil derivar o *traçado de caminhos* diretamente a partir da equação de transporte de luz, encará-la a partir da forma da integral de caminho ajuda a compreender a equação da integral de caminho, e fará mais compreensível a generalização do *traçado bidirecional de caminhos* (Bidirectional path tracing). O *traçado bidirecional de caminhos* é uma técnica em que os caminhos são gerados a partir da iluminação, bem como a partir da câmera, é discutido no final desta seção.

4.2.3.1 Visão geral

Dada a forma de integral de caminho da *LTE*, gostaríamos de estimar o valor da radiância partindo do ponto p_1 , de interseção de um raio da câmera,

$$L(p_1 \rightarrow p_0) = \sum_{i=1}^{\infty} P(\bar{p}_i),$$

para um raio de câmera desde p_0 , que intersesta a cena pela primeira vez em p_1 . Temos dois problemas que devem ser resolvidos de modo a obter esta estimativa:

1. Como é que podemos estimar o valor do somatório infinito dos termos $P(\bar{p}_i)$ com uma quantidade finita de cálculos?
2. Dado um determinado termo $P(\bar{p}_i)$, como é que vamos gerar um ou mais caminhos \bar{p} , a fim de calcular uma estimativa de Monte Carlo para a integral multidimensional?

No caso do traçado de caminhos, é possível tirar proveito do fato de que para cenas fisicamente válidas, os caminhos com mais vértices dispersam menos luz do que caminhos com um número menor vértices. Isto é uma consequência natural da conservação de energia em *BSDFs*. Por este motivo, é bom proceder estimando os primeiros termos $P(\bar{p}_i)$ e, em seguida, aplicar roleta russa para parar a amostragem após um número finito de termos sem introduzir desvio. é importante lembrar a seção 3.6, em que a Roleta Russa nos permite parar probabilisticamente a computação de termos de uma soma com tanto que seja feita a reponderação dos termos que não foram computados. Por exemplo, se sempre forem computadas as estimativas de $P(\bar{p}_1)$, $P(\bar{p}_2)$, e $P(\bar{p}_3)$, mas parou sem computar mais termos de probabilidade q , então uma estimativa imparcial da soma seria:

$$P(\bar{p}_1) + P(\bar{p}_2) + P(\bar{p}_3) + \frac{1}{1-q} \sum_{i=4}^{\infty} P(\bar{p}_i).$$

Utilizando Roleta Russa desta maneira, o problema de calcular um somatório infinito não é resolvido, mas fica mais próximo.

Se estendermos esta idéia, podemos considerar terminada a avaliação da soma para cada termo com uma probabilidade q_i ,

$$\frac{1}{1-q_1} \left(P(\bar{p}_1) + \frac{1}{1-q_2} \left(P(\bar{p}_2) + \frac{1}{1-q_3} (P(\bar{p}_3) + \dots \right) \right),$$

e eventualmente encerrar o cálculo da soma. Ainda porque, para um determinado valor de i há uma probabilidade maior do que zero de avaliar o termo $P(\bar{p}_i)$, e porque vai ser devidamente ponderado, se é que vamos proceder à sua avaliação, o resultado final é uma estimativa da soma imparcial.

4.2.3.2 Amostragem de caminhos

Dado este método para calcular somente um número finito de termos do somatório infinito, é necessário ter uma forma de calcular a contribuição de um termo $P(\bar{p}_i)$ em particular. É preciso $i+1$ vértices para especificar o caminho, onde o último vértice p_i está na fonte de luz e o primeiro vértice p_0 é determinado pela primeira interseção do raio partindo da câmera (figura 4.10). Olhando para a forma do termo $P(\bar{p}_i)$, uma integral múltipla de área sobre os objetos da cena, a idéia mais natural é amostrar vértices p_i segundo a área da superfície dos objetos da cena, de maneira que seja igualmente provável

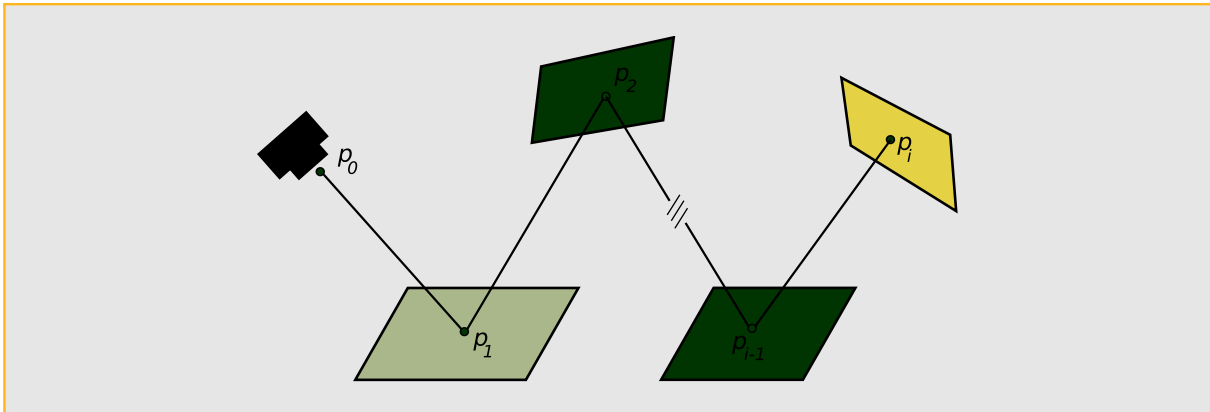


Figura 4.10: O caminho \bar{p}_i de $i+1$ vértices desde a câmera em p_0 , intersecando uma série de posições em superfícies da cena, até um ponto p_i na fonte de luz. Acontece um espalhamento de acordo a *BSDF* em cada vértice desde p_1 até p_{i-1} , de maneira que a radiância que chega na câmera através deste caminho é dado pelo produto do rendimento do caminho $T(\bar{p}_i)$ vezes a radiância emitida pela luz.

amostrar qualquer ponto p_i num objeto da cena⁶.

Segundo o visto na seção 3.1.1, podemos definir a probabilidade discreta sobre os n objetos na cena. Se cada um tem uma superfície A_i , então a probabilidade de amostrar um vértice do caminho na superfície do i -ésimo objeto deve ser:

$$p_i = \frac{A_i}{\sum_j A_j}.$$

Então, dado um método para amostrar um ponto no i -ésimo objeto com probabilidade uniforme, a *PDF* para amostrar um ponto no objeto i é $1/A_i$. Assim, a densidade de probabilidade global de amostrar o ponto é

$$\frac{A_i}{\sum_j A_j} \frac{1}{A_i}.$$

Todas as amostras p_i tem o mesmo valor de *PDF*:

$$p_A(p_i) = \frac{1}{\sum_j A_j}.$$

Dado o conjunto de vértices p_0, p_1, \dots, p_{i-1} amostrados desta maneira, podemos, amostrar o vértice p_i em uma fonte de luz na cena, definindo sua *PDF*, da mesma forma.

Embora seja possível usar a técnica usada para a amostragem vértices do caminho

⁶A implementação do `PathIntegrator` do `PBRT` não utiliza esta abordagem. Detalhes sobre a implementação podem ser vistos em [PH04b].

para amostrar pontos sobre as luzes, isso levaria a uma variância alta, uma vez que para todos os caminhos onde p_i não estiver na superfície de um emissor, o caminho teria valor zero. O valor esperado ainda seria o valor correto da integral, mas a convergência seria extremamente lenta. A melhor abordagem é amostrar apenas ao longo das áreas dos objetos emissivos, com probabilidades devidamente atualizadas. Dado um caminho completo, temos todas as informações necessárias para calcular a estimativa de $P(\bar{p}_i)$, é só uma questão de avaliar cada um dos termos.

Se soubéssemos que a iluminação indireta de alguns poucos objetos contribuíram para grande parte da iluminação na cena, poderíamos atribuir uma maior probabilidade de gerar vértices p_i do caminho sobre esses objetos, atualizando os pesos adequadamente.

No entanto, existem dois problemas interligados com amostragem caminhos feita dessa maneira. O primeiro pode levar a uma grande variância, enquanto a segunda pode levar a resultados incorretos. O primeiro problema é que muitos dos caminhos não terão contribuições se tiverem pares de vértices adjacentes que não são mutuamente visíveis. Considere a aplicação deste método de amostragem de área num modelo complexo: vértices adjacentes no caminho quase sempre têm uma ou duas paredes entre eles, portanto não contribui com o caminho e obtém-se variância excessiva na estimativa.

O segundo problema é que, se o integrando tem funções delta na mesma, esta técnica de amostragem nunca será capaz de escolher vértices do caminho tais que as distribuições delta sejam não-nulas. E mesmo que não existem distribuições delta, como as *BSDFs* tornam-se cada vez mais brilhantes, quase todos os caminhos vão ter baixa contribuição desde que os pontos de $f(p_{i+1} \rightarrow p_i \rightarrow p_{i-1})$ farão com que a *BSDF* tenha um valor pequeno ou nulo e, novamente, vamos ter variância elevada. De maneira similar, pequena área de fontes luminosas também podem ser fontes de variância não se amostrar explicitamente.

4.2.3.3 Construção incremental do caminho

Uma solução que resolve ambos os problemas é a construção incremental do caminho, começando pelo vértice p_0 na câmera. Em cada vértice, a *BSDF* é amostrada para gerar uma nova direção; o próximo vértice p_{i+1} é encontrado traçando um raio desde p_i na direção amostrada e encontrando a primeira intersecção. Estamos tentando encontrar um caminho com contribuição global grande, fazendo escolhas que permitam encontrar direções com importantes contribuições locais. Embora é possível imaginar situações em que esta abordagem poderia ser ineficaz, em geral é uma boa estratégia.

Devido a que esta abordagem constrói o caminho por amostragem das *BPDFs* de acordo com o ângulo sólido, e porque a integral de caminho *LTE* é uma integral de superfície sobre a cena, é preciso aplicar a correção para converter a densidade de probabilidade segundo um ângulo sólido p_ω para uma densidade segundo a área p_A :

$$p_A = p_\omega \frac{\|p_i - p_{i+1}\|^2}{|\cos \theta_i|}.$$

A correção desta situação provoca alguns cancelamentos do termo geométrico $G(p_i \leftrightarrow p_{i+1})$ em $P(\bar{p}_i)$. Além disso, sabemos que p_i e p_{i+1} devem ser mutuamente visíveis desde que o raio foi traçado para encontrar p_{i+1} , de modo que o termo de visibilidade é igual a um. Portanto, se usarmos esta técnica de amostragem, mas amostramos o último vértice p_i de alguma distribuição correspondente à superfície das fontes luminosas $p_A(p_i)$, o valor da estimativa de Monte Carlo para o caminho é :

$$\frac{L_e(p_i \rightarrow p_{i-1})}{p_A(p_i)} \left(\prod_{j=1}^{i-1} \frac{f(p_{j+1} \rightarrow p_j \rightarrow p_{j-1}) |\cos \theta_j|}{p_\omega(p_{j+1} - p_j)} \right).$$

4.3 Mapeamento de fótons

O *Mapeamento de fótons* (Photon mapping) [JEN96] [JEN01] provavelmente seja o algoritmo mais bem-sucedido de iluminação global até o presente. Embora tendencioso, o algoritmo é capaz de produzir imagens de qualidade próxima a de uma fotografia em um período de tempo razoável. Também pode manipular elegantemente tarefas de renderização que foram muito difíceis antes do algoritmo ter sido introduzido, tais como espalhamento de sub-superfícies e renderização de meios participativos. O *Mapeamento de Fótons* resolve o problema de iluminação global em duas passadas. Na primeira passada o renderizador lança partículas de fótons desde as fontes de luz e guarda registros, numa estrutura de dados (mapa de fótons), onde os fótons batem nos objetos da cena. Na segunda passada, traçado de raios desde a câmera é utilizado para gerar a imagem. Durante a passada do traçado de raios, o renderizador consulta o *mapa de fótons* para estimar a iluminação indireta, tornando o processo eficiente.

4.3.1 Construção do mapa de fótons

Conforme mencionado, a primeira fase do *mapeamento de fótons* lança partículas desde as fontes luminosas para a cena. A simulação da trajetória de um fóton é feita da seguinte maneira: uma partícula é emitida desde uma fonte luminosa, escolhendo-se um ponto na fonte de luz e uma direção inicial aleatória. A partícula emitida pode rebater um certo número de vezes na cena, criando um caminho, que chamamos *caminho do fóton*. A medida que o fóton bate na cena, durante seu percorrido, é criado um registro para cada interseção (não especular) ao longo do caminho do fóton. O renderizador armazena os registros numa *kd-tree* que chamamos de *mapa de fótons* para permitir a posterior pesquisa espacial.

4.3.1.1 Potência do fóton

Uma interpretação do mapa de fótons é que cada fóton representa alguma fração da potência emitida pelas fontes de luz, e os registros de choque de fótons representam a iluminação incidente sobre as superfícies na cena. O poder atribuído a um determinado registro de choque, Φ_i , decorre diretamente da probabilidade com a qual o caminho do fóton foi gerado e as características emissivas das fontes luminosas. A figura 4.11 mostra o cálculo do valor Φ_i da equação 4.6 no caso geral.

$$\Phi_i = \frac{L_e(p_0, \omega_0) |\cos(\phi_0)|}{p_{\text{area}}(p_0) p_{\text{angulo}}(\phi_0)} \times \prod_{j=1}^{m-1} \frac{f(p_j, \omega_j, \omega_{j+1}) |\cos(\phi_j)|}{p_{\text{angulo}}(p_j, \omega_j, \omega_{j+1}) p_{\text{proleta}}(p_j)}. \quad (4.6)$$

Jensen [JEN01] salienta que, idealmente, todos os registros de choque dos fótons de-

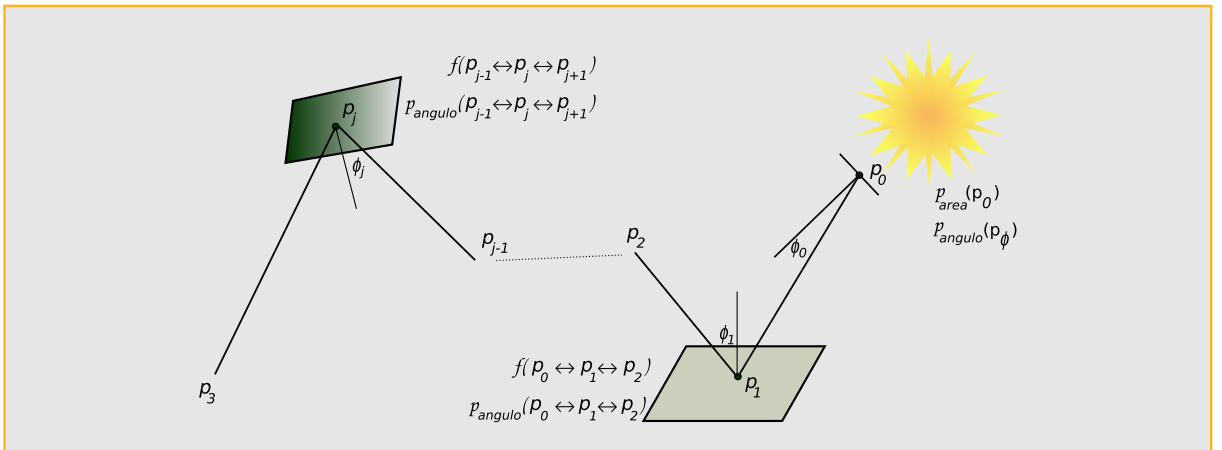


Figura 4.11: Trajetória do fóton utilizada para o cálculo da potência Φ_i .

vem conter igual potência. Isto pode ser alcançado por: **(1)** escolhendo um ponto de emissão e uma direção inicial proporcional às propriedades emissivas das fontes luminosas, **(2)** escolhendo as direções de reflexão proporcionalmente com o termo da $BSDF$ e **(3)** colocando p_{roleta} proporcional com reflectância da superfície. Se esses passos forem seguidos, Φ_i se reduz para Φ_{total}/n , onde Φ_{total} é a potência total emitida pelas fontes de luz e n é o número de fótons emitidos a partir de fontes de luz.

4.3.2 Estimando a radiância com o mapa de fótons

Dado um *mapa de fótons* da cena, o renderizador pode estimar a radiância refletida num ponto x de uma superfície sem traçar raios, utilizando um processo de estimativa de densidade dos K -vizinhos mais próximos. A idéia é procurar os k registros de choque mais próximos a x no *mapa de fótons* e usá-los para reconstruir a iluminação que chega em x . A estrutura kd -tree do *mapa de fótons* faz com que a procura dos registros de choque vizinhos seja eficiente. Uma vez que os vizinhos foram encontrados, a radiância refletida $L_r(x \rightarrow \Psi)$ pode ser estimada mediante a equação 4.7, como na figura 4.12

$$L_r(x \rightarrow \Psi) \approx \frac{1}{\pi r^2} \sum_{i=1}^k \Phi_i f_s(\Psi \leftrightarrow x \leftrightarrow \Theta_i). \quad (4.7)$$

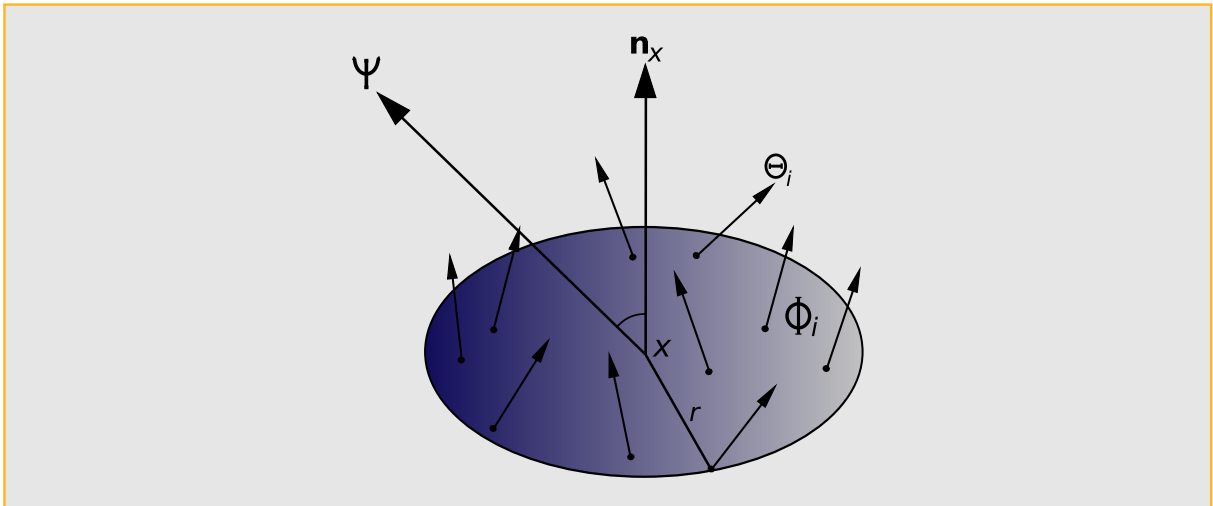


Figura 4.12: Estimativa de radiância baseada no mapa de fótons. A radiância refletida em x pode ser aproximada como a soma das contribuições dos registros dos k fótons mais próximos, dividida pela área em que os vizinhos foram encontrados (πr^2 onde r é a distância até o k fóton mais próximo). Note que não é preciso um termo do cosseno uma vez que a densidade de choque de fótons sobre a superfície é naturalmente proporcional ao cosseno do ângulo de incidência.

4.3.3 Visão direta do mapa de fótons e reunião final (gather)

Visão direta do mapa de fótons

A segunda passada do algoritmo de mapeamento de fótons cria uma imagem de exibição traçando raios desde o ponto do olho (câmera). A utilização mais óbvia do mapa de fótons, seria para substituir a radiância refletida nos pontos atingidos pelos raios provenientes do olho com a estimativa da figura 4.12. Na prática, isso não funciona bem, uma vez que a estimativa da radiância geralmente não é suficientemente precisa para a visualização direta. O lado esquerdo da imagem na figura 4.13 demonstra o típico embaçamento resultante da visualização direta do mapa de fótons.

Reunião final e o mapa cáustico

A maioria dos artefatos de embaçamento causados pela visualização direta do mapa de fótons pode ser eliminada através da execução de uma etapa final de reunião, que estima a radiância utilizando o mapa fótons no segundo salto do fóton e não o primeiro. Neste esquema, a radiância em x é determinada enviando raios de sondagem, e estimando a radiância nas localizações das sondas utilizando o mapa de fótons. Embora o último passo de reunião seja caro, na maioria das situações ainda é muito mais eficiente do que o tradicional traçado de caminhos para a mesma qualidade de imagem.

Um problema notável com a reunião final é que ele não funciona para superfícies muito brilhantes ou especulares. Jensen soluciona este problema definindo um mapa de fótons cáustico separado que somente inclui somente os choques de fótons que aconteceram sobre superfícies especulares. A iluminação cáustica pode então ser estimada diretamente a partir do mapa cáustico, enquanto a iluminação restante é computada no passo de reunião (gather) final. A idéia de mapa cáustico funciona porque os choques dos fótons se concentram naturalmente em regiões cáusticas, levando a uma estimativa mais precisa da radiância. A metade direita da imagem na figura 4.13 mostra a melhora da qualidade que resulta da renderização com uma reunião final e separando o mapa de fótons cáustico. Embora a qualidade é muito melhor que num traçado de caminhos com um número semelhante de amostras, algum embaçamento é evidente no cáustico abaixo da bola de vidro. Além disso, as esferas difusas dentro da bola de vidro, aparecem manchadas porque a densidade de amostras no mapa de fótons cáusticos fóton é baixa nessa região.

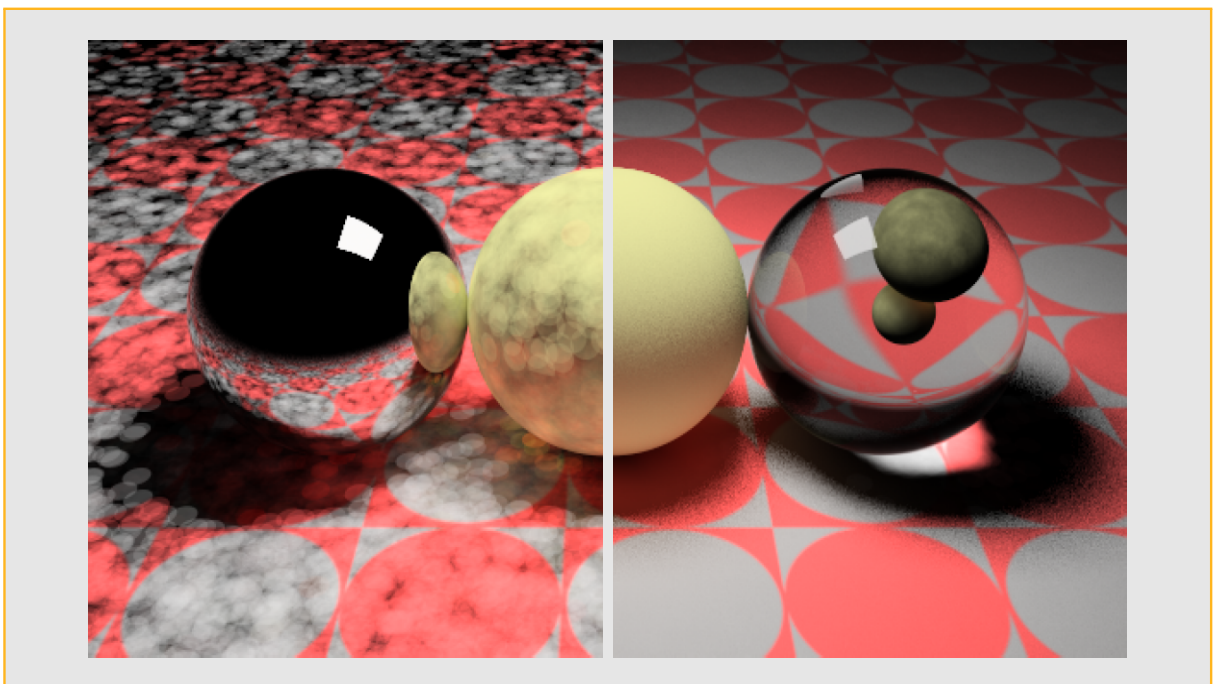


Figura 4.13: A visualização direta do mapa de fótons (esquerda) produz uma imagem muito embaçada. Um passo de reunião final (direita) elimina a maioria dos artefatos da imagem.

5 *Mapas de iluminação*

5.1 *Imagens HDR versus LDR*

As câmeras tradicionais foram criadas para registrar a luz de maneira que pudesse ser reproduzida sobre papel fotográfico, enquanto que as câmeras digitais, por outro lado, foram desenvolvidas para poderem reproduzir a luz capturada num monitor. Nem os monitores nem os papéis fotográficos conseguem representar toda a faixa dinâmica de iluminação. Entretanto, se forem tiradas varias fotografias com diferentes tempos de exposição, essas imagens podem ser combinadas para que a faixa dinâmica seja recuperada aproximadamente. Com essa técnica, uma serie de imagens pode ser combinada em uma única imagem com alta faixa dinâmica (*HDR* - High Dynamic Range), denominada mapa de irradiação.

Atualmente, a maioria das imagens coloridas são representadas com um byte por pixel em cada um dos canais vermelho, verde e azul. Com três bytes por pixel, mais de 16 milhões de cores diferentes podem ser atribuídos a cada pixel. Se bem isto parece muito, na verdade só temos 256 valores diferentes possíveis para cada canal de cor *RGB* dos pixels. Estas imagens são conhecidas como imagens de baixa faixa dinâmica ou *LDR* (Low Dynamic Range). Dispor de somente 256 valores por canal de cor resulta inadequado para representar alguns tipos de cenas. Por exemplo, isto pode ser apreciado em cenas fotografadas no interior de um recinto com uma janela, ou uma paisagem no interior de uma floresta com uma clareira. Em ambos os casos temos uma cena onde há regiões com pouca iluminação e outras com muita iluminação.

O intervalo de valores que conformam uma imagem convencional é aproximadamente de duas ordens de magnitude, armazenados num byte para cada um dos canais *RGB* por cada pixel. Não é possível imprimir diretamente imagens com alta faixa dinâmica; por isso, para simular o efeito de reduzir uma imagem *HDR* para o intervalo de saída normal, reduzimos a faixa dinâmica de uma fotografia convencional em duas ordens de magnitude.

Na figura 5.1 mostramos duas imagens *LDR* de 8 bits por canal por pixel, e as mesmas imagens reduzidas para 4 bits por canal por pixel. Assim com menor quantidade de bits temos menor qualidade. Para algumas cenas, 8 bits por canal de cor é suficiente, mas em algumas aplicações isto não basta. Uma das razões para isto é que o mundo real produz uma faixa muito maior que as duas ordens de magnitude comuns nas imagens digitais. Por exemplo, o Sol é arredor de 100 milhões de vezes mais brilhante que a luz de uma estrela [FER01, SW90] . Luminância de ambientes típicos estão na tabela 5.1.

Tabela 5.1: Alguns ambientes comuns

Condição	Iluminação (em cd/m^2)
luz de estrelas	10^{-3}
luz de lua	10^{-1}
luz dentro de um quarto	10^2
luz do sol	10^5
Intensidade máxima dos monitores CRT	10^2

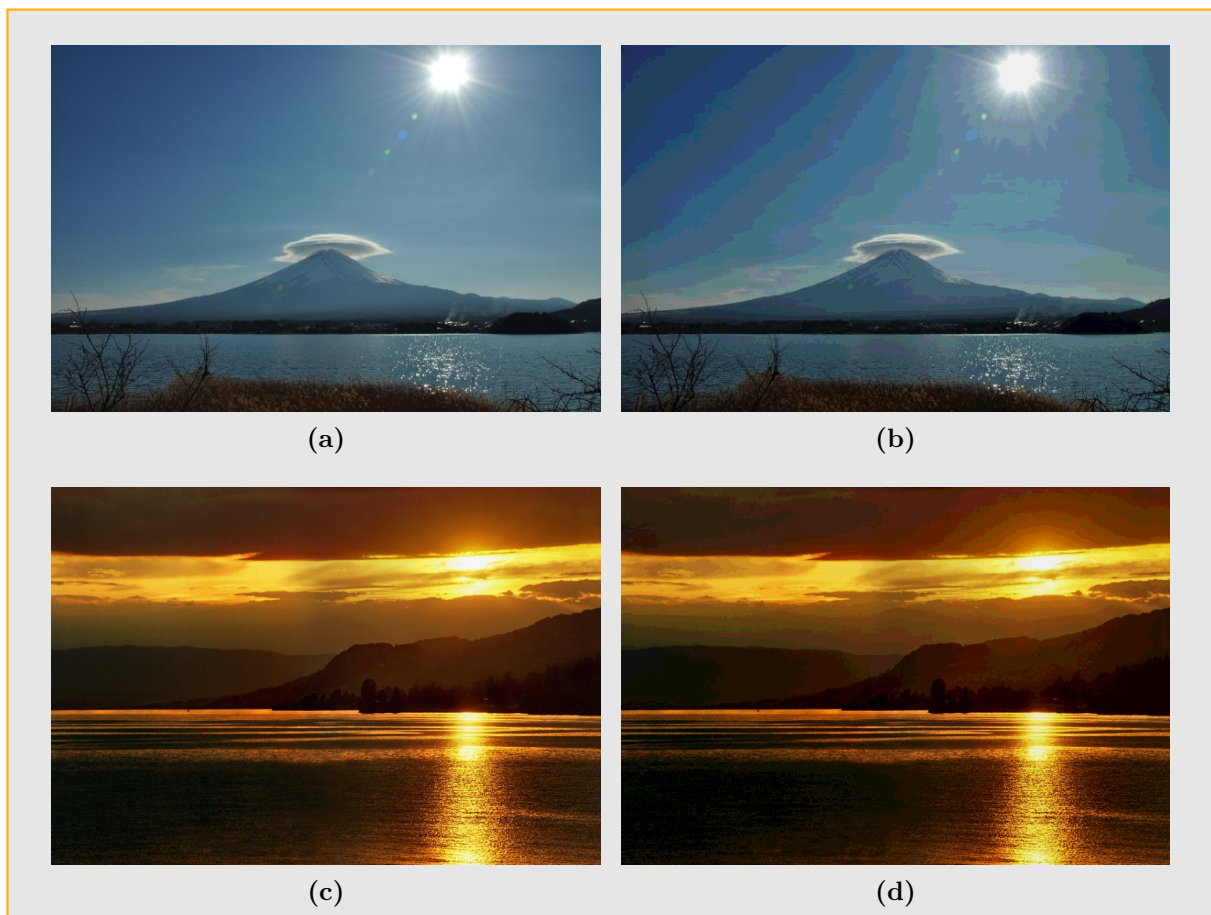


Figura 5.1: (a) e (c) imagens com 8-bits por canal, (b) e (d) imagens com 4-bits por canal.

O sistema visual humano adapta-se a condições de iluminação que podem variar perto de 10 ordens de magnitude [FER01]. Numa cena, o sistema visual humano pode captar simultaneamente de aproximadamente cinco ordens de magnitude. Isto está em forte contraste com os típicos monitores *CRT* (cathode-ray tube, tubo de raios catódicos) capazes de reproduzir aproximadamente duas ordens de magnitude de variação. É por isto que em geral as imagens são codificadas com 8 bits, dado que os dispositivos tradicionais não conseguem representar uma amplitude maior.

Uma representação em imagem *HDR* consegue reduzir os erros de precisão a níveis abaixo dos humanamente detectáveis. As imagens *HDR* também podem ser utilizadas como fontes de luz complexas em algoritmos de renderização convencionais [DEB98]. Neste caso, não podemos utilizar imagens *LDR* convencionais e obter resultados foto-realistas, devido a que a faixa dinâmica de luz emitida por várias partes de uma cena é muito maior do que as duas ordens de grandeza disponíveis nas imagens *LDR* convencionais. Se pretendemos iluminar uma cena artificial com uma representação de uma cena real, temos que capturar a iluminação da cena real em *HDR*.

Devido a que imagens *HDR* armazenam a faixa completa de luz que chega num ponto do espaço, as mesmas contêm informações sobre a forma, a cor e a intensidade das fontes de iluminação direta, bem como a cor e distribuição da iluminação indireta a partir de superfícies no resto da cena. Utilizando algoritmos de renderização adequados, e técnicas de amostragem, podemos usar eficientemente imagens *HDR* para simular com precisão como os objetos e ambientes ficariam se fossem iluminados pela luz do mundo real. Este processo de utilização de imagens como fontes de luz é chamado de *iluminação baseada em imagens* (*IBL*), e foi introduzido por Paul Debevec [DEB98].

5.2 Mapas de iluminação

O primeiro passo para trabalhar com iluminação de cenas artificiais mediante iluminação extraída de um ambiente real, é armazenar a iluminação do ambiente real. Uma forma de fazer isto é obtendo uma imagem de sondagem de luz (*Light Probe*). Uma imagem de sondagem de luz (*Light Probe*) é uma imagem omnidirecional de alta faixa dinâmica que registra as condições de incidência da iluminação num ponto determinada do espaço.

Uma vez capturada uma imagem de sondagem de luz (*Light Probe*), ela precisa ser armazenada num arquivo de imagem utilizando um mapeamento de imagem omnidirecional, obtendo assim uma representação digital de toda a luz da cena real numa imagem

formatada, denominada *mapa de iluminação*. Entre os mapeamentos mais utilizados podemos citar: mapa de esfera espelhada ideal, mapa angular, mapa de latitude-longitude e mapa de cubo. Durante a pesquisa realizada para esta dissertação utilizamos o mapa latitude-longitude, que será apresentado nesta seção. A seguir apresentamos as fórmulas para determinar adequadamente as coordenadas (u, v) na imagem, que corresponde a uma direção unitária do mundo $D = (x, y, z)$, e vice-versa. As coordenadas do mundo real estão no sistema de coordenadas canônico $e_1 = (1, 0, 0)$, $e_2 = (0, 1, 0)$ e $e_3 = (0, 0, 1)$.

5.2.1 Mapeamento Latitude-Longitude

Para realizar o mapeamento de latitude-longitude utilizamos um domínio da imagem retangular dado por $(u, v) \in [0, 1] \times [0, 1]$. Em primeiro lugar fazemos uma transformação das coordenadas reais (x, y, z) para coordenadas esféricas (ϕ, θ) . As equações de mapeamento de coordenadas reais do mundo para coordenadas esféricas, e vice-versa são:

$$(\phi, \theta) = \left(\arctan \frac{y}{x}, \arccos z \right); \quad (5.1)$$

$$\begin{aligned} x &= \sin \theta \cdot \cos \phi, \\ y &= \sin \theta \cdot \sin \phi, \\ z &= \cos \theta. \end{aligned} \quad (5.2)$$

Agora, a partir das coordenadas esféricas dadas pela equação 5.1 podemos obter facilmente as coordenadas (u, v) no domínio da imagem

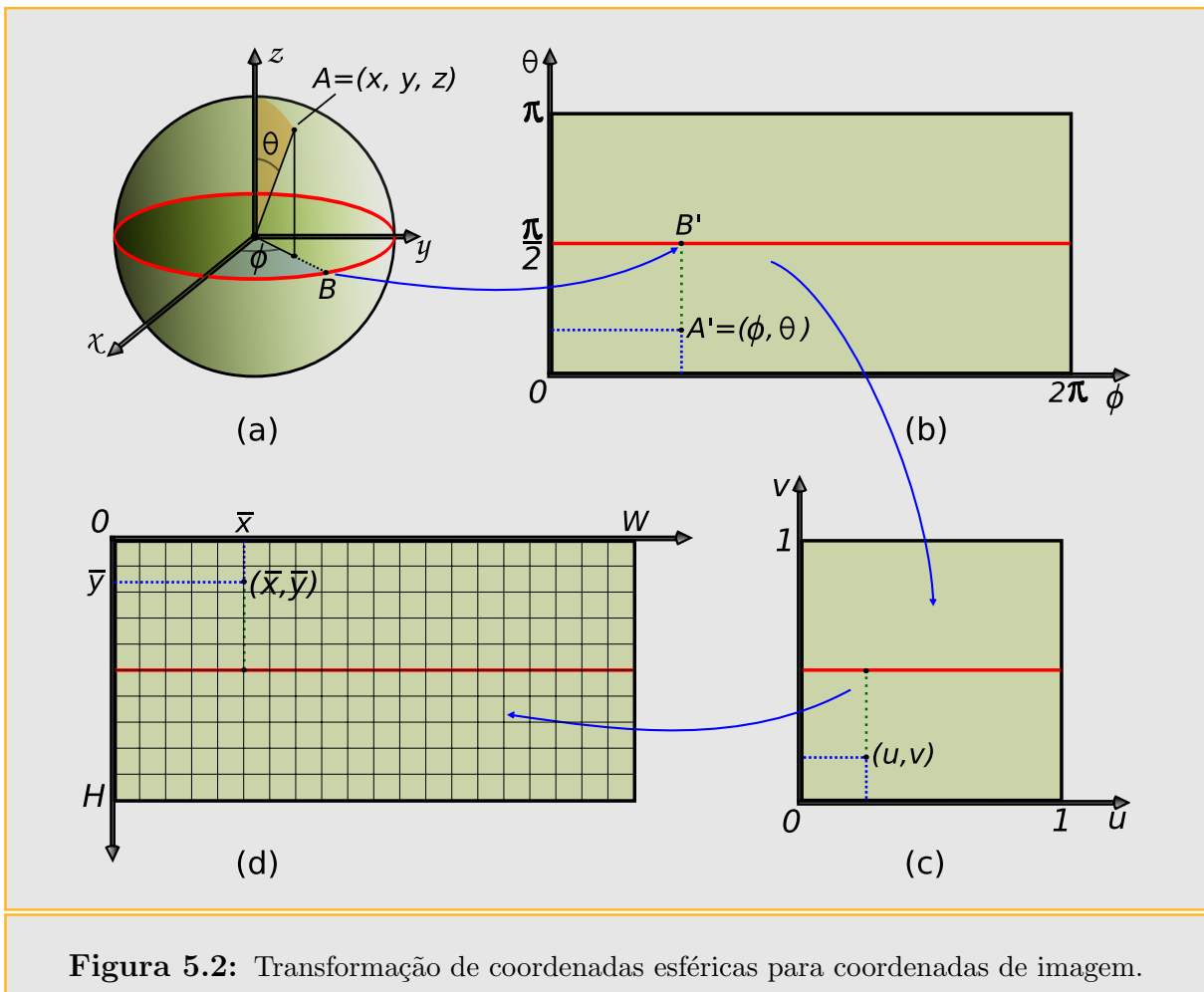
$$(u, v) = \left(\frac{\phi}{2\pi}, \frac{\theta}{\pi} \right), \quad (5.3)$$

Substituindo θ e ϕ segundo a equação 5.1, obtemos:

$$(u, v) = \left(\frac{\arctan \frac{y}{x}}{2\pi}, \frac{\arccos z}{\pi} \right). \quad (5.4)$$

Da equação 5.3, vê-se que um ponto (u, v) no domínio da imagem corresponde ao ponto $(\phi, \theta) = (2\pi u, \pi v)$ em coordenadas esféricas. Na figura 5.2 é possível observar o mapeamento da esfera de direções num mapa latitude-longitude utilizando as transformações dadas pelas equações 5.1–5.4.

Como pode ser apreciado na figura 5.2, num mapeamento do tipo latitude-longitude



uma direção azimute sobre o plano xy da esfera de direções, é medido a partir da direção $e_1 = (1, 0, 0)$, e mapeado sobre a coordenada horizontal; a elevação z é mapeada na coordenada vertical da imagem. Este mapeamento achata a esfera numa área retangular. A borda horizontal superior da imagem corresponde à direção $e_3 = (0, 0, 1)$ da esfera de direções, ou pólo Norte; a borda horizontal inferior da imagem corresponde com a direção $-e_3 = (0, 0, -1)$ da esfera, ou pólo Sul. O formato, naturalmente tem uma relação de aspecto 2:1, e introduz a menor distorção para regiões próximas à linha do horizonte (plano equatorial da esfera de direções). As áreas de igual elevação absoluta $|z|$ na esfera de direções, sofrem mesma deformação, que aumenta de forma progressiva a medida que a elevação se aproxima dos pólos.

O mapeamento latitude-longitude é conveniente porque é retangular e não tem costuras (o único problema são os pontos singulares, pólos Norte e Sul), e porque as fórmulas do mapeamento são simples e intuitivas. Embora geralmente as linhas retas tornam-se curvas neste formato, os paralelos e meridianos da esfera de direções correspondem a linhas retas horizontais e verticais respectivamente no mapa latitude-longitude.

Nesta dissertação trabalhamos exclusivamente com mapas de iluminação no formato latitude-longitude armazenados em imagens *HDR* (formato OpenEXR¹) com tamanho $W \times H$ pixels (onde $W = 2H$, geralmente 1024×512 pixels).

Dado um mapa de iluminação de tamanho $W \times H$ pixels, e uma posição (\bar{x}, \bar{y}) no mesmo (a coordenada $(0,0)$ na imagem corresponde ao canto superior esquerdo), podemos recuperar as coordenadas esféricas (θ, ϕ) :

$$(\phi, \theta) = \left(\frac{2\pi\bar{x}}{W}, \frac{\pi\bar{y}}{H} \right). \quad (5.5)$$

As coordenadas euclidianas (x, y, z) podem ser obtidas facilmente mediante a equação 5.2. A equação 5.5 será de grande utilidade prática nas implementações dos diferentes métodos de amostragem propostos para mapas de iluminação nesta dissertação.

5.2.2 Deformação de área e representação do mapa de iluminação

Ao fazer um mapeamento da esfera de direções do mundo real para um formato plano como o mapa latitude-longitude, existe uma deformação de área como consequência do achatamento da esfera sobre o plano. Em aplicações como iluminação baseada em imagens e amostragem de mapas de iluminação, a quantidade de irradiância está associada com a área emissora, portanto devemos conhecer exatamente as deformações de área sofridas ao passar de um *Light Probe* para um mapa de iluminação Latitude-Longitude. Afortunadamente a transformação é bem comportada e o cálculo da deformação de área sofrida na transformação é fácil de ser calculada.

Basicamente o que devemos fazer é expressar a relação entre a diferencial de área de um conjunto de direções $d\omega$ e a diferencial de área do par (ϕ, θ) (figura 5.3). A diferencial de área $d\omega$ é o produto dos diferenciais dos seus lados, $\sin\theta d\theta$ e $d\phi$

$$d\omega = \sin\theta d\theta d\phi.$$

Dado um *Light Probe* $M(x, y, z)$, considere a função luminância $L : \mathbb{R}^3 \rightarrow \mathbb{R}$ definida por $L(R, G, B) = 0.2125R + 0.7154G + 0.0721B$. Então a função $L_M(x, y, z) := L(M(x, y, z))$ é a radiância na direção $\omega = (x, y, z)$, e a irradiância total do *Light Probe* é dada por

¹OpenEXR é um formato para arquivos de imagem *HDR* desenvolvido pela Industrial Light & Magic. Veja <http://www.openexr.com/>

$$E = \int_{S^2} L_M(\omega) d\omega, \quad (5.6)$$

ou equivalentemente

$$E = \int_0^{2\pi} \int_0^\pi L_M(\phi, \theta) \sin(\theta) d\theta d\phi, \quad (5.7)$$

se considerarmos coordenadas esféricas.

Em geral temos $L_{LL}(u, v) = L_{LP}(\phi, \theta)$, i.e., a radiância de um ponto do mapa Latitude-Longitude é igual à radiância do ponto correspondente no *Light Probe*. Isto acontece porque os softwares de mapeamento calculam a transformação dos pontos e associam a mesma radiância do *Light Probe*, sem fazer uma compensação pela deformação de área, o que é muito natural, devido às diversas aplicações dos mapas. Fazer a correção do mapa pela deformação de área é interessante em algumas situações, especificamente quando desejamos utilizar um mapa de iluminação para construir uma *PDF* a fim de aplicar técnicas de amostragem e realizar implementações eficientes. Nos próximos capítulos veremos com mais detalhes as situações onde é conveniente trabalhar com o mapa corrigido

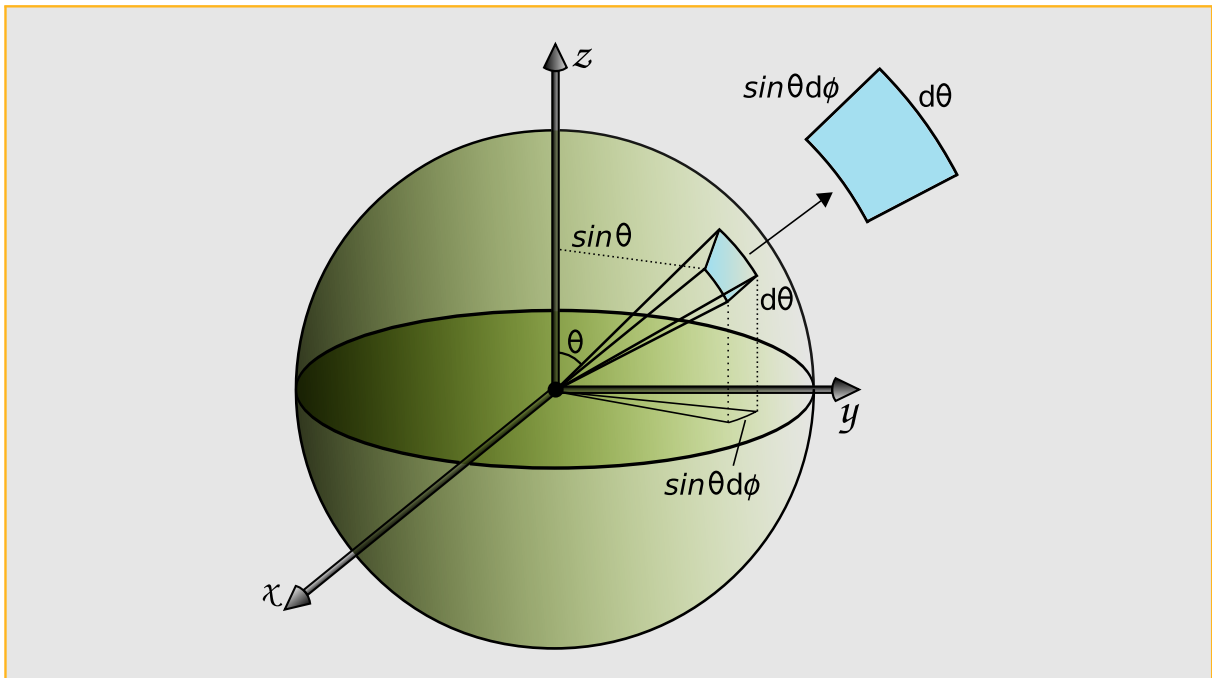


Figura 5.3: O diferencial de área dA subtendido por um diferencial de ângulo sólido é o produto do diferencial dos comprimentos das duas arestas $\sin \theta d\phi$ e $d\theta$. A relação resultante, $d\omega = \sin \theta d\phi d\theta$, é utilizada para a conversão entre integrais sobre ângulos sólidos e integrais sobre ângulos esféricos.

e onde é preciso o mapa original.

No caso de mapas de iluminação de tamanho $W \times H$ pixels, se $L_{M_o}(\bar{x}, \bar{y})$ é a função de radiância do mapa original, coincidente com a radiância do *Light Probe*, a função de compensação pela deformação de área é definida por

$$F_{M_o}(\bar{x}, \bar{y}) := M_o(\bar{x}, \bar{y}) \cdot \sin\left(\frac{\bar{y}}{H}\pi\right) = M_o(\bar{x}, \bar{y}) \cdot \sin(\theta). \quad (5.8)$$

Assim, dado um mapa original $M_o(\bar{x}, \bar{y})$ no formato Latitude-Longitude, a relação com seu transformado $M_c(\bar{x}, \bar{y})$, pela função $F_{M_o}(\bar{x}, \bar{y})$ de correção de área, é dada pela equação

$$M_c(\bar{x}, \bar{y}) = F_{M_o}(\bar{x}, \bar{y}), \quad (5.9)$$

Da mesma forma, temos para a luminância

$$L_{M_c}(\bar{x}, \bar{y}) = L(F_{M_o}(\bar{x}, \bar{y})) = L_{M_o}(\bar{x}, \bar{y}) \cdot \sin(\theta). \quad (5.10)$$

Na Figura 5.4 vemos o efeito de amostrar um mapa original isoluminante e sua transformação pela deformação de área. Como resultado, as posições das amostras mudam quando considerarmos a deformação de área induzida pelo mapeamento. No mapa sem a correção, as amostras estão distribuídas uniformemente oque significa que na esfera de direções correspondente tem uma concentração maior de amostras na região dos pólos.

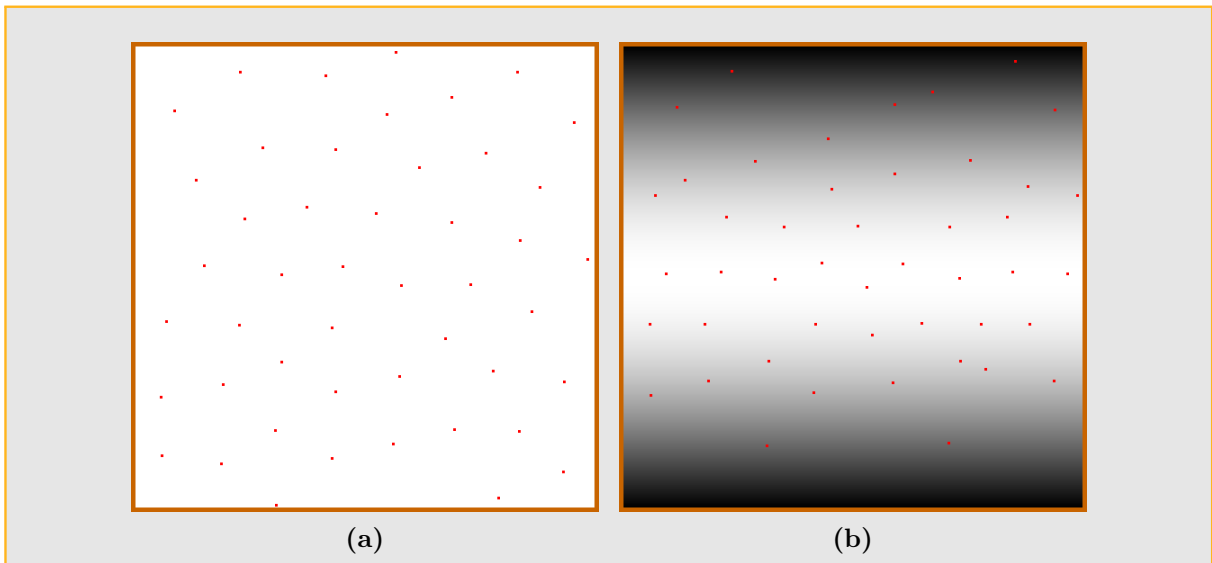


Figura 5.4: Amostragem com mosaicos de Penrose [ODJ04]: 5.4a mapa homogêneo, 5.4b mapa homogêneo com correção de área.

Por outro lado, no mapa com correção, é possível notar que a distribuição de amostras é uniforme na direção horizontal, mas na direção vertical tem-se maior quantidade de amostras na região central do mapa, isto faz com que a distribuição das amostras seja uniforme na esfera de direções correspondente.

6 *Amostragem de mapas de iluminação*

6.1 Amostragem por Importância

Uma interface básica de amostragem de mapas de iluminação é implementada no *PBRT* através do plugin *InfiniteArealight*, e utiliza uma distribuição que tem como pesos o cosseno sobre as direções do hemisfério quando amostra as direções de iluminação incidente no ponto a ser sombreado, ou seja, realiza uma amostragem uniforme sobre o hemisfério visível. Enquanto que esta distribuição de amostragem garante convergência ao resultado correto no limite (devido a que se tem probabilidade não nula de selecionar qualquer direção), pode levar a grande variância em estimativas de iluminação direta, se o mapa de iluminação utilizado tem muito mais luminosidade em algumas partes e pouca em outras. Em geral, sabemos que mapas de iluminação de ambientes realistas tem esta característica.

Atualmente, o *PBRT* possui um novo plugin *InfiniteAreaLightIS*, desenvolvido por Matt Pharr e Greg Humphreys [PH04a], baseado em amostragem por importância do mapa de iluminação.

A seguir vamos mostrar como utilizar o mapa de iluminação para definir uma função de distribuição de probabilidade para amostragem por importância sobre a esfera de direções. Este enfoque é simples de ser implementado e reduz consideravelmente a variância em imagens renderizadas utilizando mapas de iluminação, devido fornecer uma boa forma de aproximar a distribuição de um dos termos no integrando de iluminação direta (equação 4.2). Em conjunto com amostragem múltipla por importância para dar peso às amostras recolhidas, pode-se gerar imagens de alta qualidade com uma baixa taxa de amostragem. Com um mapa de iluminação onde as luzes de brilho intenso estão ocupando pequenas regiões, utilizar amostragem por importância resulta muito efetivo. Com o método do cosseno muitas vezes a maior parte das amostras pode cair fora das pequenas

regiões com alta luminância, e o resultado final é uma imagem com alta variância.

6.1.1 Esquema prático de amostragem por importância com técnicas unidimensionais

Basicamente podemos pensar a construção do esquema de amostragem a partir de três passos fundamentais:

1. Definir uma função constante por partes em coordenadas de imagem (u, v) , $p(u, v)$, baseada na distribuição de radiância do mapa de iluminação.
2. Construir um método de amostragem para transformar coordenadas (ξ_1, ξ_2) 2D uniformemente distribuídas em $[0, 1] \times [0, 1]$ em amostras tomadas da distribuição constante por partes $p(u, v)$.
3. Definir a Função de Densidade de Probabilidade (*PDF*) sobre as direções na esfera unitária baseada na densidade de probabilidade (*PDF*) sobre (u, v) .

A combinação destes três passos permite obter amostras na esfera de direções de acordo com a distribuição, que aproxima muito bem a função de radiância, obtendo-se uma redução substancial na variância da imagem.

Passo 1: Definir a função $p(u, v)$

Seja M_o , de $W \times H$ pixels (figura 6.1), um mapa de iluminação obtido através do mapeamento Latitude-Longitude de um *Light Probe*, e chamemos de $M_o(u, v)$ sua representação em coordenadas de imagem, onde $(u, v) \in [0, W] \times [0, H]$.

A seguir definimos a função $F(u, v)$,

$$F(u, v) := L_{M_o}(u, v) \cdot \sin(v/H) = L(M_o(u, v)) \cdot \sin(v/H). \quad (6.1)$$

Observe que $F(u, v)$ é a luminância do mapa $M_o(u, v)$ corrigida pela deformação de área sofrida no mapeamento Latitude-Longitude. A multiplicação pelo $\sin(v/H)$ permite ajustar a função de importância para considerar a distorção do mapeamento Latitude-Longitude. Como a função $M_o(u, v)$ é constante por partes dentro das regiões $[[u] + 1) \times [[v] + 1)$, que representam a área do pixel $([u], [v])$ da imagem M_o , também podemos considerar a modo de simplificação a função $\sin(v/H)$ como sendo constante dentro da área

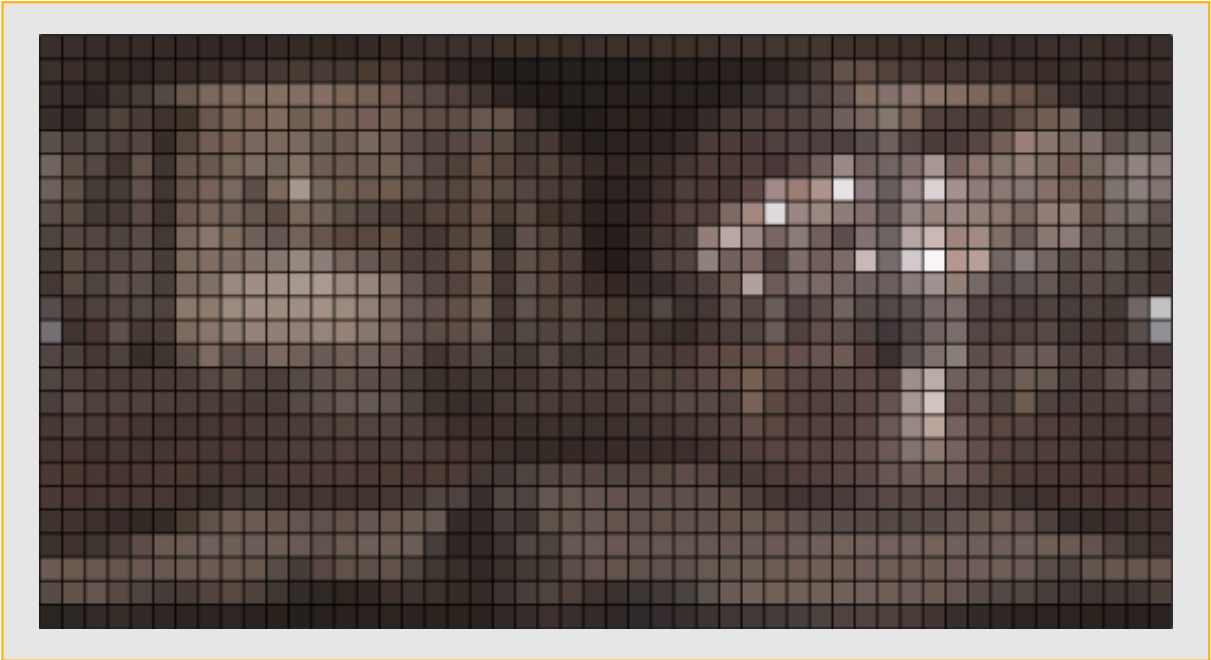


Figura 6.1: Mapa *Galileo* $M_o(u, v)$ redimensionado com 100×50 pixels.

de cada pixel. Para isto, vamos trocar a função contínua $\sin(v/H)$ pela função constante por partes $\sin(\tilde{v}/H)$, $\tilde{v} = \lfloor v \rfloor + \frac{1}{2}$. É fácil ver que a função $\sin(\tilde{v}/H)$ é constante por partes nas mesmas regiões que a função M_o . Logo podemos definir a função constante por partes $f(u, v)$ da figura 6.2,

$$f(u, v) := L_{M_o}(u, v) \cdot \sin(\tilde{v}/H) = L(M_o(u, v)) \cdot \sin(\tilde{v}/H), \quad \tilde{v} = \lfloor v \rfloor + \frac{1}{2}. \quad (6.2)$$

Para entender como amostrar de uma distribuição 2D $p(u, v)$, lembramos (seção 3.5) que para distribuições gerais de probabilidade multidimensional *conjunta*, cada dimensão deve ser amostrada numa forma baseada nas amostras obtidas para as dimensões prévias. Para este caso, considere a função de importância $f(u, v)$, constante por partes no domínio $[0, W] \times [0, H]$, cuja imagem é determinada por um conjunto de WH valores. Podemos representar os valores adotados pela função $f(u, v)$ como $f_{\hat{u}, \hat{v}}$, onde $f_{\hat{u}, \hat{v}}$ é o valor de $f(u, v)$ sobre o intervalo $[\hat{u}, \hat{u} + 1) \times [\hat{v}, \hat{v} + 1)$, para $\hat{u} = \lfloor u \rfloor$, $\hat{v} = \lfloor v \rfloor$ (parte inteira).

A distribuição *conjunta* da função $f(u, v)$ 2D é definida como

$$p(u, v) = \frac{f(u, v)}{\int \int f(u, v) \, du \, dv}.$$

Como $f(u, v)$ é uma função constante por partes, pois foi definida a partir da luminosidade do mapa de iluminação M_o , as integrais resultam simplesmente em somatórios de

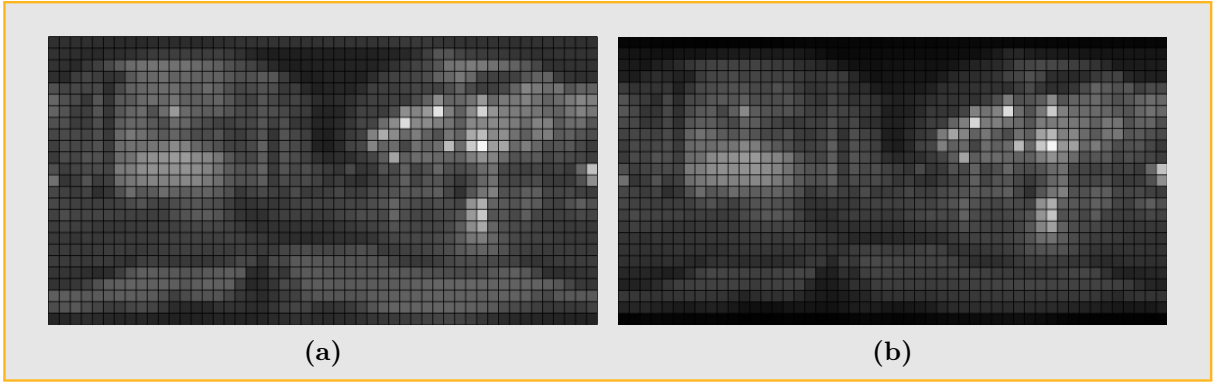


Figura 6.2: (a) Função de luminância $L_{M_o}(u, v)$ do Mapa da figura 6.1, (b) Função de luminância (a) com mapeamento do seno, $f(u, v)$.

valores $f_{\hat{u}, \hat{v}}$,

$$p(u, v) = \frac{f(u, v)}{\sum_{\hat{u}=0}^{W-1} \sum_{\hat{v}=0}^{H-1} f_{\hat{u}, \hat{v}}},$$

i.e., a probabilidade de um ponto do mapa é igual ao valor da sua luminância sobre a luminância total do mapa. É possível definir

$$I_f = \iint f(u, v) du dv = \sum_{\hat{u}=0}^{W-1} \sum_{\hat{v}=0}^{H-1} f_{\hat{u}, \hat{v}}.$$

Passo 2: Amostragem 2D por inversão

A *função de densidade marginal* $p_u(u)$, obtida integrando sobre a variável v , é facilmente calculada somando os valores $f_{\hat{u}, \hat{v}}$

$$p_u(u) = \int p(u, v) dv = \frac{\sum_{\hat{v}=0}^{H-1} f_{\hat{u}, \hat{v}}}{I_f},$$

para $\hat{u} = \lfloor u \rfloor$. Observe que $p_u(u)$ é uma função constante por partes com W valores. Como $p_u(u)$ é uma função 1D constante por partes, é possível tomar uma amostra u pelo método de inversão para funções constantes por partes descrito na seção 3.3.2. Para aplicar o método de inversão, definimos a *CDF* da função $p_u(u)$ como:

$$CDF_u(u) = \frac{\int_0^u \int_0^H f(u, v) dv du}{I_f} = \frac{\sum_{\hat{i}=0}^{\hat{u}-1} \sum_{\hat{j}=0}^{H-1} f_{\hat{i}, \hat{j}}}{I_f} + \frac{\left(\sum_{\hat{v}=0}^{H-1} f_{\hat{u}, \hat{v}} \right) (u - \hat{u})}{I_f}. \quad (6.3)$$

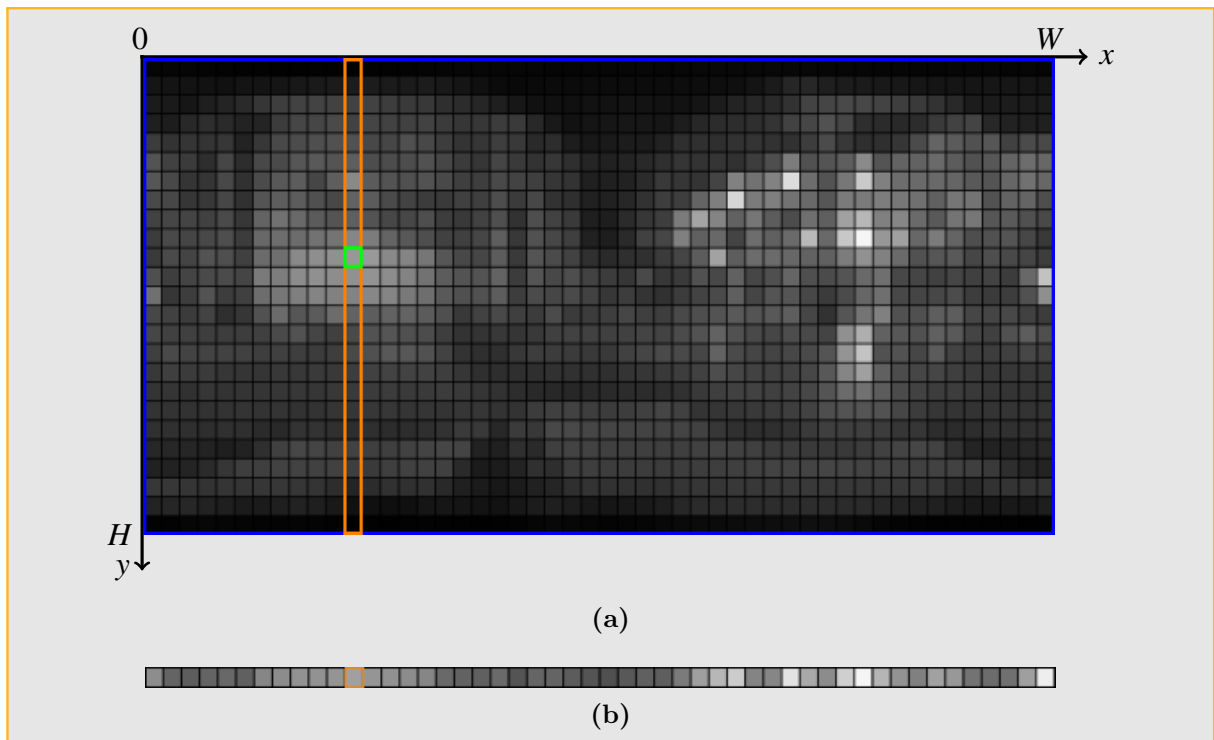


Figura 6.3: (a) Distribuição conjunta $p(u,v)$ da função $f(u,v)$ da figura 6.2b, (b) Soma dos valores das colunas da distribuição conjunta.

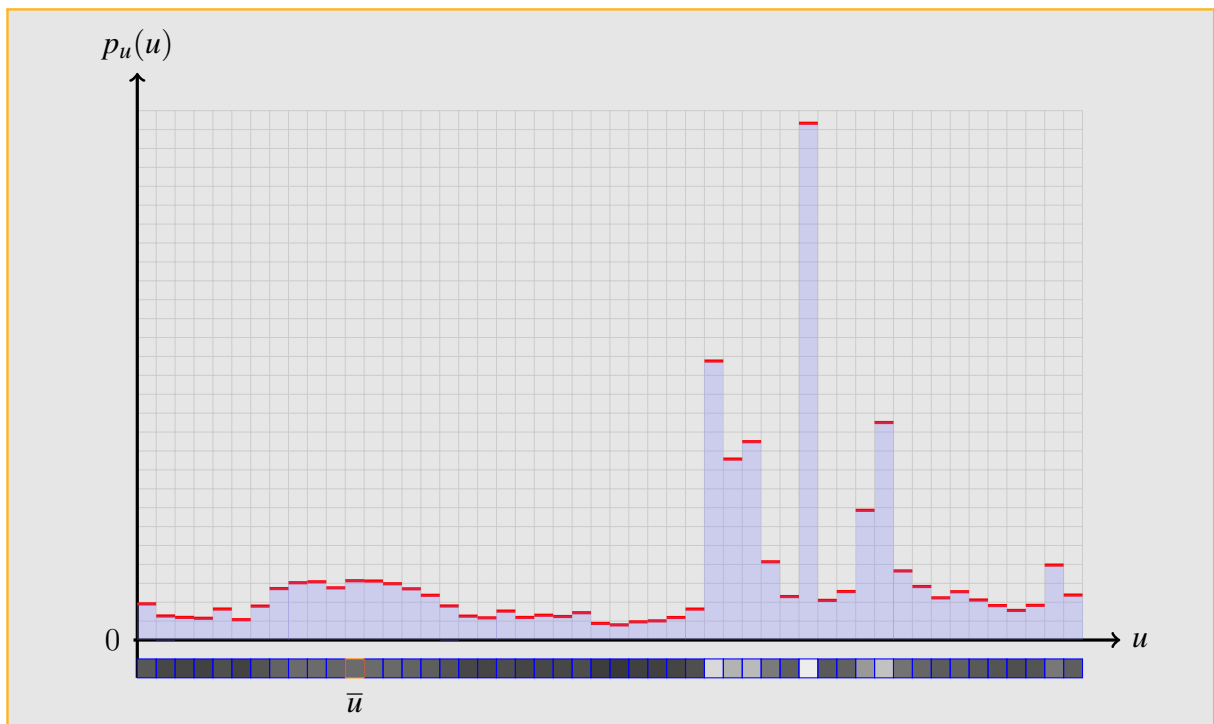


Figura 6.4: Função de probabilidade marginal $p_u(u)$ da função de probabilidade da figura 6.2b.

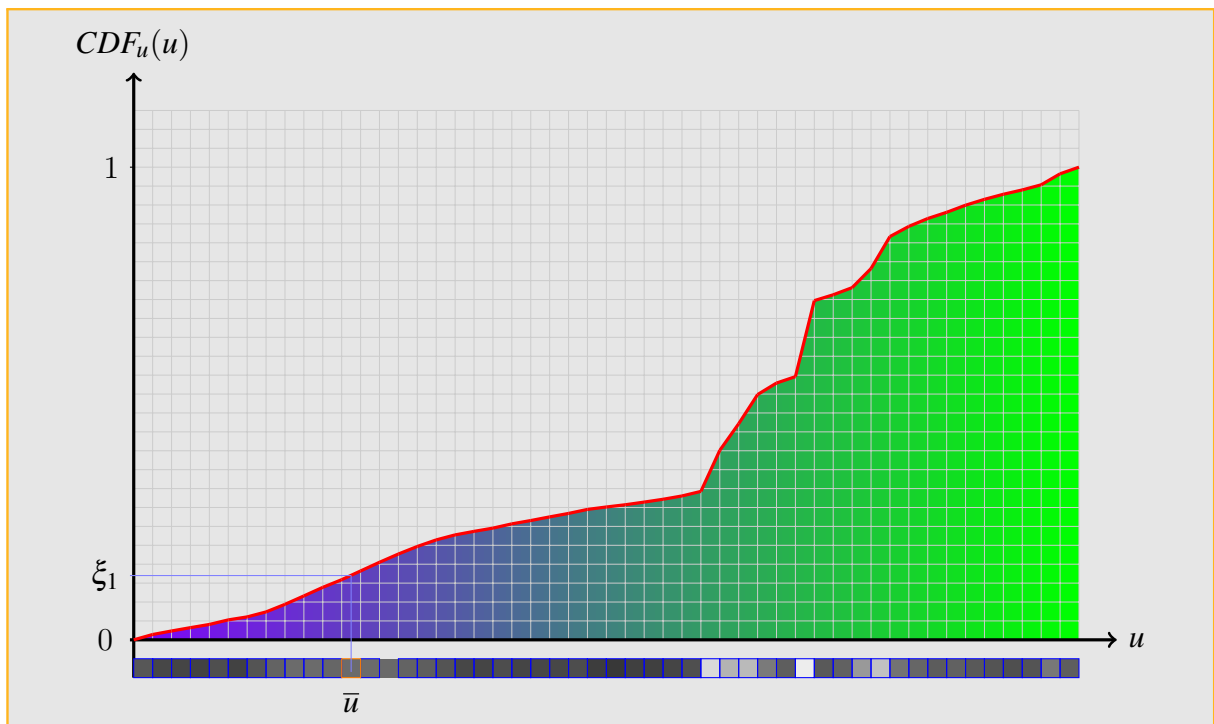


Figura 6.5: Função de densidade acumulada (CDF) da função de densidade marginal $p_v(v|\bar{u})$ da figura 6.4.

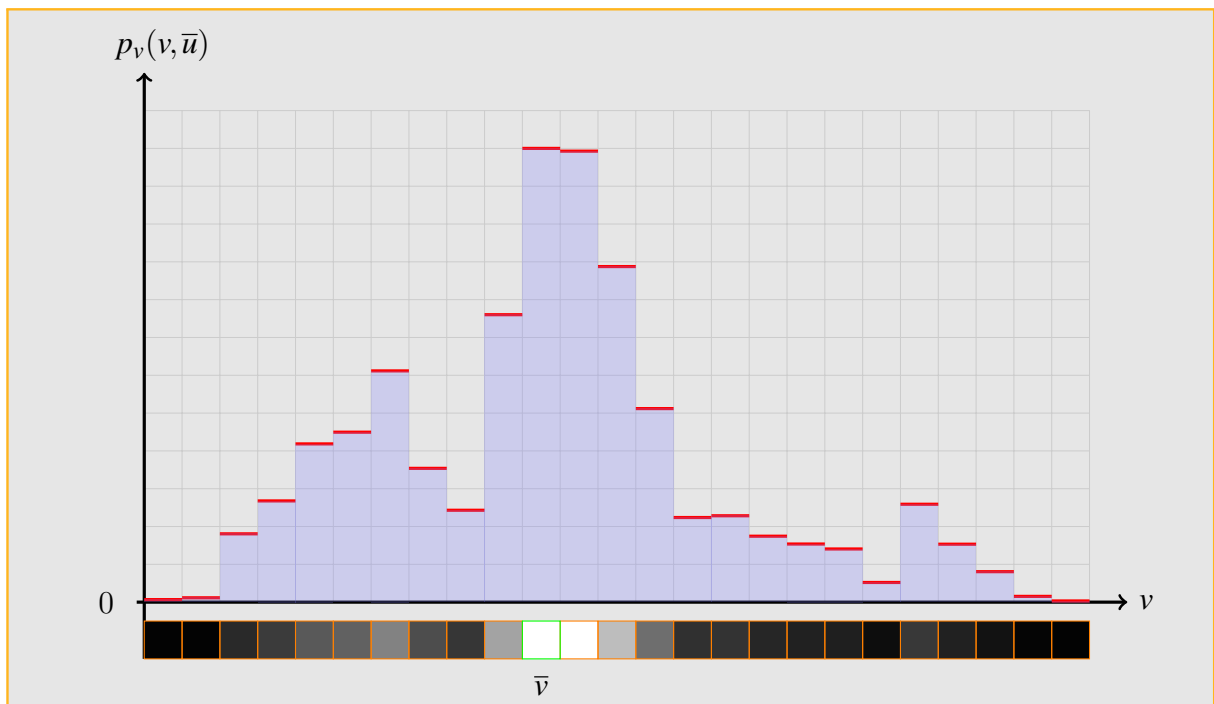


Figura 6.6: Função de probabilidade condicional $p_v(v|\bar{u})$ da função de densidade de probabilidade 6.2b, para uma amostra \bar{u} .

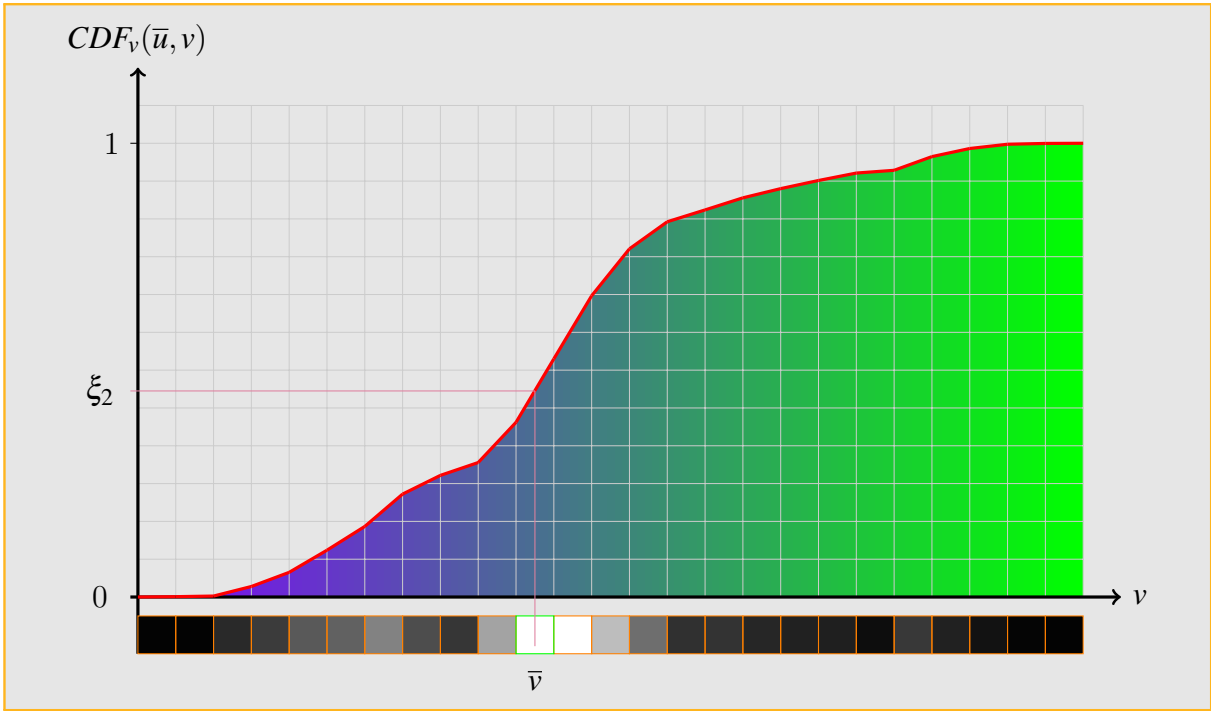


Figura 6.7: Função de densidade acumulada (CDF) da função de densidade condicional $p_v(v|\bar{u})$ da figura 6.6.

A função de densidade condicional $p_v(v|u)$ para v , dada uma amostra particular u , é

$$p_v(v|u) = \frac{p(u, v)}{p_u(u)} = \frac{f_{\hat{u}, v}}{I_f}.$$

Se a função constante por partes $p_u(u)$ é representada como conjunto de valores $g_{\hat{u}}$ com as convenções anteriores, teremos:

$$p_v(v|u) = \frac{p(u, v)}{p_u(u)} = \frac{f_{\hat{u}, v}}{g_{\hat{u}}}.$$

Dado \bar{u} , a função $p_v(v|\bar{u}) = \frac{p(\bar{u}, v)}{p_u(\bar{u})}$ também é constante por partes, e pode ser amostrada pelo método de inversão para funções constantes por partes descrito na seção 3.3.2.

Definimos a *CDF* da função $p_v(v|\bar{u})$ como segue

$$CDF_v(\bar{u}, v) = \frac{\int_0^v f(\bar{u}, v) dv}{I_f}. \quad (6.4)$$

Dada uma amostra (ξ_1, ξ_2) uniformemente distribuída em $[0, 1] \times [0, 1]$, podemos transformá-

la numa amostra (u, v) tomada da distribuição $p(u, v)$.

Primeiro, a distribuição $p_u(u)$ é calculada para $u = 0, \dots, W - 1$. Com $CDF(u)$ procura-se resolver $CDF(u) = \xi_1$ pelo método de inversão, e obtemos uma amostra u , tal que $\hat{u} = \lfloor u \rfloor$. Utilizando $CDF_{\hat{u}}(v)$, aplicamos o método de inversão para resolver $CDF_{\hat{u}}(v) = \xi_2$, obtendo a amostra v .

Assim com a amostra (u, v) , podemos agora transformá-la numa direção sobre a esfera unitária utilizando transformação de coordenadas.

Passo 3: Cálculo da PDF

Dada a amostra (u, v) , calculada na seção anterior, vejamos agora como transformá-la numa direção e calcular adequadamente sua PDF , a partir da PDF , da distribuição $p(u, v)$.

Como $(u, v) \in [0, W] \times [0, H]$ temos da seção que a transformação em coordenadas esféricas é:

$$(\phi, \theta) = \left(\frac{2\pi u}{W}, \frac{\pi v}{H} \right).$$

Logo é possível utilizar a fórmula de coordenadas esféricas 5.2 para obter a direção $\omega = (x, y, z)$.

Transformar a amostra (u, v) do mapa Latitude-Longitude numa amostra sobre a esfera de direções resulta simples utilizando a transformação de coordenadas para obter $\omega = (x, y, z)$. Ainda é preciso calcular a PDF para a amostra $\omega = (x, y, z)$, isto pode ser feito calculando a transformação entre a densidade utilizada para amostrar (u, v) e a densidade correspondente depois de mapear a imagem para a esfera unitária. Vale lembrar que a parametrização Latitude-Longitude de uma imagem (ϕ, θ) é $x = r \sin \phi \cos \theta$, $y = r \sin \theta \cos \phi$, $z = r \cos \theta$.

Consideramos a função g , que faz o mapeamento de (u, v) para (ϕ, θ) ,

$$g(u, v) = \left(\frac{2\pi u}{W}, \frac{\pi v}{H} \right).$$

O valor absoluto do determinante do Jacobiano $|J_g|$ é $2\pi^2/(WH)$. Realizando uma mudança de variáveis multidimensional (seção 3.4.1), é possível obter a densidade em

termos das coordenadas esféricas (ϕ, θ) .

$$p(\phi, \theta) = p(u, v) \frac{WH}{2\pi^2}.$$

Pela definição de coordenadas esféricas, resulta facilmente calcular o determinante do Jacobiano do mapeamento de (r, ϕ, θ) para (x, y, z) , que é $r^2 \sin \theta$. Como estamos trabalhando com a esferas de direções, temos $r = 1$. Aplicando novamente uma mudança de variáveis multidimensionais (seção 3.4.1), determina-se a relação final entre as densidades de probabilidade de direções na esfera em termos da densidade de probabilidade para a amostras da função constante por parte (ϕ, θ) ,

$$p(\omega) = \frac{p(\phi, \theta)}{\sin \theta} = p(u, v) \frac{WH}{2\pi^2 \sin \theta}.$$

Esta relação é a chave para a aplicação desta técnica: nos permite amostrar a partir de uma distribuição constante por partes definida pelo mapa de iluminação e transformar a amostra e sua densidade de probabilidade em termos das direções sobre a esfera unitária.

6.2 Amostragem por Algoritmo de Corte Mediano

Paul Debevec apresentou em 2005 um método simples para obter amostras de um mapa de iluminação, utilizando a idéia concebida por Paul Heckbert no algoritmo de corte mediano para quantização. A idéia consiste em subdividir sucessivamente o mapa de iluminação uma quantidade de vezes e extrair uma amostra de cada região obtida no final das subdivisões.

6.2.1 O algoritmo

Dado um Light Probe M_o no formato Latitude-Longitude, o algoritmo para n iterações é o seguinte:

1. Para corrigir a deformação sofrida pelo mapeamento latitude-longitude do mapa M_o , aplicamos a função de correção de área F_{M_o} ao mapa M_o e obtemos o mapa corrigido M_c .
2. Utilizando como função de amostragem a luminância do mapa M_c , calcula-se um mapa de somas de áreas M_s da luminância.

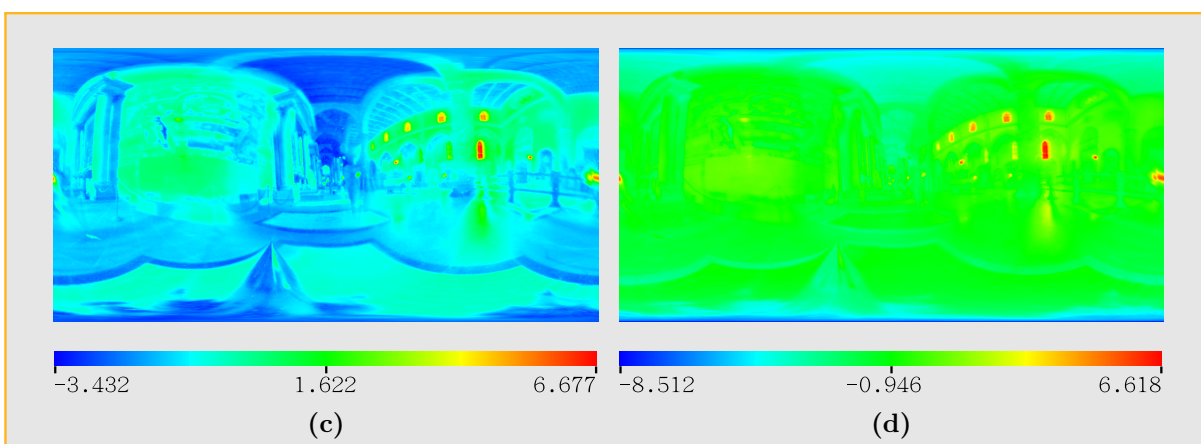
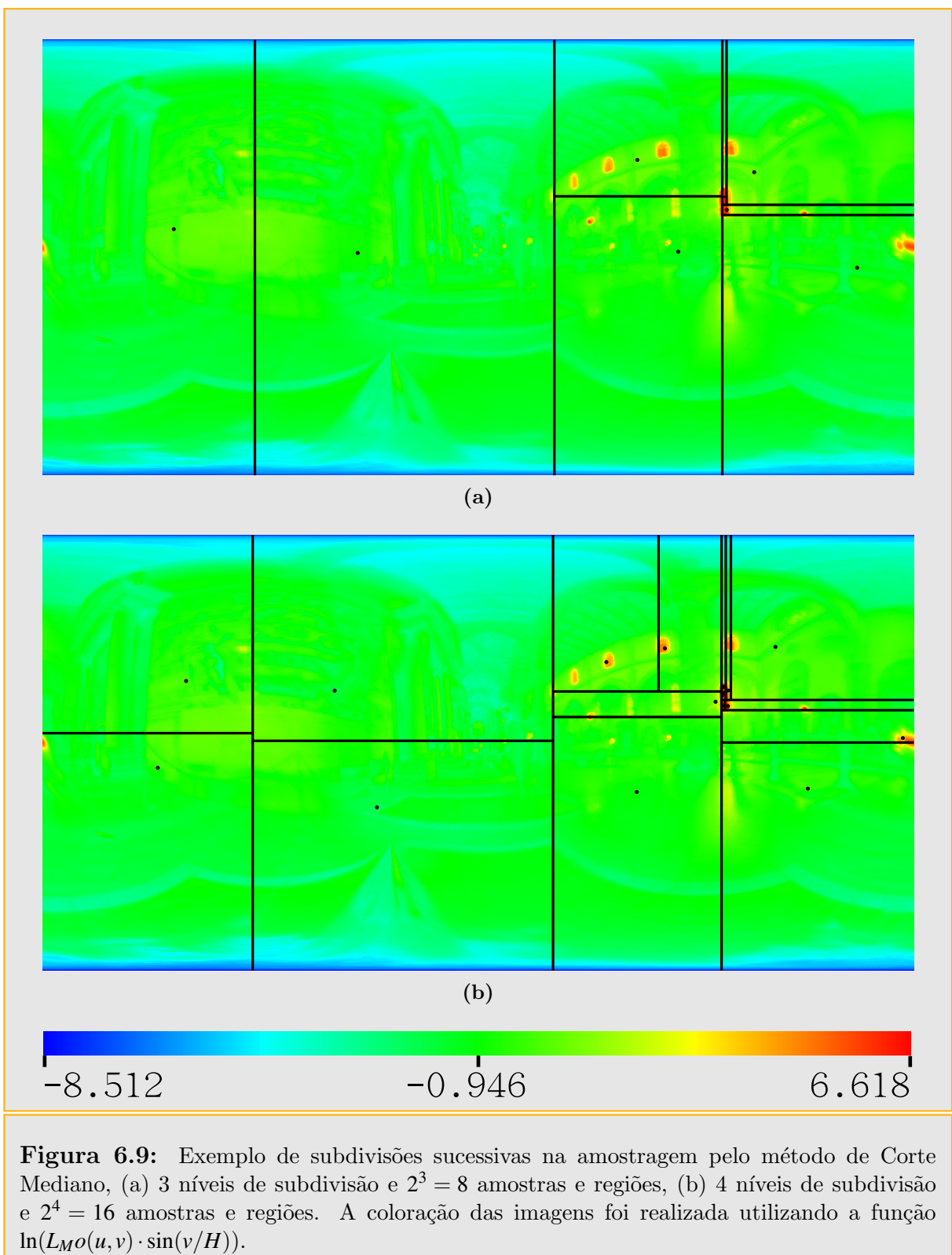


Figura 6.8: (c) Função de densidade de probabilidade determinada pela luminância do mapa *Galileo* A escala de cores utilizada na representação é $\ln(L_{M_o}(u, v))$, i.e, o logaritmo da luminância do mapa.(d) Função de densidade de probabilidade determinada pela luminância do mapa *Galileo* corrigida pela deformação de área sofrida pelo mapeamento esférico. A escala de cores utilizada na representação é $\ln(L_{M_o}(u, v) \cdot \sin(v/H))$, i.e, o logaritmo da luminância do mapa corrigida pela deformação de área.

3. Começa a *Lista* de regiões tomando o mapa completo M_c como uma região;
4. Divide todas as regiões da *Lista* ao longo da maior dimensão, em regiões de luminância similar.
5. Se o número de iterações é menor que n volta para 4.
6. Cada região da *Lista*, é convertida numa luz posicionada no centro da região, e a cor é calculada como a soma da cor dos pixels da região no mapa M_c .

A técnica é extremamente rápida, porém tem como inconveniente que somente é possível obter 2^n amostras em n iterações. Para problemas que requerem uma quantidade média de amostras isto pode ser um problema, pois por exemplo com $n = 6$ obtém-se 64 luzes, e com $n = 7$ obtém-se 128 luzes, portanto não temos como trabalhar com 80 luzes, por exemplo. Outro fato é que as luzes são determinadas de maneira única para cada n , i.e., mesmo rodando o algoritmo várias vezes para um valor n , o resultado é sempre o mesmo.



6.3 Amostragem hierárquica por importância

Apresentaremos agora uma estratégia eficiente para gerar mapas de amostras a partir de uma função de importância sobre um domínio 2D. A técnica que apresentaremos

consiste em realizar amostragem por importância a partir de um mosaico de Penrose modificado, que é subdividido recursivamente.

6.3.1 Amostragem hierárquica baseada no mosaico de Penrose

A princípios da década de 70, Roger Penrose encontrou uma forma de cobrir o plano com um mosaico aperiódico consistente de duas peças simples em forma de losango. Na figura 6.10 (lado esquerdo) vemos os dois tipos de azulejos de Penrose, e a regra de construção de um mosaico de Penrose utilizando os azulejos (lado esquerdo). Ostromoukhov et al. [ODJ04], utilizaram um conjunto alternativo de azulejos para construir um mosaico aperiódico similar ao mosaico original de Penrose, e cobrir um domínio plano 2D, realizando subdivisões sucessivas de maneira aperiódica segundo uma função de importância do domínio.

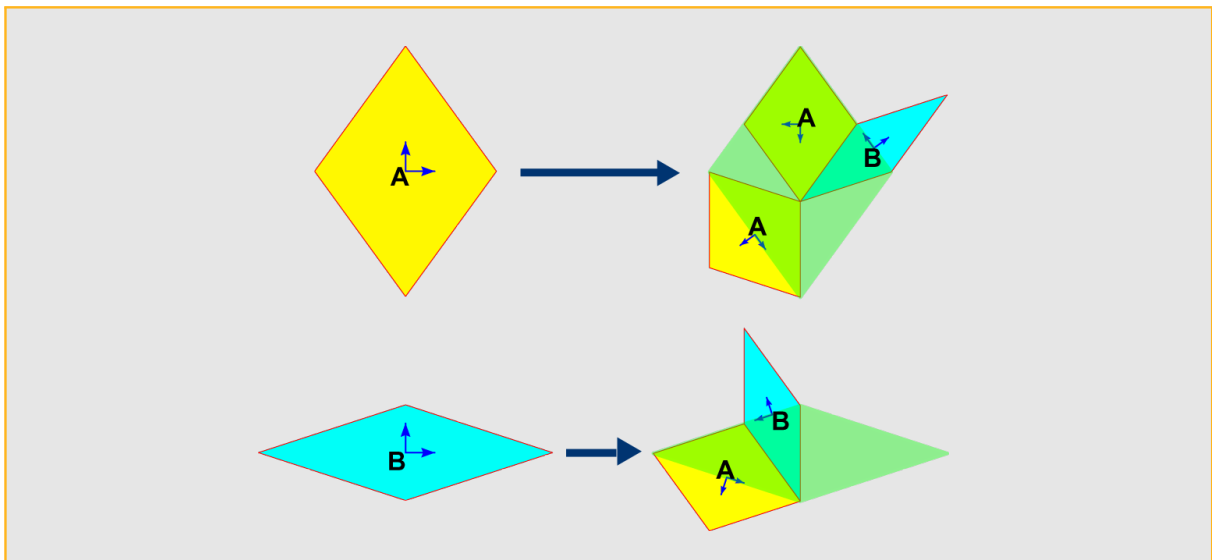


Figura 6.10: Os azulejos de Penrose são dois tipos de losangos, o mais fino com ângulos iguais a 36° e 144° e o maior com ângulos de 72° e 108° . Do lado direito é possível observar o processo de subdivisão utilizado por Penrose para construir um mosaico aperiódico com estes dois tipos de azulejos.

A forma alternativa do mosaico de Penrose, utilizada por Ostromoukhov et al. [ODJ04], pode ser representada com 6 tipos de azulejos, dos quais 4 peças são triangulares e duas peças são pentagonais. As peças pentagonais tem a finalidade de marcar a aperiodicidade do mosaico, e são chamadas de *azulejos de amostragem*. As peças pentagonais são consideradas infinitesimais, pois o único que interessa na prática é a localização do seu centro. A posição e orientação dos azulejos é representada por pares de vetores ortogonais (figura 6.11), os quais são importantes no processo de amostragem, como será visto mais

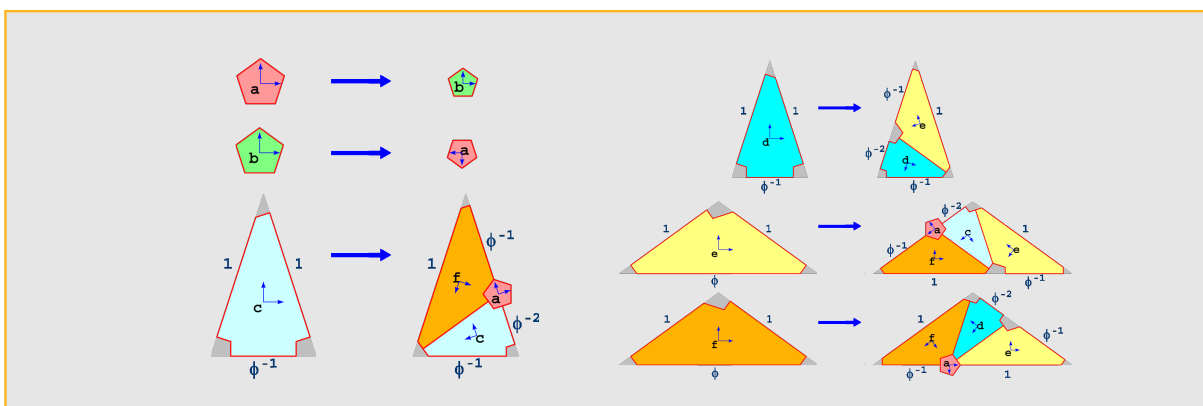


Figura 6.11: Regras de produção do mosaico de Penrose modificado, utilizado por Ostromoukhov [ODJ04]. ϕ é o número de ouro $\phi = \frac{1+\sqrt{5}}{2} \approx 1.61803$. Pares de vetores ortogonais formam a bases para cada azulejo.

adiante.

Considera-se o processo de subdivisão de Penrose da figura 6.11 como um processo recursivo. Um código binário, F-código, no sistema de Fibonacci descrito na seção 6.3.3 é dado a cada azulejo durante o processo de subdivisão.

O processo de subdivisão é realizado mediante as seguintes regras de produção:

$$Penrose := \begin{cases} a_* \mapsto \{b_{00*}\} \\ b_* \mapsto \{a_{00*}\} \\ c_* \mapsto \{f_{00*}, c_{10*}, a_{10*}\} \\ d_* \mapsto \{e_{00*}, d_{10*}\} \\ e_* \mapsto \{f_{00*}, c_{10*}, e_{01*}, a_{10*}\} \\ f_* \mapsto \{e_{00*}, d_{10*}, f_{01*}, a_{01*}\}, \end{cases} \quad (6.5)$$

onde x_y representa um azulejo do tipo x , com um código F igual a y . O símbolo $*$ representa o F-código já acumulado no azulejo pelas subdivisões anteriores. Cada subdivisão concatena 2 símbolos ao F-código atual. Assim depois de n subdivisões o F-código tem $2n$ símbolos. Os F-códigos são números inteiros expressados no sistema de Fibonacci, como descrito na seção 6.3.3.

Algumas observações importantes:

- Os números decimais que correspondem aos F-códigos atribuídos para os azulejos de amostragem pentagonais do tipo a e b estão todos no intervalo $[1, (F_{2(n+1)} - 1)]$, onde n é o nível de subdivisão dos azulejos iniciais, e F_i é o i -ésimo número de Fibonacci.

- As subdivisões sucessivas enriquecem o conjunto de amostras obtidos com as subdivisões anteriores, colocando novos pontos de amostragem entre as amostras anteriores.
- Os Números decimais já atribuídos a um nível de subdivisão n permaneceram na mesma posição durante todas as subdivisões seguintes. Observe que ao obter um azulejo de amostragem do tipo a ou b , o valor decimal do seu F-código permanecerá constante nas subdivisões sucessivas, como consequência das regras de produção 6.5.

Vejamos agora como funciona o sistema de amostragem adaptativa por importância utilizando o mosaico de Penrose com as regras de produção 6.5. Em primeiro lugar, a área de interesse é coberta com um par de azulejos do tipo e e f , como na figura 6.12. A continuação é aplicada a subdivisão recursivamente, mediante as regras de produção 6.5. A subdivisão de um azulejo acaba quando o nível de subdivisão κ é atingido. Neste caso, a saída é a posição dos centros dos azulejos de tipo a e b , cujo valor de importância local é maior que o valor decimal do F-código do azulejo atual. A fim de obter a quantidade desejada de amostras no final do processo de subdivisão, a função de importância $I(x,y)$ é escalada por um fator constante mag .

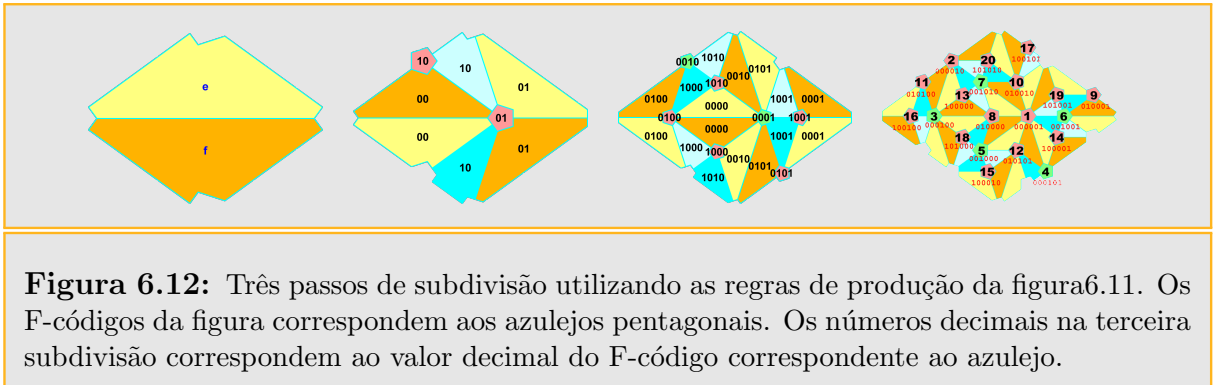


Figura 6.12: Três passos de subdivisão utilizando as regras de produção da figura 6.11. Os F-códigos da figura correspondem aos azulejos pentagonais. Os números decimais na terceira subdivisão correspondem ao valor decimal do F-código correspondente ao azulejo.

O nível de subdivisão de um azulejo é determinado como

$$\kappa = \lceil \log_{\phi^2} \max_{tile} (mag \cdot I(x,y)) \rceil, \quad (6.6)$$

onde $\lceil x \rceil$ é o primeiro inteiro maior ou igual que x , $I(x,y)$ é a função de importância na posição (x,y) , e $\phi = \frac{1+\sqrt{5}}{2}$ é o número de ouro. O fator $\log \phi^2$ pode ser entendido como o fator de similaridade do mosaico de Penrose, pois a cada subdivisão a área dos azulejos decaem num fator ϕ^2 . O valor $\max_{tile}(\cdot)$ representa o valor máximo dentro do azulejo $tile$, que geralmente na implementação prática do algoritmo somente é calculado em alguns pontos do azulejo.

6.3.2 Relaxação com dados tabelados

A fim de melhorar a distribuição espacial das amostras obtidas, Ostromoukhov et al. [ODJ04] utiliza uma tabela de *vetores corretores*, para realocar as posições das amostras. Estes vetores corretores estão dados em termos da base ortogonal de cada azulejo e tem um efeito de relaxamento na distribuição de amostras. O fato do mosaico de Penrose ser aperiódico, inviabiliza o cálculo de *vetores corretores* para cada amostra. No entanto, a natureza de similaridade no mosaico permite obter um número limitado destes vetores corretores e ainda conseguir um bom resultado. Para construir a tabela de vetores, os azulejos são etiquetados com um *índice estrutural* i_s que corresponde ao valor decimal dos 6 primeiros símbolos do F-código. Isto dá um total de 21 etiquetas diferentes, e cada uma destas etiquetas é associada a um vetor corretor. Ostromoukhov et al. [ODJ04] determinaram esta indexação experimentalmente.

Como os *vetores corretores* devem resultar representativos para qualquer função de densidade, foi feita uma otimização para diferentes valores de importância. Ostromoukhov et al. [ODJ04] otimizaram sobre n valores de importância representados por um *índice de importância* i_v calculado como:

$$i_v = \lfloor n \cdot \Psi(\text{mag} \cdot I(x,y)) \rfloor, \quad (6.7)$$

onde $\lfloor \cdot \rfloor$ denota o inteiro inferior, e $I(x,y)$ é a função de importância na posição (x,y) , e $\Psi(\cdot)$ mapeia um número real no intervalo $[0,1]$. Ostromoukhov et al. [ODJ04] que com valor $n = 8$ são obtidos os melhores resultados.

Os vetores corretores resultantes otimizados sobre cada valor de importância são armazenados numa tabela de 8×21 elementos.

O procedimento realizado para obter a tabela é o seguinte:

- A tabela é inicializada com todos os vetores iguais a zero. Depois, para cada *índice de importância*, i_v , aplica-se os seguintes passos:
 1. É criado um conjunto grande de amostras utilizando a amostragem hierárquica aqui descrita, utilizando a importância atual, e os vetores corretores da última versão da tabela corretora.
 2. É aplicado relaxação de Lloyd neste conjunto, mantendo os pontos da borda fixos.

3. Para cada amostra é calculado a diferença entre a sua posição original e depois do relaxamento.
4. É calculada a média dos vetores corretores para todos os *azulejos de amostragem* com igual *índice estrutural* i_s . O resultado é armazenado na posição (i_v, i_s) na tabela.
5. São repetidos os passos (1)–(4) até obter convergência (geralmente de 5 a 10 iterações).

Ao final da otimização, os pontos corrigidos se correspondem muito bem com a verdadeira relaxação de Lloyd para todos os níveis de importância.

6.3.3 Sistema numérico de Fibonacci

O sistema de base ϕ , é um sistema posicional, onde $\phi = \frac{1+\sqrt{5}}{2}$ é o número de ouro. Qualquer número racional x pode ser escrito neste sistema de maneira exata como no sistema decimal ou binário, somente que no lugar de utilizar potências de 10 ou 2, o sistema utiliza potências de ϕ . Por exemplo, o número $(101.001)_\phi$ na base ϕ é

$$(101.001)_\phi = \phi^2 + \phi^0 + \phi^{-3} \approx 3.8541_{10}$$

O sistema de base ϕ é semelhante ao sistema de Fibonacci que veremos a continuação. No caso do sistema de Fibonacci, todo inteiro n pode ser representado no sistema de Fibonacci como uma soma de termos F_j da sequência de Fibonacci multiplicados pelos coeficientes posicionais, que podem ser 0 ou 1. Assim, um número n pode ser representado pelo seu F-código $(b_m b_{m-1} \cdots b_3 b_2)_F$:

$$n = (b_m b_{m-1} \cdots b_3 b_2)_F \iff n = \sum_{j=2}^m b_j F_j. \quad (6.8)$$

O primeiro índice no somatório é $j = 2$ pela convenção adotada para os números de Fibonacci F_j ;

$$F_0 = 0, \quad F_1 = 1, \quad F_2 = 1, \quad F_3 = 2, \quad F_4 = 3, \quad F_5 = 5, \quad F_6 = 8, \quad F_7 = 13, \quad \dots$$

A representação de um número no sistema de Fibonacci não é única, mas pode ser unicamente determinada colocando uma imposição chamada de *forma normal*, que não

permite dois símbolos 1 adjacentes na codificação de um número. A demonstração deste fato é dado pelo teorema de Zeckendorf, o qual pode ser encontrado em [KNU97] junto com o procedimento para transformar na forma normal uma sequência arbitrária de 0 e 1. Alguns números na forma normal no sistema de Fibonacci são

$$\begin{aligned} 1 &= (000001)_F, & 2 &= (000010)_F, & 3 &= (000100)_F, & 4 &= (000101)_F, \\ 5 &= (001000)_F, & 6 &= (001001)_F, & 7 &= (001010)_F, & 8 &= (010000)_F, \\ 9 &= (010001)_F, & 10 &= (010010)_F, & 11 &= (010100)_F, & 12 &= (010101)_F, \\ 13 &= (100000)_F, & 14 &= (100001)_F, & 15 &= (100010)_F, & 16 &= (100100)_F, \\ 17 &= (100101)_F, & 18 &= (101000)_F, & 19 &= (101001)_F, & 20 &= (101010)_F, \end{aligned}$$

A função $\Psi(x)$ que mapeia números reais positivos x no intervalo $[0,1]$, é definida como

$$\Psi(x) = (\log_{\phi^2} \sqrt{5} \cdot x) \bmod 1.$$

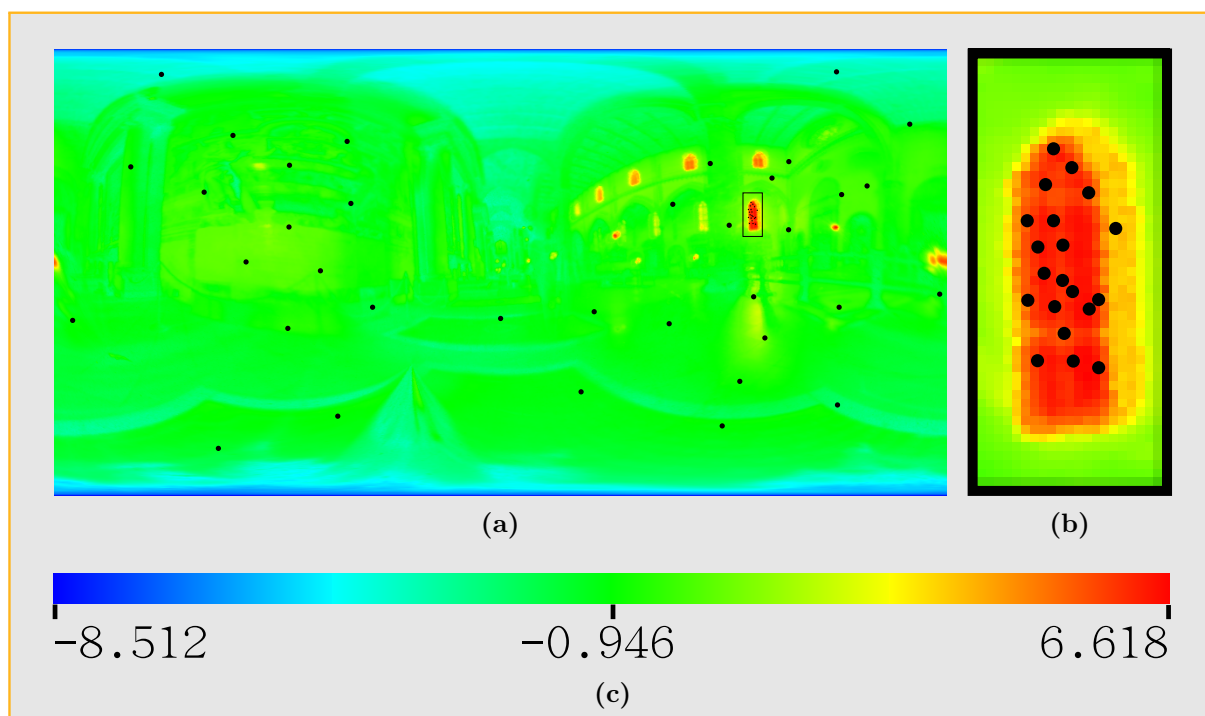


Figura 6.13: (a) Amostragem do mapa *Galileo* com 56 amostras pelo método de amostragem hierárquica por importância, (b) Detalhe ampliado da figura (a), (c) Escala logarítmica de cores, $\ln(L_{MO}(u,v) \cdot \sin(v/H))$, utilizada para representar a função de importância $L_{MO}(u,v) \cdot \sin(v/H)$.

Parte III

Nossa proposta

7

*Abordagem combinada***7.1 Diagramas de Voronoi**

Um diagrama de Voronoi é uma decomposição de um espaço métrico em regiões de acordo com a distância a determinados pontos. Dado um conjunto A de n pontos no espaço métrico $\Omega \subseteq \mathbb{R}^n$, queremos determinar para cada ponto $p_i \in A$, qual é a região $\mathcal{V}(A, p_i)$ dos pontos de Ω , que estão mais próximos de p_i que qualquer outro ponto de A .

A região de Voronoi $\mathcal{V}(A, p_i)$ correspondente ao ponto p_i é definida por:

$$\mathcal{V}(A, p_i) = \{x \in \Omega : d(x, p_i) \leq d(x, p_j) \quad \forall j = 1, \dots, n, \quad j \neq i\}.$$

Os pontos $p_i \in A$ são chamados de pontos geradores ou simplesmente geradores. O conjunto $\mathcal{V}(A, p_i)_{i=1}^n$ é o Diagrama de Voronoi de Ω , e $\mathcal{V}(A, p_i)$ é chamado de célula de Voronoi correspondente a p_i .

7.1.1 Diagramas de Voronoi na esfera unitária \mathbb{S}^2

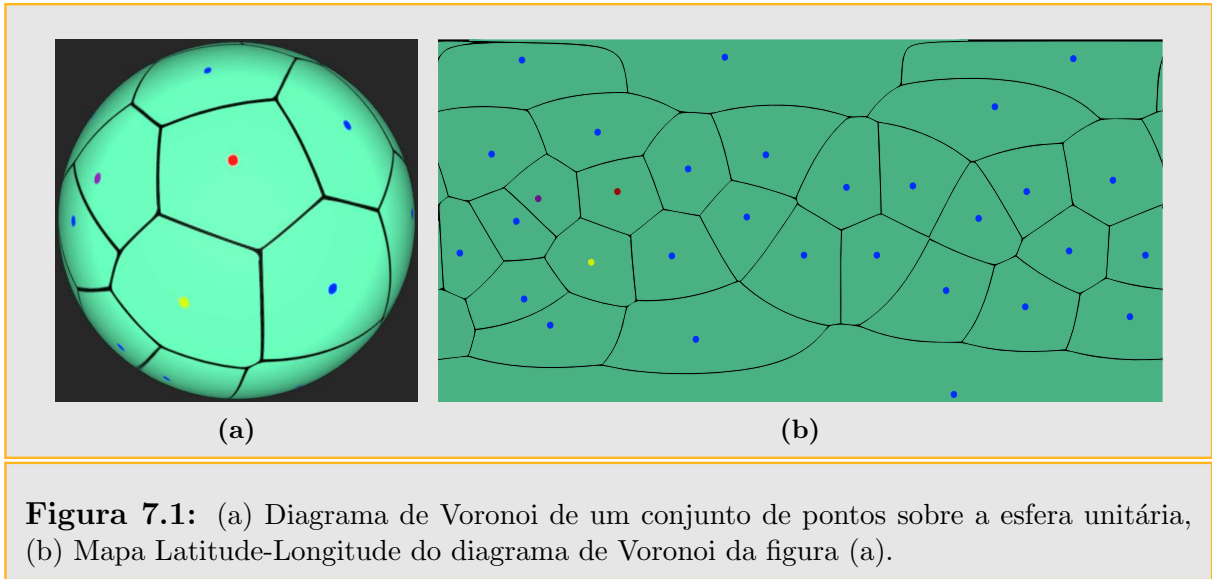
Seja $\mathbb{S}^2 = \{x \in \mathbb{R}^3 : \langle x, x \rangle = 1\} = \{x \in \mathbb{R}^3 : x_1^2 + x_2^2 + x_3^2 = 1\}$, a esfera unitária em \mathbb{R}^3 . Considere a métrica $d_s : \mathbb{S}^2 \times \mathbb{S}^2 \rightarrow \mathbb{R}$, definida por:

$$d_s(x, y) = \arccos(x \cdot y),$$

que é o comprimento do segmento de geodésica que une os pontos x e y .

Para $x, y, z \in \mathbb{S}^2$ são satisfeitas:

1. $d_s(x, x) = 0$;
2. Se $x \neq y$ então $d_s(x, y) > 0$;



$$3. d_s(x, y) = d_s(y, x);$$

$$4. d_s(x, z) \leq d_s(x, y) + d_s(y, z);$$

portanto, d_s é uma métrica para \mathbb{S}^2 .

Definição 1. Dados dois pontos $p_i, p_j \in \mathbb{S}^2$, a distância entre eles é definida como o comprimento de arco geodésico $d_s(p_i, p_j)$ que conecta p_i e p_j .

Definição 2. Dado um conjunto $A = \{p_1, p_2, \dots, p_n\}$, $p_i \in \mathbb{S}^2$, as regiões de Voronoi $\mathcal{V}(A, p_1), \mathcal{V}(A, p_2), \dots, \mathcal{V}(A, p_n)$ para a métrica d_s são definidas por:

$$\mathcal{V}(A, p_i) = \{p \in \mathbb{S}^2 : d_s(p, p_i) \leq d_s(p, p_j) \quad \forall j = 1, \dots, n, \quad j \neq i\}.$$

A fim de obter uma representação alternativa do diagrama de Voronoi, é possível definir o envelope inferior de funções sobre a esfera \mathbb{S}^2 de maneira similar à definição de envoltura inferior de funções bivariadas no espaço [HAL04]:

Definição 3. Dado um conjunto de funções bivariadas $F = f_1, f_2, \dots, f_n$, onde $f_i: \mathbb{S}^2 \rightarrow \mathbb{R}$, sua envoltura inferior $\Psi(x, y, z)$ é definida como o mínimo pontual $\Psi(x, y, z) = \min_{1 \leq i \leq n} f_i(x, y, z)$.

O diagrama de minimização $\mathcal{M}(F)$ do conjunto F é o mapa bidimensional obtido por projeção central da envoltura inferior sobre a esfera \mathbb{S}^2 .

Edelsbruner e Seidel [ES86] observaram a conexão entre os diagramas de Voronoi em \mathbb{R}^d e a envoltura inferior das correspondentes funções de distância aos sítios em \mathbb{R}^{d+1} . Isto

também vale para o caso esférico. Das definições acima, é claro que se $f_i : \mathbb{S}^2 \rightarrow \mathbb{R}$ é dado por $f_i(x) = d_s(x, p_i)$, para $i = 1, \dots, n$, então o diagrama de minimização de $\{f_1, \dots, f_n\}$ sobre \mathbb{S}^2 é exatamente o diagrama de Voronoi de A sobre \mathbb{S}^2 .

Um novo esquema baseado no algoritmo de envoltura do *CGAL* [MEY06] foi desenvolvido para computar diferentes tipos de diagramas Voronoi. O esquema provê uma reduzida e conveniente interface entre a construção dos diagramas e a construção de envolturas, que por sua vez, são calculados utilizando o pacote `envelope_3` do *CGAL* [MWZ08]. A obtenção de um novo tipo de diagramas de Voronoi depende basicamente do fornecimento de uma classe com as curvas de separação de fronteiras do novo tipo de diagrama [HSS09].

Ao utilizar o método de amostragem de Ostromoukhov, o resultado é um conjunto A de amostras pertencentes à esfera unitária \mathbb{S}^2 . Dependendo da forma que o conjunto A será utilizado, não é suficiente somente conhecer a posição das amostras. Por exemplo, se desejarmos utilizar o conjunto A para criar um conjunto de luzes direcionais a ser utilizado para iluminar a cena, devemos proceder de forma tal que a iluminação total do grupo de luzes direcionais seja igual à iluminação do mapa. Para lograr isto devemos particionar o mapa em regiões, somar a iluminação de cada região e associar a cada região uma luz direcional. Uma idéia natural de encarar este problema é construir luzes direcionais P_i com direções p_i , dadas pelas amostras p_i de A e associar a cada luz direcional P_i a iluminação que corresponde a célula de Voronoi sobre a esfera unitária \mathbb{S}^2 correspondente ao ponto p_i . Durante a implementação temos que resolver o problema de calcular uma partição da superfície da esfera unitária em regiões de Voronoi segundo o conjunto A de amostras obtido na etapa de amostragem. Dado que os algoritmos de amostragem são extremadamente rápidos, o maior tempo de processamento é gasto no cálculo da partição de Voronoi e na classificação dos pixels do mapa de iluminação segundo as regiões obtidas. Existem diversas formas de implementar o cálculo da partição de Voronoi para a esfera, aproveitando estruturas de dados que aceleram muito a classificação dos pontos do mapa de iluminação. No estado atual do nosso estudo somente realizamos o cálculo uma única vez para cada conjunto de amostras, e dado que trabalhamos com conjuntos com menos de 3000 amostras, um algoritmo de força bruta resolve bem o problema, pois não estamos interessados no tempo de pré-processamento de amostras, dado que os esquemas de amostragem utilizados são ultra-rápidos e sabemos que existem soluções eficientes e rápidas para calcular partições de Voronoi sobre a esfera [FSH08]. Não deixamos de mencionar este ponto, pois ele deve ser trabalhado futuramente para estender a abordagem híbrida para um esquema de amostragem em dois estágios, i.e., amostrar a fonte de luz e a *BRDF* a fim de obter uma aproximação melhor do produto da equação de iluminação

direta.

7.2 Metodologias tradicionais de renderização com mapas de iluminação

É comum ver na bibliografia atual sobre renderização foto-realista novas e interessantes abordagens do problema de iluminação. É possível inclusive escolher entre vários tipos de integradores para realizar o cálculo de iluminação. Porém o problema central nesta dissertação permanece: como escolher as fontes de luz para obter uma iluminação realista quando a informação que dispomos para trabalhar é um mapa de iluminação omnidirecional? Basicamente podemos dividir a literatura em duas abordagens sobre o problema: a primeira consiste em utilizar o mapa de iluminação para aplicar algum algoritmo de seleção de amostras, e utilizá-las para criar um conjunto de luzes pontuais. A segunda opção é utilizar o mapa de iluminação como uma luz de área infinita que envolve a cena e aplicar algoritmos de seleção de amostras durante a integração de Monte Carlo.

7.2.1 Geração de luzes direcionais

A geração de luzes direcionais pode ser realizada utilizando métodos de amostragem tais como o Corte Mediano ou Amostragem hierárquica com mosaicos de Penrose:

Algoritmo de Corte Mediano: quando for utilizado o método do Corte Mediano 6.2, o algoritmo implementado para este trabalho fornece a localização das amostras no mapa e também a iluminação total da célula a qual pertence cada amostra. Cada posição (u, v) de uma amostra no mapa de iluminação é transformada na sua correspondente direção (x, y, z) na esfera unitária, obtendo-se assim um conjunto de luzes direcionais que representa o mapa de iluminação.

Amostragem hierárquica com mosaicos de Penrose: utiliza o método de amostragem descrito na seção 6.3. O resultado obtido é uma lista da localização das amostras no mapa de iluminação. Com a localização das amostras no mapa de iluminação, calculamos sua posição (x, y, z) na esfera unitária mediante as equações 5.5 e 5.2. Em seguida calculamos as células de Voronoi sobre a esfera, e finalmente somamos a radiância em cada célula para obter a radiância que representará cada luz direcional.

O conjunto de luzes obtido é utilizado para renderizar a cena. Para ambos os métodos,

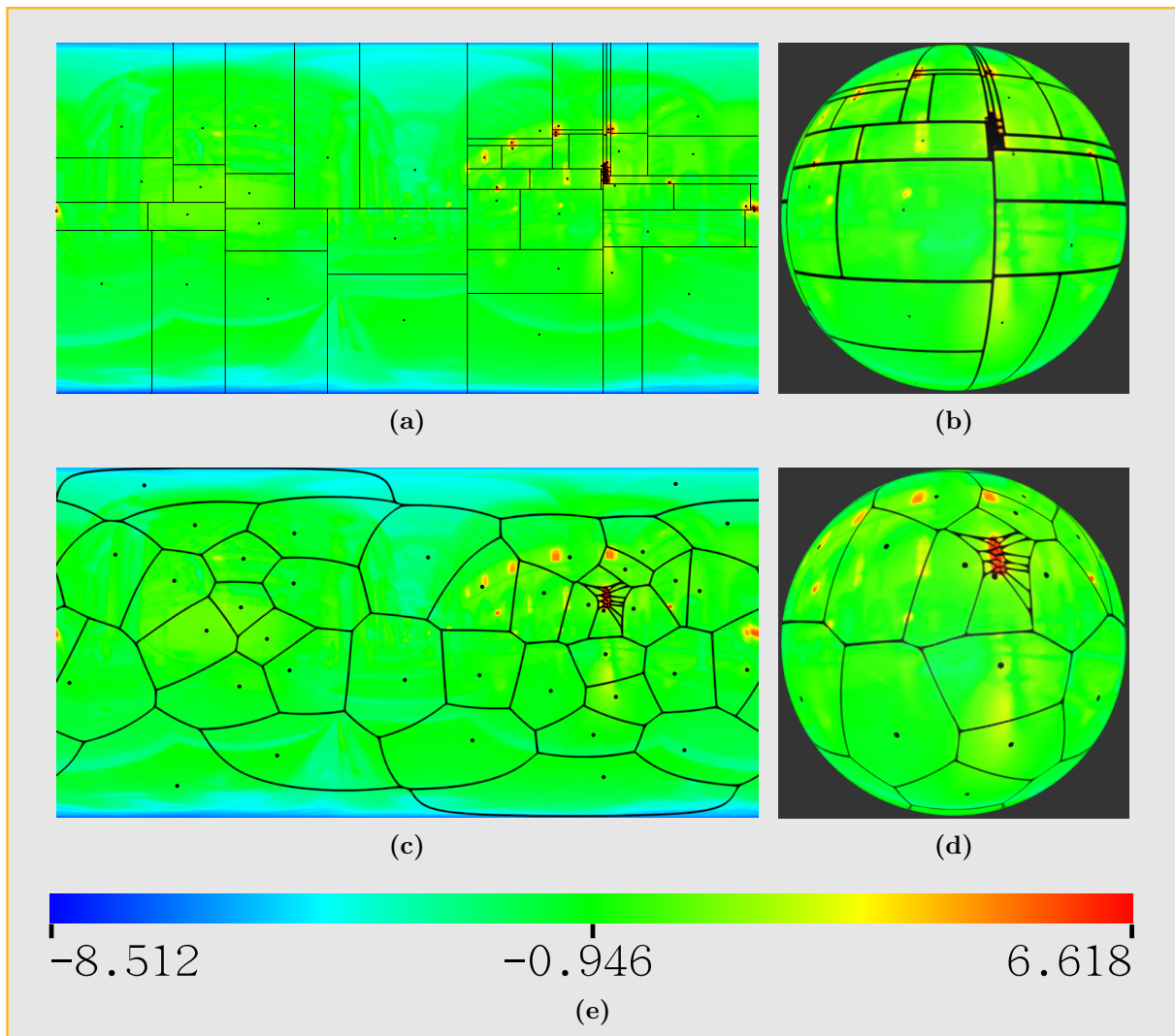


Figura 7.2: (a) Amostragem do mapa *Galileo* com o método de Corte Mediano utilizando 64 amostras. Os pontos pretos representam a localização das luzes. É possível observar que várias células retangulares tem uma grande diferença entre seus lados (retângulos achatados), (b) Mapeamento esférico do mapa de amostragem (a). As células de amostragem que tem a borda superior ou inferior do mapa como um dos seus lados compartilham o pólo norte ou sul, respectivamente, no mapeamento esférico. Muitas células tem como ponto em comum os pólos da esfera, (c), Amostragem hierárquica por importância do mapa *Galileo* utilizando 64 amostras. Os pontos pretos representam a localização das luzes e as células de Voronoi a zona do mapa, que cuja radiância será contabilizada para cada luz, (d) Mapeamento esférico do mapa de amostragem (c), onde é possível observar uma boa distribuição das amostras e células sobre a esfera, (e) Escala logarítmica de cores, $\ln(L_M \circ(u, v) \cdot \sin(v/H))$, utilizada para representar a função de importância $L_M \circ(u, v) \cdot \sin(v/H)$ do mapa *Galileo*.

o cálculo da radiância de cada luz é feito da seguinte maneira:

Sabe-se que a radiância total é dada por:

$$E = \int_{\mathbb{S}^2} R_M(\omega) d\omega = \int_0^{2\pi} \int_0^\pi R_M(\phi, \theta) \sin(\theta) d\theta d\phi,$$

em particular, para um mapa de iluminação discreto M de dimensão $W \times H$

$$\begin{aligned} E &= \int_0^{2\pi} \int_0^\pi R_M(\phi, \theta) \sin(\theta) d\theta d\phi \\ &= \frac{2\pi\pi}{W \cdot H} \sum_{i=0}^{W-1} \sum_{j=0}^{H-1} R_M(i, j) \cdot \sin\left(\frac{(j+0.5) \cdot \pi}{H}\right). \end{aligned}$$

Desta forma, para uma célula $C(p_k)$ correspondente à amostra p_k , temos que a radiância é dada por:

$$R_{C(p_k)} = \frac{2\pi\pi}{W \cdot H} \sum_{(i+0.5, j+0.5) \in C(p_k)} R_M(i, j) \cdot \sin\left(\frac{(j+0.5) \cdot \pi}{H}\right). \quad (7.1)$$

7.2.1.1 Vantagens e desvantagens

A renderização com luzes direcionais é simples de ser implementada, a iluminação na cena é uniforme, pois todas as luzes são utilizadas na renderização de cada pixel, portanto a cena apresenta pouca variância local. Entre os inconvenientes podemos mencionar a alta quantidade de luzes necessárias para obter efeitos realistas na renderização e o problema de penumbras e sombras com bordas definidas por causa da discretização do ambiente de iluminação. Este último problema pode ser evitado, na maioria dos casos, aumentando o número de luzes. Porém, isto pode acarretar em tempo de computação excessivo.

7.2.2 Amostragem direta

Dado um mapa de iluminação M_o , a amostragem direta é utilizada quando o mapa é considerado como uma luz de área infinita envolvendo a cena. O método de amostragem direta utilizado nesta dissertação é o de amostragem por importância descrito na seção 6.1, mas também pode ser utilizado amostragem por estratificação ou até mesmo uma amostragem combinada. A figura 7.3 mostra a distribuição das amostras no mapa *Galileo* durante uma renderização do modelo *Killeroo* (cena do dinossauro).

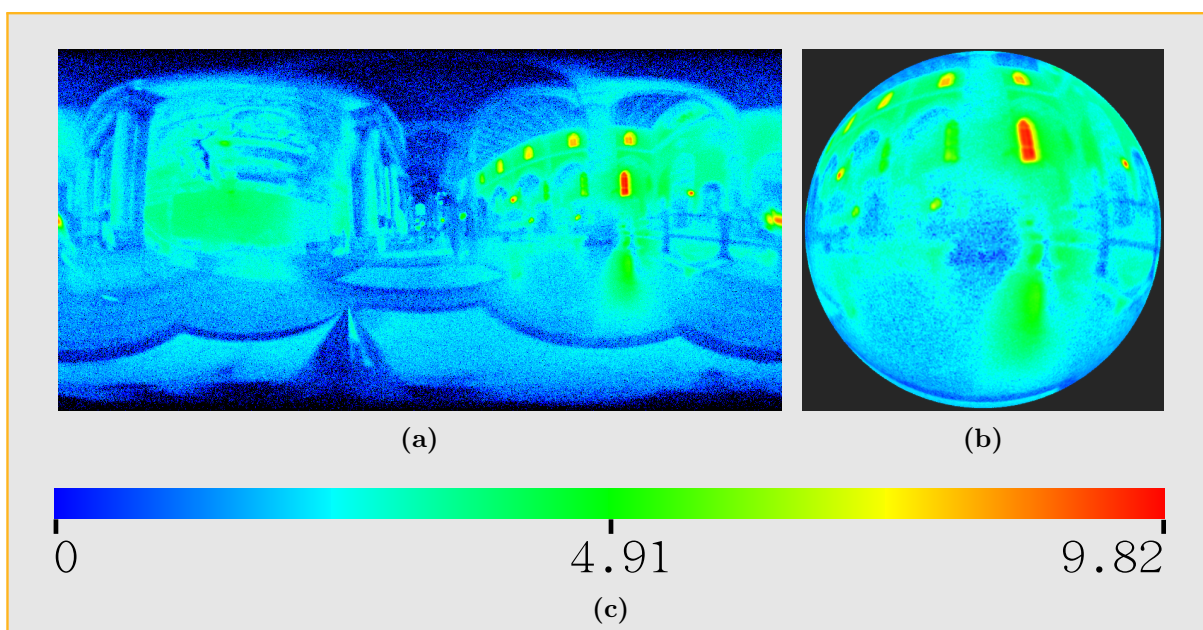


Figura 7.3: (a) Mapa de densidade de amostras durante uma renderização utilizando o mapa *Galileo* 1024×512 . O mapa de densidade foi construído contando a quantidade de vezes que a posição de um pixel do mapa *Galileo* foi amostrada durante a renderização. Foram tomadas 10434404 amostras durante a renderização. (b) Mapeamento esférico do mapa (a). (c) Escala logarítmica de cores utilizada para representar o mapa (a). A escala de cores é obtida calculando o logaritmo natural dos números de amostras coletado para cada posição do mapa *Galileo*.

7.2.2.1 Vantagens e desvantagens

O método de amostragem direta por importância tem como principal vantagem a obtenção de renderizações fisicamente corretas e de boa qualidade com uma quantidade relativamente baixa de amostras, dependendo da distribuição das fontes de iluminação na cena. Este método também proporciona sombras com bordas suaves e efeitos realmente realistas em penumbras. A principal desvantagem é a granulosidade na imagem final devido à variância associada aos métodos probabilísticos, que se não pode ser eliminada completamente, pode ser reduzida aumentando o número de amostras. Porém esta solução resulta excessivamente cara computacionalmente.

7.3 Geração de mapas de pré-amostragem

Uma alternativa às técnicas tradicionais de geração de luzes direcionais e à amostragem direta do mapa por métodos de Monte Carlo, que chamamos de mapa de pré-amostragem, consiste em gerar um número relativamente grande de amostras a partir do mapa de

iluminação, e passar estas amostras ao integrador para utilizá-las durante a renderização, em lugar de utilizar o mapa ou mesmo uma quantidade fixa de luzes direcionais. Durante a renderização, o integrador de Monte Carlo solicitará n amostras de luz para renderizar um pixel P , que serão selecionadas por importância do conjunto Ω , obtido pela pré-amostragem.

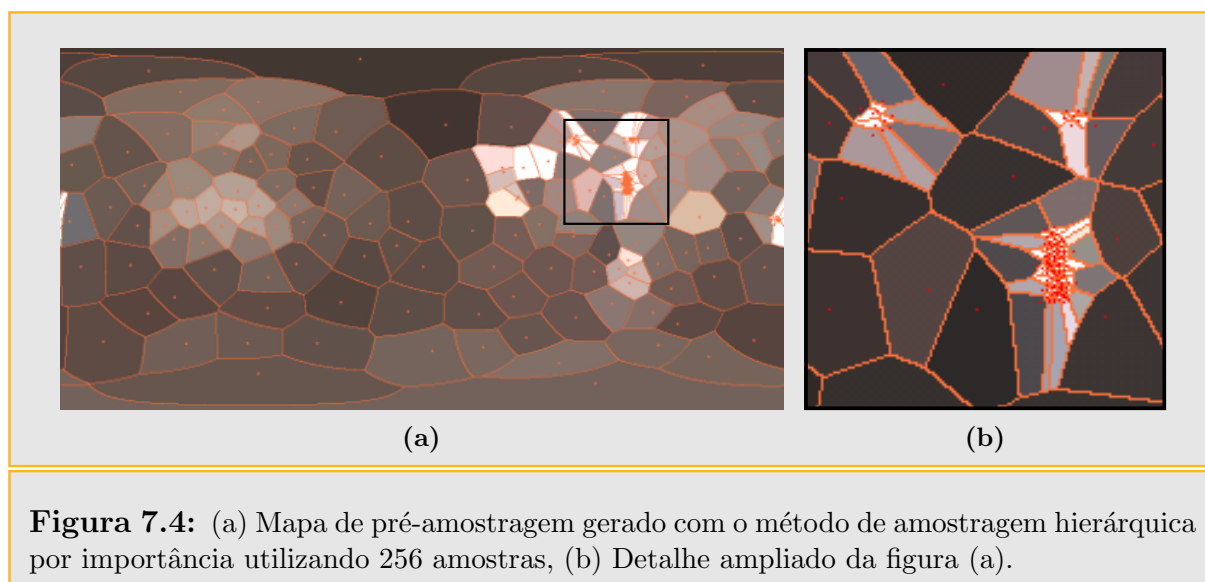


Figura 7.4: (a) Mapa de pré-amostragem gerado com o método de amostragem hierárquica por importância utilizando 256 amostras, (b) Detalhe ampliado da figura (a).

Desenvolvemos o plugin *PRESAMP* para o *PBRT* [PH04b], que utiliza um mapa de pré-amostragem para realizar a renderização. Na criação do mapa de pré-amostragem é preciso guardar as seguintes informações:

- a posição das amostras no mapa de iluminação;
- uma imagem latitude-longitude da partição do mapa em regiões de influência das amostras obtidas;
- uma lista da radiância total da região de influência de cada amostra, calculada mediante a equação 7.1;

O nosso software é capaz de criar mapas de pré-amostragem utilizando duas técnicas de amostragem: algoritmo de Corte Mediano, e amostragem hierárquica por importância com mosaicos de Penrose:

Algoritmo de Corte Mediano: o mapa de pré-amostragem pode ser gerado facilmente, pois o algoritmo nos provê da lista das posições das amostras e a lista da radiância de cada região de influência. A imagem da partição pode ser calculada durante a execução do algoritmo, utilizando as divisões sucessivas feitas no mapa, e guardada no final.

Algoritmo de amostragem hierárquica por importância com Mosaicos de Penrose: o mapa de pré-amostragem será gerado da seguinte maneira:

- Obtém-se o conjunto de amostras e suas posições;
- É calculado a partição de Voronoi da esfera unitária para o conjunto de amostras obtido;
- A partição de Voronoi obtida é guardada no formato latitude-longitude;
- Para cada amostra é calculada a radiância total da sua célula de Voronoi mediante a equação 7.1;

Esta aproximação do mapa de iluminação original por um mapa de pré-amostragem diminui entre 20 e 25 por cento o tempo de renderização da cena, além de contribuir para uma pequena melhora na qualidade visual da imagem final, devido à redução de variância local por causa da diminuição da quantidade de amostras. Basicamente este método proporciona uma aproximação mais suave da renderização devido que na etapa de amostragem, são utilizadas menos amostras que com a amostragem por importância.

A obtenção de mapas de pré-amostragem baseados no algoritmo de amostragem hierárquica por importância pode ser melhorada utilizando estratégias de clusterização 7.4.3 com a finalidade evitar aglutinamento de amostras e assim diminuir o conjunto de amostras, permitindo maior eficiência na renderização.

7.4 Esquema híbrido para renderização

Como vimos nas seções anteriores, as abordagens tradicionais do problema de iluminação baseada em imagens é feita de duas maneiras: uma gerando um conjunto de luzes direcionais ou pontuais a partir de um mapa de iluminação e renderizando a cena com este conjunto. A segunda alternativa é utilizar o mapa de iluminação para obter uma função de importância e tomar amostras durante a renderização para calcular posições numa luz de área definida pelo mapa de iluminação. Cada uma destas abordagens tem vantagens e desvantagens, como foi possível observar nas seções precedentes. Nossa proposta é uma abordagem intermediária, sistema híbrido, que permita utilizar na mesma renderização tanto luzes direcionais quanto amostragem direta do mapa, segundo a necessidade do problema, a fim de ter um controle fino da situação e obter melhoras nos resultados.

Além das propostas tradicionais de utilização dos mapas de iluminação, apresentaremos aqui uma nova idéia que visa aproveitar o melhor de cada abordagem anterior e obter resultados. Gostaríamos de aproveitar os benefícios de ambas as abordagens vistas anteriormente na seção 7.2, i.e., obter diminuição da variância na cena utilizando luzes direcionais importantes e suavizar as bordas das sombras, realizando amostragem por importância durante a renderização com métodos de Monte Carlo nas regiões do mapa onde a iluminação é mais fraca.

7.4.1 Separação do mapa de iluminação

Primeiramente, estratificamos o mapa de iluminação M_o utilizando a função de luminância para definir duas regiões A e B , tais que $A \cup B = [0, 2\pi] \times [0, \pi]$ e $A \cap B = \Phi$. A estratificação ou separação é feita utilizando a função de luminância para separar o mapa em duas regiões, uma com luminância alta e outra com luminância baixa.

Consideremos o mapa de iluminação latitude-longitude $M_o : [0, 2\pi] \times [0, \pi] \rightarrow \mathbb{R}^3$ e a função de luminância $L : \mathbb{R}^3 \rightarrow \mathbb{R}$, definida na seção 5.2.2, temos que luminância total do mapa M_o é:

$$E_{M_o} = \int_0^{2\pi} \int_0^\pi L_{M_o}(\phi, \theta) \sin(\theta) d\theta d\phi,$$

Dado um limiar real $p \in [0, 1]$, as regiões A e B são determinadas de forma tal a satisfazer as seguintes condições:

$$\int_0^{2\pi} \int_0^\pi \xi_A(\phi, \theta) \cdot L_{M_o}(\phi, \theta) \sin(\theta) d\theta d\phi = p \cdot E_{M_o}, \quad (7.2)$$

$$\int_0^{2\pi} \int_0^\pi \xi_B(\phi, \theta) \cdot L_{M_o}(\phi, \theta) \sin(\theta) d\theta d\phi = (1 - p) \cdot E_{M_o}, \quad (7.3)$$

$$\forall (\phi_A, \theta_A) \in A, \forall (\phi_B, \theta_B) \in B, L_{M_o}(\phi_A, \theta_A) \geq L_{M_o}(\phi_B, \theta_B). \quad (7.4)$$

Assim, o estrato A concentra $p \cdot 100\%$ da luminância total do mapa M_o nas regiões mais brilhantes, e B contém $(1 - p) \cdot 100\%$ da luminância nas regiões mais escuras.

A área dos estratos A e B considerados sobre a esfera unitária é dada por:

$$Area_{\mathbb{S}^2}(A) = \int_{\mathbb{S}^2} \xi_A(x) dx = \int_0^{2\pi} \int_0^\pi \xi_A(\phi, \theta) \cdot \sin(\theta) d\theta d\phi. \quad (7.5)$$

$$Area_{\mathbb{S}^2}(B) = \int_{\mathbb{S}^2} \xi_B(y) dy = \int_0^{2\pi} \int_0^\pi \xi_B(\phi, \theta) \cdot \sin(\theta) d\theta d\phi. \quad (7.6)$$

onde: $\xi_\Omega : \Delta \rightarrow \{0, 1\}$ é a função característica de $\Omega \subset \Delta$, e tem-se $\xi_\Omega(x) = 1$ se $x \in \Omega$; e $\xi_\Omega(x) = 0$ se $x \notin \Omega$.

Na prática, como o mapa de iluminação M_o é representado por uma imagem de $W \times H$ pixels, trabalhamos com uma discretização da esfera, obtida pelo mapeamento esférico da representação discreta do mapa de iluminação, o que facilita muito a implementação computacional. A implementação é realizada seguindo os seguintes passos:

- Dada a imagem do mapa M_o com $W \times H$ pixels, criamos um array $\bar{M}[W \times H, 2]$, de maneira que seus elementos sejam:

$$\bar{M}[k, 1] = L_{M_o}(\text{mod}(k, W), \lfloor k/W \rfloor),$$

$$\bar{M}[k, 2] = k.$$

- Ordena-se em forma decrescente o array \bar{M} segundo a primeira coordenada, i.e segundo $\bar{M}[x, 1]$. Para obter A , tal que:

$$\int_0^{2\pi} \int_0^\pi \xi_A(\phi, \theta) \cdot L_{M_o}(\phi, \theta) \sin(\theta) d\theta d\phi \approx p \cdot E_{M_o},$$

basta achar o menor inteiro r tal que:

$$\sum_{i=0}^r \bar{M}[i, 1] \cdot \sin\left(\pi \frac{\lfloor \bar{M}[i, 2] / W \rfloor + 0.5}{H}\right) \geq p \cdot E_{M_o},$$

Enquanto o valor $r < W \times H$ representa a área do estrato A em pixels na representação plana, pois consideramos cada pixel como sendo um quadrado de lado 1, o correspondente valor da área do estrato A depois do mapeamento esférico é obtido aplicando a deformação de área a cada $\bar{M}[i, 2]$ para $0 \leq i \leq r$ e realizando o somatório:

$$Area_{esferica}(A) = \sum_{i=0}^r \sin\left(\frac{\lfloor \bar{M}[i, 2] / W \rfloor + 0.5}{H} \cdot \pi\right). \quad (7.7)$$

Agora é possível definir dois mapas M_A e M_B , tais que:

$$M_A(\phi, \theta) = M_o(\phi, \theta) \cdot \xi_A(\phi, \theta),$$

$$M_B(\phi, \theta) = M_o(\phi, \theta) \cdot \xi_B(\phi, \theta),$$

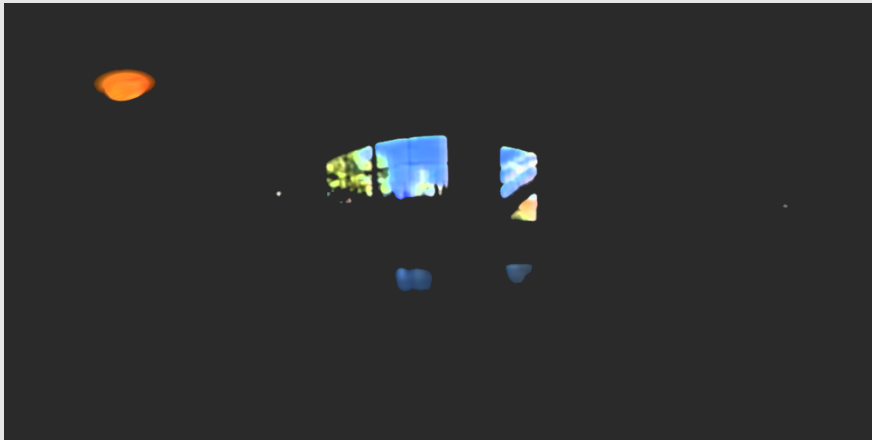
em função da imagem digital do mapa de iluminação M_o e do valor r , obtendo uma forma de fácil implementação computacional:

$$M_A(x, y) = \begin{cases} M_o(x, y) & \text{se } \exists 0 \leq k \leq r : \overline{M}[k, 2] = x + W \cdot y, \\ 0 & \text{se } \nexists 0 \leq k \leq r : \overline{M}[k, 2] = x + W \cdot y, \end{cases}$$

$$M_B(x, y) = \begin{cases} M_o(x, y) & \text{se } \exists r < k < W \times H : \overline{M}[k, 2] = x + W \cdot y, \\ 0 & \text{se } \nexists r < k < W \times H : \overline{M}[k, 2] = x + W \cdot y, \end{cases}$$

Em geral, quando dispomos de um mapa de iluminação proveniente de um ambiente real, a área da região A para um valor de $p = 0.5$ representa menos que o 5% da área total do mapa (por exemplo, no caso do Mapa *Kitchen* a área da região A para $p = 0.6371$ é aproximadamente 2.8% da área do mapa, figura 7.5a). Esta acumulação de luminância numa pequena área pode ser aproveitada tirando amostras da região (A) e convertendo-as num conjunto Ω de luzes direcionais, figura 7.5c. Como a área é pequena, podemos representar bem a região com poucas amostras, i.e., poucas luzes direcionais. A região complementar (B), cuja área representa a maior parte do mapa de iluminação, pode ser tratada de duas formas:

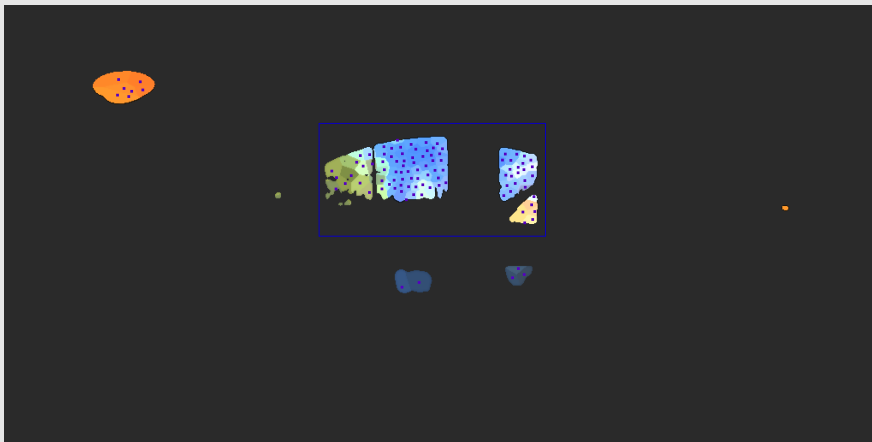
- Guardar o estrato B como um mapa HDR M_B , onde os pixels do estrato A têm radiância nula (figura 7.5b). Neste caso a renderização da cena será feita utilizando o plugin *InfiniteSample (IS)* do *PBRT* [PH04b] com o mapa M_B , e a renderização do conjunto Ω de luzes direcionais obtidas a partir da região A .
- Amostrar o estrato B com o método de amostragem hierárquica por importância com mosaicos de Penrose [ODJ04] ou de Corte Mediano [DEB05], e guardar um mapa de pré-amostragem M_{PA} , figura 7.5d. Aqui a renderização final será feita com o plugin *PRESAMP*, criado para o *PBRT* [PH04b], para renderizar a iluminação dada pelo mapa M_{PA} , juntamente com a renderização do conjunto Ω de luzes direcionais.



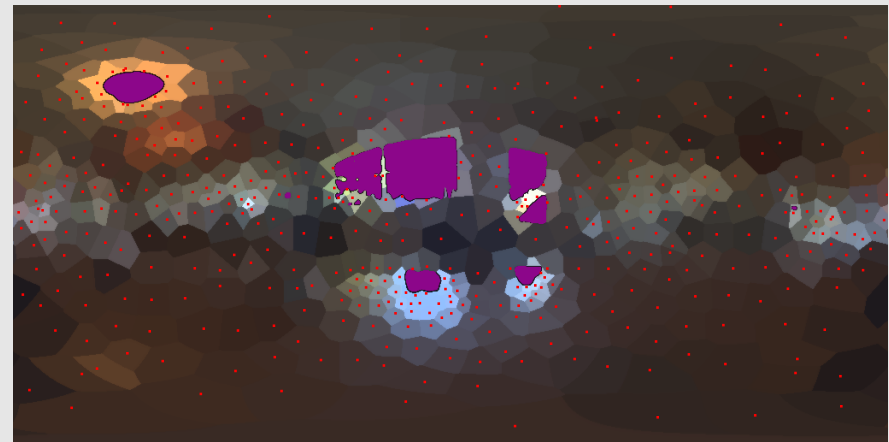
(a) Estrato A , com limiar $p = 0.6371$, área 14730 pixels.



(b) Estrato $B = \text{Mapa} - A$, área 509558 pixels.



(c) amostragem hierárquica por importância sobre o estrato A .



(d) Mapa de pré-amostragem do estrato B .

Figura 7.5: (a) e (b) Estratificação do Mapa HDR *Kitchen* nos estratos A e B utilizando o limiar de variação máxima $p = 0.6371$. (c) Geração de 92 luzes com amostragem hierárquica por importância (Penrose Tiling) [ODJ04] do estrato A utilizando clusterização com frequência $F_i = 1$, (d) Geração de um mapa de pré-amostragem a partir do estrato B com 382 amostras.

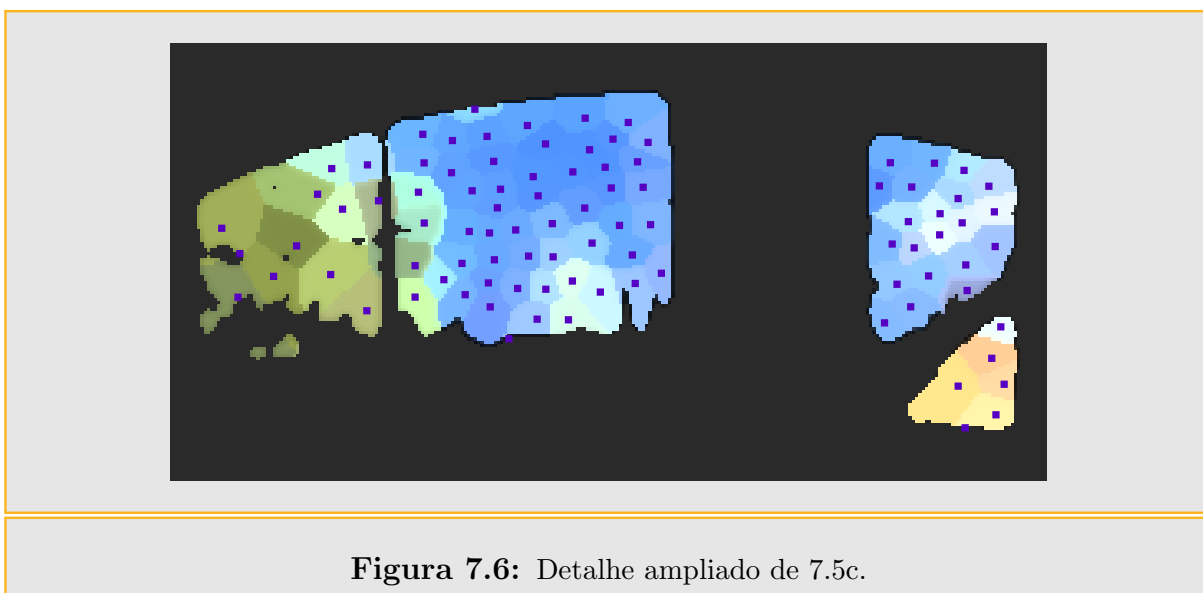


Figura 7.6: Detalhe ampliado de 7.5c.

7.4.2 Estratégias para determinação dos estratos e limiar de separação p

Um aspecto crucial na abordagem híbrida é a escolha certa do limiar p utilizado para realizar a separação do mapa de iluminação, devido a que o valor de p está relacionado diretamente com a área do mapa destinada à criação de luzes direcionais e a área a ser utilizada para amostragem direta ou pré-amostragem. Uma boa escolha de p ajudará a reduzir o número de luzes direcionais utilizadas e a maximizar a iluminação total das mesmas. Inicialmente trabalhamos com uma escolha manual do limiar p , mas recentemente temos estudado alternativas para realizar estimativas do limiar p , para dar ao usuário uma aproximação prévia junto com a possibilidade de ajustar este valor. Além da escolha direta do limiar p por parte do usuário, apresentamos a continuação duas formas rápidas e de grande ajuda para calcular uma estimativa inicial para o limiar p :

7.4.2.1 Estimativa por quantidade de luzes desejadas e ângulo de separação mínima

Dado um valor angular α de separação mínima desejada entre as luzes direcionais e um valor inteiro N para a quantidade de luzes desejadas pelo usuário, é possível determinar os estratos A e B e obter conjuntamente o valor do limiar p . Considerando que a área de uma calota esférica é:

$$Area_{calota} = 2\pi \cdot R \cdot h \quad (7.8)$$

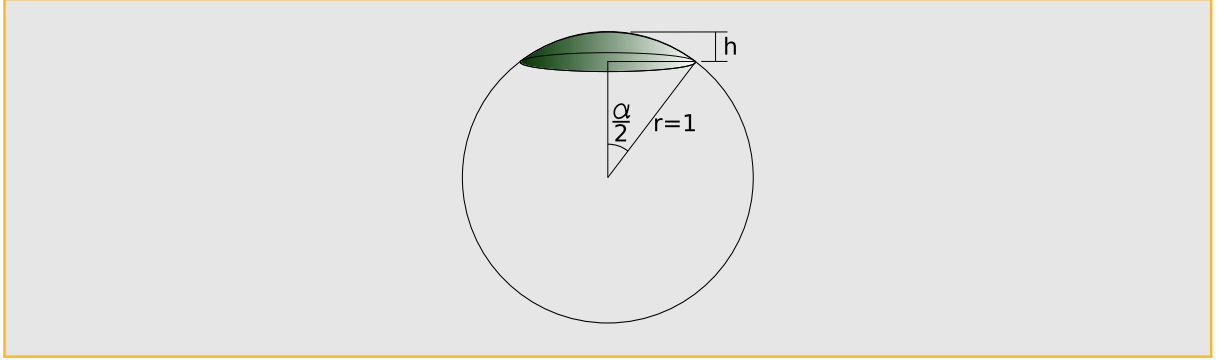


Figura 7.7: Calota esférica.

onde R é o raio da esfera e h é a altura da calota (figura 7.7).

Considerando $R = 1$, por estarmos trabalhando na esfera unitária, podemos pensar que a zona de influência de uma luz é dada por uma calota esférica com centro na luz e abertura angular igual a $\alpha/2$. Com estas considerações temos que a área de influência de uma luz k é

$$Area_{luz}(k) = 2\pi \cdot h = 2\pi \cdot (1 - \cos(\alpha/2)) \quad (7.9)$$

Logo, se temos N luzes a área total mínima ocupada pelas luzes sobre a esfera unitária é:

$$Area_{Luzes} = N \cdot 2\pi \cdot (1 - \cos(\alpha/2)) \quad (7.10)$$

Levando em conta que dado um mapa de iluminação M_o , o trabalho computacional é feito sobre uma imagem I de $W \times H$ pixels, portanto a relação de áreas entre o formato latitude-longitude e o mapeamento esférico é a seguinte:

$$\frac{Area_{plana}(I)}{Area_{plana}(M_o)} = \frac{Area_{esferica}(I)}{Area_{esferica}(M_o)}, \quad (7.11)$$

ou equivalentemente

$$\frac{W \cdot H}{2\pi\pi} = \frac{Area_{esferica}(M_o)}{4\pi}, \quad (7.12)$$

obtendo-se desta forma:

$$Area_{esferica}(M_o) = \frac{W \cdot H}{2\pi\pi} \cdot 4\pi. \quad (7.13)$$

Utilizando o fator anterior, obtemos a estimativa para a área do estrato A sobre a imagem I depois do mapeamento esférico

$$Area_{esferica}(A) = \frac{W \cdot H}{2\pi\pi} \cdot N \cdot 2\pi \cdot (1 - \cos(\alpha/2)). \quad (7.14)$$

Agora o trabalho de calcular o estrato A se reduz a achar o menor inteiro r tal que:

$$\frac{W \cdot H}{\pi} \cdot N \cdot (1 - \cos(\alpha/2)) \leq \sum_{i=0}^r \sin\left(\frac{\lfloor \bar{M}[i,2]/W \rfloor + 0.5}{H} \cdot \pi\right). \quad (7.15)$$

O valor do limiar p pode ser aproximado como a relação entre a luminância do estrato A e a luminância total do mapa:

$$p \approx \frac{\sum_{i=0}^r \bar{M}[i,1] \cdot \sin\left(\pi \frac{\lfloor \bar{M}[i,2]/W \rfloor + 0.5}{H}\right)}{\sum_{i=0}^{W \cdot H - 1} L_{M_o}(\text{mod}(i, W), \lfloor i/W \rfloor) \sin\left(\pi \frac{\lfloor i/W \rfloor + 0.5}{H}\right)}. \quad (7.16)$$

Agora o usuário pode ajustar este valor de p manualmente para obter a distribuição desejada. Devemos esclarecer também que mesmo desejando N luzes direcionais, o método de amostragem escolhido pode introduzir no estrato A $M > N$ amostras. Neste caso o sistema realizará uma clusterização para eliminar $M - N$ amostras antes de proceder ao cálculo da luzes direcionais finais.

7.4.2.2 Estimativa por taxa de variação máxima

Neste caso o valor do limiar de separação p é determinado a partir do valor mínimo sobre uma função.

Primeiramente consideremos a função constante por partes obtida a partir do array \bar{M}

$$f(x) = \begin{cases} \bar{M}[0,1] & : 0 \leq x \leq \sin\left(\frac{\lfloor \bar{M}[0,2]/W \rfloor + 0.5}{H} \cdot \pi\right) \\ \bar{M}[i,1] & : x_{i-1} < x \leq x_i \quad i = 1, \dots, W \cdot H - 1 \end{cases} \quad (7.17)$$

onde:

$$x_i = \sum_{j=0}^i \sin\left(\frac{\lfloor \bar{M}[j,2]/W \rfloor + 0.5}{H} \cdot \pi\right). \quad (7.18)$$

Seja $\psi = W \cdot \sum_{i=0}^{W \cdot H - 1} \sin\left(\frac{i+0.5}{H} \pi\right)$, a função $f : [0, \psi] \rightarrow \mathbb{R}$ está definida para $0 \leq x \leq \psi$. Definimos agora a função integral linear por partes:

$$g : [0, \psi] \rightarrow \mathbb{R},$$

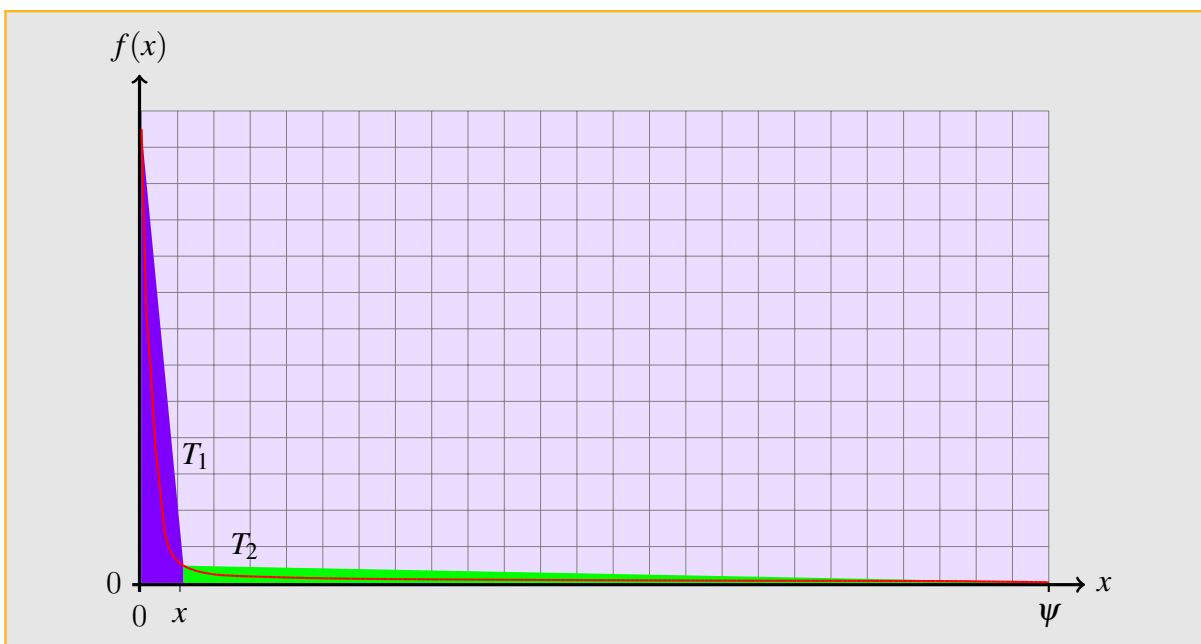


Figura 7.8: A função $\zeta(x) = T_1 + T_2$, onde T_1 é a área do trapézio lilaz e T_2 a área do trapézio verde.

$$g(x) = \int_0^x f(x) dx.$$

Definimos a seguir a função

$$\zeta(x) = \frac{f(0) + f(x)}{2} \cdot x + \frac{f(x) + f(\psi)}{2} (\psi - x).$$

A idéia de separar um mapa de iluminação em estratos A e B foi concebida para determinar A com o mínimo de área e máxima iluminação possível, e obter um conjunto mínimo de luzes direcionais importantes. Minimizar a área de A é equivalente a maximizar a área de B ao tempo que tentamos maximizar a iluminação de A . A função $\zeta(x)$ representa uma aproximação da integral da luminância do mapa de iluminação por dois trapézios, como na figura 7.8. Neste esquema, a melhor relação entre área e iluminação para o estrato $A(x)$ é obtido com o \bar{x} , que minimiza a função $\zeta(x)$. O cálculo do valor \bar{x} é simples de ser realizado a nível de implementação, pois a função ζ é avaliada numa quantidade finita de pontos x_i , para $i = 0, \dots, W \cdot H - 1$, associados aos pixels da imagem que representa o mapa de iluminação M_o .

Calculamos o conjunto de mínimos globais

$$\Psi = \{\bar{x}_j : \zeta(\bar{x}_j) \leq \zeta(x_i), \forall i, j = 0, \dots, W \cdot H - 1, i \neq j\}, \quad (7.19)$$

onde os x_k 's são definidos pela equação 7.18. Claramente o conjunto $\Psi \neq \emptyset$, e no caso de termos $\#(\Psi) > 1$ tomamos $\bar{x} = \min_x \Psi$, dado que $Area_{esf.}(A(x)) = x$ e principalmente pretendemos minimizar a área de $A(x)$.

Uma vez determinado o valor máximo \bar{x} , o valor da área do estrato A será o próprio \bar{x} . Agora o cálculo do valor do limiar de separação de luminância p decorre do cálculo do mínimo inteiro r tal que:

$$\bar{x} \leq x_r = \sum_{j=0}^r \sin\left(\frac{\lfloor \bar{M}[j, 2] \rfloor / W + 0.5}{H} \cdot \pi\right). \quad (7.20)$$

Calculado r , obtemos p como sendo

$$p = \frac{\sum_{i=0}^r \bar{M}[i, 1] \cdot \sin\left(\pi \frac{\lfloor \bar{M}[i, 2] \rfloor / W + 0.5}{H}\right)}{\sum_{i=0}^{W \cdot H - 1} L_{M_o}(\text{mod}(i, W), \lfloor i/W \rfloor) \sin\left(\pi \frac{\lfloor i/W \rfloor + 0.5}{H}\right)}. \quad (7.21)$$

Na continuação vemos algumas imagens do processo de separação pela taxa máxima de variação para um mapa de iluminação.

Tabela 7.1: Mapas de iluminação utilizados. As imagens dos mapas utilizados nas renderizações podem ser encontrados no Apêndice IV.

Mapa	<i>Galileo</i>	<i>Grace</i>	<i>Rnl</i>	<i>Kitchen</i>
Area Lat-Long	1024×512	1024×512	1024×512	1024×512
Area esférica	333772.72	333772.72	333772.72	333772.72
Energia *	384651.69	628952.81	393835.03	158041.83
Mínimo $\zeta(x)$	1882379	8904683	2088810.875	573606.37
Limiar p	0.53997	0.515	0.46	0.6371
Area Lat-Long A	3293	7173	20297	14730
Area Esférica A	2883.89	4640.10	12383.28	12885.93
Area Lat-Long B	520995	517115	503991	509558
Area Esférica B	330888.83	329132.62	321389.44	320886.79
Energia A	207699.64	323885.37	181148.23	100694.83
Energia B	176952.05	305067.44	212686.8	57347
% Area Esférica A	0.864%	1.39%	3.71%	3.86%
% Area Esférica B	99.136%	98.61%	96.29%	96.14%
% Energia A	53.997%	51.5%	46%	63.71%
% Energia B	43.003%	48.5%	54%	36.29%

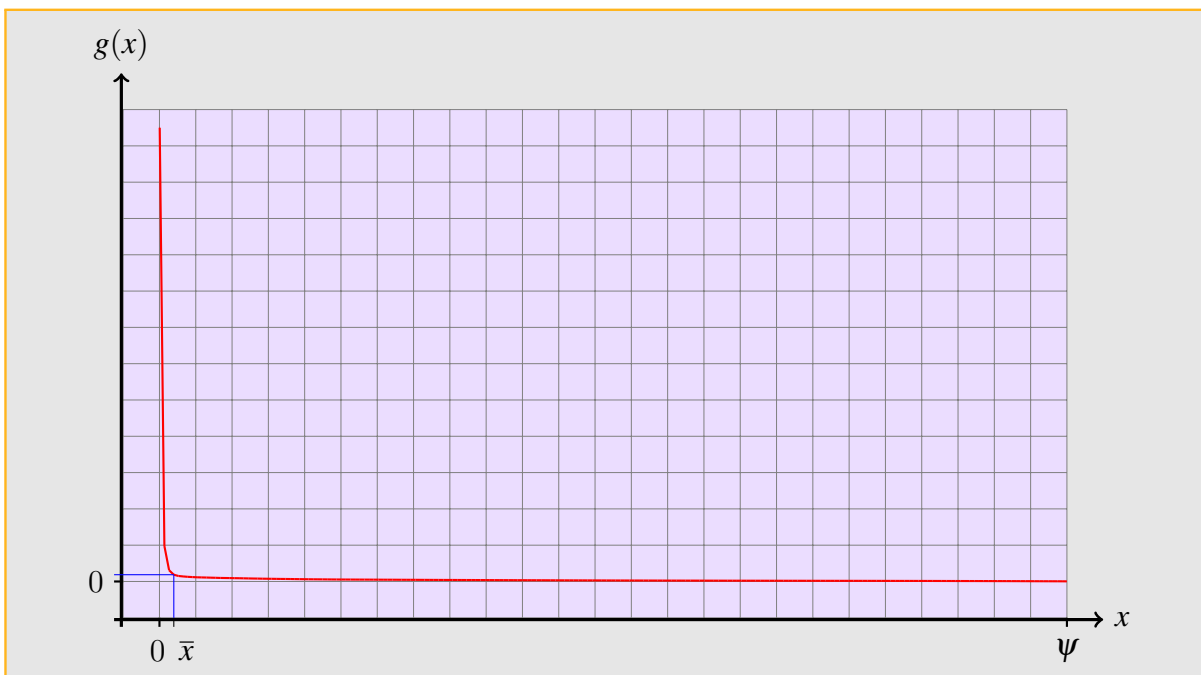


Figura 7.9: Função monôtona não-crescente de luminância do mapa *Galileo*, $f(x)$.

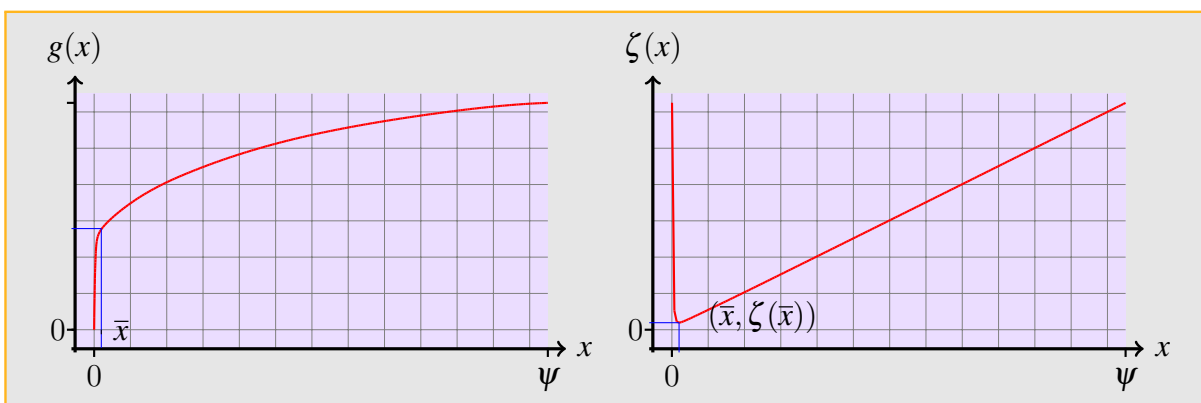


Figura 7.10: Esquerda: Integral $g(x)$ da função de luminância da figura 7.9. Direita: Função $\zeta(x)$ para o mapa *Galileo*.

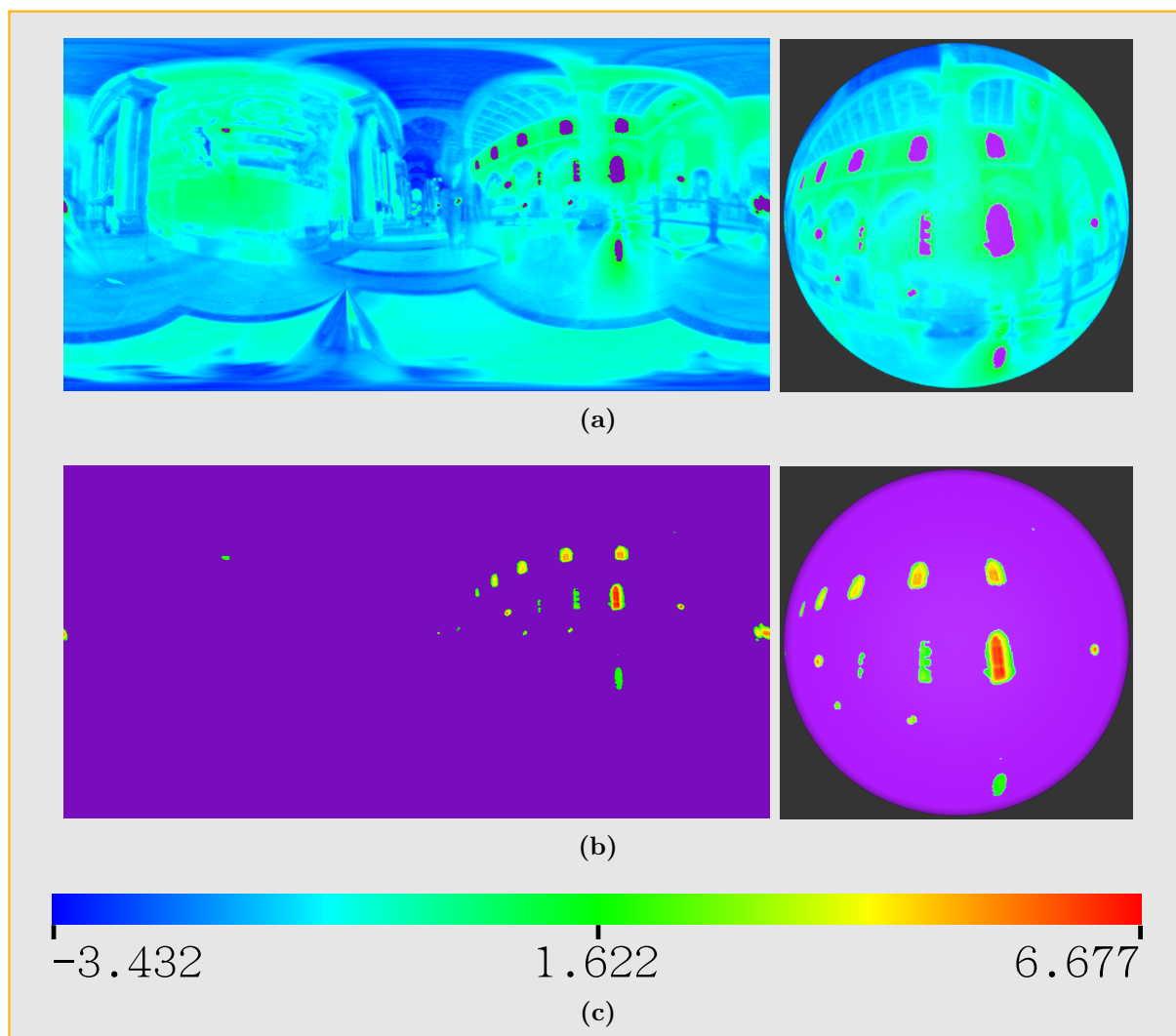


Figura 7.11: (a) Estrato A do mapa *Galileo* com limiar $p = 0.53997$, (b) Estrato B do mapa *Galileo* com limiar $p = 0.53997$, (c) Escala logarítmica de cores, $\ln(L_{Mo}(u, v) \cdot \sin(v/H))$, utilizada para representar a função de importância $L_{Mo}(u, v) \cdot \sin(v/H)$ do mapa *Galileo*.

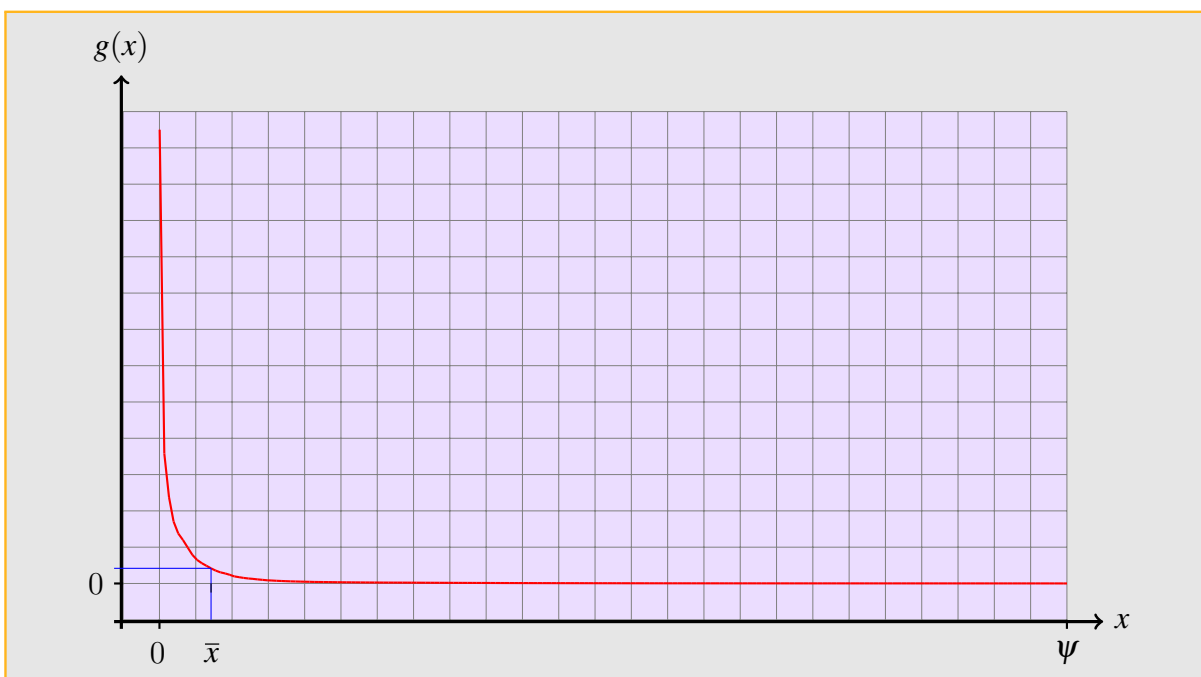


Figura 7.12: Função monôtona não-crescente de luminância do mapa *Grace*, $f(x)$.

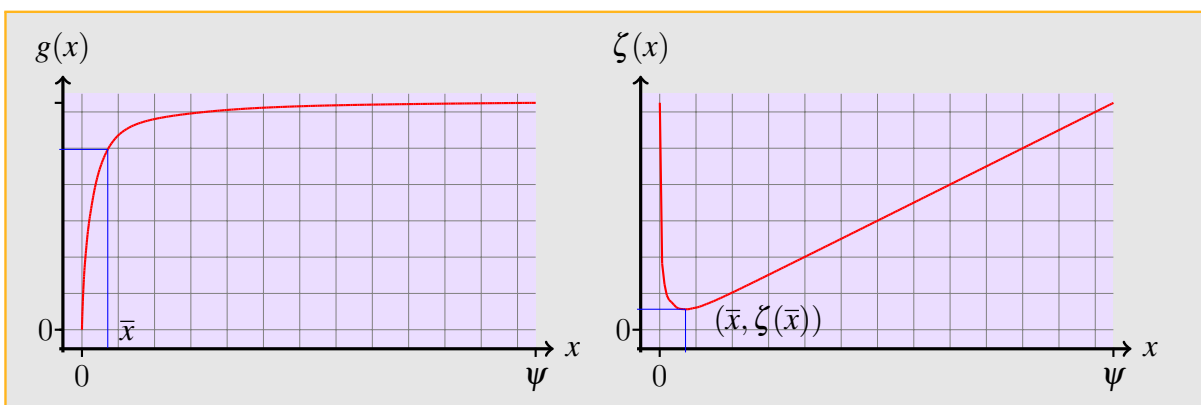


Figura 7.13: **Esquerda:** Integral $g(x)$ da função de luminância da figura 7.12. **Direita:** Função $\zeta(x)$ para o mapa *Grace*.

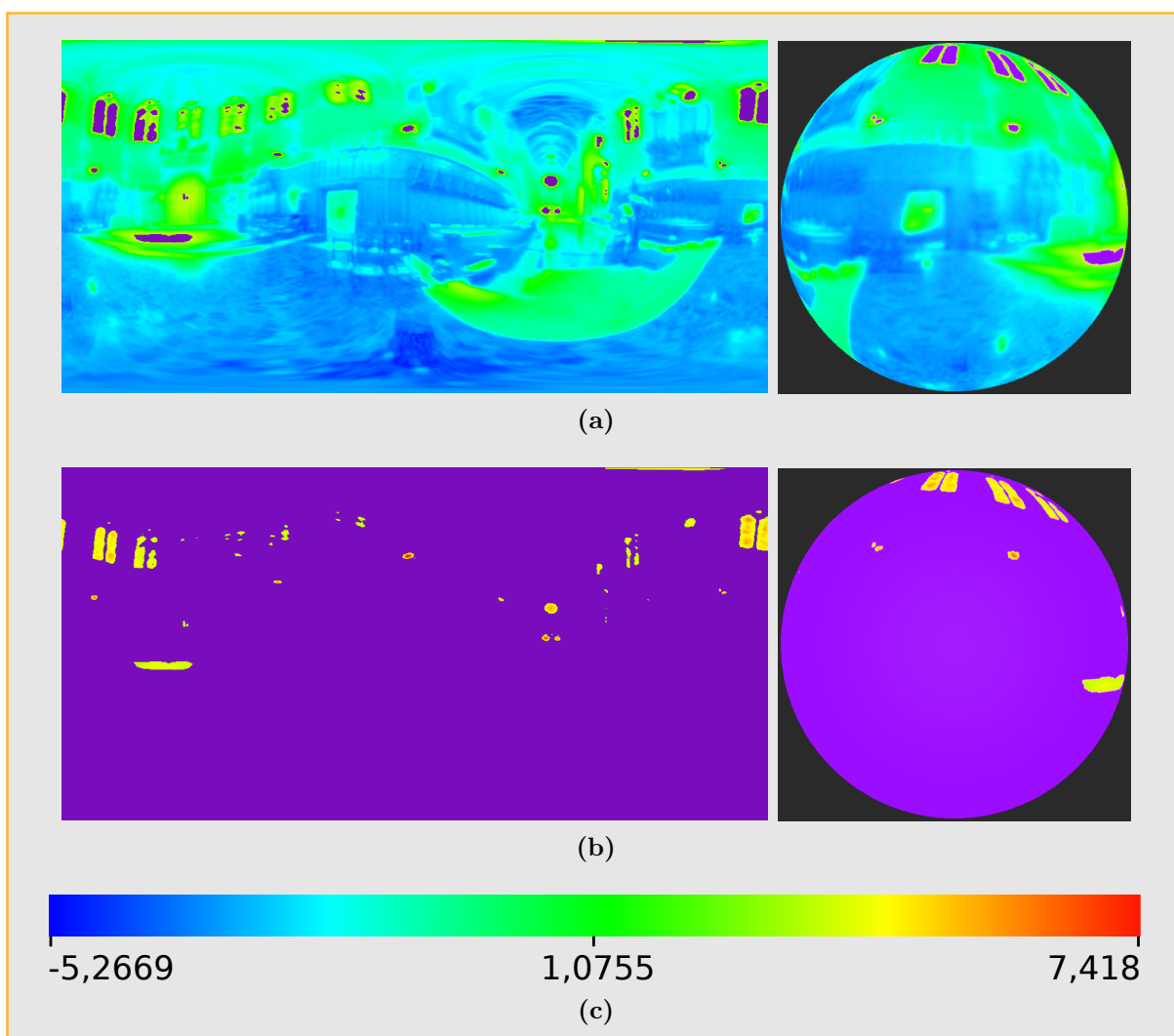


Figura 7.14: (a) Estrato A do mapa *Grace* com limiar $p = 0.515$, (b) Estrato B do mapa *Grace* com limiar $p = 0.515$, (c) Escala logarítmica de cores, $\ln(L_{Mo}(u, v) \cdot \sin(v/H))$, utilizada para representar a função de importância $L_{Mo}(u, v) \cdot \sin(v/H)$ do mapa *Grace*.

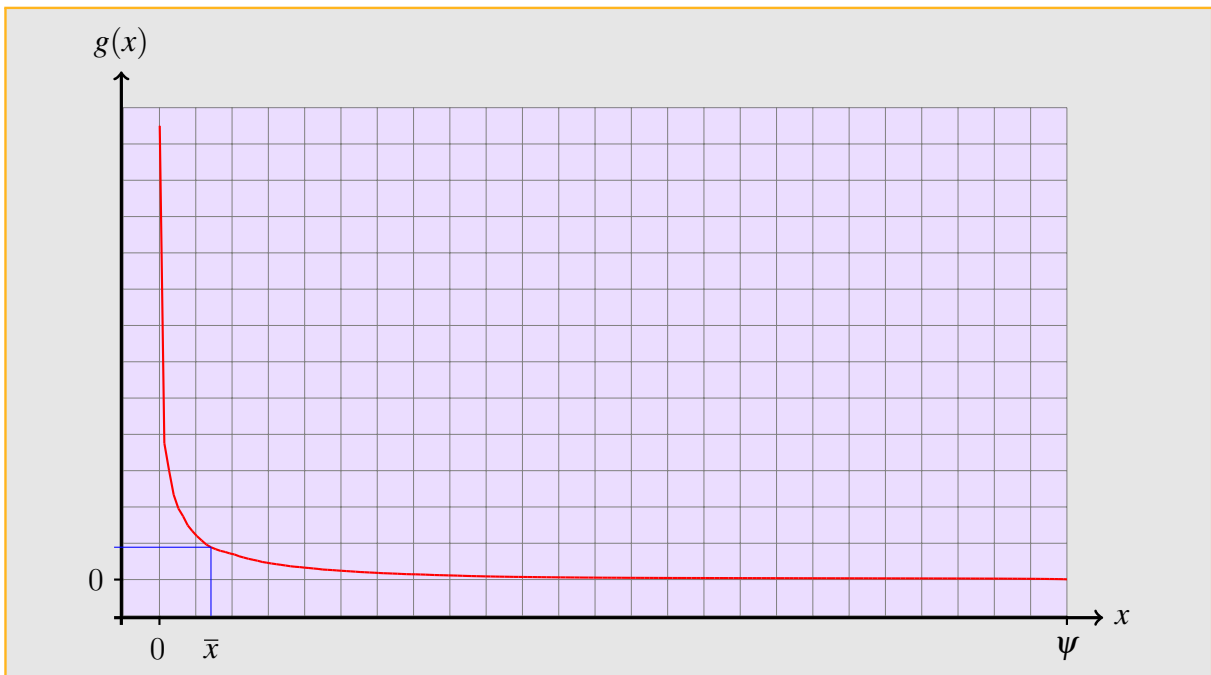


Figura 7.15: Função monótona não-crescente de luminância do mapa Rnl , $f(x)$.

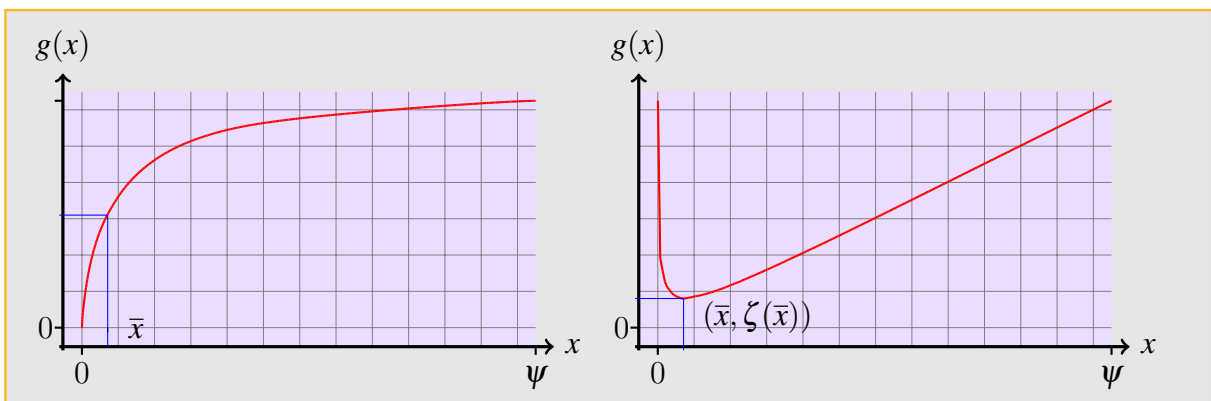


Figura 7.16: Esquerda: Integral $g(x)$ da função de luminância da figura 7.15. Direita: Função $\zeta(x)$ para o mapa Rnl .

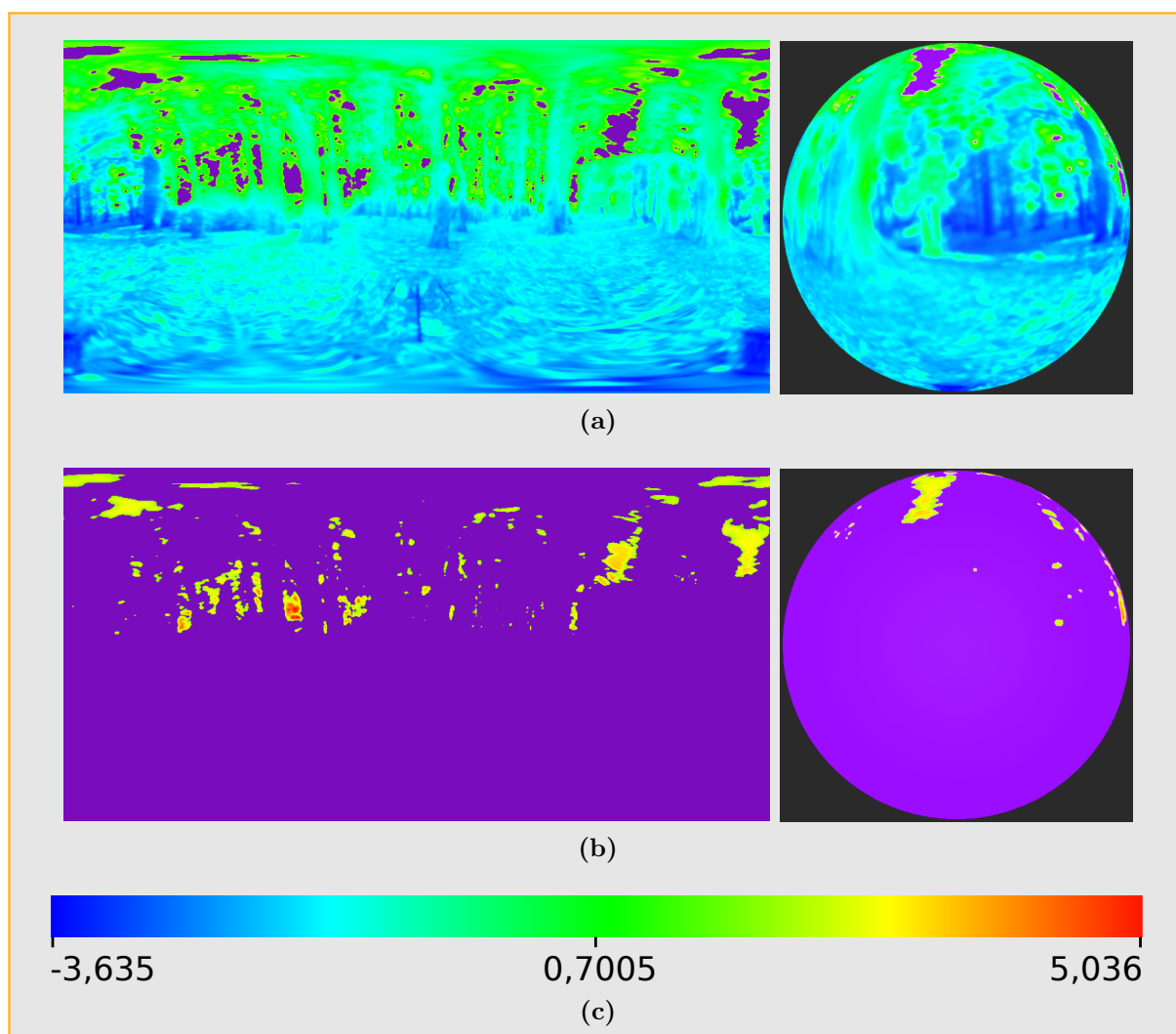


Figura 7.17: (a) Estrato *A* do mapa *Rnl* com limiar $p = 0.46$, (b) Estrato *B* do mapa *Rnl* com limiar $p = 0.46$, (c) Escala logarítmica de cores, $\ln(L_{Mo}(u, v) \cdot \sin(v/H))$, utilizada para representar a função de importância $L_{Mo}(u, v) \cdot \sin(v/H)$ do mapa *Rnl*.

7.4.3 Clusterização de amostras

Juntamente com a estratificação, também tivemos cuidado de introduzir controles adicionais, dado que o método Penrose Tiling de Ostromoukhov [ODJ04] apresenta alguns inconvenientes. Como o método de amostragem hierárquica por importância com mosaicos de Penrose [ODJ04] aplicado a um mapa de iluminação utiliza uma região retangular contínua do plano com as dimensões do mapa, e função de importância constante por partes igual a um múltiplo escalar da luminância do mapa, tem-se frequentemente uma alta condensação de amostras em regiões de área muito pequena. Por exemplo, muitas vezes temos duas ou mais amostras numa área que corresponde a um pixel do mapa (figura 7.18). Este fenômeno acontece porque nos mapas extraídos de ambientes reais a variação da luminância é muito grande. Como o objetivo é obter luzes direcionais que iluminem a cena, precisamos de amostras devidamente espalhadas para cobrir bem a região A . Na prática não temos interesse em ter amostras cujas proximidades seja inferior a 1 pixel, pois não influenciaram no resultado final. Por fim, agrupamos estas amostras aplicando um método de clusterização sugerido por Velho et al. [VGS97], eliminando desta forma as amostras redundantes. Implementamos em nosso sistema a clusterização com duas opções de parada e três opções para o valor da frequência ou peso F da amostra definidas abaixo.

Dado um cluster com dois elementos $K = \{c_i, c_j\}$, o nível de quantização ótima utilizando a métrica do quadrado da distância geodésica sobre a esfera é:

$$c = \frac{F_i}{F_i + F_j} c_i + \frac{F_j}{F_i + F_j} c_j,$$

$$E(c_i c_j) = \frac{F_i F_j^2 + F_j F_i^2}{(F_i + F_j)^2} \|\arccos(c_i \cdot c_j)\|^2,$$

onde as opções para F_i são:

7.4.3.1 Frequência $F_i = 1$

Esta opção de frequência leva em conta somente a distância entre as amostras e ignora completamente a luminância das mesmas e a área da sua célula de Voronoi. É recomendada quando se deseja suprimir amostras próximas sem levar em conta sua luminosidade. Pode ser utilizada por exemplo quando desejarmos obter um conjunto de luzes direcio-

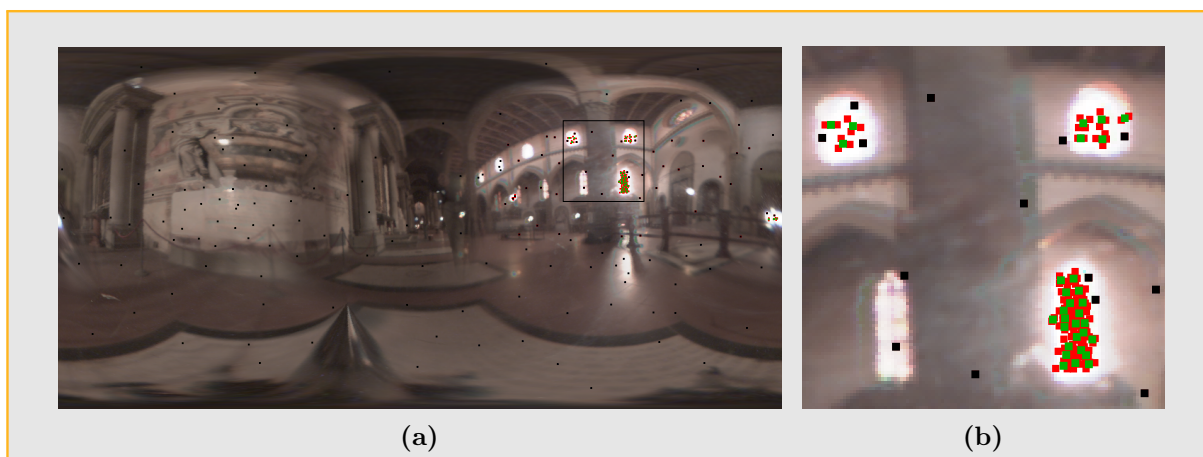


Figura 7.18: (a) Clusterização de 304 amostras hierárquicas por importância do mapa *Galileo* utilizando frequência $F_i = 1$ para obter 200 amostras clusterizadas. As amostras pretas e vermelhas são obtidas pela amostragem hierárquica por importância. Depois da clusterização as amostras pretas permaneceram inalteradas, e as amostras vermelhas dão lugar às novas amostras verdes. Assim o resultado da clusterização é o conjunto de amostras pretas e verdes. (b) Detalhe ampliado da clusterização (a).

nais a partir de um mapa de iluminação com alta concentração de luminância em regiões pequenas pelo método de Ostromoukhov. Dependendo da distribuição de luminância do mapa, é possível que para conseguir uma densidade adequada de amostras nas regiões de pouca luminosidade, obtenha-se redundância de amostras nas regiões mais brilhantes do mapa. Para diminuir a quantidade de amostras eliminando as que estão muito próximas entre si, é conveniente utilizar a frequência $F_i = 1$. Vemos a seguir um exemplo de utilização da clusterização no mapa *Galileo*.

7.4.3.2 Frequência $F_i = \int_{Vor(c_i)} L(x) dx$

Esta frequência é calculada como a luminância da célula de Voronoi da amostra. Pode ser utilizada quando temos interesse em juntar amostras com baixa luminância.

7.4.3.3 Frequência $F_i = \frac{\int_{Vor(c_i)} L(x) dx}{Área \ Vor(c_i)}$

Esta frequência é dada pela luminância média da célula de Voronoi da amostra. Em termos de distribuição probabilística, é a mais adequada, pois leva em conta a distância das amostras como sua importância em função da luminância média.

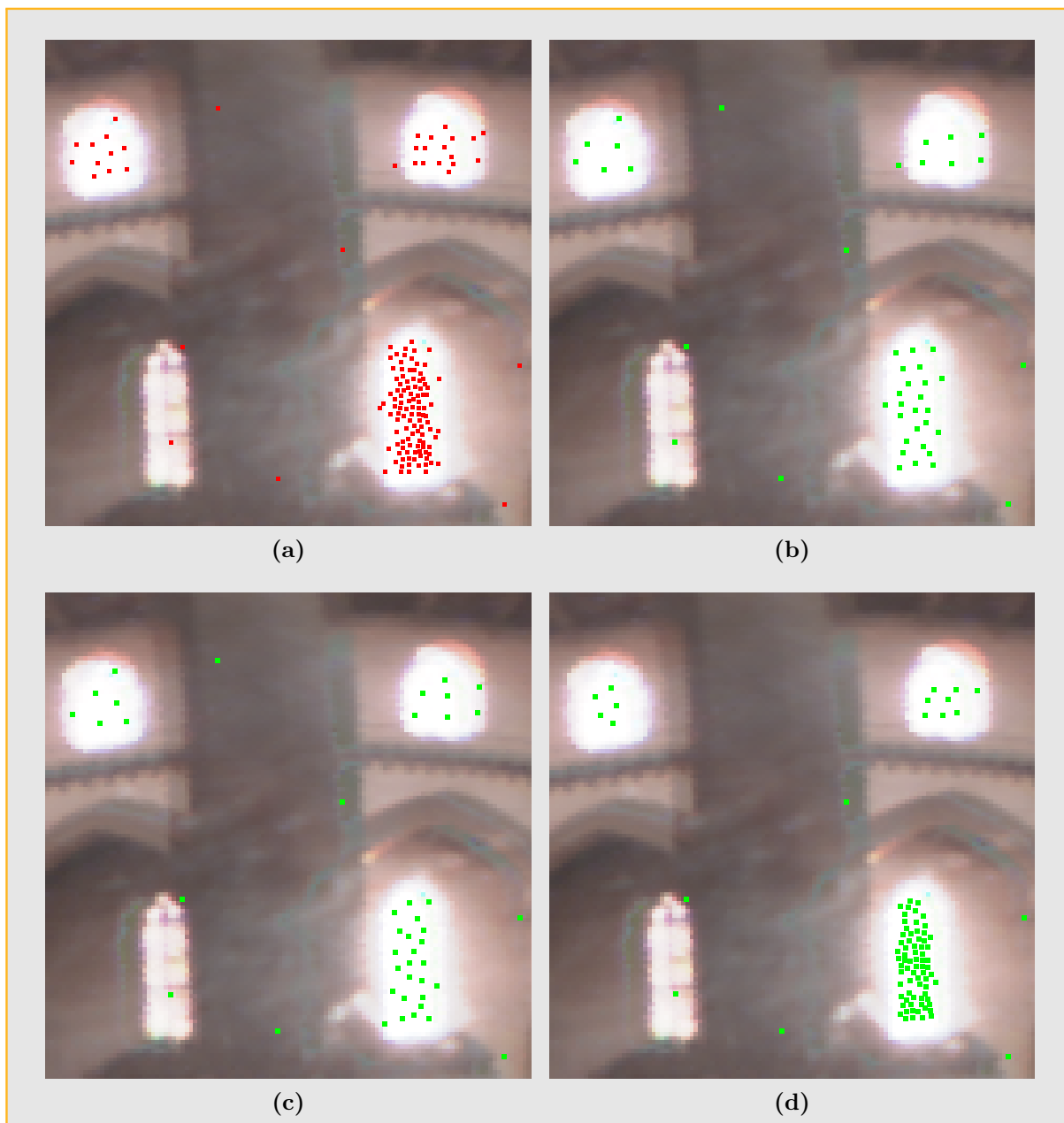


Figura 7.19: (a) Detalhe da amostragem hierárquica por importância da região retangular da figura 7.18a , (b) Clusterização da amostras da figura (a) com $F_i = 1$, (c) Clusterização das amostras da figura (a) com $F_i = \int_{Vor(c_i)} L(x)dx$, (d) Clusterização das amostras da figura (a) com $F_i = \frac{\int_{Vor(c_i)} L(x)dx}{Área \ Vor(c_i)}$.

7.4.3.4 Critérios de parada

Incluimos no sistema desenvolvido dois tipos de parada:

Parada por número de clusters: Dado um conjunto Ω com N amostras, a clusterização é realizada sucessivamente até atingir o valor $M \leq N$ desejado pelo usuário. Como no método de clusterização por pares de pontos utilizado, aqui a quantidade de clusters diminui numa unidade a cada interação, serão realizadas $N - M$ junções.

Parada por separação mínima: Dado um conjunto Ω com N amostras e um limiar α de separação angular mínima, a clusterização é realizada até que a distância angular entre os clusters seja maior ou igual a α , isto é, até termos:

$$\alpha \leq \arccos(c_i \cdot c_j) \quad \forall i \neq j. \quad (7.22)$$

8 *Resultados*

8.1 Medição da convergência dos algoritmos de renderização

Descreveremos agora as métricas de erro que utilizaremos para medir a propriedades de convergência dos algoritmos propostos nesta dissertação.

Como os principais métodos de renderização utilizados na dissertação são principalmente não tendenciosos, adotamos uma métrica objetiva em lugar de uma métrica perceptual como principal ferramenta para validar os resultados.

8.1.1 Diferença de cores

O primeiro passo para definir a diferença entre duas imagens é definir a diferença de cor entre dois pixels. Existem diversas maneiras de calcular a diferença ou distância entre duas cores. Pode-se utilizar uma norma $L1$ com pesos onde os pesos são proporcionais à contribuição da luminância nas diferentes componentes de cor. No espaço de cor RGB , a luminância de uma cor $C = (r, g, b)$ é definida como $|C|_c = 0.212671r + 0.715160g + 0.072169b$. Similarmente, se define a diferença entre duas cores $C_1 = (r_1, g_1, b_1)$ e $C_2 = (r_2, g_2, b_2)$ como

$$|C_2 - C_1|_c = 0.212671|r_2 - r_1| + 0.71516|g_2 - g_1| + 0.072169|b_2 - b_1|,$$

onde $|\cdot|_c$ denota a diferença entre duas cores.

Uma segunda forma válida de definir a distância é a traves da norma Euclidiana no espaço RGB , para a qual a distância entre duas cores é definida como

$$|C_2 - C_1|_e = |r_2 - r_1| + |g_2 - g_1| + |b_2 - b_1|.$$

8.1.2 Diferença de imagens

A métrica de erro padrão para a diferença entre duas imagens é o erro quadrado médio (MSE - mean squared error), também chamado de variância (σ^2). Para duas imagens A e B com as mesmas dimensões, a variância é definida como:

$$\sigma^2 = \frac{1}{N} \sum_{i=1}^N |A_i - B_i|_{\delta}^2,$$

onde N é o número de pixels em cada imagem e $|A_i - B_i|_{\delta}$ é a diferença de cor entre o pixel i da imagem A e o pixel i da imagem B , segundo a norma $|\cdot|_{\delta}$. Note como esta definição é similar à da variância descrita na equação 3.8. Por isto é apropriado chamar a equação anterior medida da variância se uma das imagens é a imagem esperada, e a outra uma aproximação obtida por algum algoritmo de renderização de Monte Carlo.

8.2 Resultados

Desenvolvemos o plugin *FHIS* (Fast Hierarchical Importance Sampling) para o *PBRT* [PH04b] que realiza amostragem por importância utilizando um mapa de pré-amostragem do mapa de iluminação, obtido pelo método de amostragem hierárquica com mosaicos de Penrose [ODJ04]. Com o plugin *FHIS*, mais os plugins do *PBRT* é possível implementar os diversos esquemas de amostragem descritos anteriormente, utilizando-os separadamente ou combinados. Realizamos comparações numéricas utilizando quatro métodos:

- **D.L.:** Aproximação do mapa de iluminação por luzes direcionais, obtidas com algoritmos de Corte Mediano (**MC**) [DEB05] e amostragem hierárquica com mosaicos de Penrose (**OS**) [ODJ04];
- **I.S.:** Amostragem direta por importância utilizando o plugin *IS*, *PBRT* [PH04b];
- **M.P.A.:** Amostragem direta por importância utilizando mapas de pré-amostragem, obtidos com **MC** e **OS**, utilizando o plugin *FHIS*, *PBRT* [PH04b];
- **HIB.:** Método híbrido; separação do mapa original e renderização combinada utilizando luzes direcionais + amostragem por importância direta ou sobre mapas de pré-amostragem.

Referência das siglas utilizadas nesta seção:

- **MC:** Amostragem por Corte Mediano (Median Cut), [DEB05].
- **OS:** Amostragem Hierárquica por importância com mosaicos de Penrose, [ODJ04].
- **DL:** Luzes direcionais. Nas tabelas a coluna **DL** indica a utilização de luzes direcionais.
- **MPA:** Mapa de pré-amostragem. Nas tabelas a coluna **MPA** é utilizada para indicar a utilização de um mapa de pré-amostragem pelo plugin *FHIS*.
- **IS:** Amostragem por importância. Nas tabelas a coluna **IS** indica o número de amostras utilizada pelo plugin *IS*.
- **CI:** Indica a frequência utilizada no cálculo de Clusterização.

Tabela 8.1: Códigos utilizados nas tabelas para representar os parâmetros utilizados na renderização das imagens.

Siglas	Formato	Significado
DL	MET : NUM	MET indica o método utilizado, que pode ser MC ou OS ; NUM indica a quantidade de luzes direcionais utilizadas.
MPA	MET(AM) : NUM	MET indica o método utilizado, que pode ser MC ou OS . (AM) indica o número total de amostras que contém o MPA. NUM indica o número de amostras por raio na renderização.
IS	NUM	NUM indica o número de amostras por raio na renderização.
CI	FREQ	FREQ Frequência utilizada na Clusterização. Pode ser: $\mathbf{F}_1 = 1$, $\mathbf{F}_2 = \int_{Vor(c_i)} L(x)dx$, $\mathbf{F}_3 = \frac{\int_{Vor(c_i)} L(x)dx}{Área \ Vor(c_i)}$.

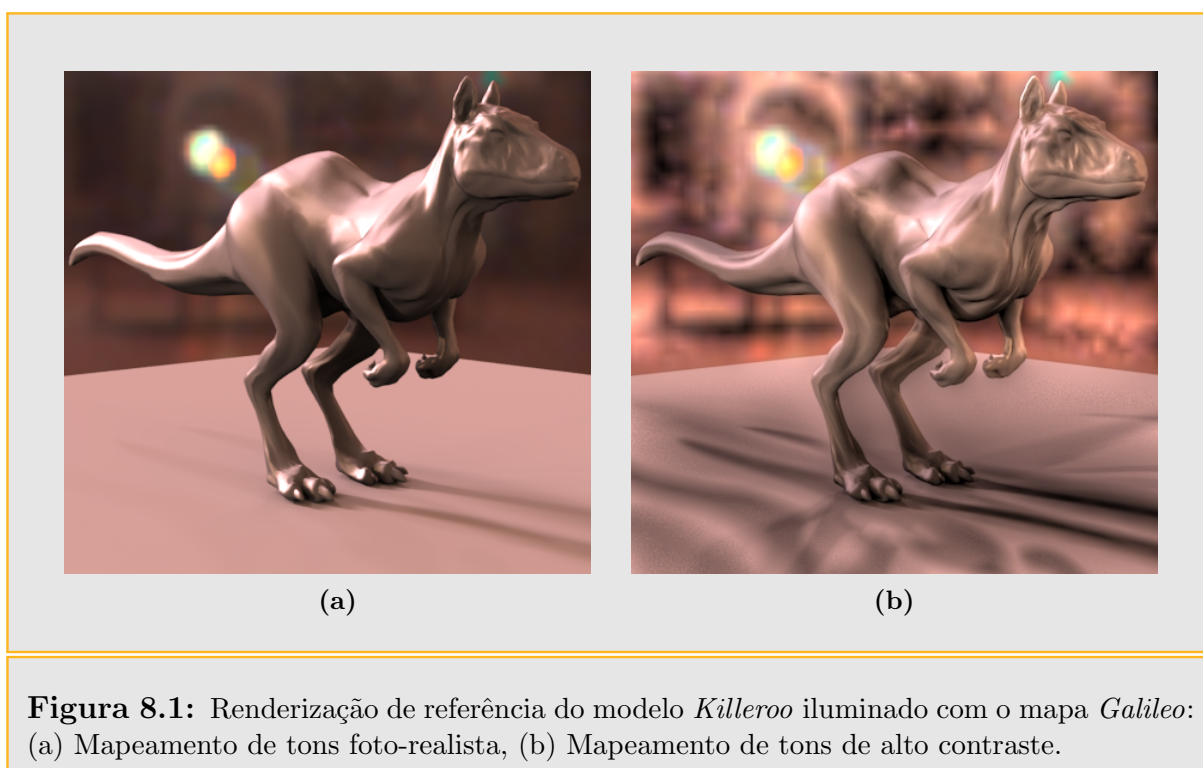
8.2.1 Resultados para Iluminação Direta (Direct Lighting)

Nas comparações realizadas para métodos de iluminação direta utilizamos dois modelos, *Killeroo* e *Buddha*, e três mapas de iluminação, *Galileo*, *Grace* e *Rnl*.

Na tabela 8.2 incluímos os resultados dos testes mais relevantes. Na Figura 8.2 pode-se apreciar a comparação do tempo de renderização e o correspondente erro MSE euclidiano para os principais resultados da tabela 8.2. A fim de poder colocar em evidência os erros e as diferenças entre as renderizações, optamos por utilizar um mapeamento de tons com alto contraste em lugar de um efeito foto-realista, dado que por estarmos trabalhando com imagens HDR e erros relativamente pequenos, os mesmos passam despercebidos com

mapeamentos tonais mais realistas. Uma comparação entre um mapeamento foto-realista e o mapeamento de alto contraste pode ser apreciado na figura 8.1.

Nas figuras 8.4,8.5,8.6 e 8.7 mostramos detalhes de algumas renderizações da tabela 8.2 (lado esquerdo) e o respectivo mapeamento logarítmico do erro MSE com respeito à imagem de referência (lado direito), que permite observar melhor a diferença entre as renderizações. Finalmente, nas figuras 8.8,8.9,8.10 e 8.11 mostramos algumas renderizações do modelo *Killeroo* com mapeamento tonal de alto contraste (lado esquerdo) e o correspondente mapeamento logarítmico do erro MSE com respeito à imagem de referência (lado direito).



8.2.1.1 Interpretação das tabelas de resultados para iluminação direta

A fim de entender a codificação utilizada nas tabelas de resultados para iluminação direta desta seção, mostraremos a continuação a interpretação de três resultados da tabela 8.2. Esta interpretação é válida também para as tabelas 8.3 e 8.4:

- **N=1** : Figura renderizada com o método de amostragem direta por importância (MET=I.S.). O tempo de renderização foi de 81.1 segundos, com erro quadrado médio com métrica euclideana de 0.051807 e erro quadrado médio com a métrica induzida pela luminância de 0.01542. Na renderização foi utilizado o mapa de ilu-

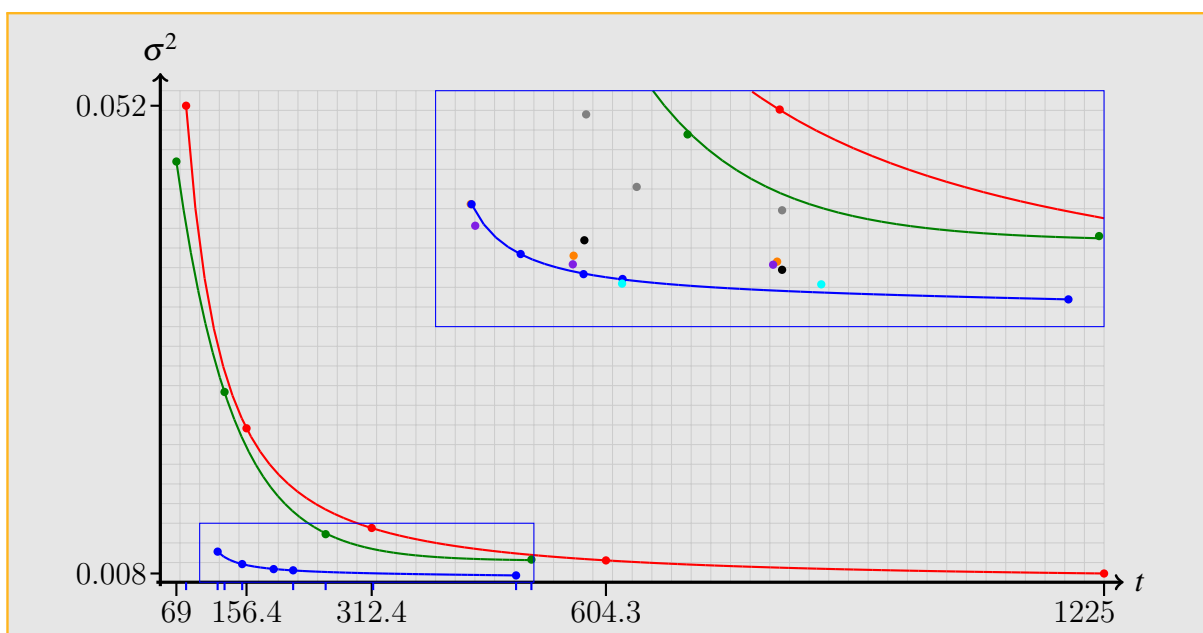


Figura 8.2: Aproximação por ajuste de curvas das funções de erro quadrado médio (MSE) euclidiano para alguns valores da tabela 8.2. ●: Amostragem por importância com plugin *IS*; ●: Amostragem por importância de Mapas de pré-amostragem, plugin *FHIS*; ●: Método híbrido utilizando mapas de pré-amostragem + luzes direcionais. No detalhe ampliado (retângulo azul), os pontos coloridos representam a correspondente coordenada (tempo, σ^2) da tabela 8.2.

minação completo e trabalhou-se com 16 amostras por pixel na amostragem direta por importância (MAPA B / IS=16).

- **N=6** : Figura renderizada com o método de amostragem direta por importância do mapa de pré-amostragem (MET=M.P.A.). O tempo de renderização foi de 69 segundos, com erro quadrado médio com métrica euclideana de 0.046599 e erro quadrado médio com a métrica induzida pela luminância de 0.01399. O mapa de iluminação foi transformado num mapa de pré-amostragem com 1310 amostras obtidas pelo método de amostragem hierárquica por importância com mosaicos de Penrose (OS) e na renderização foram utilizados 16 amostras por pixel tomados com amostragem direta por importância a partir do mapa de pré-amostragem (OS(1310):16).
- **N=16** : Figura renderizada com o método híbrido. O tempo de renderização foi de 191.8 segundos, com erro quadrado médio com métrica euclideana de 0.012291 e erro quadrado médio com a métrica induzida pela luminância de 0.00371. O mapa de iluminação foi dividido em dois mapas: Mapa A com 0.331% da área mais brilhante do mapa original, Mapa B com os 99.669% restantes. O Mapa A foi transformado num conjunto de 48 luzes direcionais utilizando amostragem hierárquica por impor-

tância com mosaicos de Penrose (OS) e clusterização com frequência F_3 . O Mapa B foi transformado num mapa de pré-amostragem com 569 amostras obtidas pelo método de amostragem hierárquica por importância com mosaicos de Penrose (OS). A renderização foi realizada utilizando as 48 luzes obtidas do Mapa A mais 32 amostras amostras por pixel tomados com amostragem direta por importância a partir do mapa de pré-amostragem B.

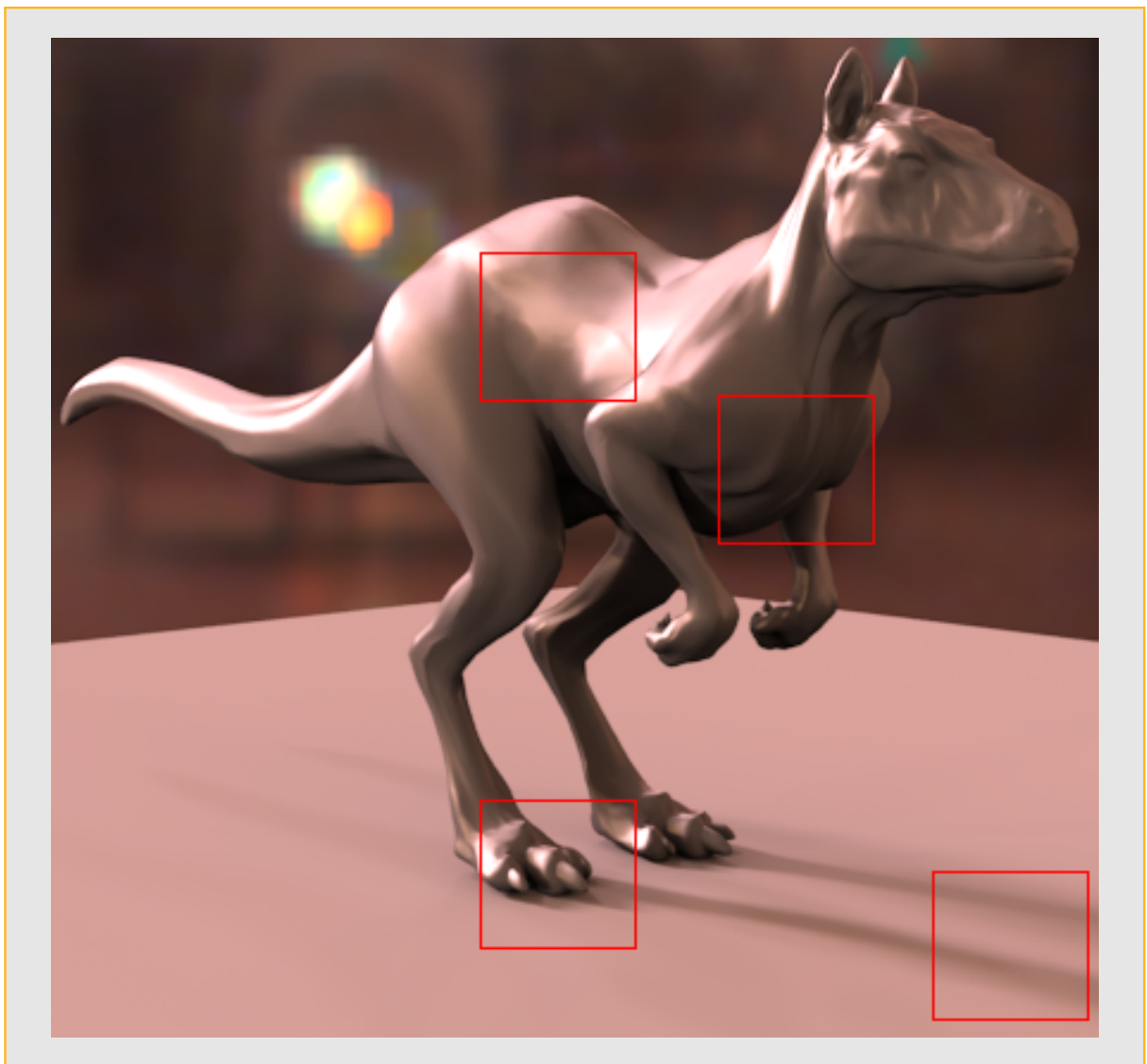
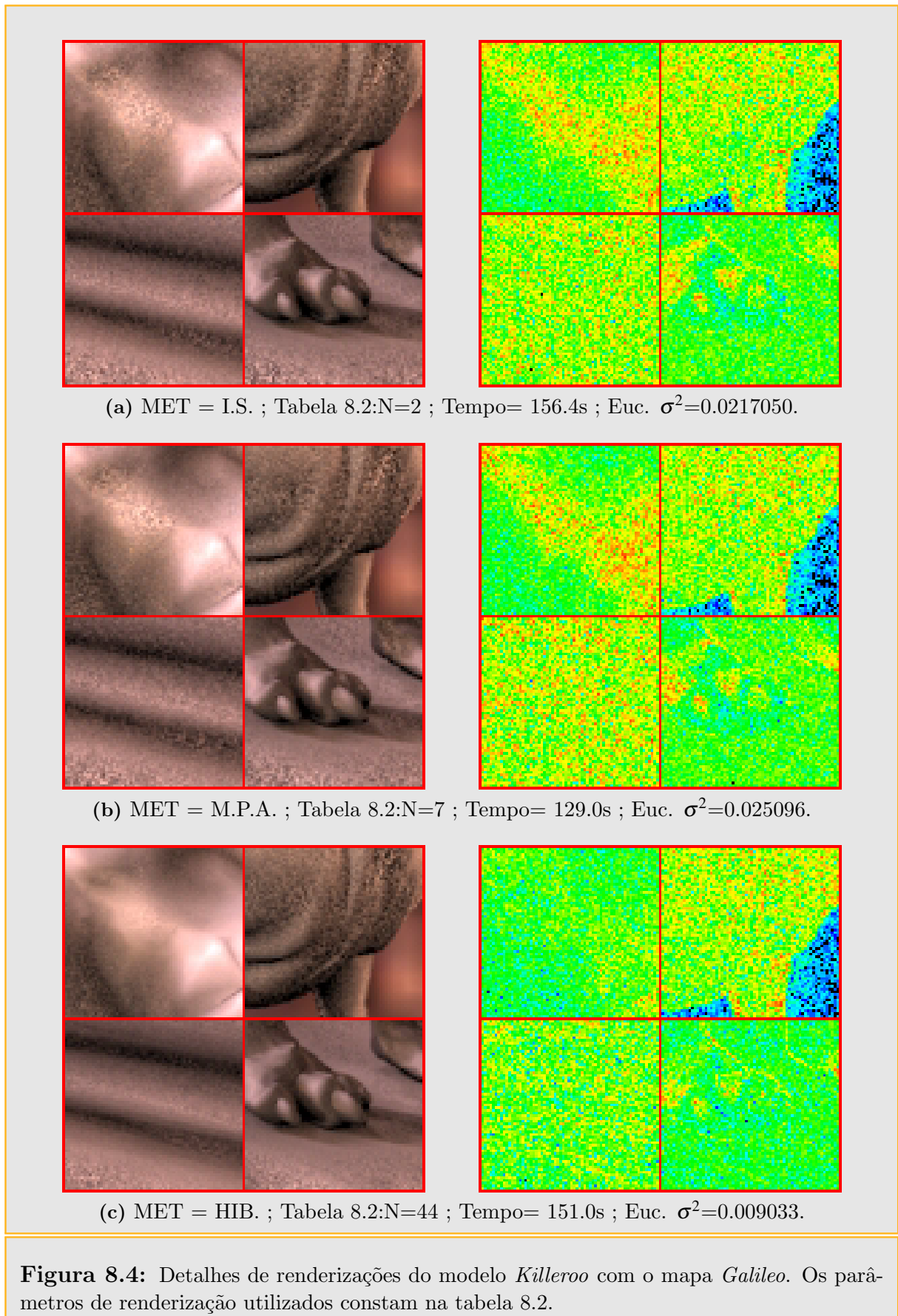


Figura 8.3: Renderização de referência do modelo *Killeroo* iluminado com o mapa *Galileo*, utilizando 32768 (64x512) amostras por pixel. Os retângulos vermelhos delimitam as regiões utilizadas nas figuras 8.4, 8.5, 8.6 e 8.7 para comparar alguns resultados da tabela 8.2.

Tabela 8.2: Comparação de renderizações do modelo *Killeroo* com o mapa *Galileo*. A Imagem de referência utilizada para o cálculo do erro foi renderizada com *IS* no PBRT [PH04b], com 32768 (64x512) amostras (Figura 8.3). (AMD64 +3000, 2Gb RAM).

N	Fig.	MET	Tempo Seg.	Euc. MSE σ^2	Lum. MSE σ^2	% Área A	Mapa A		Mapa B	
							Cl	DL	IS	MPA
1	●	I.S.	81.1	0.051807	0.01542	0			16	
2	8.4a●	I.S.	156.4	0.021705	0.00654	0			32	
3	8.5a●	I.S.	312.4	0.012404	0.00373	0			64	
4	8.6a●	I.S.	604.3	0.009375	0.00284	0			128	
5	8.6c●	I.S.	1225.0	0.008154	0.00247	0			256	
6	●	MPA	69.0	0.046599	0.01399	0				OS(1310):16
7	8.4b●	MPA	129.0	0.025096	0.00762	0				OS(1310):32
8	8.5b●	MPA	255.0	0.011824	0.00360	0				OS(1310):64
9	●	MPA	511.4	0.009452	0.00289	0				OS(1310):128
10	●	HIB.	223.3	0.010599	0.07155	0.331	F_1	OS:48	32	
11	●	HIB.	191.8	0.012291	0.00371	0.331	F_1	OS:48		IS/OS(569):32
12	●	HIB.	313.9	0.010054	0.00305	0.331	F_1	OS:48		IS/OS(569):64
13	●	HIB.	190.7	0.009354	0.00284	0.331	F_1	OS:48		OS(569):32
14	●	HIB.	313.9	0.008665	0.00264	0.331	F_1	OS:48		OS(569):64
15		HIB.	128.8	0.010747	0.00329	0.864	F_2	OS:48		OS(519):16
16	8.5c●	HIB.	190.2	0.008564	0.00262	0.864	F_2	OS:48		OS(519):32
17		HIB.	314.2	0.008591	0.00262	0.864	F_2	OS:48		OS(519):64
18		HIB.	128.9	0.011434	0.00350	0.864	F_3	OS:48		OS(519):16
19	●	HIB.	119.9	0.010198	0.00312	0.864	F_1	OS:40		OS(512):16
20	●	HIB.	184.0	0.008994	0.00275	0.864	F_1	OS:40		OS(512):32
21	●	HIB.	310.8	0.008856	0.00271	0.864	F_1	OS:40		OS(512):64
22	●	HIB.	120.4	0.010195	0.00312	0.864	F_2	OS:40		OS(512):16
23		HIB.	185.4	0.008919	0.00273	0.864	F_2	OS:40		OS(512):32
24		HIB.	309.3	0.008788	0.00268	0.864	F_2	OS:40		OS(512):64
25		HIB.	133.1	0.010961	0.00334	0.864	F_1	OS:50		OS(512):16
26		HIB.	196.1	0.008761	0.00268	0.864	F_1	OS:50		OS(512):32
27		HIB.	328.0	0.009141	0.00279	0.864	F_1	OS:50		OS(512):64
28		HIB.	197.8	0.008869	0.00271	0.864	F_2	OS:50		OS(512):32
29		HIB.	327.0	0.009208	0.00281	0.864	F_2	OS:50		OS(512):64
30		HIB.	151.6	0.009663	0.00296	0.864	F_1	OS:64		OS(512):16
31	●	HIB.	214.5	0.008451	0.00258	0.864	F_1	OS:64		OS(512):32
32		HIB.	340.2	0.008437	0.00258	0.864	F_1	OS:64		OS(512):64
33		HIB.	153.0	0.009625	0.00294	0.864	F_2	OS:64		OS(512):16
34		HIB.	215.9	0.008376	0.00256	0.864	F_2	OS:64		OS(512):32
35		HIB.	338.9	0.008370	0.00256	0.864	F_2	OS:64		OS(512):64
36		HIB.	120.8	0.009715	0.00298	0.864	F_1	OS:40		OS(360):16
37		HIB.	182.3	0.008888	0.00272	0.864	F_1	OS:40		OS(360):32
38		HIB.	310.6	0.008872	0.00271	0.864	F_1	OS:40		OS(360):64
39		HIB.	91.8	0.013426	0.00414	0.864	F_2	OS:40		OS(360):8
40	●	HIB.	122.7	0.009693	0.00297	0.864	F_2	OS:40		OS(360):16
41	●	HIB.	183.5	0.008793	0.00269	0.864	F_2	OS:40		OS(360):32
42	●	HIB.	308.3	0.008784	0.00268	0.864	F_2	OS:40		OS(360):64
43		HIB.	196.7	0.008658	0.00264	0.864	F_1	OS:50		OS(360):32
44	8.4c●	HIB.	151.0	0.009033	0.00277	0.864	F_1	OS:64		OS(360):16
45		HIB.	339.7	0.008389	0.00256	0.864	F_1	OS:64		OS(360):64
46	8.6b●	HIB.	214.2	0.008343	0.00255	0.864	F_2	OS:64		OS(360):32
47	●	HIB.	338.3	0.008326	0.00254	0.864	F_2	OS:64		OS(360):64
48	8.7b●	HIB.	492.3	0.007975	0.00242	1.275	F_2	OS:128	64	



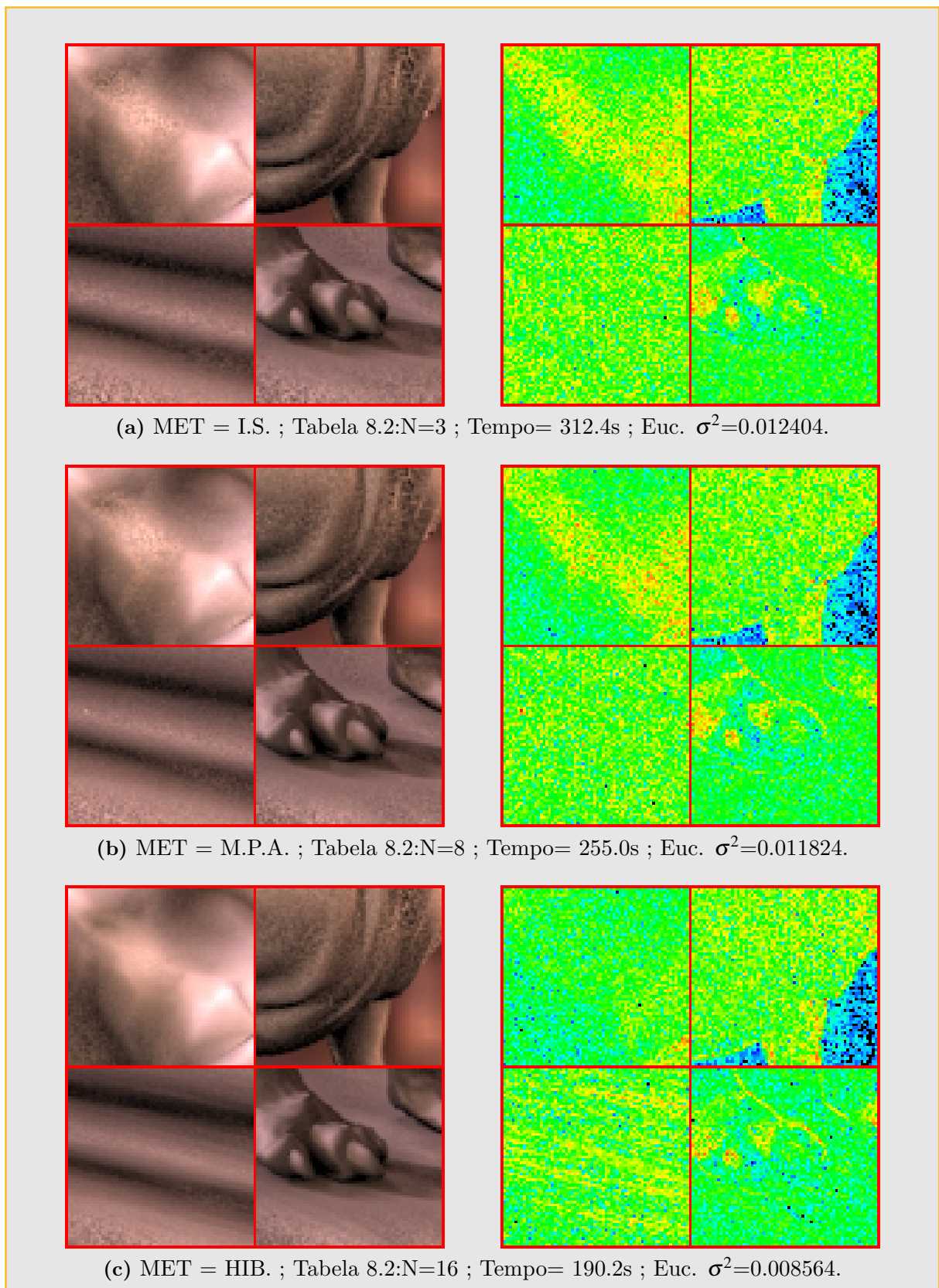
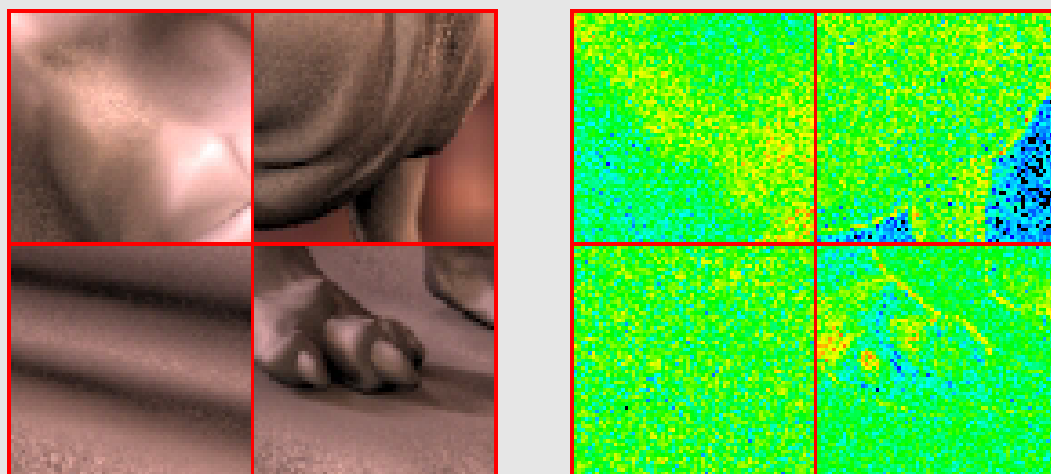
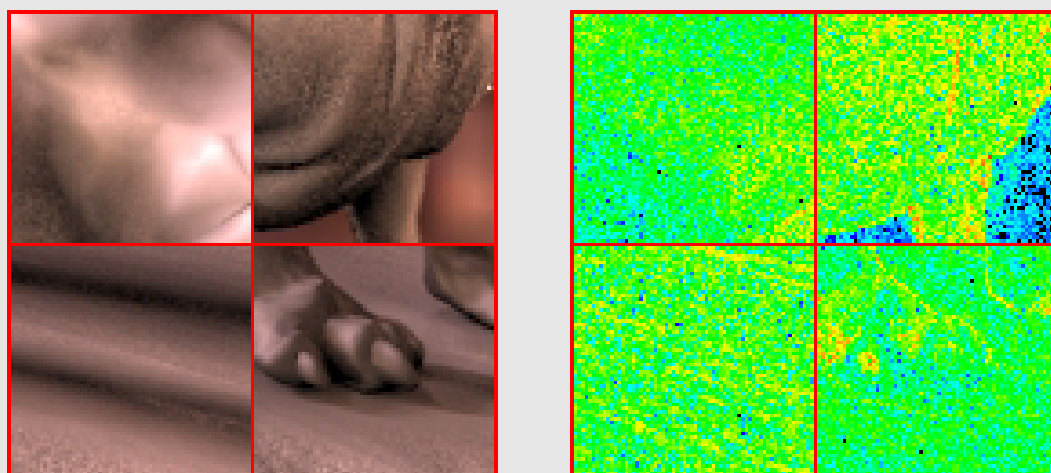


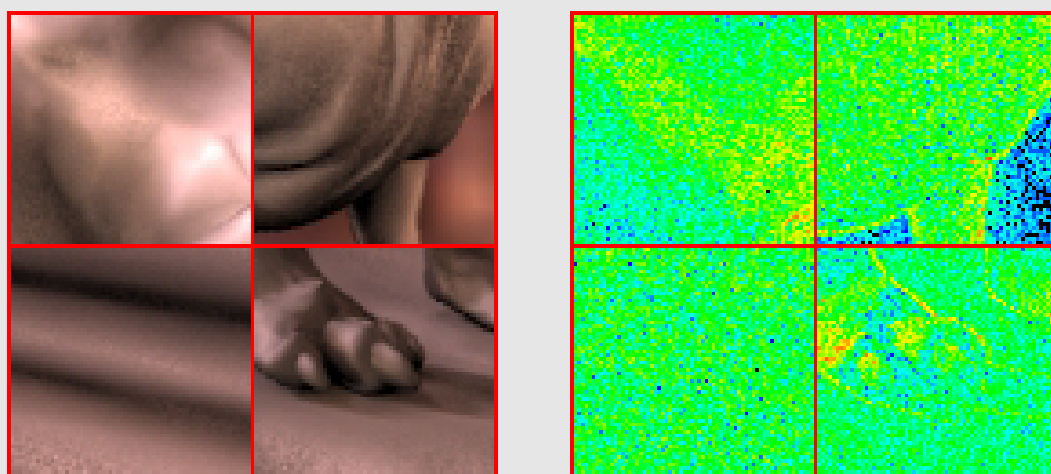
Figura 8.5: Detalhes de renderizações do modelo *Killeroo* com o mapa *Galileo*. Os parâmetros de renderização utilizados constam na tabela 8.2.



(a) MET = I.S. ; Tabela 8.2:N=4 ; Tempo= 604.3s ; Euc. $\sigma^2=0.009375$.

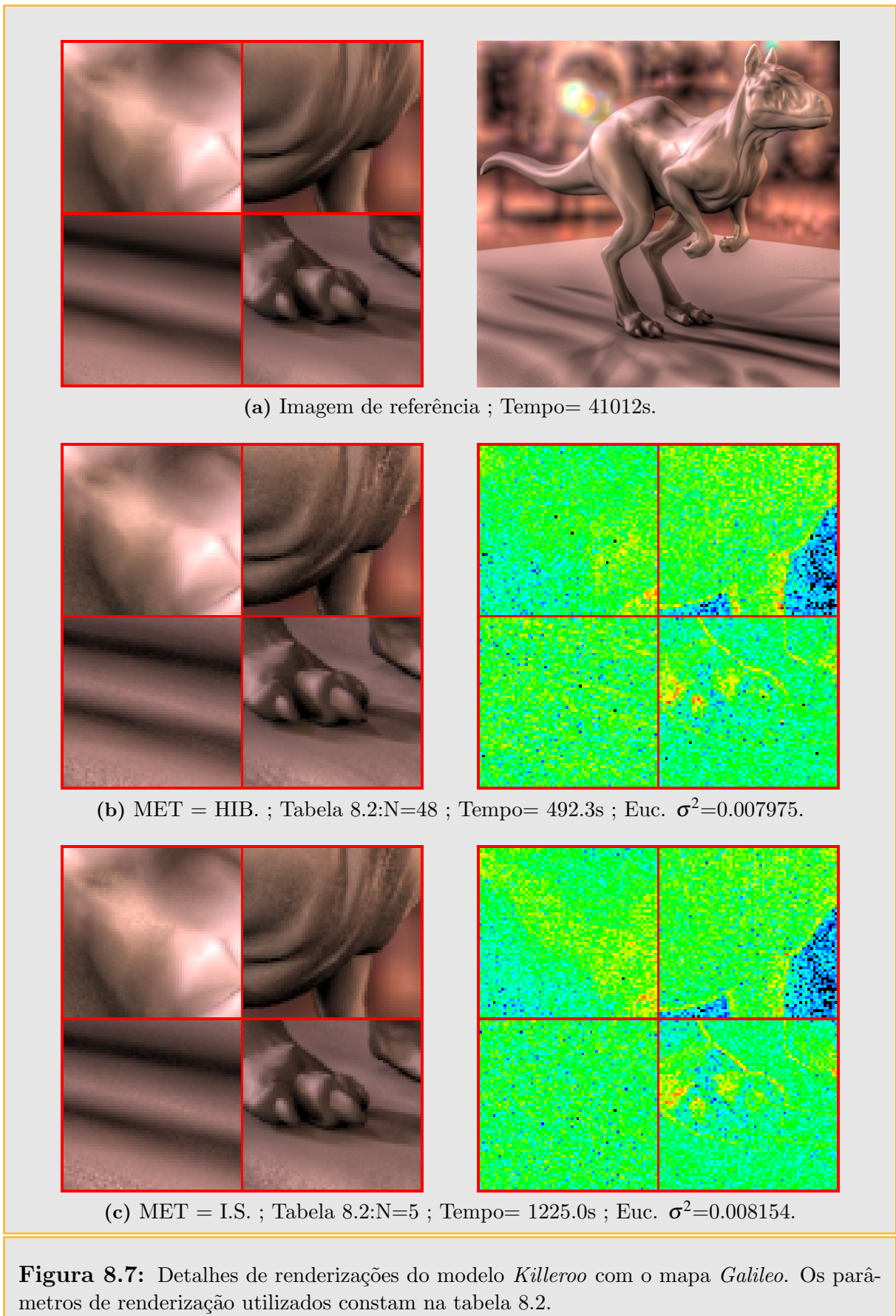


(b) MET = HIB. ; Tabela 8.2:N=46 ; Tempo= 214.2s ; Euc. $\sigma^2=0.008343$.



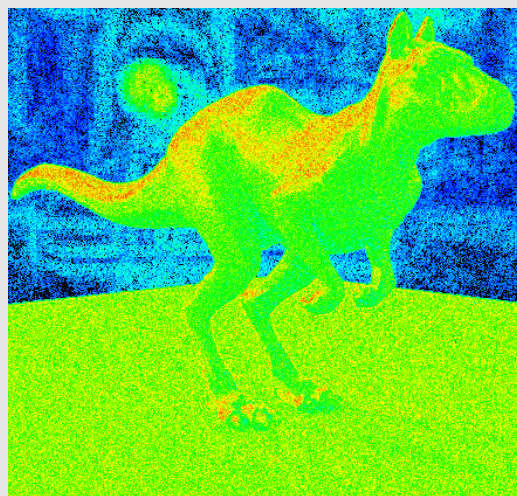
(c) MET = I.S. ; Tabela 8.2:N=5 ; Tempo= 1225.0s ; Euc. $\sigma^2=0.008154$.

Figura 8.6: Detalhes de renderizações do modelo *Killeroo* com o mapa *Galileo*. Os parâmetros de renderização utilizados constam na tabela 8.2.

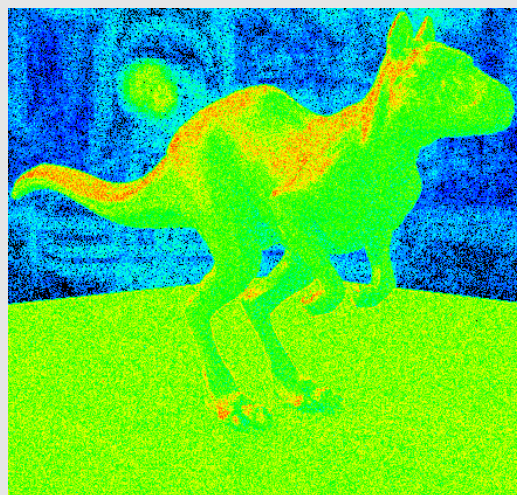




(a) MET = I.S. ; Tabela 8.2:N=1 ; Tempo= 81.1s ; Euc. $\sigma^2=0.051807$



(b) MET = M.P.A. ; Tabela 8.2:N=6 ; Tempo= 69.0s ; Euc. $\sigma^2=0.046599$.



(c) MET = HIB. ; Tabela 8.2:N=39 ; Tempo= 91.8s ; Euc. $\sigma^2=0.013426$.

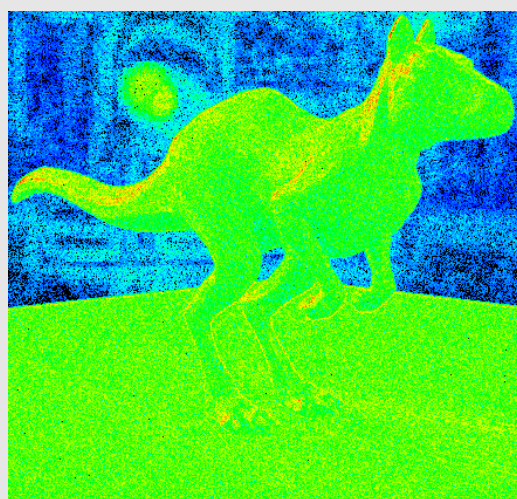
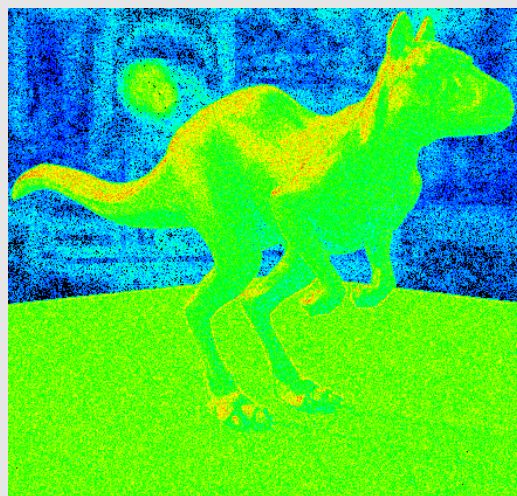


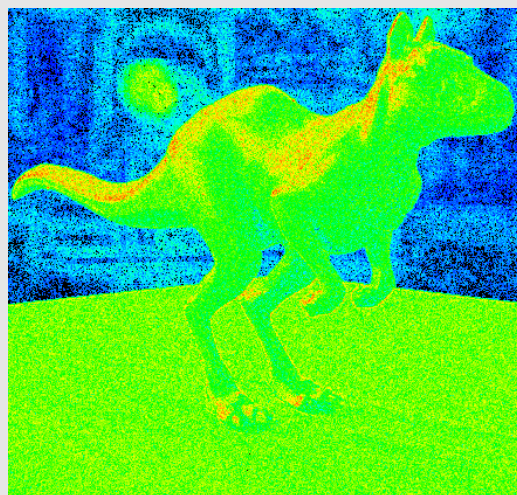
Figura 8.8: Detalhes de renderizações do modelo *Killeroo* com o mapa *Galileo*. Os parâmetros de renderização utilizados constam na tabela 8.2.



(a) MET = I.S. ; Tabela 8.2:N=2 ; Tempo= 156.4s ; Euc. $\sigma^2=0.021705$



(b) MET = M.P.A. ; Tabela 8.2:N=7 ; Tempo= 129.0s ; Euc. $\sigma^2=0.025096$.



(c) MET = HIB. ; Tabela 8.2:N=40 ; Tempo= 122.7s ; Euc. $\sigma^2=0.009693$.

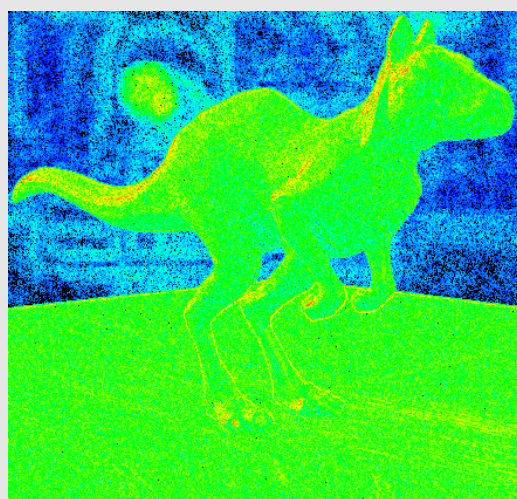
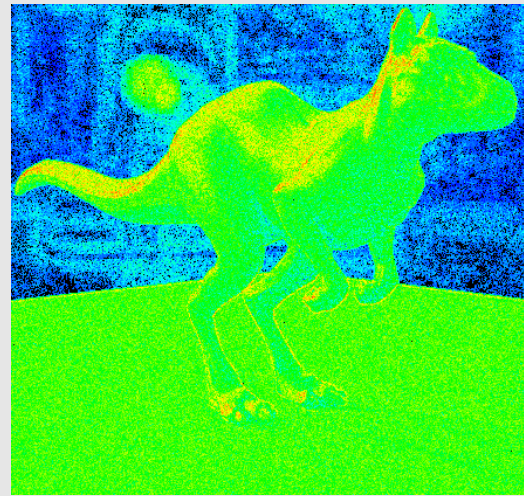
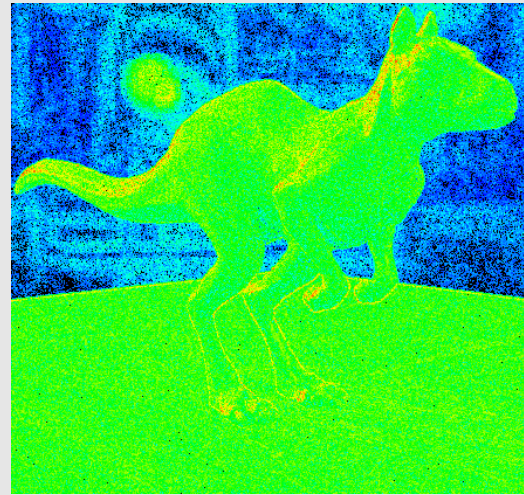


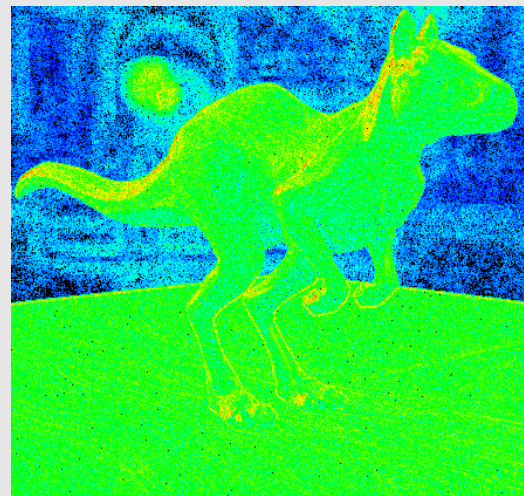
Figura 8.9: Detalhes de renderizações do modelo *Killeroo* com o mapa *Galileo*. Os parâmetros de renderização utilizados constam na tabela 8.2.



(a) MET = I.S. ; Tabela 8.2:N=3 ; Tempo= 312.4s ; Euc. $\sigma^2=0.012404$



(b) MET = HIB. ; Tabela 8.2:N=44 ; Tempo= 151.0s ; Euc. $\sigma^2=0.009033$.

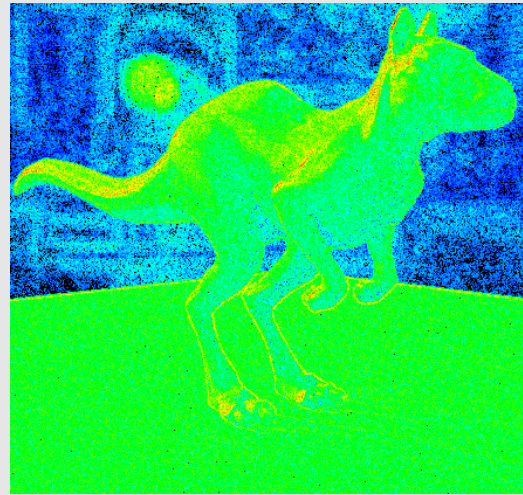


(c) MET = HIB. ; Tabela 8.2:N=46 ; Tempo= 214.2s ; Euc. $\sigma^2=0.008343$.

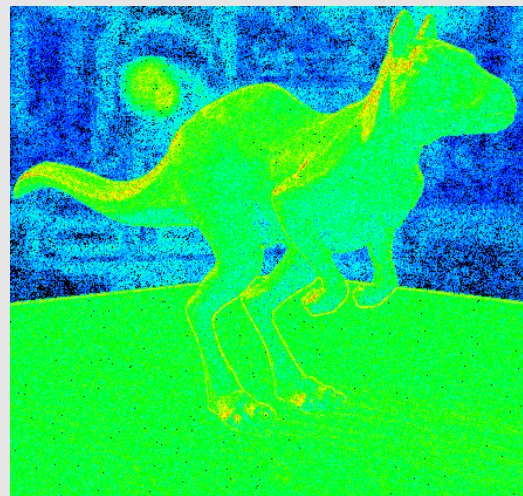
Figura 8.10: Detalhes de renderizações do modelo *Killeroo* com o mapa *Galileo*. Os parâmetros de renderização utilizados constam na tabela 8.2.



(a) Imagem de referência ; Tempo= 41012s.



(b) MET = I.S. ; Tabela 8.2:N=5 ; Tempo= 1225.0s ; Euc. $\sigma^2=0.008154$.



(c) MET = HIB. ; Tabela 8.2:N=48 ; Tempo= 492.3s ; Euc. $\sigma^2=0.007975$.

Figura 8.11: Detalhes de renderizações do modelo *Killeroo* com o mapa *Galileo*. Os parâmetros de renderização utilizados constam na tabela 8.2.

Tabela 8.3: Comparação de renderizações do modelo *Buddha* com o mapa *Grace*. A Imagem de referência utilizada para o cálculo dos erros foi renderizada com *IS* no PBRT [PH04b], com 32768 (64x512) amostras (Figura 8.15a). (AMD64 +3000, 2Gb RAM).

N	Fig.	MET	Tempo Seg.	Euc.MSE σ^2	Lum.MSE σ^2	% Área A	Mapa A		Mapa B	
							Cl	DL	IS	MPA
1	•	I.S.	80.5	0.129832	0.024133	0			16	
2	8.16a •	I.S.	163.6	0.077130	0.014590	0			32	
3	8.17a •	I.S.	311.2	0.059068	0.010880	0			64	
4	8.18a •	I.S.	611.4	0.045997	0.008769	0			128	
5	•	I.S.	1224.5	0.041945	0.008017	0			256	
6	8.16b •	HIB.	149.3	0.065939	0.012014	1.39		MC:64		OS(476):16
7	8.17b •	HIB.	210.7	0.051115	0.009802	1.39		MC:64		OS(476):32
8	•	HIB.	337.1	0.049371	0.009554	1.39		MC:64		OS(476):64
9	•	HIB.	151.2	0.063070	0.011258	1.39	F_1	OS:64		OS(476):16
10	•	HIB.	213.6	0.048474	0.009083	1.39	F_1	OS:64		OS(476):32
11	8.18b •	HIB.	334.5	0.046654	0.008810	1.39	F_1	OS:64		OS(476):64
12	8.16c	HIB.	148.9	0.062757	0.011162	1.39	F_2	OS:64		OS(476):16
13	8.17c	HIB.	209.0	0.048542	0.009077	1.39	F_2	OS:64		OS(476):32
14		HIB.	149.5	0.063629	0.011403	1.39	F_3	OS:64		OS(476):16
15		HIB.	209.6	0.049120	0.009256	1.39	F_3	OS:64		OS(476):32
16		HIB.	232.2	0.050088	0.009268	1.39	F_1	OS:80		OS(476):32
17		HIB.	231.6	0.050394	0.009324	1.39	F_2	OS:80		OS(476):32

Tabela 8.4: Comparação de renderizações do modelo *Buddha* com o mapa *Rnl*. A Imagem de referência utilizada para calcular os erros foi renderizada com *IS* no PBRT [PH04b], com 32768 (64x512) amostras (Figura 8.15b). (AMD64 +3000, 2Gb RAM).

N	Fig.	MET	Tempo Seg.	Euc. MSE σ^2	Lum. MSE σ^2	% Área A	Mapa A		Mapa B	
							Cl	DL	IS	MPA
1	•	I.S.	81.9	0.029390	0.008745	0			16	
2	8.21a •	I.S.	160.1	0.016434	0.005006	0			32	
3	8.22a •	I.S.	316.9	0.010649	0.003241	0			64	
4	8.23a •	I.S.	623.1	0.008355	0.002566	0			128	
5	•	MPA	67.9	0.021242	0.006399	0				OS(613):16
6	8.21b •	MPA	131.8	0.014224	0.004332	0				OS(613):32
7	•	MPA	255.8	0.009639	0.002945	0				OS(613):64
8	8.21c •	HIB.	157.4	0.010408	0.003083	3.71	F_3	OS:70		OS(437):16
9	8.22b •	HIB.	223.9	0.008653	0.002624	3.71	F_3	OS:70		OS(437):32
10	8.22c •	HIB.	259.2	0.008332	0.002527	3.71	F_2	OS:92		OS(437):32
11	•	HIB.	253.6	0.008305	0.002516	3.71	F_3	OS:92		OS(437):32
12	8.23c •	D.L.	348.4	0.007966	0.002497	100		MC:256		
13	•	D.L.	341.2	0.008744	0.002751	100	F_3	OS:256		

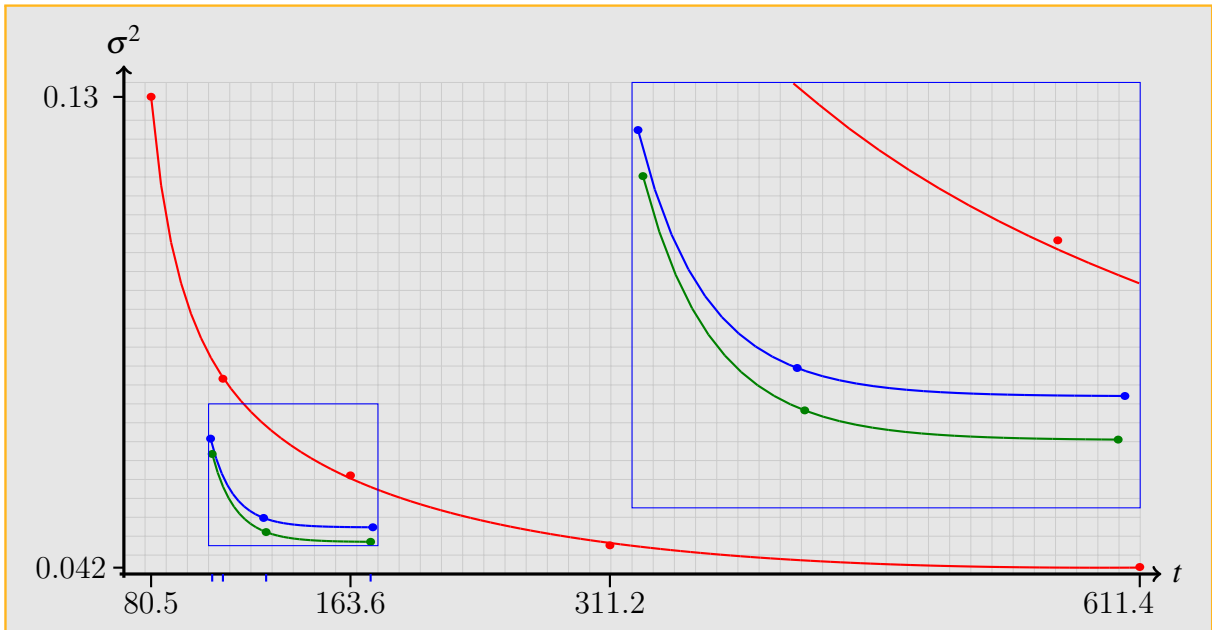


Figura 8.12: Aproximação por ajuste de curvas das funções de erro quadrado médio (MSE) euclidiano para alguns valores da tabela 8.3. ●: Amostragem por importância com plugin *IS*; ●: Amostragem por importância de Mapas de pré-amostragem, plugin *FHIS*; ●: Método híbrido utilizando mapas de pré-amostragem + luzes direcionais.

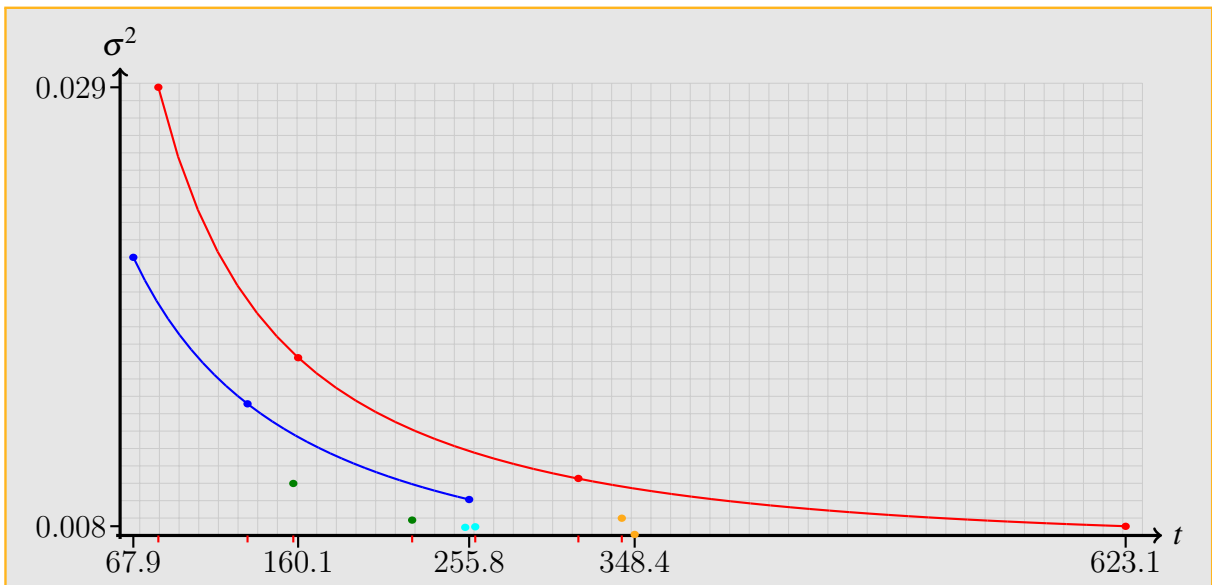


Figura 8.13: Aproximação por ajuste de curvas das funções de erro quadrado médio (MSE) euclidiano para alguns valores da tabela 8.4. ●: Amostragem por importância com plugin *IS*; ●: Amostragem por importância de Mapas de pré-amostragem, plugin *FHIS*. Os pontos coloridos representam a correspondente coordenada (tempo, σ^2) da tabela 8.4



(a) *Buddha* iluminado com o mapa *Grace*



(b) *Buddha* iluminado com o mapa *Rnl*

Figura 8.14: Renderizações de referência do modelo *Buddha* com 32768 (64x512) amostras por pixel. Em cada caso foi utilizado um mapeamento tonal de alto contraste.



(a) *Buddha* iluminado com o mapa *Grace*



(b) *Buddha* iluminado com o mapa *Rnl*

Figura 8.15: Renderizações de referência do modelo *Buddha* com 32768 (64x512) amostras por pixel. Em cada caso os retângulos vermelhos nas figuras (a) e (b) delimitam as regiões utilizadas nas figuras 8.16,8.17,8.18 e figuras 8.21, 8.22, 8.23 respectivamente para comparar alguns resultados das tabelas 8.3 e 8.4.

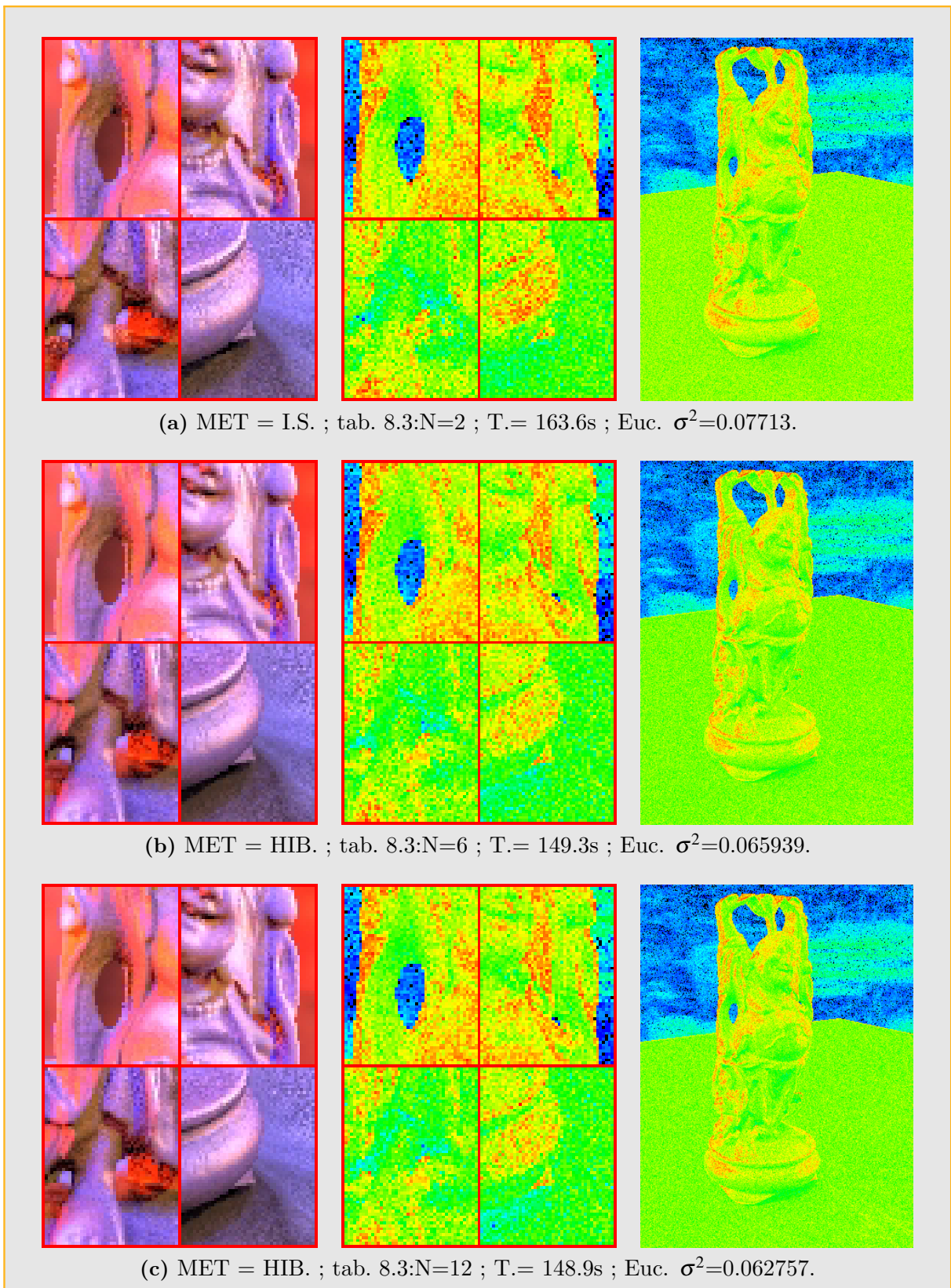


Figura 8.16: Detalhes de renderizações do modelo *Buddha* com o mapa *Grace*. Os parâmetros de renderização utilizados constam na tabela 8.3.

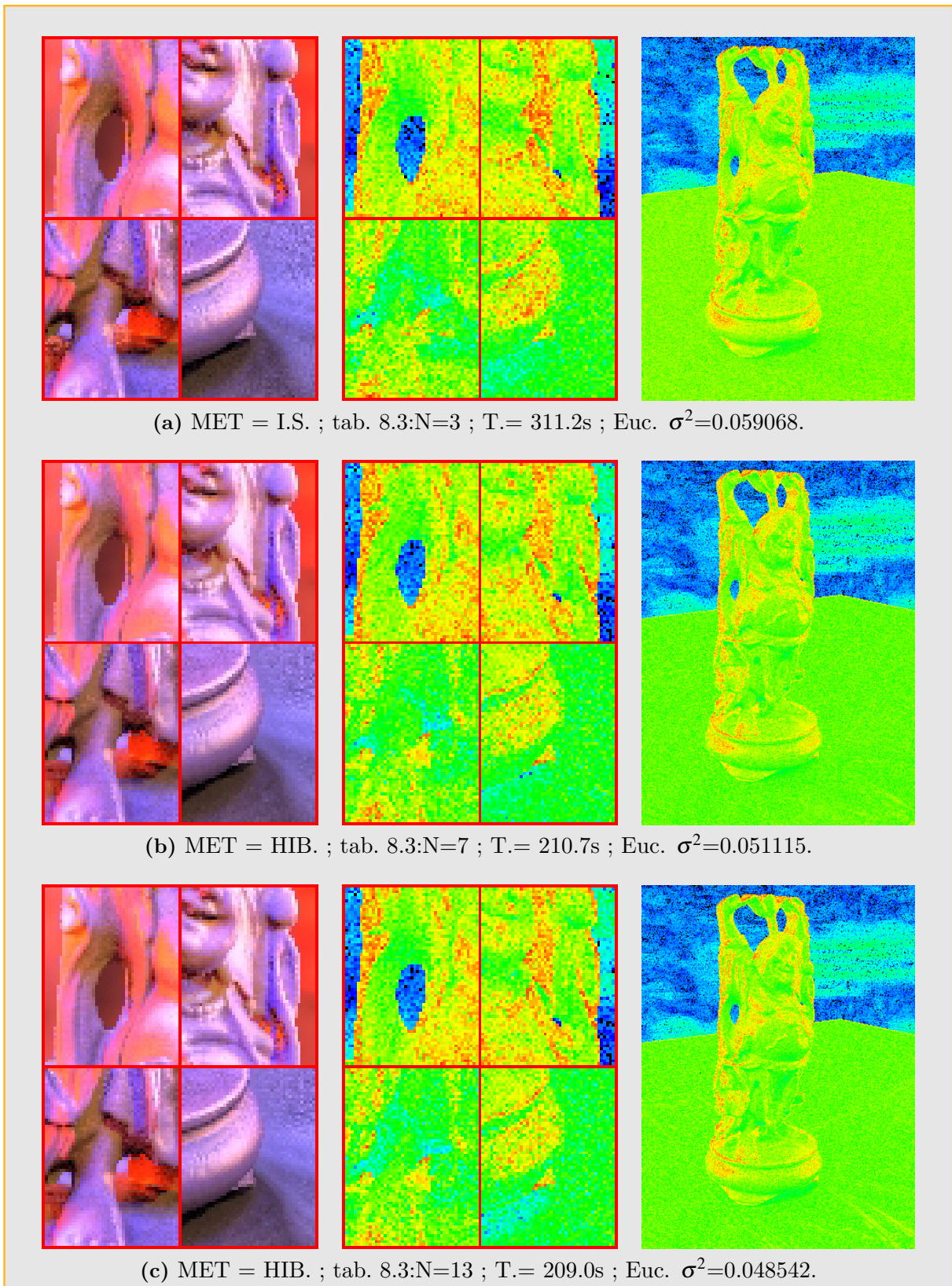


Figura 8.17: Detalhes de renderizações do modelo *Buddha* com o mapa *Grace*. Os parâmetros de renderização utilizados constam na tabela 8.3.

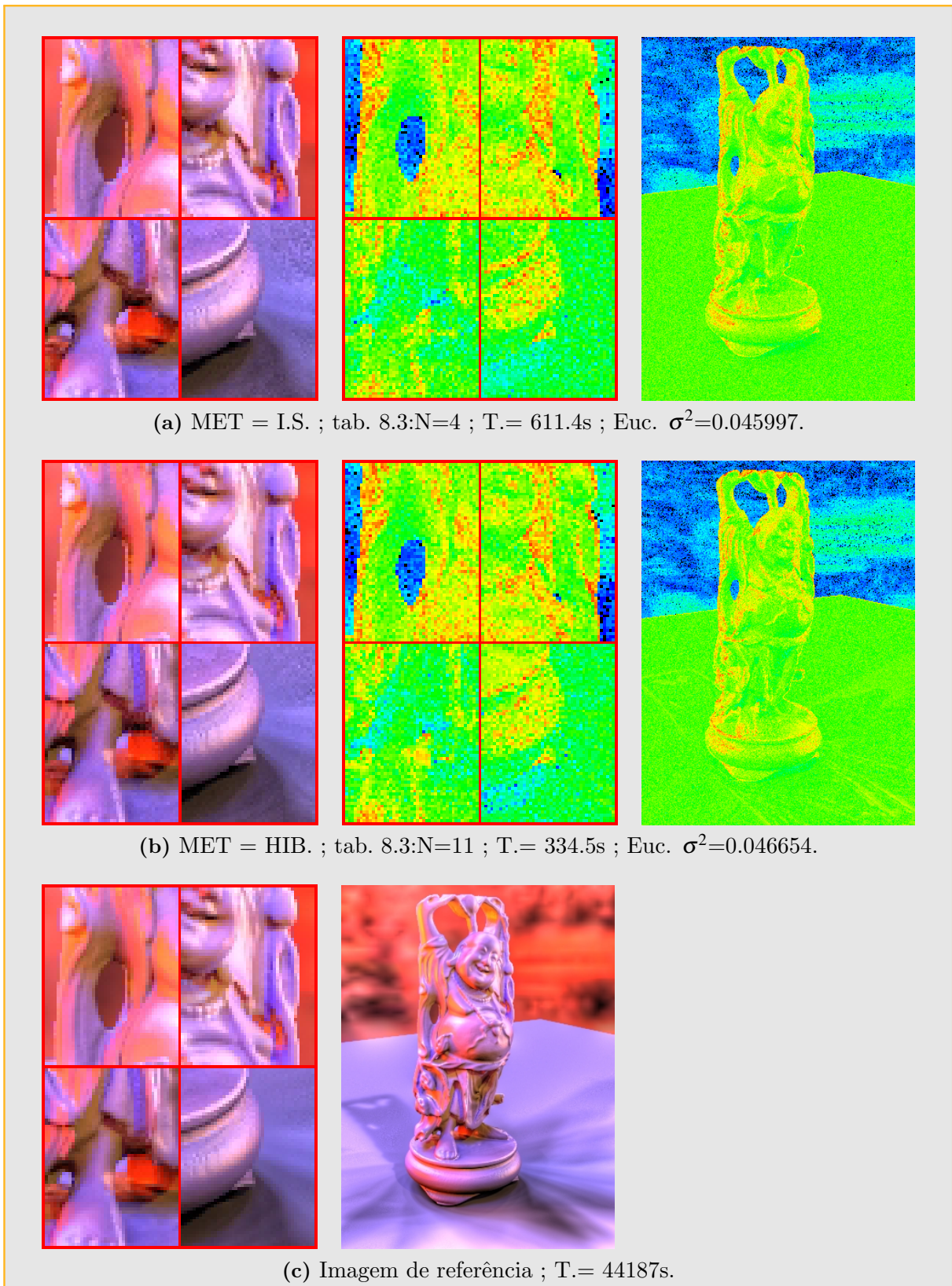
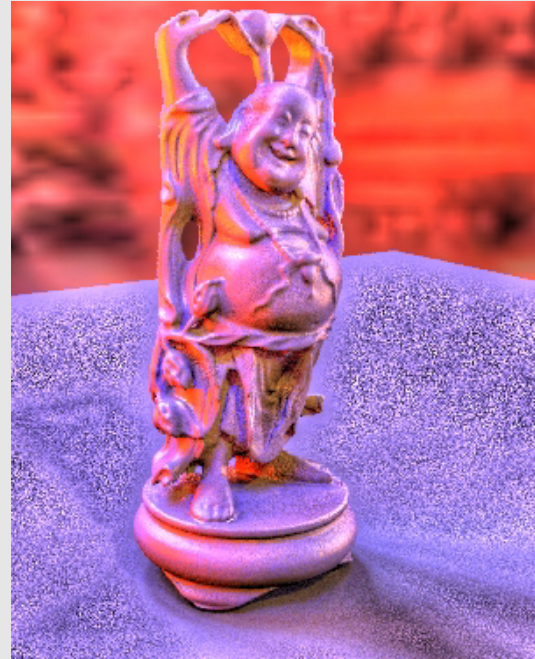
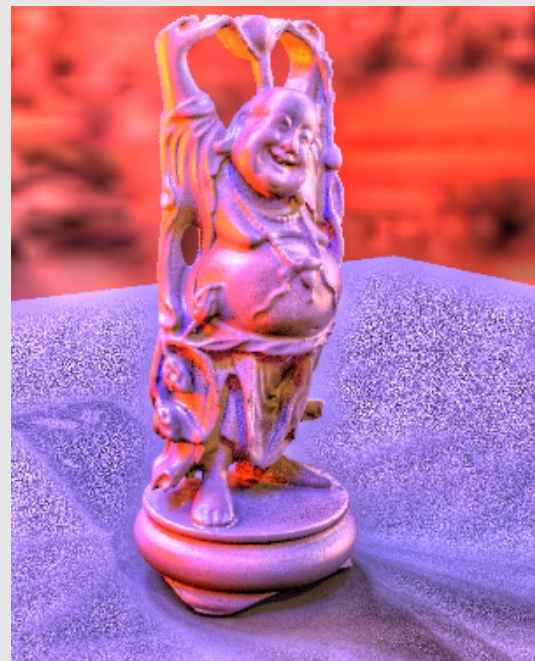


Figura 8.18: Detalhes de renderizações do modelo *Buddha* com o mapa *Grace*. Os parâmetros de renderização utilizados constam na tabela 8.3.



(a) MET = I.S. ; tabela 8.3:N=2 ; Tempo= 163.6s.



(b) MET = HIB. ; tabela 8.3:N=12 ; Tempo= 148.9s.

Figura 8.19: Detalhe do modelo *Buddha* iluminado com o mapa *Grace*. Os parâmetros de renderização utilizados constam na tabela 8.3.

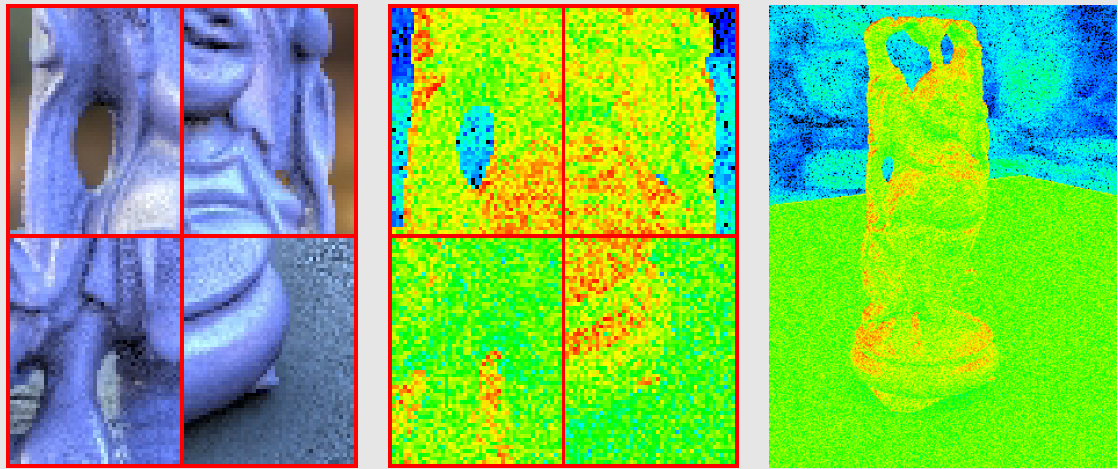


(a) MET = I.S. ; tabela 8.3:N=4 ; Tempo= 611.4s.

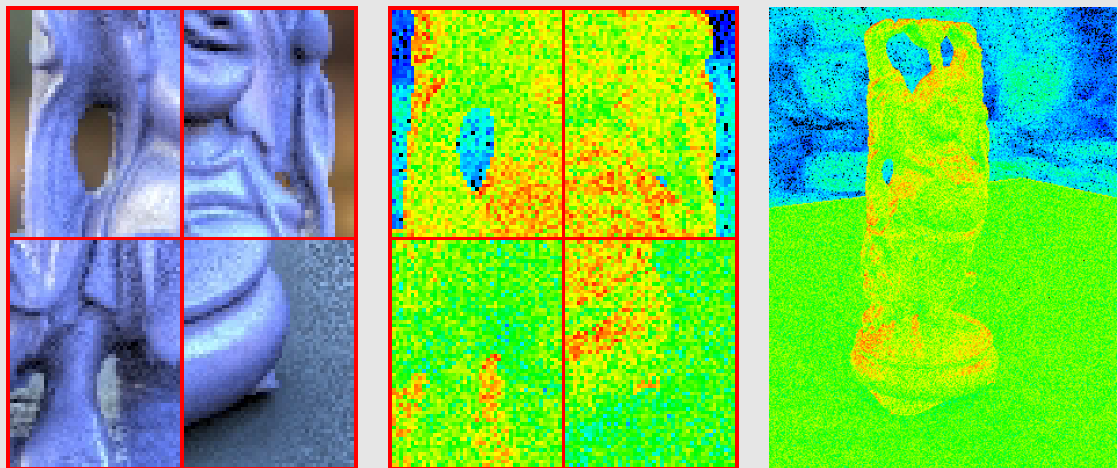


(b) MET = HIB. ; tabela 8.3:N=11 ; Tempo= 334.5s.

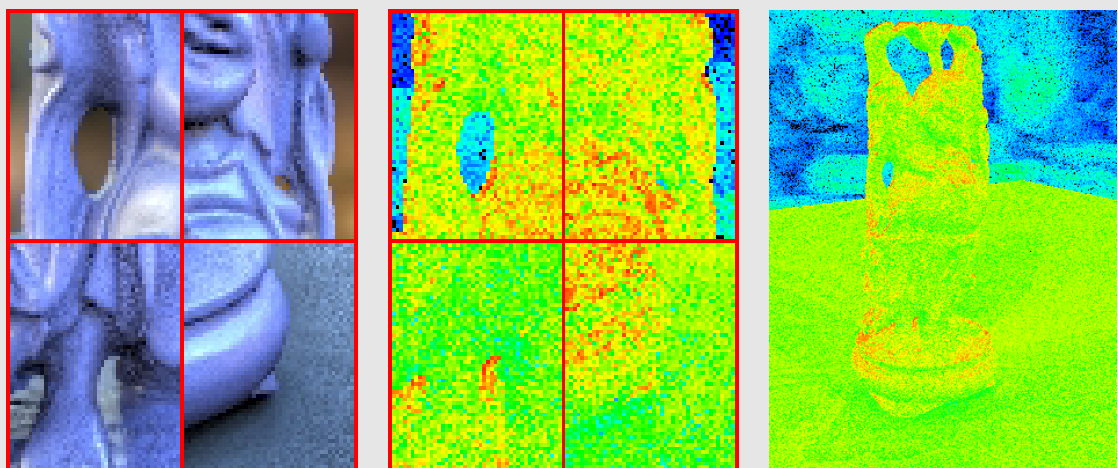
Figura 8.20: Detalhe do modelo *Buddha* iluminado com o mapa *Grace*. Os parâmetros de renderização utilizados constam na tabela 8.3.



(a) MET = I.S. ; tab. 8.4:N=2 ; T.= 160.1s ; Euc. $\sigma^2=0.016434$.



(b) MET = HIB. ; tab. 8.4:N=6 ; T.= 131.8s ; Euc. $\sigma^2=0.014224$.



(c) MET = HIB. ; tab. 8.4:N=8 ; T.= 157.4s ; Euc. $\sigma^2=0.010408$.

Figura 8.21: Detalhes de renderizações do modelo *Buddha* com o mapa *Rnl*. Os parâmetros de renderização utilizados constam na tabela 8.4.

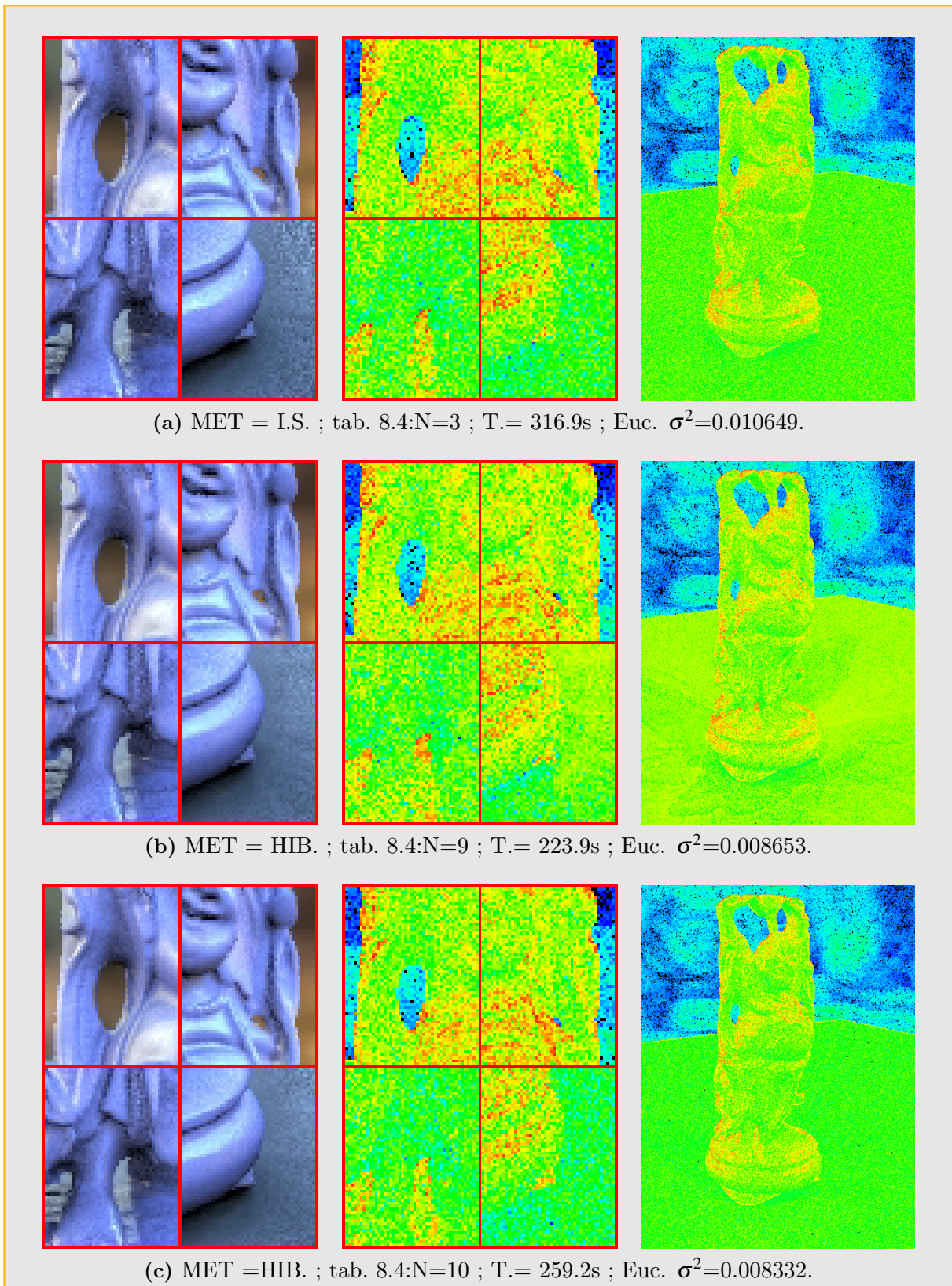


Figura 8.22: Detalhes de renderizações do modelo *Buddha* com o mapa *Rnl*. Os parâmetros de renderização utilizados constam na tabela 8.4.

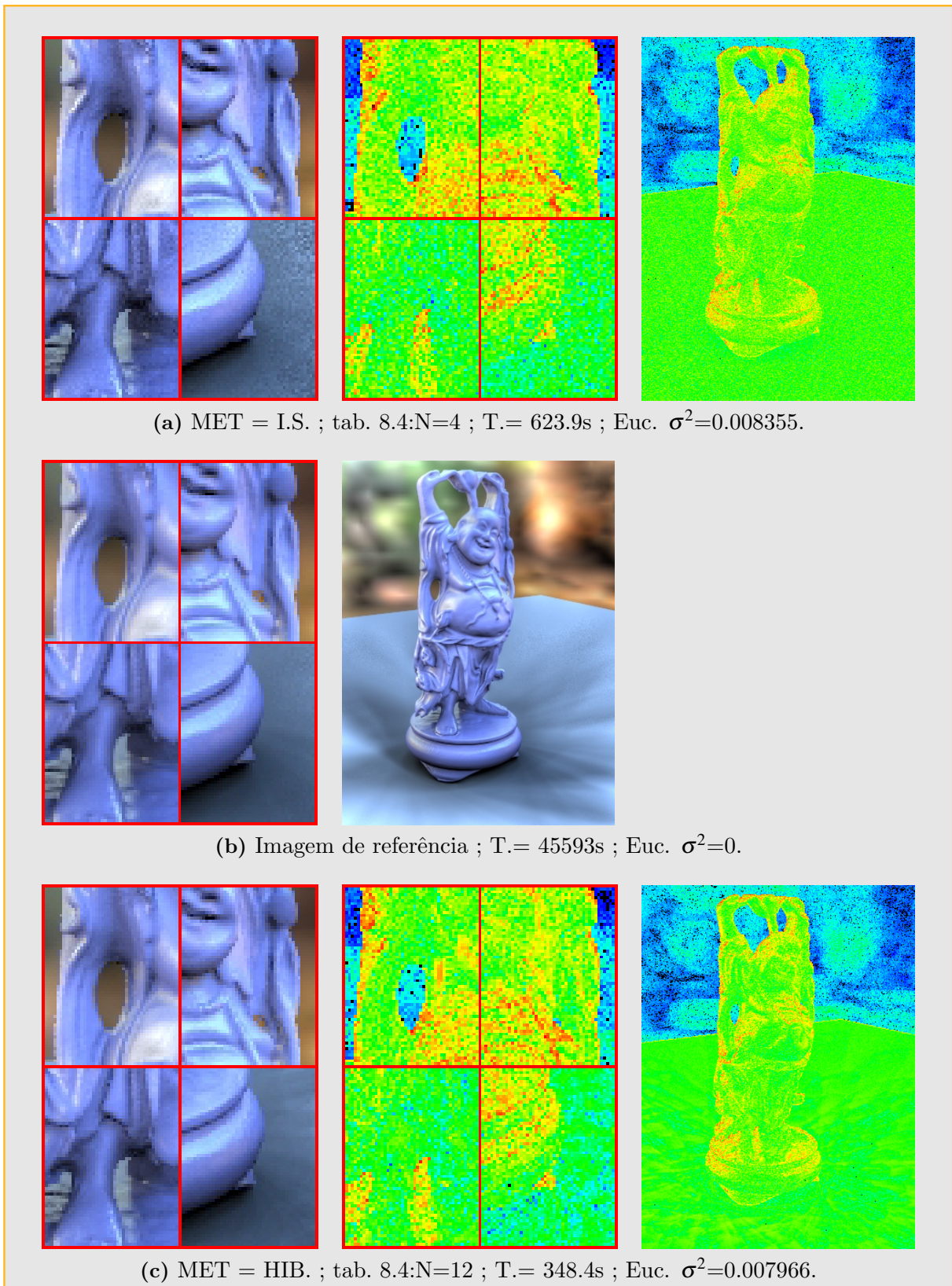


Figura 8.23: Detalhes de renderizações do modelo *Buddha* com o mapa *Rnl*. Os parâmetros de renderização utilizados constam na tabela 8.4.

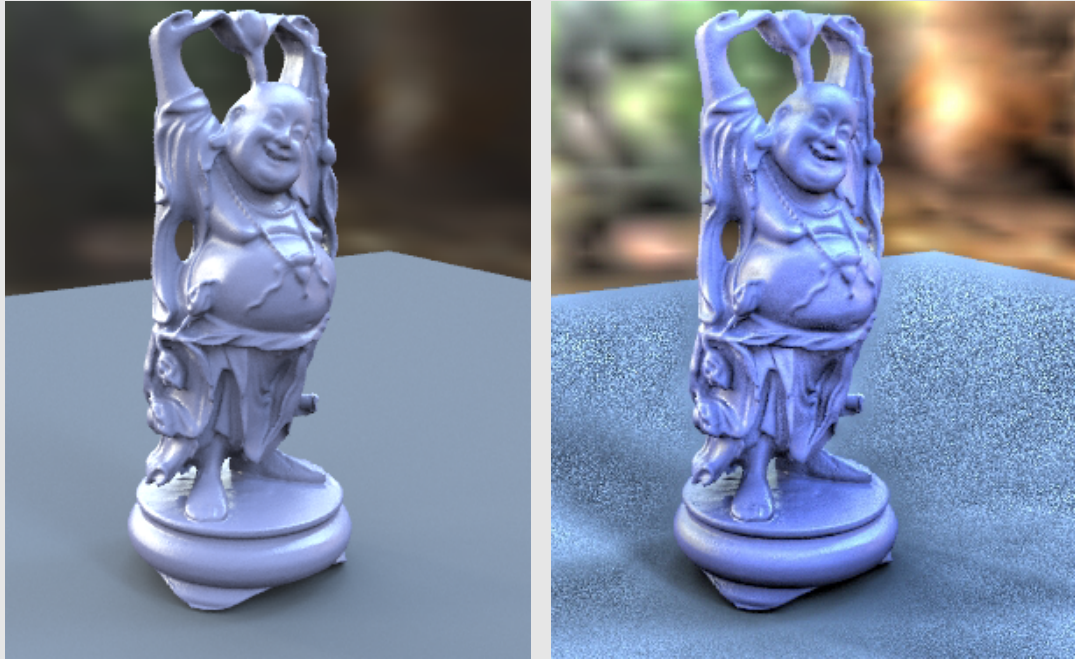


(a) MET = I.S. ; tabela 8.4:N=3 ; Tempo= 316.9s.

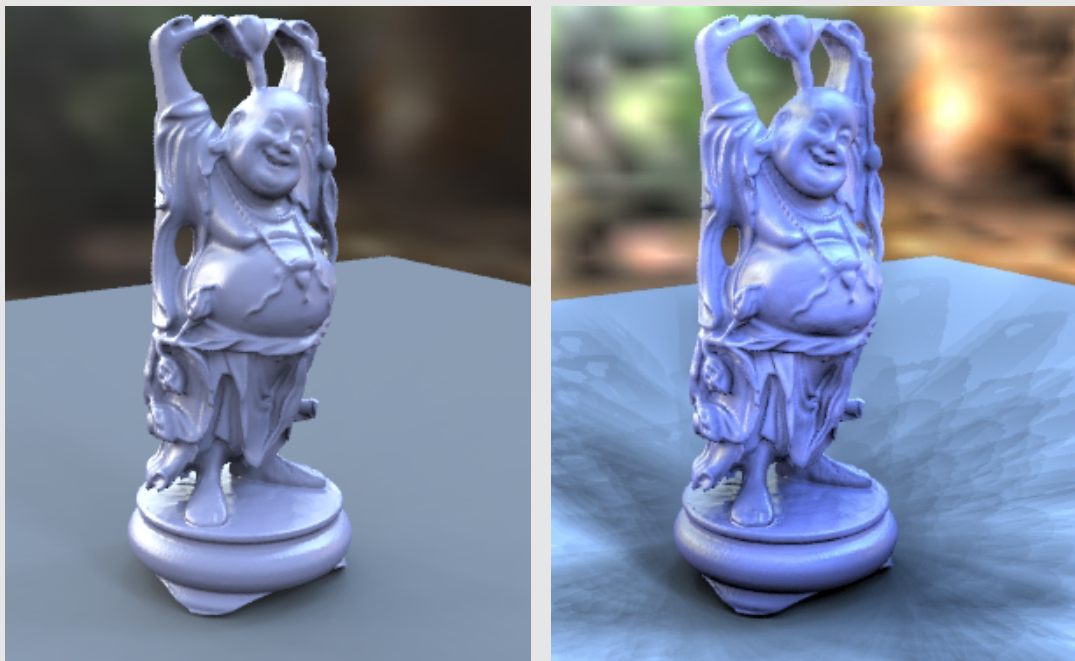


(b) MET = HIB. ; tabela 8.4:N=10 ; Tempo= 259.2s.

Figura 8.24: Detalhe do modelo *Buddha* iluminado com o mapa *Rnl*. Os parâmetros de renderização utilizados constam na tabela 8.4.



(a) MET = I.S. ; tabela 8.4:N=4 ; Tempo= 623.9s.



(b) MET = HIB. ; tabela 8.4:N=12 ; Tempo= 348.4s.

Figura 8.25: Detalhe do modelo *Buddha* iluminado com o mapa *Rnl*. Os parâmetros de renderização utilizados constam na tabela 8.4.

8.2.2 Resultados para iluminação global com Traçado de Caminhos (Path Tracing)

No cálculo de iluminação global mediante a equação de renderização, boa parte do tempo de renderização é gasto no cálculo das contribuições da iluminação indireta. A fim de poder mostrar aqui alguns exemplos de utilização e desempenho da nossa proposta híbrida para problemas de iluminação global, adaptamos e reescrevemos 2 versões (**PT1** e **PT2**, figura 8.26) do integrador *Tracing* (**PT**) do *PBRT* para poder realizar os cálculos com nosso esquema de iluminação.

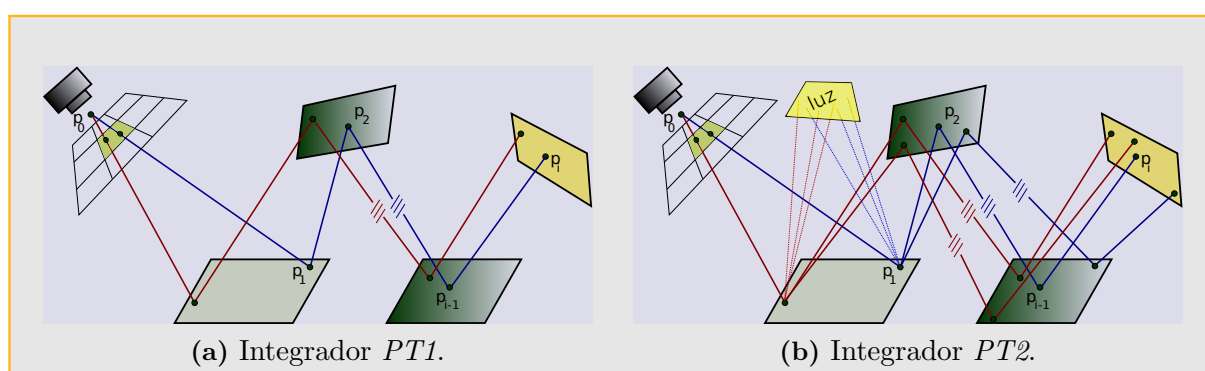


Figura 8.26: Integradores *PT1* e *PT2*. O integrador *PT1* é semelhante ao integrador *PT* do *PBRT*, adaptado para realizar amostragem de caminhos no método híbrido a partir de um conjunto de luzes direcionais e mapas de iluminação ou pré-amostragem. O integrador *PT2* foi escrito para trabalhar em duas etapas: na primeira é calculada a contribuição da iluminação direta (linhas de pontos) e na segunda a contribuição da iluminação indireta (linhas contínuas).

8.2.2.1 Interpretação das tabelas de resultados para Traçado de Caminhos

Para entender a codificação utilizada nas tabelas de resultados obtidos para Traçado de Caminhos, mostraremos a continuação a interpretação de três resultados da tabela 8.5. Esta interpretação é válida também para a tabela 8.6:

- **N=3** : Figura renderizada com o método híbrido (MET=HIB.), utilizando o plugin *PT1*. O tempo de renderização foi de 761.3 segundos e erro quadrado médio com métrica euclidiana de 0.015543. O mapa de iluminação foi dividido em dois mapas: Mapa A com 0.864% da área mais brilhante do mapa original, Mapa B com os 99.136% restantes. O Mapa A foi transformado num conjunto de 64 luzes direcionais utilizando amostragem hierárquica por importância com mosaicos de Penrose (OS) e clusterização com frequência F_2 . O Mapa B foi transformado num mapa de

pré-amostragem com 360 amostras obtidas pelo método de amostragem hierárquica por importância com mosaicos de Penrose (OS). A renderização foi realizada amostrando 256 caminhos por pixel (Amost. P.T.). Para cada caminho foi escolhida uma amostra por importância no conjunto formado pelas 64 luzes obtidas do Mapa A e o mapa de pré-amostragem B.

- **N=4** : Figura renderizada com o método de amostragem direta por importância (MET=I.S.) e plugin de integração *PT*. O tempo de renderização foi de 1068.9 segundos, com erro quadrado médio com métrica euclideana de 0.016286. Na renderização foi utilizado o mapa de iluminação completo e trabalhou-se com 256 caminhos por pixel e amostragem direta por importância de uma amostra por caminho (IS=1).
- **N=5** : Figura renderizada com o método híbrido (MET=HIB.), utilizando o plugin *PT2*. O tempo de renderização foi de 657.7 segundos, com erro quadrado médio com métrica euclideana de 0.015988. O mapa de iluminação foi dividido em dois mapas: Mapa A com 0.864% da área mais brilhante do mapa original, Mapa B com os 99.136% restantes. O Mapa A foi transformado num conjunto de 40 luzes direcionais utilizando amostragem hierárquica por importância com mosaicos de Penrose (OS) e clusterização com frequência F_2 . O Mapa B foi transformado num mapa de pré-amostragem com 360 amostras obtidas pelo método de amostragem hierárquica por importância com mosaicos de Penrose (OS). A renderização foi realizada utilizando 4 grupos de 64 caminhos (Amost P.T. 4×64) de forma que os 64 caminhos de cada grupo compartilhem o primeiro segmento de caminho. Para cada um dos 4 grupos foi calculada a iluminação direta, e para cada um dos 4×64 caminhos foi calculada apenas a contribuição indireta, finalmente ambas contribuições foram somadas.

Tabela 8.5: Comparação de renderizações de uma caixa com duas esferas iluminadas com o mapa *Galileo*, utilizando path tracing. A imagem de referência utilizada para calcular os erros foi renderizada com *IS* no PBRT [PH04b], com 32768 amostras (Figura 8.29a). (AMD64 +3000, 2Gb RAM).

N	Fig.	MET.	INT	Tempo Seg.	MSE EUC σ^2	Amost. P.T.	Área A (%)	Mapa A		Mapa B	
								Cl	DL	IS	MPA
1	8.30a	HIB.	PT1	387.6	0.029357	128	0.864	F_2	OS:64		OS(360)
2	8.30b	I.S.	PT	567.8	0.030331	128	0			1	
3	8.29c	HIB.	PT1	761.3	0.015543	256	0.864	F_2	OS:64		OS(360)
4	8.30c	I.S.	PT	1068.9	0.016286	256	0			1	
5	8.30d	HIB.	PT2	657.7	0.015988	4×64	0.864	F_2	OS:40		OS(360):16
6	8.29b	HIB.	PT2	1269.3	0.009003	4×61	0.864	F_2	OS:64		OS(360):16
7	8.29d	I.S.	PT	2236.8	0.009123	512	0			1	

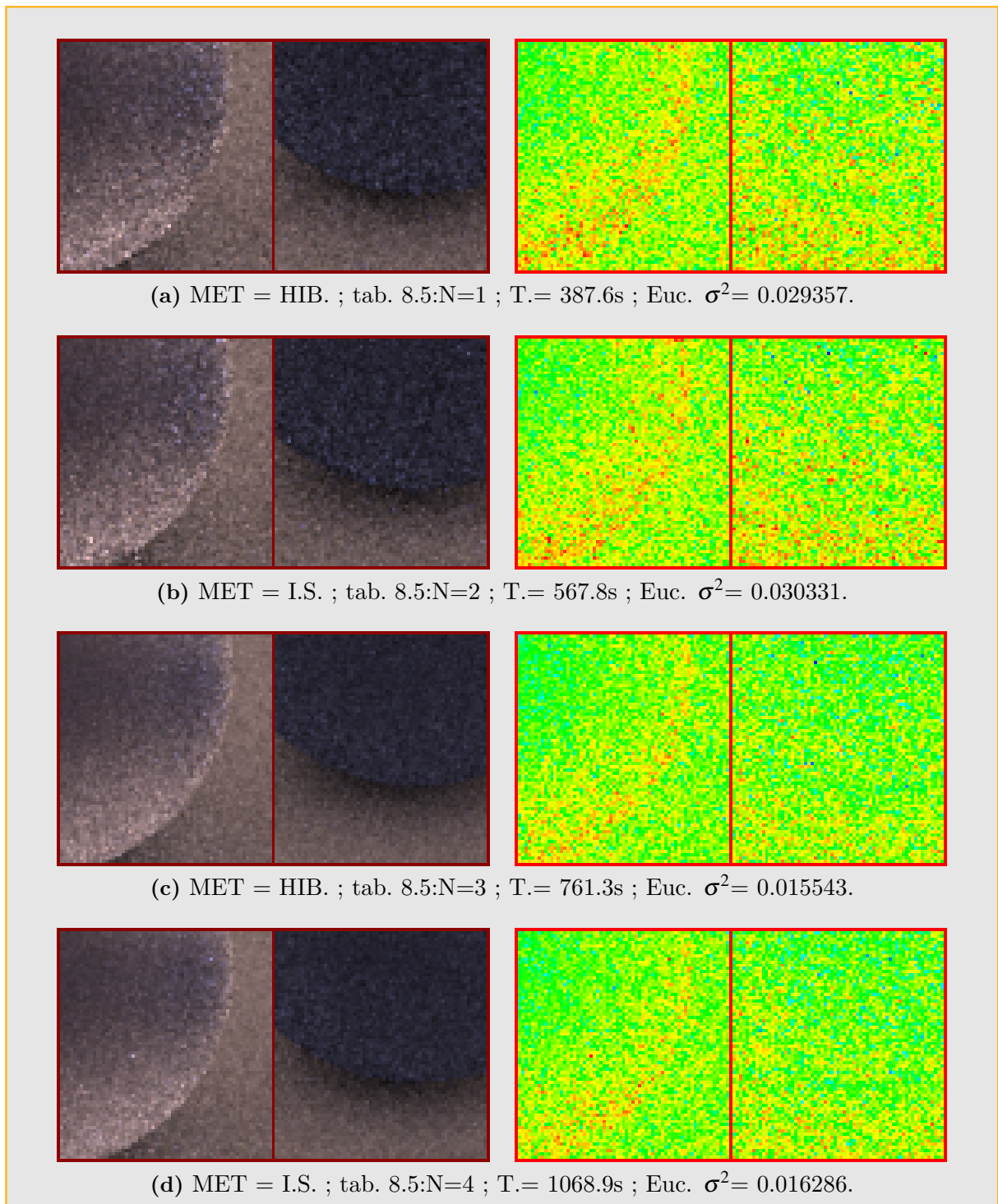


Figura 8.27: Detalhes de renderizações do modelo *2 esferas* com o mapa *Galileo*. Os parâmetros de renderização utilizados constam na tabela 8.5.

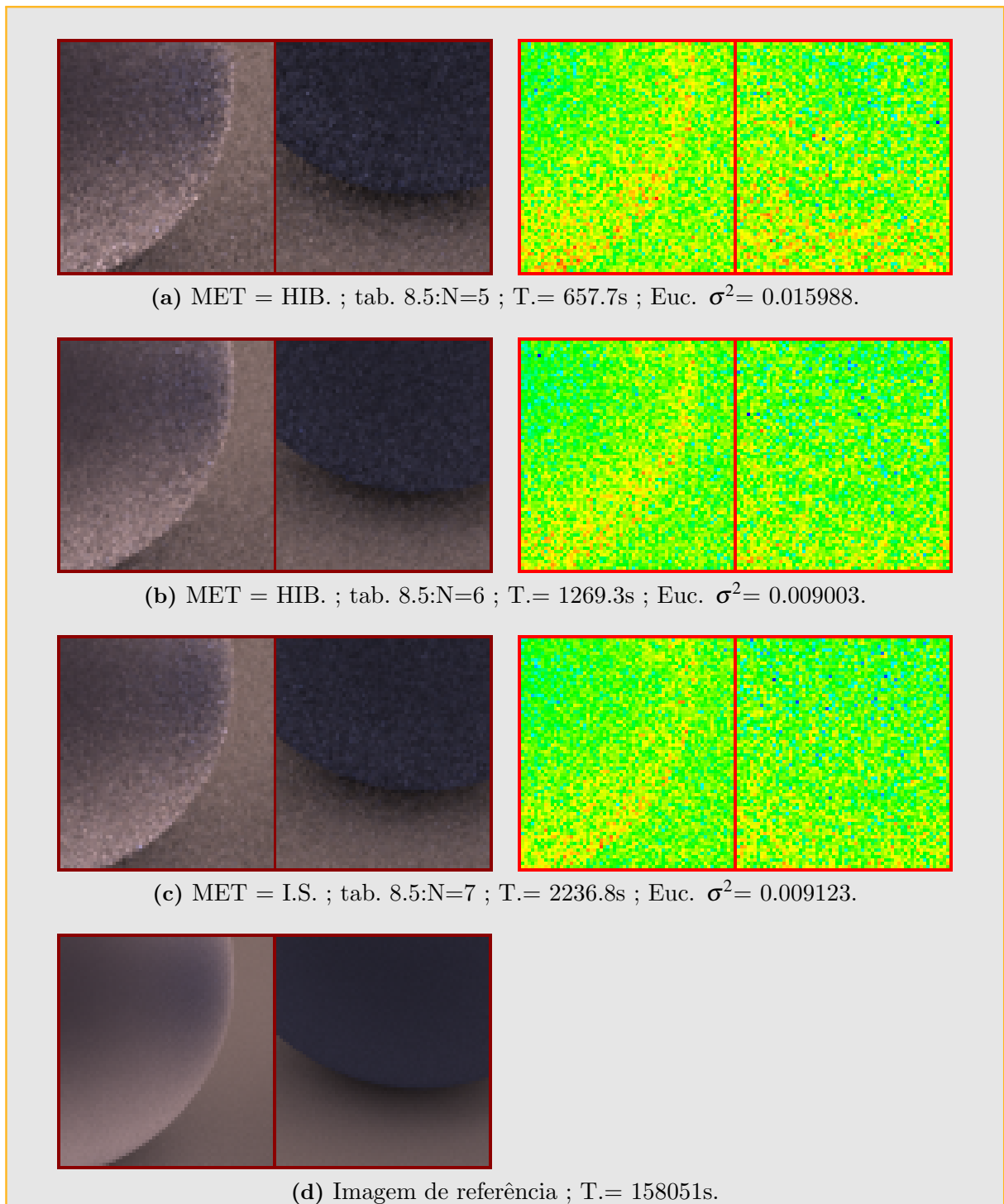
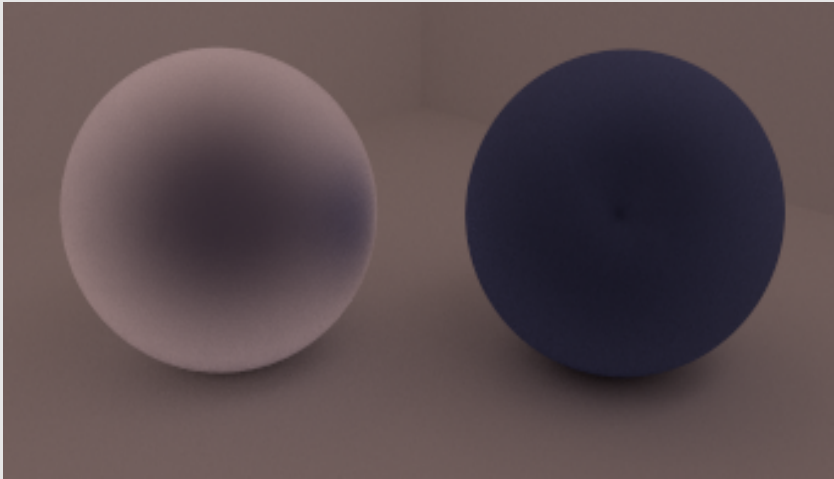
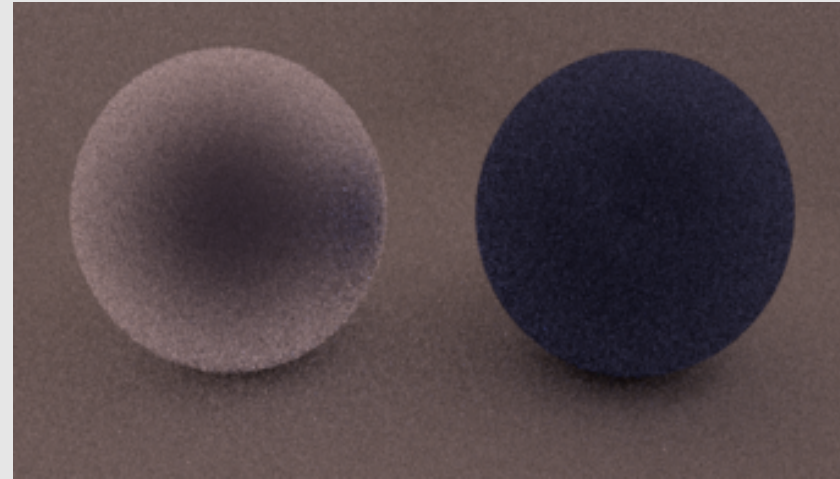


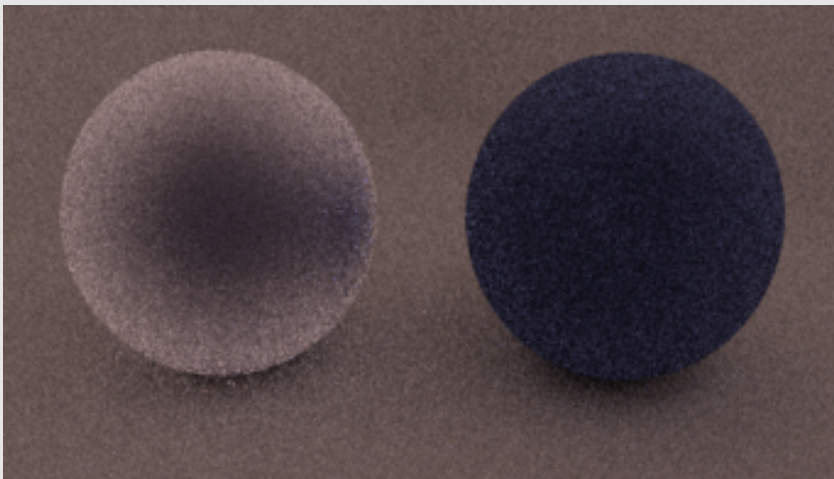
Figura 8.28: Detalhes de renderizações do modelo *2 esferas* com o mapa *Galileo*. Os parâmetros de renderização utilizados constam na tabela 8.5.



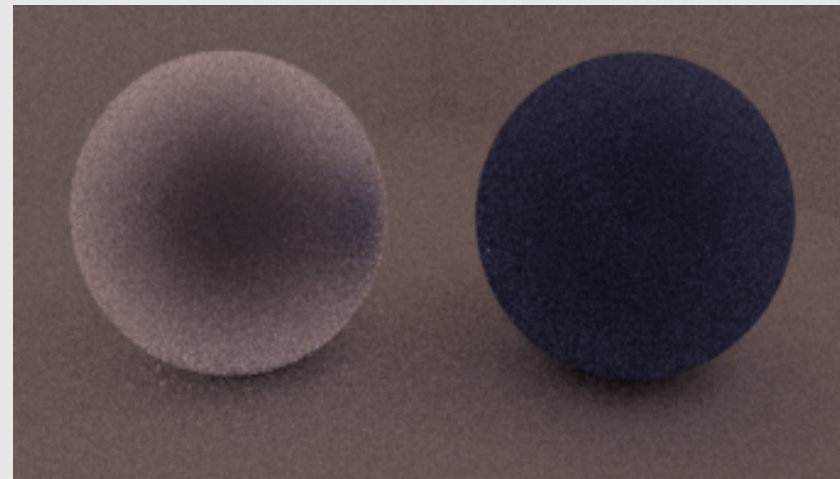
(a) Imagem de referência; Tempo= 158051s.



(b) MET = HIB. ; tabela 8.5:N=6 ; Tempo= 1269.3s.

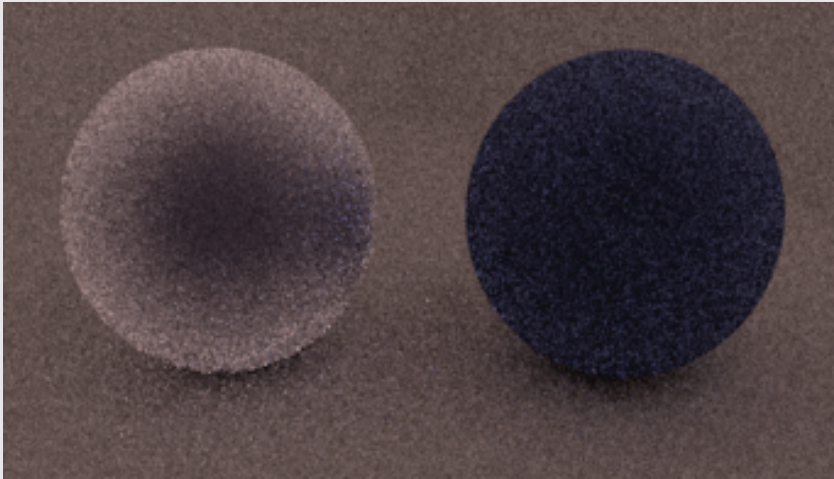


(c) MET = HIB. ; tabela 8.5:N=3 ; Tempo= 761.3s.

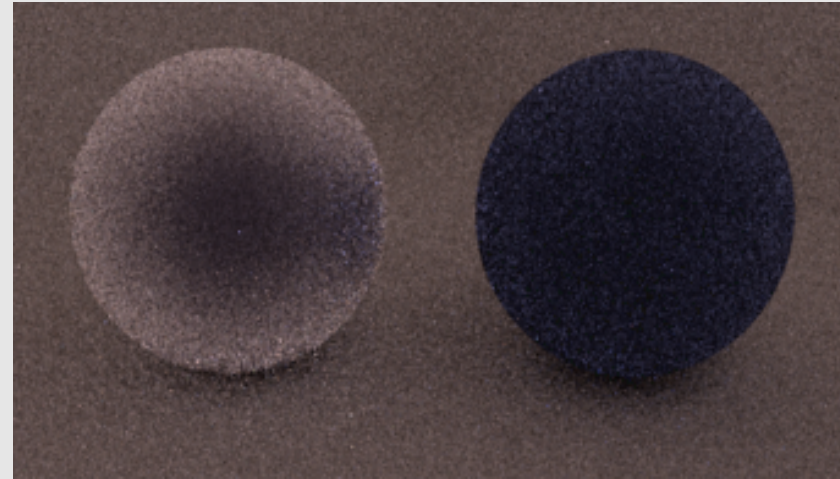


(d) MET = I.S. ; tabela 8.5:N=7 ; Tempo= 2236.8s.

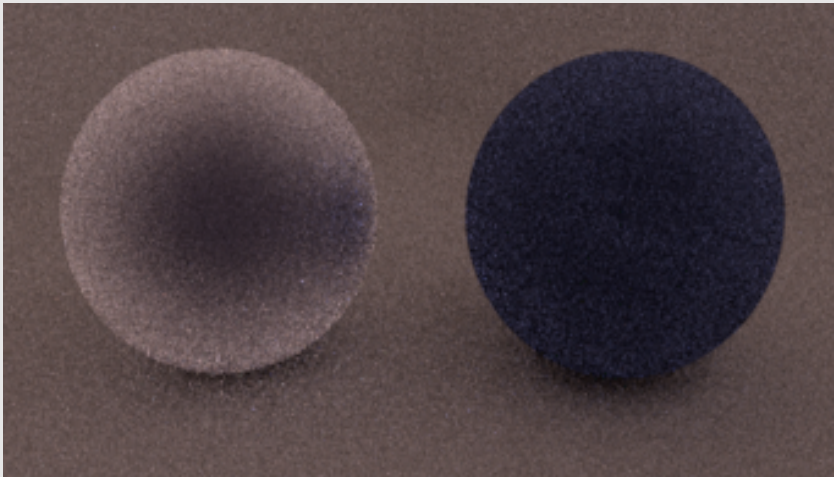
Figura 8.29: Modelo 2 esferas iluminado com o mapa *Galileo*. Os parâmetros de renderização utilizados constam na tabela 8.5.



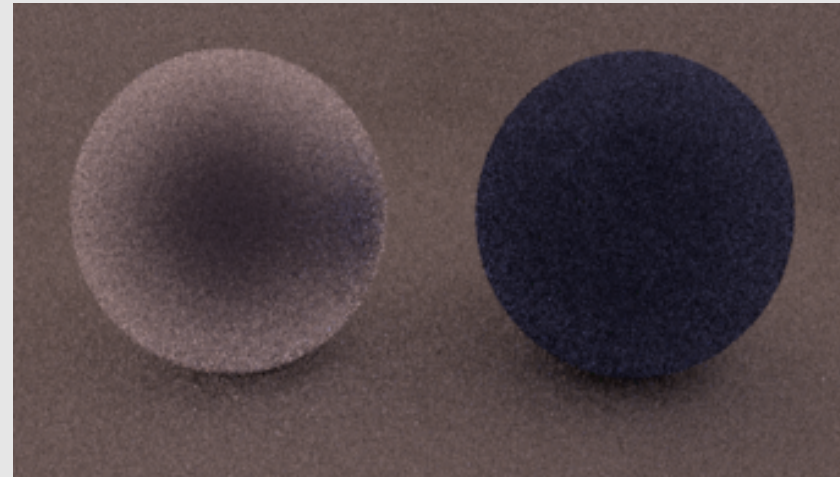
(a) MET = HIB. ; tabela 8.5:N=1 ; Tempo= 387.6s.



(b) MET = I.S. ; tabela 8.5:N=2 ; Tempo= 567.8s.



(c) MET = I.S. ; tabela 8.5:N=4 ; Tempo= 1068.9s.

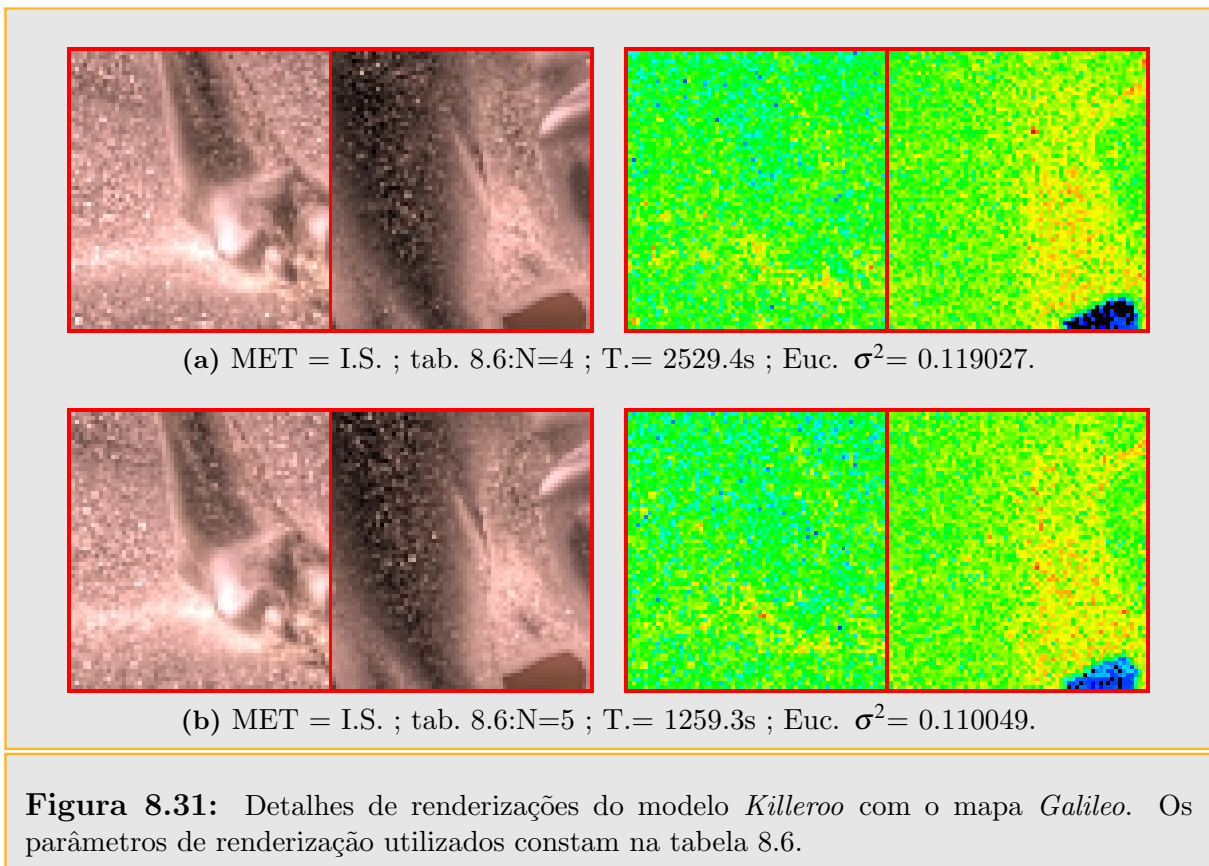


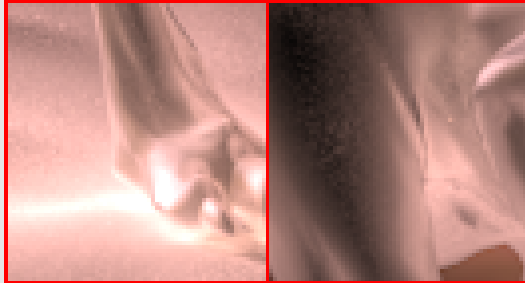
(d) MET = HIB. ; tabela 8.5:N=5 ; Tempo= 657.7s.

Figura 8.30: Modelo 2 esferas iluminado com o mapa *Galileo*. Os parâmetros de renderização utilizados constam na tabela 8.5.

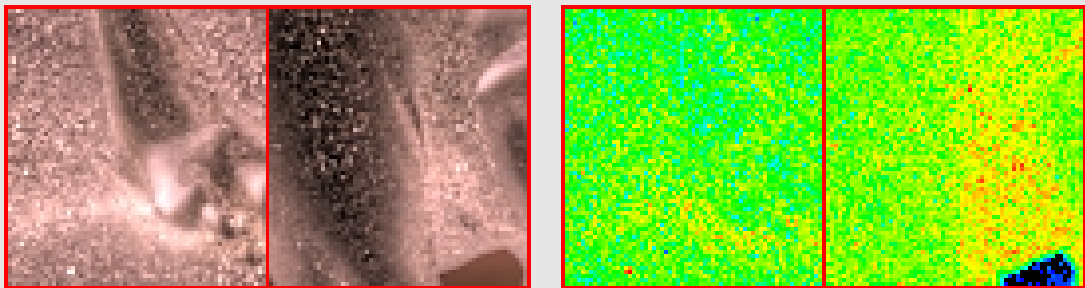
Tabela 8.6: Comparação de renderizações do modelo *Killeroo* iluminado com o mapa *Galileo*, utilizando path tracing. A imagem de referência utilizada para calcular os erros foi renderizada com *IS* no *PBRT* [PH04b], com 32768 amostras (Figura 8.34a). (AMD64 +3000, 2Gb RAM).

N	Fig.	MET.	INT	Tempo Seg.	MSE EUC σ^2	Amost. P.T.	Área A (%)	Mapa A		Mapa B	
								Cl	DL	IS	MPA
1	8.33a	I.S.	PT	1293.2	0.237848	256	0			1	
2	8.33b	HIB.	PT2	689.6	0.214314	8×29	0.864	F_2	OS:40		OS(512):8
3	8.34b	HIB.	PT1	1060.9	0.234260	256	0.864	F_2	OS:64		OS(360)
4	8.33c	I.S.	PT	2529.4	0.119027	512	0			1	
5	8.33d	HIB.	PT2	1259.3	0.110049	8×63	0.864	F_2	OS:40		OS(360):8

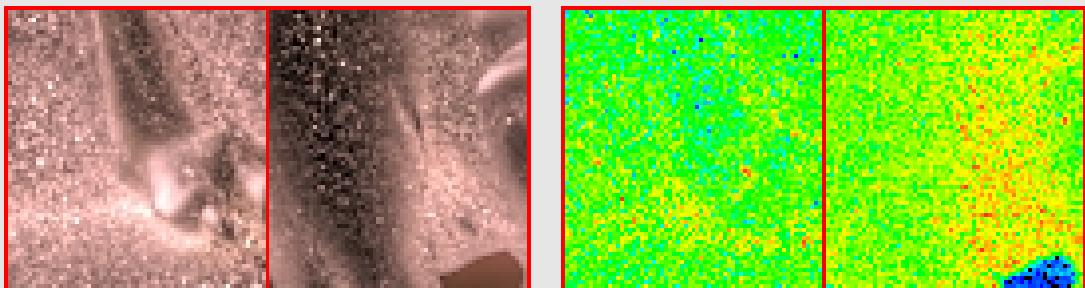




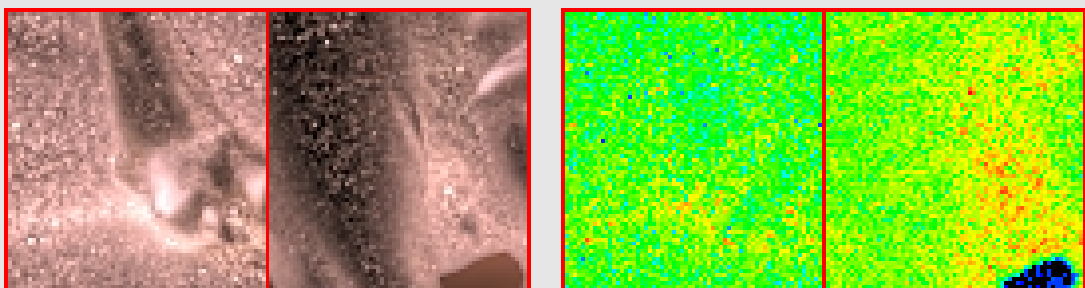
(a) Imagem de referência ; T.= 169774s.



(b) MET = I.S. ; tab. 8.6:N=1 ; T.= 1293.2s ; Euc. $\sigma^2= 0.237848$.

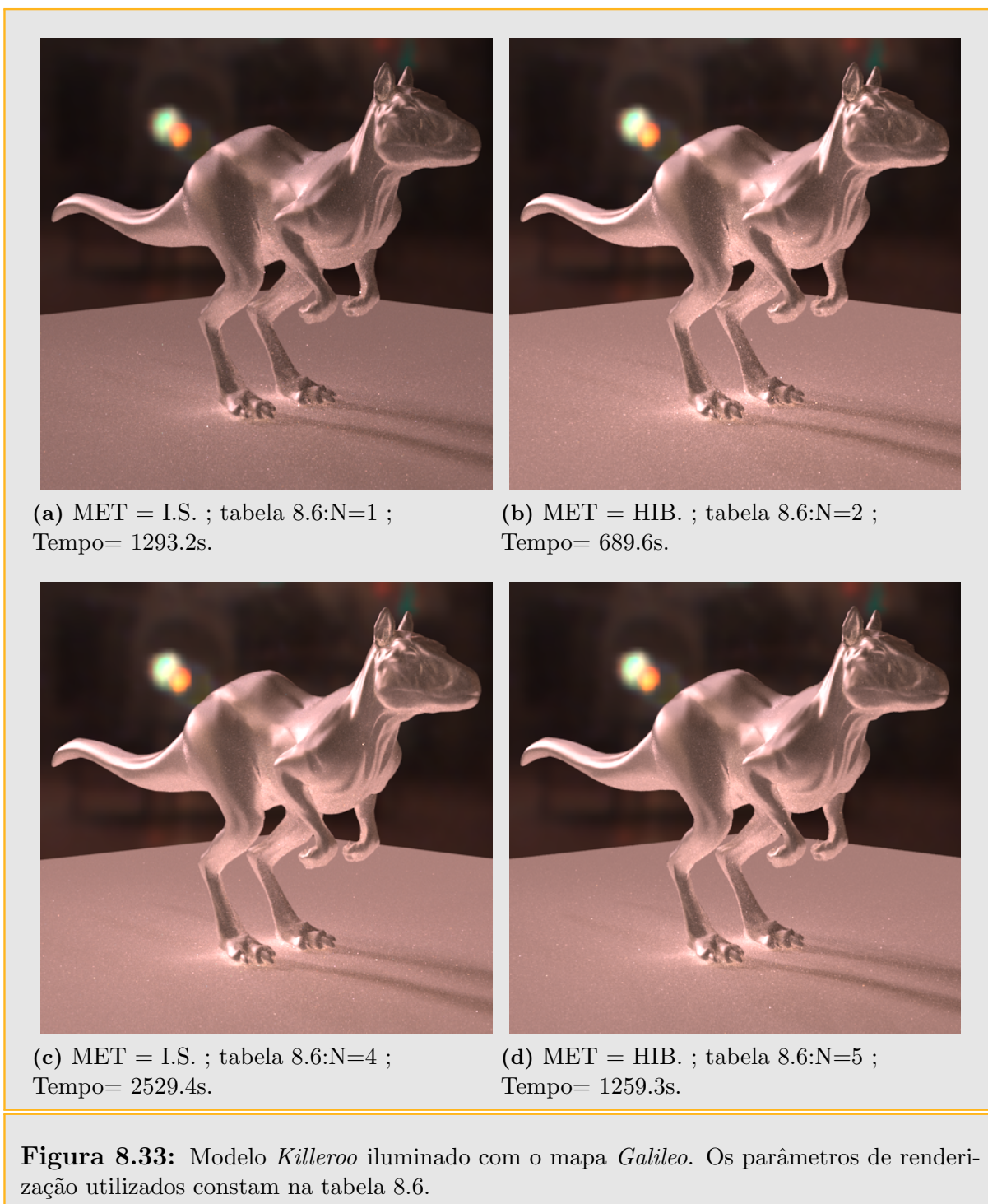


(c) MET = HIB. ; tab. 8.6:N=2 ; T.= 689.6s ; Euc. $\sigma^2= 0.214314$.



(d) MET = HIB. ; tab. 8.6:N=3 ; T.= 1060.9s ; Euc. $\sigma^2= 0.234260$.

Figura 8.32: Detalhes de renderizações do modelo *Killeroo* com o mapa *Galileo*. Os parâmetros de renderização utilizados constam na tabela 8.6.





(a) Imagem de referência ; Tempo= 169774s.



(b) MET = HIB. ; tabela 8.6:N=3 ; Tempo= 1060.9s.

Figura 8.34: Modelo *Killeroo* iluminado com o mapa *Galileo*. Os parâmetros de renderização utilizados constam na tabela 8.6.

8.2.3 Resultados para iluminação global com Mapeamento de Fótons (Photon Mapping)

Apresentamos nesta seção alguns exemplos renderizados com o método de iluminação global de Mapeamento de Fótons. Dado que este método é tendencioso a comparação dos resultados será visual.

Para todos os exemplos das tabelas 8.7, 8.8 e 8.9 o mapa de pré-amostras (*M.P.A.*) utilizado contém 360 amostras obtidas pelo método de Amostragem Hierárquica por Importância de Ostromukhov, [ODJ04].

Referências das tabelas 8.7, 8.8 e 8.9:

- **AA**= Quantidade de amostras para Antialiasing,
- **Fótons Ind.**= Número total fótons para calcular a iluminação indireta,
- **Fótons Ca.**= Número total fótons para calcular a iluminação caústica,
- **RF**= Quantidade de amostras utilizadas na reunião final de fótons (Final gathering),
- **DM**= Distância máxima permitida na coleta de fótons para interpolar a radiância num ponto p .

Os quatro últimos parâmetros são utilizados pelo integrador Photonmap do software *PBRT*. Para mais detalhes veja [PH04b](capítulo 16).

8.2.3.1 Interpretação das tabelas de resultados para Mapeamento de Fótons

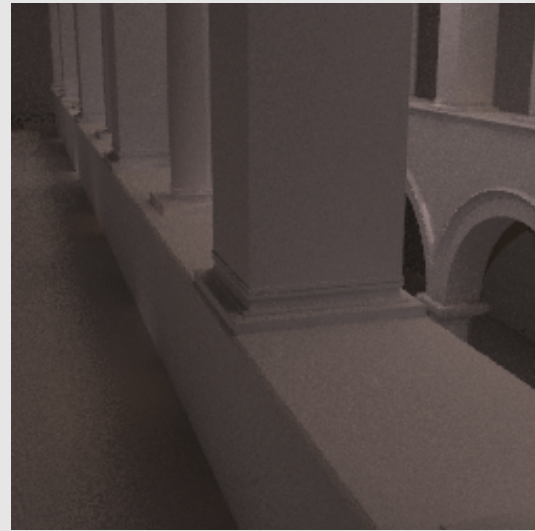
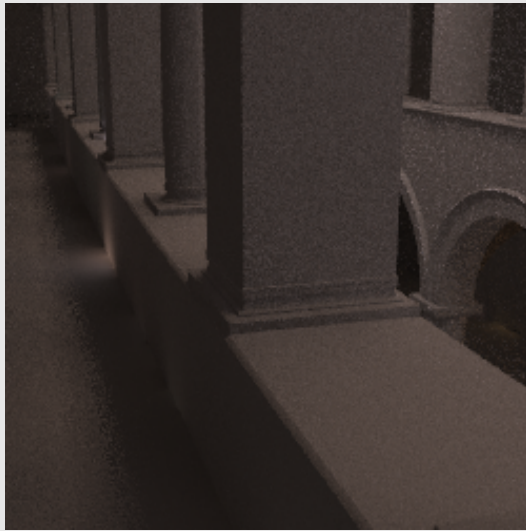
Para entender a codificação utilizada nas tabelas de resultados obtidos para Mapeamento de Fótons, mostraremos a continuação a interpretação de três resultados da tabela 8.7. Esta interpretação é válida também para as tabelas 8.8 e 8.9:

- **N=1** : Figura renderizada com o método de amostragem direta por importância (MET=I.S.). O tempo de renderização foi de 525.9 segundos. Foram usadas 4 amostras para antialiasing (AA). Na renderização foi utilizado o mapa de iluminação completo e trabalhou-se com 64 amostras por pixel na amostragem direta por importância (MAPA B / IS=64). No total foram usadas 4×64 amostras por pixel.

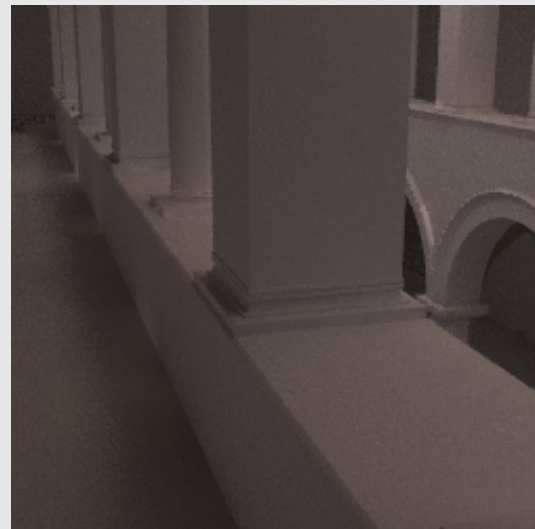
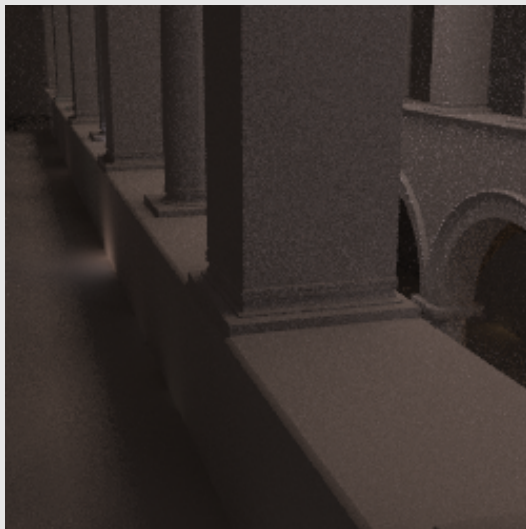
- **N=2** : Figura renderizada com o método híbrido. O tempo de renderização foi de 462.8 segundos. Foram usadas 4 amostras para antialiasing (AA). O mapa de iluminação foi dividido em dois mapas: Mapa A com 0.864% da área mais brilhante do mapa original, Mapa B com os 99.136% restantes. O Mapa A foi transformado num conjunto de 64 luzes direcionais utilizando amostragem hierárquica por importância com mosaicos de Penrose (OS) e clusterização com frequência F_2 . O Mapa B foi transformado num mapa de pré-amostragem com 360 amostras obtidas pelo método de amostragem hierárquica por importância com mosaicos de Penrose (OS). A renderização foi realizada utilizando as 64 luzes obtidas do Mapa A mais 32 amostras amostras tomados com amostragem direta por importância a partir do mapa de pré-amostragem B. No total foram usadas $4 \times (64+32)$ amostras por pixel.
- **N=8** : Figura renderizada com luzes direcionais (MET=D.L.). O tempo de renderização foi de 1016 segundos. Foram usadas 8 amostras para antialiasing (AA). O mapa de iluminação foi transformado num conjunto de 256 luzes direcionais utilizando amostragem de corte mediano. A renderização foi realizada utilizando as 256 luzes obtidas. No total foram usadas $8 \times (256)$ amostras por pixel.

Tabela 8.7: Comparação de renderizações do modelo *Sponza 1* iluminado com o mapa *Galileo*, utilizando mapeamento de fótons.

N	Fig.	MET.	Tempo Seg.	AA	Fótons		RF	DM	Área A (%)	Mapa A	Mapa B	
					Ind.	Ca.				DL	IS	MPA
1	8.35a	I.S.	525.9	4	1×10^5	0	64	1.5	0		64	
2	8.35b	HIB.	462.8	4	1×10^5	0	64	1.5	0.864	OS(F_2):64		32
3	8.35c	I.S.	702.5	4	1×10^5	0	64	1.5	0		128	
4	8.35d	HIB.	530.9	4	1×10^5	0	64	1.5	0.864	OS(F_2):64		64
5	8.36a	I.S.	1379.4	8	1×10^5	0	64	1.5	0		128	
6	8.36b	HIB.	1080.6	8	1×10^5	0	64	1.5	0.864	OS(F_2):64		64
7	8.36c	I.S.	2129.6	8	1×10^5	0	64	1.5	0		256	
8	8.36d	D.L.	1016.0	8	1×10^5	0	64	1.5	100	MC:256		
9	8.37a	D.L.	854.5	4	2×10^5	0	64	2.5	100	MC:256		
10	8.37b	HIB.	906.9	4	2×10^5	0	64	2.5	0.864	OS(F_2):64		64

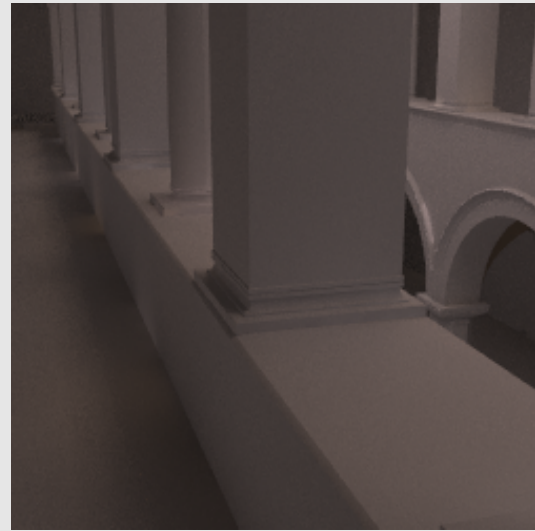


(a) MET=I.S.; Tab. 8.7:N=1; T=525.9s. (b) MET=HIB.; Tab. 8.7:N=2; T=462.8s.

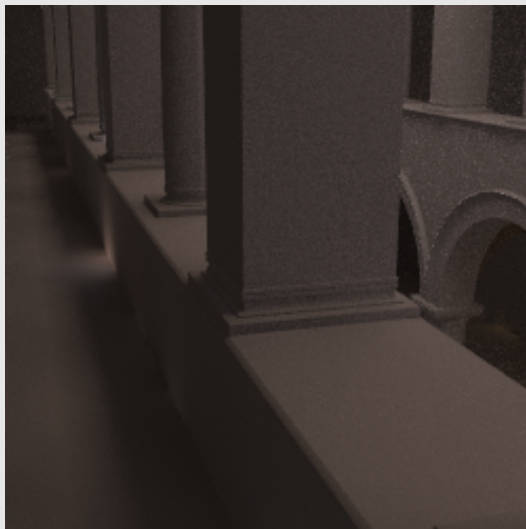


(c) MET=I.S.; Tab. 8.7:N=3; T=702.5s. (d) MET=HIB.; Tab. 8.7:N=4; T=530.9s

Figura 8.35: Modelo *Sponza 1* iluminado com o mapa *Galileo*, utilizando a técnica de mapeamento de fótons. Os parâmetros de renderização utilizados constam na tabela 8.7.



(a) MET=I.S.; Tab. 8.7:N=5 ; T=1379.4s. (b) MET=HIB.; Tab. 8.7:N=6; T=1080.6s.



(c) MET=I.S.; Tab. 8.7:N=7; T=2129.6s. (d) MET=D.L.; Tab. 8.7:N=8; T=1016s.

Figura 8.36: Modelo *Sponza 1* iluminado com o mapa *Galileo*, utilizando a técnica de mapeamento de fótons. Os parâmetros de renderização utilizados constam na tabela 8.7.

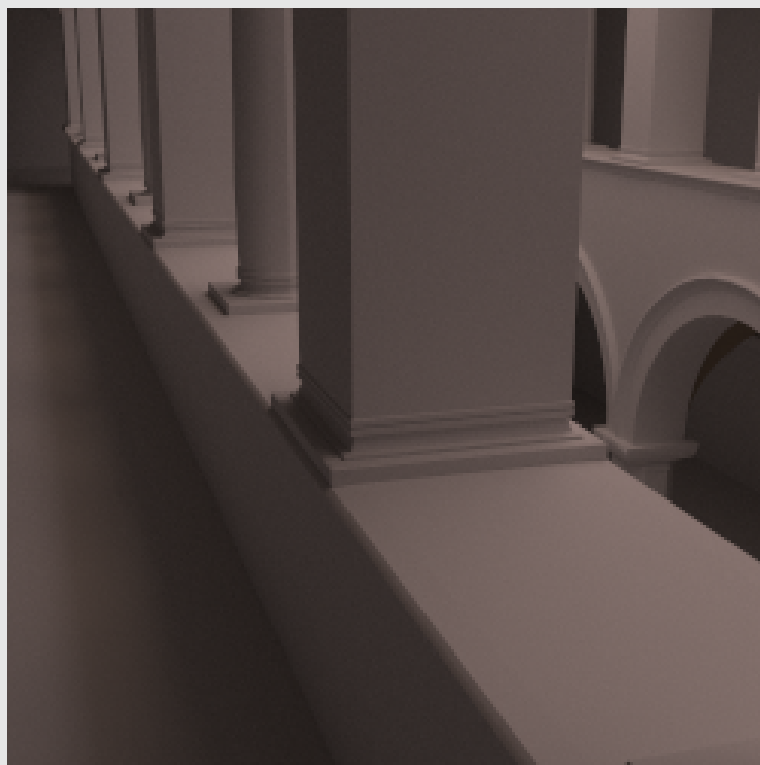


(a) MET=D.L.; Tab. 8.7:N=9; T=854.5s. (b) MET=HIB.; Tab. 8.7:N=10; T=906.9s.

Figura 8.37: Modelo *Sponza 1* iluminado com o mapa *Galileo*, utilizando a técnica de mapeamento de fótons. Os parâmetros de renderização utilizados constam na tabela 8.7.

Tabela 8.8: Comparação de renderizações do modelo *Sponza 2* iluminado com o mapa *Galileo*, utilizando mapeamento de fótons.

N	Fig.	MET.	Tempo Seg.	AA	Fótons		RF	DM	Área A (%)	Mapa A		Mapa B	
					Ind.	Ca.				DL	IS	MPA	
1	8.39a	HIB.	137.8	4	1×10^5	0	N	1.5	0.864	OS(F_2):64			32
2	8.39b	HIB.	823.4	4	1×10^5	0	64	1.5	0.864	OS(F_2):64			32
3	8.39c	I.S.	197.3	4	1×10^5	0	N	1.5	0			64	
4	8.39d	I.S.	946.6	4	1×10^5	0	64	1.5	0			64	
5	8.40a	D.L.	173.3	4	1×10^5	0	N	1.5	100	MC:256			
6	8.40b	D.L.	889.7	4	1×10^5	0	64	1.5	100	MC:256			
7	8.40c	I.S.	380.9	4	1×10^5	0	N	1.5	0			128	
8	8.40d	I.S.	1142.5	4	1×10^5	0	64	1.5	0			128	
9	8.41a	HIB.	89.3	4	1×10^5	0	N	1.5	0.864	OS(F_2):64			16
10	8.41b	HIB.	782.6	4	1×10^5	0	64	1.5	0.864	OS(F_2):64			16
11	8.41c	HIB.	199.8	4	1×10^5	0	N	1.5	0.864	OS(F_2):64			64
12	8.41d	HIB.	910.1	4	1×10^5	0	64	1.5	0.864	OS(F_2):64			64



(a) *Sponza 1* ; T= 283963s.



(b) *Sponza 2* ; T= 275095s.

Figura 8.38: Modelos *Sponza 1* e *Sponza 2* iluminados com o mapa *Galileo*, utilizando a técnica de Traçado de Caminhos e 32768 amostras por pixel.



(a) MET=HIB.; Tab. 8.8:N=1; T=137.8s. (b) MET=HIB.; Tab. 8.8:N=2; T=823.4s.



(c) MET=I.S.; Tab. 8.8:N=3; T=197.3s. (d) MET=I.S.; Tab. 8.8:N=4; T=946.6s.

Figura 8.39: Modelo *Sponza 2* iluminado com o mapa *Galileo*, utilizando a técnica de mapeamento de fótons. **Esquerda:** sem reunião final, **Direita:** com reunião final. Os parâmetros de renderização utilizados constam na tabela 8.8.



(a) MET=D.L.; Tab. 8.8:N=5; T=173.3s. (b) MET=D.L.; Tab. 8.8:N=6; T=889.7s.



(c) MET=I.S.; Tab. 8.8:N=7; T=380.9s. (d) MET=I.S.; Tab. 8.8:N=8; T=1142.5s.

Figura 8.40: Modelo *Sponza 2* iluminado com o mapa *Galileo*, utilizando a técnica de mapeamento de fótons. **Esquerda:** sem reunião final, **Direita:** com reunião final. Os parâmetros de renderização utilizados constam na tabela 8.8.



(a) MET=HIB.; Tab. 8.8:N=9; T=89.3s. (b) MET=HIB.; Tab. 8.8:N=10; T=782.6s.

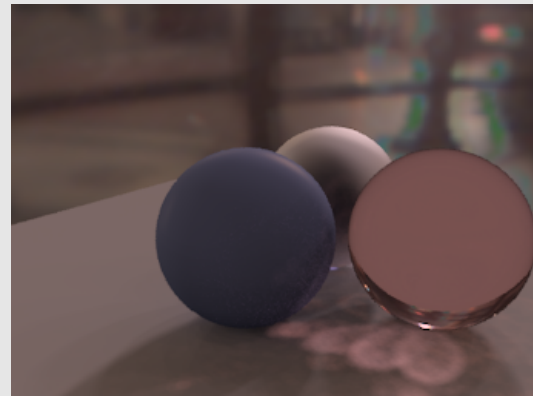
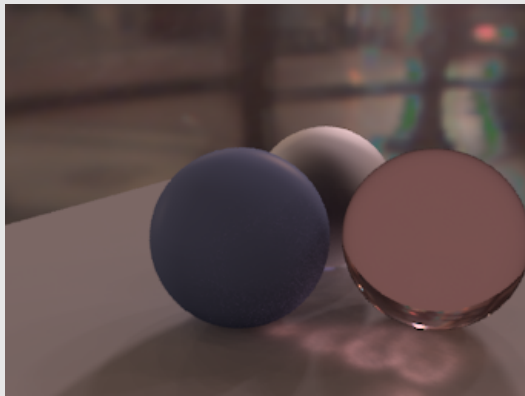


(c) MET=HIB.; Tab. 8.8:N=11; T=199.8s. (d) MET=HIB.; Tab. 8.8:N=12; T=910.1s.

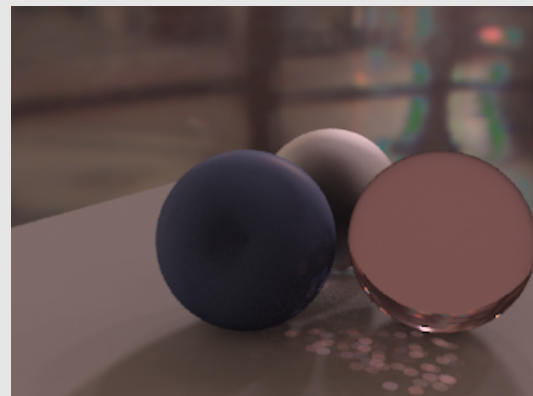
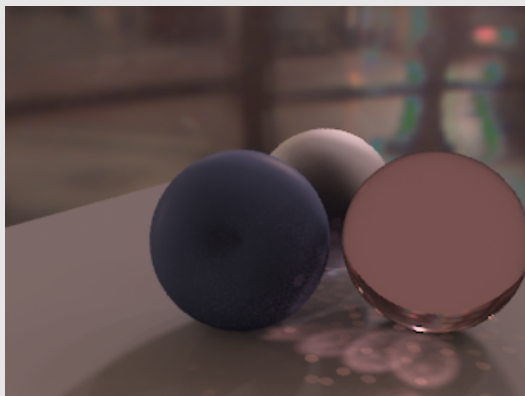
Figura 8.41: Modelo *Sponza 2* iluminado com o mapa *Galileo*, utilizando a técnica de mapeamento de fótons. **Esquerda:** sem reunião final, **Direita:** com reunião final. Os parâmetros de renderização utilizados constam na tabela 8.8.

Tabela 8.9: Comparação de renderizações do modelo *3 esferas* iluminado com o mapa *Galileo*, utilizando mapeamento de fótons.

N	Fig.	MET.	Tempo Seg.	AA	Fótons		RF	DM	Área A (%)	Mapa A		Mapa B	
					Ind.	Ca.				D.L.	I.S.	MPA	
1	8.43a	HIB.	768.7	4	10^5	10^5	100	50	0.864	OS(F_2):64			32
2	8.43b	HIB.	794.8	4	10^5	10^5	100	50	0.864	OS(F_2):64			64
3	8.43c	I.S.	853.4	4	10^5	10^5	100	50	0		128		
4	8.43d	D.L.	735.3	4	10^5	10^5	100	50	100	MC:256			
5	8.43e	HIB.	673.6	4	10^5	20000	100	50	0.864	OS(F_2):64			64
6	8.43f	I.S.	801.6	4	10^5	20000	100	50	0		128		
7	8.42a	D.L.	680.5	4	10^5	20000	100	50	100	MC:256			
8	8.42b	D.L.	445.6	4	10^5	20000	100	5	100	MC:256			
9	8.42c	HIB.	435.5	4	10^5	20000	100	5	0.864	OS(F_2):64			64
10	8.42d	I.S.	571.6	4	10^5	20000	100	5	0		128		

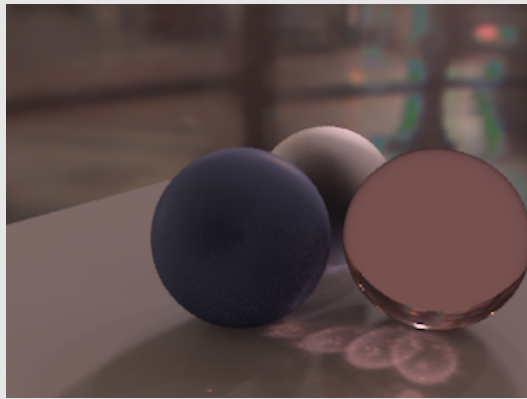


(a) MET=D.L.; Tab. 8.9:N=7; T=680.5s. (b) MET=D.L.; Tab. 8.9:N=8; T=445.6s.

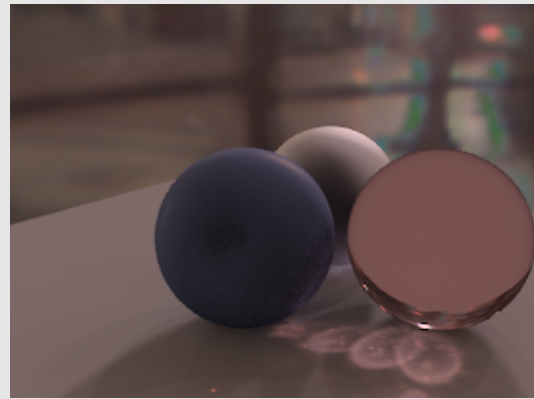


(c) MET=HIB.; Tab. 8.9:N=9; T=435.5s. (d) MET=I.S.; Tab. 8.9:N=10; T=571.6s.

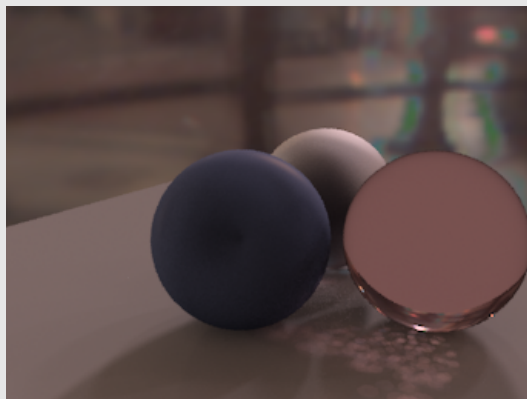
Figura 8.42: Modelo *3 esferas* iluminado com o mapa *Galileo*, utilizando a técnica de mapeamento de fótons. Os parâmetros de renderização utilizados constam na tabela 8.9.



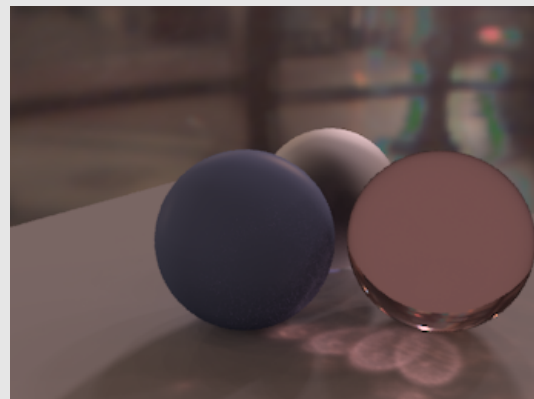
(a) MET=HIB.; Tab. 8.9:N=1; T=768.7s.



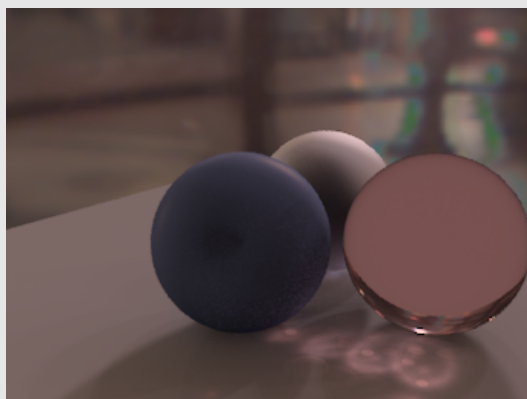
(b) MET=HIB.; Tab. 8.9:N=2; T=794.8s.



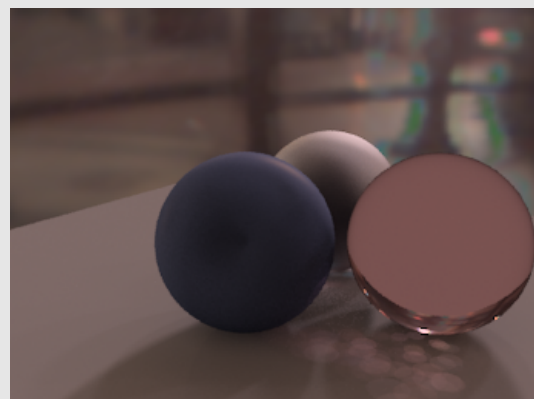
(c) MET=I.S.; Tab. 8.9:N=3; T=853.4s.



(d) MET=D.L.; Tab. 8.9:N=4; T=735.3s.



(e) MET=HIB.; Tab. 8.9:N=5; T=673.6s.



(f) MET=I.S.; Tab. 8.9:N=6; T=801.6s.

Figura 8.43: Modelo 3 esferas iluminado com o mapa *Galileo*, utilizando a técnica de mapeamento de fótons. Os parâmetros de renderização utilizados constam na tabela 8.9.

8.3 Conclusões

Propusemos uma abordagem alternativa para o tratamento de problemas de renderização foto-realista baseados em iluminação real que oferece ao usuário liberdade de escolha dos parâmetros para ter um controle mais fino do resultado da renderização. O *Esquema Híbrido* apresentado neste trabalho utiliza o método de amostragem hierárquica por importância, desenvolvido por Ostromoukhov et al. [ODJ04], e o método de amostragem por Corte Mediano, apresentado por Paul Debevec [DEB05].

Na seção 8.2 realizamos comparações dos métodos tradicionais, de geração de luzes direcionais a partir da amostragem do mapa, e amostragem direta por importância, com diversas configurações do *Esquema Híbrido* proposto. Em todas as situações o esquema híbrido foi igual ou mais eficiente que as técnicas tradicionais apresentadas. O grau de eficiência varia com a escolha dos parâmetros, o modelo e o mapa utilizado.

Para conseguir um *Esquema Híbrido* eficiente tivemos que superar vários obstáculos, entre eles:

- Conseguir uma boa aproximação das regiões brilhantes do mapa utilizando poucas luzes direcionais. Para isto foi implementado um esquema de clusterização de amostras para evitar o aglutinamento natural de amostras derivado dos métodos de amostragem por importância.
- Obter distribuições de amostras e luzes direcionais no plano latitude-longitude, que fossem boas na representação esférica do mapa. A estes efeitos implementamos o cálculo e células de Voronoi na esfera para delimitar as regiões do mapa que contribuiriam com cada luz direcional.

Incluímos também uma ferramenta de estimativa de parâmetros que o usuário pode utilizar, tais como o cálculo automático de limiar de variação máxima, que permite separar o mapa de iluminação de forma automatizada, e tem fornecido excelentes resultados nas renderizações. A possibilidade da determinação automática de um bom limiar de separação permite maior autonomia ao *Esquema Híbrido* fornecendo também um bom ponto de partida para os ajustes por parte do usuário.

A proposta híbrida fornece resultados competitivos para o cálculo de iluminação direta, reduzindo o tempo de renderização em até 80% em certos casos. Também obtivemos resultados significativos para os principais métodos de iluminação global com eficiência de até 50% no tempo de renderização com o método de *Traçado de Caminhos* e 75% com

Mapeamento de Fótons. A fim de implementar eficientemente o algoritmo de *Traçado de Caminhos*, tivemos que reescrever o integrador para comportar o *Esquema Híbrido*. No caso de *Mapeamento de Fótons* a integração é realizada aproveitando o integrador original do *PBRT*.

Em síntese o *Esquema Híbrido* fornece ao usuário, além da configuração automática, configurações manuais ajustáveis, mantendo um bom desempenho tanto em renderizações de iluminação direta quanto em métodos de iluminação global.

8.4 Trabalhos futuros

- Trabalhar num método para determinar automaticamente o fator *MAG* do método de amostragem hierárquica com mosaicos de Penrose a partir do número de amostras desejadas, a fim de automatizar completamente o processo de criação de mapas de pré-amostragem e a geração de luzes direcionais.
- Estender a idéia de amostragem híbrida para outros métodos de amostragem de mapas, como por exemplo o método apresentado por Ostromoukhov em [OST07].
- Implementar eficientemente o cálculo das regiões de Voronoi na esfera. Pretendemos implementar o método descrito em [FSH08], aproveitando a estrutura do *CAL*.
- Pesquisar outros métodos de clusterização de amostras, que possam melhorar o desempenho.
- Implementar um integrador para *Traçado de Caminhos* mais eficiente, explorando melhor o esquema de geração de amostras do *PBRT*.
- Implementar o *Esquema Híbrido* com outros integradores de iluminação global, tais como: *IGI* (Iluminação Global Interativa) [WKB⁺02], e *Cache de Irradiância* [WAR94].
- Trabalhar num *Esquema Híbrido* que possa ser aplicado para obter amostras do produto $f(\mathbf{p}, \boldsymbol{\omega}_o, \boldsymbol{\omega}_i)L_i(\mathbf{p}, \boldsymbol{\omega}_i)$ em lugar de amostrar somente o termo correspondente à radiância das fontes luminosas $L_i(\mathbf{p}, \boldsymbol{\omega}_i)$ como foi feito nesta dissertação.

Parte IV

Apêndice



Figura 8.44: Mapa *Galileo*. Luminância: Máx.= 768.10449, Min.= 0.038616.



Figura 8.45: Mapa *Grace*. Luminância: Máx.=1665.74231, Min.=0.0051595.

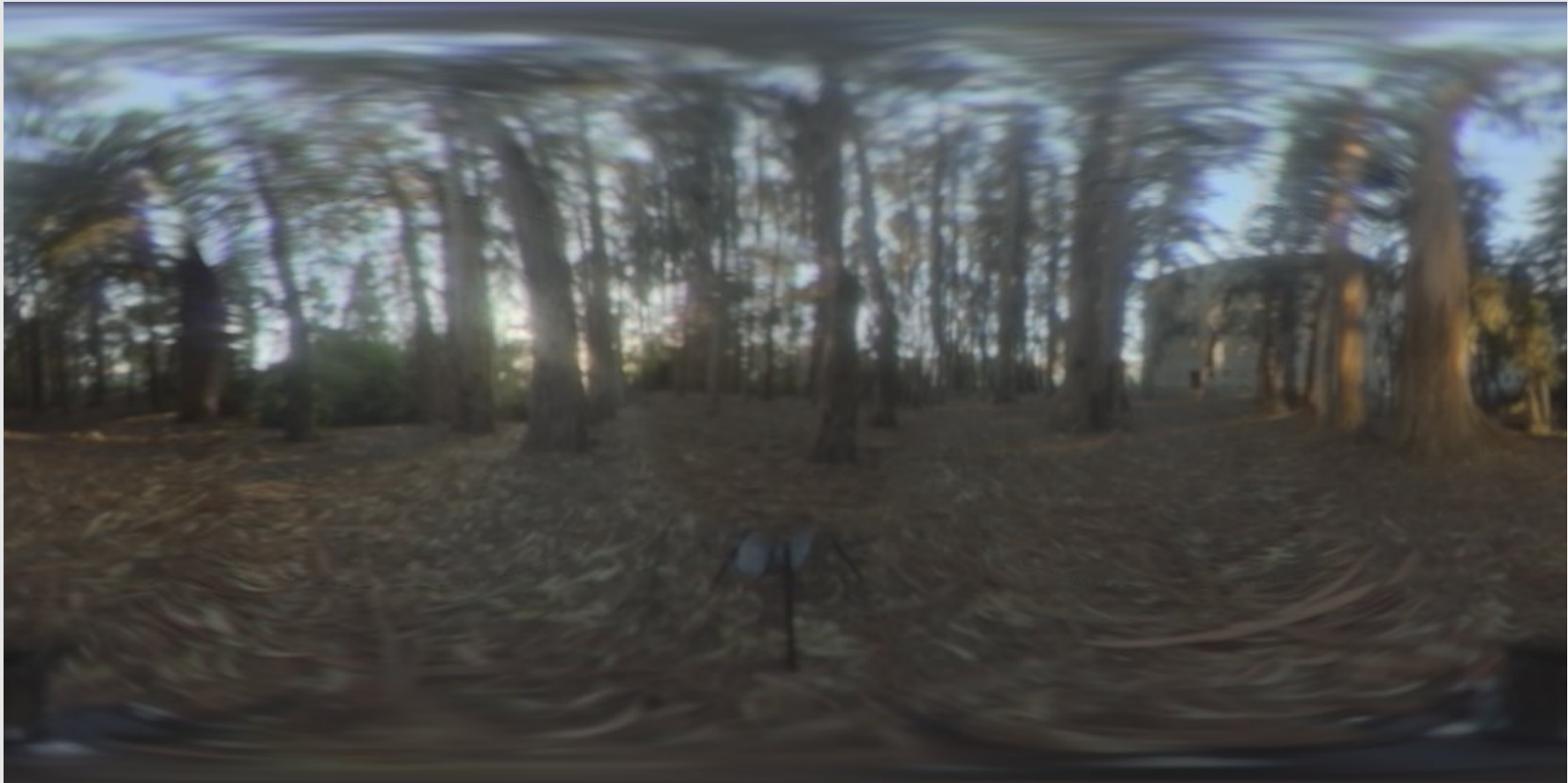


Figura 8.46: Mapa *Rnl*. Luminância: Máx.= 153.91383, Min.= 0.0263906.



Figura 8.47: Mapa *Kitchen*. Luminância: Máx.= 54.999947, Min.= 0.0216593.

Referências Bibliográficas

- [APP68] A. APPEL. Some techniques for shading machine renderings of solids. In *AFIPS Spring Joint Computing Conference*, volume 32 of *AFIPS Conference Proceedings*, pages 37–45. Thomson Book Company, Washington D.C., 1968.
- [ARBJ03] S. AGARWAL, R. RAMAMOORTHI, S. BELONGIE, and H. W. JENSEN. Structured importance sampling of environment maps. *ACM Trans. Graph.*, 22(3):605–612, 2003.
- [BGH05] D. BURKE, A. GHOSH, and W. HEIDRICH. Bidirectional importance sampling for direct illumination. In O. Deussen, A. Keller, K. Bala, P. Dutré, D. W. Fellner, and S. N. Spencer, editors, *Rendering Techniques*, pages 147–156. Eurographics Association, 2005.
- [CD01] J. COEN and P. DEBEVEC. Lightgen, hdrshop plugin. <http://www.ict.usc.edu/~jcohen/lightgen/lightgen.html>, 2001.
- [CETC06] D. CLINE, P. K. EGBERT, J. F. TALBOT, and D. L. CARDON. Two stage importance sampling for direct lighting. In *Rendering Techniques 2006 (Eurographics Symposium on Rendering)*, pages 103–113, June 2006.
- [DEB98] P. E. DEBEVEC. Rendering synthetic objects into real scenes: Bridging traditional and image-based graphics with global illumination and high dynamic range photography. In *SIGGRAPH*, pages 189–198, 1998.
- [DEB05] P. E. DEBEVEC. A median cut algorithm for light probe sampling. *SIGGRAPH Poster*, 2005.
- [DM97] P. E. DEBEVEC and J. MALIK. Recovering high dynamic range radiance maps from photographs. In *SIGGRAPH*, pages 369–378, 1997.
- [ES86] H. EDELSBRUNNER and R. SEIDEL. Voronoi diagrams and arrangements. *Discrete Comput. Geom.*, 1(1):25–44, 1986.
- [FER01] J. A. FERWERDA. Elements of early vision for computer graphics (tutorial). *IEEE Computer Graphics and Applications*, 21(5):22–33, 2001.
- [FSH08] E. FOGEL, O. SETTER, and D. HALPERIN. Exact implementation of arrangements of geodesic arcs on the sphere with applications. In *Abstracts of 24th European Workshop on Computational Geometry*, pages 83–86, 2008.
- [GDH06] A. GHOSH, A. DOUCET, and W. HEIDRICH. Sequential sampling for dynamic environment maps. In *SIGGRAPH '06: ACM SIGGRAPH 2006 Sketches*, page 157, New York, NY, USA, 2006. ACM.

- [GRE86] N. GREENE. Environment mapping and other applications of world projections. *IEEE Comput. Graph. Appl.*, 6(11):21–29, 1986.
- [HAL04] D. HALPERIN. Arrangements. In Jacob E. Goodman and Joseph O’Rourke, editors, *Handbook of Discrete and Computational Geometry*, chapter 24, pages 529–562. CRC Press LLC, Boca Raton, FL, 2004.
- [HSS09] D. HALPERIN, O. SETTER, and M. SHARIR. Constructing two-dimensional Voronoi diagrams via divide-and-conquer of envelopes in space. In *In Proceedings of the 6th annual International Symposium on Voronoi Diagrams in Science and Engineering (ISVD)*, pages 43–52, 2009.
- [ICG86] D. S. IMMEL, M. F. Cohen, and D. P. Greenberg. A radiosity method for non-diffuse environments. In *SIGGRAPH ’86: Proceedings of the 13th annual conference on Computer graphics and interactive techniques*, pages 133–142, New York, NY, USA, 1986. ACM.
- [JEN96] H. W. JENSEN. Global illumination using photon maps. In Xavier Pueyo and Peter Schröder, editors, *Rendering Techniques*, pages 21–30. Springer, 1996.
- [JEN01] H. W. JENSEN. *Realistic image synthesis using photon mapping*. A. K. Peters, Ltd., Natick, MA, USA, 2001.
- [KAJ86] J. T. KAJIYA. The rendering equation. In David C. Evans and Russell J. Athay, editors, *SIGGRAPH*, pages 143–150. ACM, 1986.
- [KK03] T. KOLLIG and A. KELLER. Efficient illumination by high dynamic range images. In Per H. Christensen, Daniel Cohen-Or, and Stephen N. Spencer, editors, *Rendering Techniques*, volume 44 of *ACM International Conference Proceeding Series*, pages 45–51. Eurographics Association, 2003.
- [KNU97] D. E. KNUTH. *The art of computer programming, volume 1 (3rd ed.): fundamental algorithms*. Addison Wesley Longman Publishing Co., Inc., Redwood City, CA, USA, 1997.
- [MEY06] M. MEYEROVITCH. Robust, generic and efficient construction of envelopes of surfaces in three-dimensional spaces. In Yossi Azar and Thomas Erlebach, editors, *ESA*, volume 4168 of *Lecture Notes in Computer Science*, pages 792–803. Springer, 2006.
- [MH84] G. S. MILLER and C. R. HOFFMAN. Illumination and reflection maps: Simulated objects in simulated and real environments. *Curse Notes for Advanced Computer Graphics Animation*, SIGGRAPH, 1984.
- [MJH06] X. MEI, M. JAEGER, and B. HU. An effective stratified sampling scheme for environment maps with median cut method. In *CGIV*, pages 384–389. IEEE Computer Society, 2006.
- [MWZ08] M. MEYEROVITCH, R. WEIN, and B. ZUKERMAN. 3d envelopes. In CGAL Editorial Board, editor, *CGAL User and Reference Manual*. 3.4 edition, 2008. Chapter 28.

- [ODJ04] V. OSTROMOUKHOV, C. DONOHUE, and P. JODOIN. Fast hierarchical importance sampling with blue noise properties. *ACM Trans. Graph.*, 23(3):488–495, 2004.
- [OST07] V. OSTROMOUKHOV. Sampling with polyominoes. *ACM Trans. Graph.*, 26(3):78, 2007.
- [PH04a] M. PHARR and G. HUMPHREYS. Infinite area light source with importance sampling., 2004. <http://www.pbrt.org/plugins/infinitesample.pdf>.
- [PH04b] M. PHARR and G. HUMPHREYS. *Physically Based Rendering: From theory to Implementation*. Morgan Kaufmann Publishers Inc., July 2004.
- [ROS03] S. M. ROSS. *Introduction to Probability Models, Eighth Edition*. Academic Press, January 2003.
- [SW90] L. SPILLMANN and J. S. WERNER. (eds.). *Visual Perception: The Neurological Foundations*. Academic Press, San Diego, 1990.
- [VGS97] L. VELHO, J. GOMES, and M. V. R. SOBREIRO. Color image quantization by pairwise clustering. In *In L. H. de Figueiredo and M. L. Netto, editors, X brazilian symposium of computer graphics and image processing*, pages 203–210. IEEE Computer Society, Los Alamitos, CA, 1997.
- [WAR94] G. J. WARD. The radiance lighting simulation and rendering system. In *SIGGRAPH*, pages 459–472. ACM, 1994.
- [WHI80] T. WHITTED. An improved illumination model for shaded display. *Commun. ACM*, 23(6):343–349, 1980.
- [WIL83] L. WILLIAMS. Pyramidal parametrics. In *SIGGRAPH '83: Proceedings of the 10th annual conference on Computer graphics and interactive techniques*, pages 1–11, New York, NY, USA, 1983. ACM.
- [WKB⁺02] I. WALD, T. KOLLIG, C. BENTHIN, A. KELLER, and P. SLUSALLEK. Interactive Global Illumination using Fast Ray Tracing. In *Proceedings of the 13th EUROGRAPHICS Workshop on Rendering*, pages 15–24. Saarland University, Kaiserslautern University, 2002.