



UFC

**UNIVERSIDADE FEDERAL DO CEARÁ
DEPARTAMENTO DE COMPUTAÇÃO
CURSO DE CIÊNCIA DA COMPUTAÇÃO**

JONAS DEYSON BRITO DOS SANTOS

**RENDERIZAÇÃO COM AMOSTRAGEM ADAPTATIVA NO
DOMÍNIO N-DIMENSIONAL**

FORTALEZA, CEARÁ

2013

JONAS DEYSON BRITO DOS SANTOS

**RENDERIZAÇÃO COM AMOSTRAGEM ADAPTATIVA NO
DOMÍNIO N-DIMENSIONAL**

Dissertação submetida à Coordenação do Curso de Pós-Graduação em Ciência da Computação da Universidade Federal do Ceará, como requisito parcial para a obtenção do grau de Mestre em Ciência da Computação.

Área de concentração: Computação Gráfica

Orientador: Prof. Dr. Joaquim Bento Cavalcante Neto

Coorientador: Prof. Dr. Creto Augusto Vidal

FORTALEZA, CEARÁ

2013

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Biblioteca de Ciências e Tecnologia

-
- S235r Santos, Jonas Deyson Brito dos.
Renderização com amostragem adaptiva no domínio N – dimensional. / Jonas Deyson Brito dos Santos. – 2013.
70f. : il. color., enc. ; 30 cm.
- Dissertação (mestrado) – Universidade Federal do Ceará, Centro de Ciências, Departamento de Computação, Programa de Pós-Graduação em Ciência da Computação, Fortaleza, 2013.
Área de Concentração: Computação Gráfica.
Orientação: Prof. Dr. Joaquim Bento Cavalcante.
Coorientação: Prof. Dr. Creto Augusto Vidal.
1. Iluminação de cena. 2. Monte Carlo, Método de. 3. Renderização. I. Título.

JONAS DEYSON BRITO DOS SANTOS

**RENDERIZAÇÃO COM AMOSTRAGEM ADAPTATIVA NO
DOMÍNIO N-DIMENSIONAL**

Dissertação submetida à Coordenação do Curso de Pós-Graduação em Ciência da Computação, da Universidade Federal do Ceará, como requisito parcial para a obtenção do grau de Mestre em Ciência da Computação. Área de concentração: Computação Gráfica

Aprovada em: __/__/____

BANCA EXAMINADORA

Prof. Dr. Joaquim Bento Cavalcante Neto
Universidade Federal do Ceará (UFC)
Orientador

Prof. Dr. Creto Augusto Vidal
Universidade Federal do Ceará (UFC)
Coorientador

Prof. Dr. Manuel Menezes de Oliveira Neto
Universidade Federal do Rio Grande do Sul
(UFRGS)

RESUMO

Este trabalho propõe melhorias em uma técnica de amostragem adaptativa multidimensional para renderização. Renderização é o processo de síntese de imagens por meio de algoritmos que simulam a iluminação em cenários virtuais. As técnicas mais gerais de renderização fotorrealística — aquelas que procuram obter imagens que se assemelham a fotografias — utilizam métodos de integração baseados em Monte Carlo para resolver a equação que descreve a distribuição de luz na cena (equação de renderização). Por ser um método probabilístico e utilizar amostras geradas randomicamente, Monte Carlo produz ruído na imagem final — resultado da variância das amostras — e portanto, pode necessitar de uma grande quantidade de amostras para que o ruído diminua a níveis aceitáveis. Com o intuito de se obter imagens de melhor qualidade com uma menor quantidade de amostras, foram propostas técnicas de amostragem adaptativa que visam concentrar o esforço de amostragem em regiões mais importantes da cena. Neste trabalho, propõe-se a modificação de uma técnica de amostragem adaptativa multidimensional por meio da adição de duas etapas: substituição de amostras e integração auxiliar. Essas etapas visam dar mais robustez à técnica, possibilitando sua utilização em uma maior variedade de situações. Além da adição de duas etapas, também propõe-se uma técnica de reconstrução mais eficiente na etapa final.

Palavras-chave: Renderização. Amostragem adaptativa. Amostragem e reconstrução.

ABSTRACT

This work proposes improvements in a multidimensional adaptive sampling technique for rendering. Rendering is the process of synthesizing images by algorithms simulating lighting in virtual scenes. The more general techniques of photorealistic rendering — those seeking images that resemble photographs — use integration methods based on Monte Carlo to solve the equation that describes the distribution of light in the scene (rendering equation). Being a probabilistic method which uses randomly generated samples, Monte Carlo produces noise in the final image — result of samples' variance — and therefore may require a large amount of samples to reduce the noise to acceptable levels. To obtain images of better quality with a lower number of samples, adaptive sampling techniques were proposed, concentrating sampling effort in the most important regions. In this work, we propose the addition of two steps to a multidimensional adaptive sampling technique: substitution of samples and auxiliary integration. These steps aim to give more strength to the technique, enabling their use in a wider variety of situations.

Keywords: Rendering. Adaptive sampling. Sampling and reconstruction.

LISTA DE FIGURAS

Figura 1.1	Renderização em tempo real. http://unigine.com/	13
Figura 1.2	Renderização <i>offline</i> . http://www.sintel.org/	14
Figura 1.3	Ruido em imagem renderizada. Típica manifestação da variância do método de integração utilizado	14
Figura 1.4	Regiões com mais ruído que outras.	15
Figura 2.1	Variáveis utilizadas na Equação de Renderização.	18
Figura 2.2	Variáveis utilizadas na segunda versão da Equação de Renderização.	18
Figura 2.3	Variáveis utilizadas na definição do fator de forma.	19
Figura 2.4	Amostragem de Importância: A linha vermelha representa a PDF usada para gerar amostras (pontos azuis). (a) Distribuição uniforme: regiões de maior importância da função são pobremente amostradas. (b) Distribuição proporcional a $f(x)$: a densidade de amostras em cada região do domínio é proporcional à importância da função na região.	21
Figura 3.1	Visão geral das etapas da técnica MDAS. (Imagem traduzida de Hachisuka et al. (2008))	31
Figura 3.2	Visão geral do processo de amostragem adaptativa da técnica MDAS. (Imagem traduzida de Hachisuka et al. (2008))	32
Figura 3.3	Visão geral da etapa de reconstrução da técnica MDAS. (Imagem traduzida de Hachisuka et al. (2008))	33
Figura 4.1	Visão geral da técnica proposta.	36
Figura 4.2	Divisão de uma célula da kd-tree (caso 2D): (a) Comparando-se as dimensões da célula, verifica-se que $\Delta_2 > \Delta_1$, portanto, escolhe-se esse eixo para subdivi-	

	são. (b) Comparando-se as coordenadas de cada amostra com relação ao eixo escolhido, encontra-se a mediana m . (c) Como ponto de divisão, escolhe-se a média entre a amostra mediana (v_m) e a amostra adjacente de maior contraste com relação à mediana (v_l).	36
Figura 4.3	Etapa de substituição de amostras iniciais: (a) Distribuição inicial de amostras — apenas uma célula (em destaque) é candidata a subdivisão. (b) Nova distribuição inicial após algumas substituições — captura-se mais informação da função subjacente, melhorando-se o processo de subdivisão adaptativa.	37
Figura 4.4	Integração Auxiliar: (a) Amostragem adaptativa pode não contabilizar certas características da função de cena. (b) Integração auxiliar permite melhorar as estimativas utilizando mais amostras em todas as regiões (cruzes vermelhas), sem necessidade de refinar a kd-tree.	40
Figura 5.1	Imagem de referência da cena “esferas multidimensionais”.	44
Figura 5.2	Cana “Fendas”: (a) Configuração da cena. (b) Imagem de referência.	44
Figura 5.3	Imagem de referência da cena “macaco”.	45
Figura 5.4	Imagem de referência da cena <i>Cornell box</i>	45
Figura 5.5	Imagem de referência da cena “ <i>Cornell box</i> com esfera”.	46
Figura 5.6	Substituição de amostras na cena “Fatia no hipercubo” com $k = n$: (a) MDAS comum. (b) Substituição de amostras com uma amostra por pixel em média (1 app). (c) Substituição com 5 app. (d) Substituição com 10 app. (e) Substituição com 100 app.	47
Figura 5.7	Substituição de amostras na cena “Fatia no hipercubo” com $k < n$: (a) MDAS comum. (b) Substituição de amostras com uma amostra por pixel em média (1 app). (c) Substituição com 5 app. (d) Substituição com 10 app. (e) Substituição com 15 app. (f) 100 app.	48
Figura 5.8	Substituição de amostras na cena “Fatia no hipercubo”: (a) Taxa de aceitação das amostras de substituição. (b) Erro quadrático médio.	49
Figura 5.9	Substituição de amostras na cena “Esferas” com $k = n$: (a) MDAS comum. (b)	

Substituição com 1 app. (c) Substituição com 5 app. (d) Substituição com 10 app. (e) Substituição com 15 app. (f) Substituição com 100 app.	50
Figura 5.10 Substituição de amostras na cena “Esferas” com $k = n$: (a) Taxa de aceitação das amostras de substituição. (b) Erro quadrático médio (EQM) das imagens obtidas.	50
Figura 5.11 Substituição de amostras na cena “Esferas” com $k < n$: (a) MDAS comum. (b) Substituição com 1 app. (c) Substituição com 5 app. (d) Substituição com 10 app. (e) Substituição com 15 app. (f) Substituição com 100 app.	51
Figura 5.12 Substituição de amostras na cena “Esferas” com $k < n$: (a) Taxa de aceitação das amostras de substituição. (b) Erro quadrático médio (EQM) das imagens obtidas.	51
Figura 5.13 Substituição de amostras na cena “Fendas”: (a) MDAS comum. (b) Substituição com 10 app. (c) 20 app (d) 30 app. (e) 40 app. (f) 50 app.	52
Figura 5.14 Substituição de amostras na cena “Fendas”: (a) Taxa de aceitação das amostras de substituição. (b) Erro quadrático médio (EQM) das imagens obtidas. ...	53
Figura 5.15 Integração auxiliar uniforme na cena “Fatia no hipercubo” com $k = n$: (a) MDAS comum. (b) 5 app. (c) 10 app. (d) 15 app. (e) 20 app. (f) 100 app.	53
Figura 5.16 Erro quadrático médio (EQM) das imagens obtidas com integração auxiliar uniforme na cena “Fatia no hipercubo” com $k < n$	54
Figura 5.17 Integração auxiliar uniforme na cena “Fatia no hipercubo” com $k < n$: (a) MDAS comum. (b) 5 app. (c) 10 app. (d) 15 app. (e) 20 app. (f) 100 app.	55
Figura 5.18 Erro quadrático médio (EQM) das imagens obtidas com integração auxiliar uniforme na cena “Fatia no hipercubo” com $k < n$	55
Figura 5.19 Comparação entre as imagens obtidas da cena “Fatia no hipercubo” utilizando-se substituição de amostras (1), integração auxiliar uniforme (2) e ambos (3). Foram utilizados os seguintes quantidades de amostras na utilização em conjunto das duas técnicas: (a) 1 app (substituição) + 5 app (integração auxiliar). (b) 5 + 10 app. (c) 10 + 15 app. (d) 15 + 20 app. (e) 100 + 100 app.(1 + 5)	56

Figura 5.20 Integração auxiliar uniforme na cena “Esferas” com $k = n$: (a) MDAS comum. (b) 5 app. (c) 10 app. (d) 15 app. (e) 20 app. (f) 100 app.	56
Figura 5.21 Erro quadrático médio (EQM) das imagens obtidas com integração auxiliar uniforme na cena “Esferas” com $k = n$	57
Figura 5.22 Integração auxiliar uniforme na cena “Esferas” com $k < n$: (a) MDAS comum. (b) 5 app. (c) 10 app. (d) 15 app. (e) 20 app. (f) 100 app.	57
Figura 5.23 Erro quadrático médio (EQM) das imagens obtidas com integração auxiliar uniforme na cena “Esferas” com $k < n$	58
Figura 5.24 Integração auxiliar uniforme na cena “Fendas”: (a) MDAS comum. (b) 10 app. (c) 20 app. (d) 30 app. (e) 40 app. (f) 50 app.	59
Figura 5.25 Erro quadrático médio das imagens obtidas da cena “Fendas” com integração auxiliar uniforme.	59
Figura 5.26 Integração auxiliar uniforme na cena Cornell box: (a) MDAS comum. (b) 10 app. (c) 20 app. (d) 30 app. (e) 40 app. (f) 50 app.	60
Figura 5.27 Erro quadrático médio das imagens obtidas da cena Cornell box com integração auxiliar uniforme.	60
Figura 5.28 Integração auxiliar uniforme na cena “Cornell box com esfera”: (a) MDAS comum. (b) 10 app. (c) 20 app. (d) 30 app. (e) 40 app. (f) 50 app.	61
Figura 5.29 Erro quadrático médio das imagens obtidas da cena “Cornell box com esfera” com integração auxiliar uniforme.	61
Figura 5.30 Reconstrução por nó da cena “Cornell box”: (a) MDAS comum. (b) Reconstrução por nó. Houve redução de 39,8 % no tempo de reconstrução.	62
Figura 5.31 Reconstrução por nó da cena “Macaco”: (a) MDAS comum. (b) Reconstrução por nó. Houve redução de 72,3 % no tempo de reconstrução.	63

LISTA DE TABELAS

Tabela 3.1	Parâmetros da técnica MDAS	34
Tabela 5.1	Descrição dos parâmetros de MDAS com as modificações propostas	42

SUMÁRIO

1	INTRODUÇÃO	13
1.1	Motivação	13
1.2	Objetivos e Contribuições	15
1.3	Organização do Trabalho	15
2	BACKGROUND	16
2.1	Radiometria	16
2.1.1	Grandezas Principais	16
2.2	Equação de Renderização	17
2.3	Integração de Monte Carlo	18
2.3.1	Variância	19
2.3.2	Amostragem de Importância	20
2.4	Solução da Equação de Renderização por <i>Ray Tracing</i> e Monte Carlo	21
2.4.1	Monte Carlo <i>Ray Tracing</i>	22
2.4.2	<i>Path Tracing</i>	22
2.5	Amostragem e Reconstrução	24
2.5.1	Amostragem	24
2.5.2	Reconstrução	25
2.6	Considerações Finais	25
3	TRABALHOS RELACIONADOS	26
3.1	Adaptatividade no Plano da Imagem	26
3.2	Adaptatividade Orientada a Efeitos Específicos	27
3.3	Adaptatividade Multidimensional	28
3.4	<i>Metropolis Light Transport</i>	29
3.5	<i>Multidimensional Adaptive Sampling and Reconstruction</i>	31
3.5.1	Amostragem Adaptativa Multidimensional	32
3.5.2	Reconstrução	32
3.6	Considerações Finais	34

4	TÉCNICA PROPOSTA	35
4.1	Visão Geral	35
4.2	Amostragem Inicial	35
4.3	Substituição das Amostras Iniciais	37
4.4	Refinamento Adaptativo	38
4.5	Integração Auxiliar	38
4.5.1	Integração Auxiliar Uniforme	39
4.5.2	Integração Auxiliar por Metropolis	39
4.6	Reconstrução por nó	41
4.7	Considerações Finais	41
5	RESULTADOS	42
5.1	Considerações sobre implementação	42
5.2	Cenas	42
5.2.1	Fatia no hipercubo	43
5.2.2	Esferas	43
5.2.3	Fendas	44
5.2.4	Macaco	44
5.2.5	<i>Cornell box</i>	45
5.2.6	<i>Cornell box com esfera</i>	45
5.3	Substituição de amostras	46
5.3.1	Fatia no hipercubo	46
5.3.2	Esferas	49
5.3.3	Fendas	49
5.4	Integração Auxiliar Uniforme	52
5.4.1	Fatia	52
5.4.2	Esferas	54
5.4.3	Fendas	54
5.4.4	<i>Cornell box</i>	58
5.4.5	<i>Cornell box com esfera</i>	58
5.5	Reconstrução por nó	58

5.6	Considerações Finais.....	62
6	CONCLUSÃO E TRABALHOS FUTUROS	64
	Referências Bibliográficas	65

1 INTRODUÇÃO

1.1 Motivação

Em computação gráfica, renderização é o processo pelo qual gera-se uma imagem a partir da descrição de um cenário virtual. A descrição do cenário consiste basicamente na geometria tridimensional dos objetos que compõem a cena, os materiais que compõem esses objetos, as propriedades das fontes de luz responsáveis pela iluminação da cena e as propriedades do sensor (câmera) responsável pela captura da imagem.

Algumas técnicas de renderização privilegiam a eficiência, sendo utilizadas em aplicações em tempo real, como jogos (Figura 1.1). Apesar de cada vez mais realistas, essas técnicas favorecem a eficiência em detrimento do realismo, desconsiderando alguns fenômenos luminosos que contribuiriam para o realismo, mas que são muito caros computacionalmente para serem calculados em tempo real. Outras técnicas visam o fotorrealismo, considerando fenômenos que tornam a cena mais fisicamente realística — iluminação indireta ou global, efeitos de focagem de luz conhecidos como *caustics*, *depth of field*, entre outros — mesmo que tais efeitos acarretem maior custo computacional. Essas técnicas são utilizadas em aplicações *offline*, ou seja, cujos resultados não precisam ser gerados e exibidos em tempo real, como a renderização de filmes de animação por exemplo (Figura 1.2).



Figura 1.1: Renderização em tempo real. <http://unigine.com/>

No contexto de renderização fotorrealística baseada em princípios físicos, a formulação clássica do problema é representada pela equação de renderização (KAJIYA, 1986), que descreve o estado de equilíbrio da luz em uma cena. Essa equação serve de base para diversas técnicas como *Path-tracing* e *Bidirectional Path-tracing* (LAFORTUNE; WILLEMS, 1993), dentre outras (PHARR; HUMPHREYS, 2010; DUTRE et al., 2006).

Sendo uma equação integral de Fredholm do segundo tipo, a equação de *rendering* apresenta dois problemas: a impossibilidade de se obter uma solução analítica para o caso geral e a alta dimensionalidade do domínio de integração. Esses problemas tornam sua solução pelos métodos de integração numérica tradicionais muito ineficiente. Portanto, o método de



Figura 1.2: Renderização *offline*. <http://www.sintel.org/>

integração de Monte Carlo é o mais usado devido à sua simplicidade, generalidade e velocidade de convergência. Entretanto, devido à sua natureza estocástica, esse método produz ruído na imagem final (Figura 1.3) em decorrência da variância das estimativas, e exige uma grande quantidade de amostras para conseguir resultados de alta qualidade.

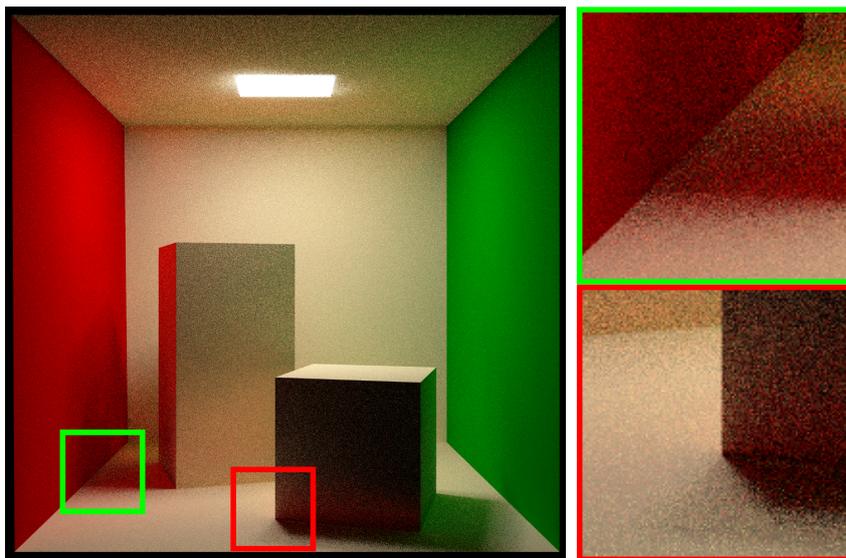


Figura 1.3: Ruído em imagem renderizada. Típica manifestação da variância do método de integração utilizado

Apesar de o ruído ser um problema inerente aos métodos probabilísticos utilizados, pode-se observar que o ruído não se distribui uniformemente pela imagem (Figura 1.4). Isso ocorre porque em certas regiões da cena, os caminhos de luz são mais difíceis de serem encontrados, resultando em maior variância nas estimativas. Além disso, nem todas as etapas dos métodos de renderização são probabilísticos, de modo que nem todos os tipos de efeitos estão sujeitos a ruído: uma cena apenas com iluminação direta e fontes de luz pontuais, por exemplo, pode ser renderizada sem a presença desse tipo de ruído.

Neste contexto, diversas técnicas de redução de ruído foram propostas, dentre elas, técnicas de amostragem adaptativa. Essas técnicas visam reduzir a quantidade de amostras necessárias para se obter uma imagem de boa qualidade, concentrando-as nas regiões onde são mais necessárias. Boa parte dessas técnicas possuem restrições quanto ao tipo de ruído tratado

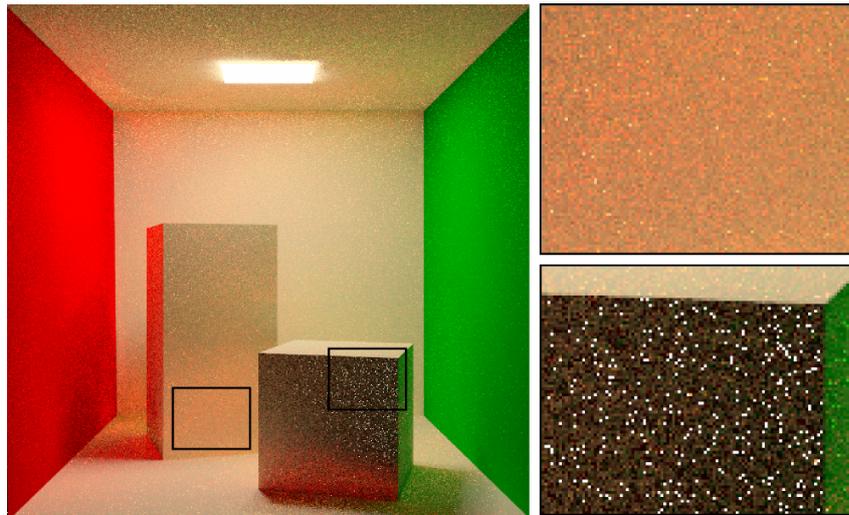


Figura 1.4: Regiões com mais ruído que outras.

(são orientadas a efeitos específicos), o que restringe seu uso e torna sua utilização, em conjunto, mais complexa.

Um técnica bastante geral quanto à sua aplicação é MDAS (*Multidimensional Adaptive Sampling and Reconstruction* (HACHISUKA et al., 2008)). Esta técnica possui uma formulação independente de efeitos específicos, possibilitando tratar uma maior gama de efeitos. Entretanto, a técnica exige que todas as amostras geradas sejam armazenadas em uma kd-tree, o que causa limitações práticas devido ao custo de manutenção dessa estrutura.

1.2 Objetivos e Contribuições

Este trabalho trata de amostragem adaptativa em renderização fisicamente realística, e visa prover uma visão geral da área, fornecendo uma classificação das técnicas quanto à sua generalidade. Além disso, são propostas algumas modificações que visam amenizar algumas das limitações da técnica MDAS. As modificações propostas consistem na utilização de um algoritmo de reconstrução mais eficiente e na adição de etapas extras de amostragem que não requerem o armazenamento de amostras: uma etapa de substituição de amostras, outra de integração auxiliar por meio de reamostragem no domínio fornecido pela kd-tree.

1.3 Organização do Trabalho

Este trabalho é organizado da seguinte forma: O Capítulo 2 apresenta alguns conceitos de renderização baseada em princípios físicos (*physically based rendering*) — radiometria, equação de renderização, técnicas de renderização baseadas em *ray tracing*, e conceitos de amostragem e reconstrução. Em seguida, no Capítulo 3, é apresentada uma visão geral das técnicas de amostragem adaptativa, além de uma descrição mais detalhada das técnicas que servem de base para as modificações propostas no Capítulo 4. A efetividade das modificações propostas é avaliada no Capítulo 5. Por fim, o Capítulo 6 apresenta algumas conclusões e possíveis trabalhos

futuros.

2 BACKGROUND

Neste capítulo é apresentada uma revisão dos principais conceitos que serão úteis no decorrer da dissertação. A Seção 2.1 apresenta as grandezas radiométricas utilizadas. A equação de renderização é apresentada na Seção 2.2. Em seguida, na Seção 2.4, são apresentados conceitos de *ray tracing* e solução da equação de renderização por Monte Carlo. Por fim, na Seção 2.5, são apresentados conceitos de amostragem e reconstrução. Como referência mais aprofundada sobre *physically based rendering* podem ser consultados Dutre et al. (2006) e Pharr e Humphreys (2010).

2.1 Radiometria

Este trabalho, assim como a maioria dos trabalhos que tratam de renderização baseada em física, utiliza um modelo radiométrico de propagação e reflexão de luz. Esse modelo age no nível de ótica geométrica, o que implica em uma série de suposições:

- *Linearidade*: O efeito de duas entradas no sistema é igual à soma dos efeitos das duas entradas individualmente.
- *Conservação de energia*: A quantidade de energia refletida (refratada) em uma superfície nunca é maior que a quantidade incidente.
- *Ausência de polarização*: Não é considerado polarização eletromagnética.
- *Ausência de fluorescência e fosforescência*: O comportamento da luz em uma comprimento de onda é completamente independente do comportamento da luz em outros comprimentos.
- *Estado de equilíbrio*: Considera-se que a luz encontra-se em um estado de equilíbrio no ambiente, ou seja, a distribuição de radiância não varia com o tempo.

Na prática, essas suposições não representam sérias deficiências, pois os efeitos visuais que decorrem desses fenômenos são geralmente sutis e ocorrem em situações bastante específicas.

2.1.1 Grandezas Principais

As principais grandezas radiométricas de interesse são:

Fluxo. O fluxo, também conhecido como fluxo radiante ou potência, é a energia total atravessando uma superfície ou região no espaço por unidade de tempo. Essa grandeza é medida em watts (W) e é denotada pelo símbolo Φ . O fluxo geralmente é usado para descrever a emissão das fontes de luz, e é dado por

$$\Phi = \frac{dQ}{dt}, \quad (2.1)$$

onde Q é a energia radiante.

Irradiância e Exitância Radiante. Irradiância (E) é a densidade de fluxo chegando em uma superfície (densidade por área). É dada por

$$E = \frac{d\Phi}{dA}. \quad (2.2)$$

Geralmente, quando o fluxo está deixando a superfície, utiliza-se a denominação *exitância radiante* (M), também conhecida como *radiosidade* (B).

Intensidade. A intensidade representa a densidade de fluxo por ângulo sólido. É dada por

$$I = \frac{d\Phi}{d\omega}. \quad (2.3)$$

Radiância. A radiância representa a densidade de fluxo por ângulo sólido por área. É dada por

$$L = \frac{d\Phi}{d\omega dA^\perp}, \quad (2.4)$$

onde dA^\perp é a área dA projetada em um plano imaginário perpendicular a ω . Radiância é a grandeza mais utilizada devido às suas propriedades: todas as outras grandezas podem ser calculadas a partir dela e a radiância permanece constante ao longo de um raio no espaço vazio. Portanto, a radiância é a grandeza mais próxima da noção de cor, sendo a mais adequada para se calcular em um sistema de *physically based rendering*.

2.2 Equação de Renderização

A equação de renderização (ou equação de transporte de luz) foi proposta por Kajiyama (1986) e representa o estado de equilíbrio da radiância na cena em termos da radiância emitida por certos objetos e as propriedades refletivas dos objetos:

$$L(p, \omega_o) = L_e(p, \omega_o) + \int_{\Omega} f_r(p, \omega_o, \omega_i) L(p, \omega_i) |\cos(\theta_i)| d\omega_i, \quad (2.5)$$

onde $L(p, \omega_o)$ é a radiância excitante no ponto p na direção ω_o , $L_e(p, \omega_o)$ é a radiância emitida pelo ponto p na direção ω_o (caso o ponto p pertença a uma fonte de luz), f_r é a BRDF (*bidirectional reflectance distribution function*) que representa as características do material no ponto p , θ_i é o ângulo entre a normal no ponto p e ω_i e $d\omega_i$ é o ângulo sólido diferencial. A Figura 2.1 esquematiza os vetores utilizados na equação.

Sendo uma equação integral de Fredholm do segundo tipo¹, a equação de renderização apresenta dois problemas: a impossibilidade de se obter uma solução analítica para o caso

¹ Apesar de ser comumente enquadrada nessa categoria, existem algumas particularidades (ARVO, 1995).

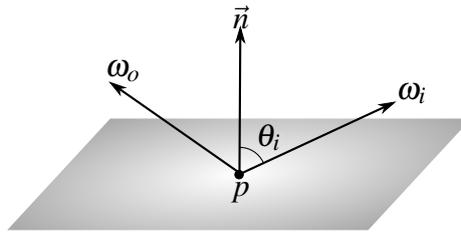


Figura 2.1: Variáveis utilizadas na Equação de Renderização.

geral e a alta dimensionalidade do domínio de integração. Esses problemas tornam sua solução pelos métodos de integração numérica tradicionais muito ineficiente. Portanto, o método de integração de Monte Carlo é o mais usado devido à sua simplicidade, generalidade e velocidade de convergência. Entretanto, devido à sua natureza estocástica, esse método produz ruído na imagem em decorrência da variância das estimativas, e exige uma grande quantidade de amostras para conseguir resultados de alta qualidade.

Uma outra forma de escrever a Equação 2.5 é substituir a integral em ângulo sólido por integral em área, resultando na seguinte equação:

$$L(p' \rightarrow p) = L_e(p' \rightarrow p) + \int_A f_r(p'' \rightarrow p' \rightarrow p) L(p'' \rightarrow p') G(p'' \leftrightarrow p') dp'', \quad (2.6)$$

onde p'' , p' e p são pontos na superfície dos objetos (Figura 2.2). Nessa versão, tem-se também o fator de forma G que reúne os termos de mudança de variável e o fator $|\cos \theta|$ da versão original:

$$G(p \leftrightarrow p') = V(p \leftrightarrow p') \frac{|\cos(\theta)| |\cos(\theta')|}{\|p - p'\|^2}, \quad (2.7)$$

onde V é o fator de oclusão ($V = 1$ se p e p' forem mutuamente visíveis e $V = 0$ caso contrário). A Figura 2.3 mostra os elementos utilizados na definição do fator de forma.

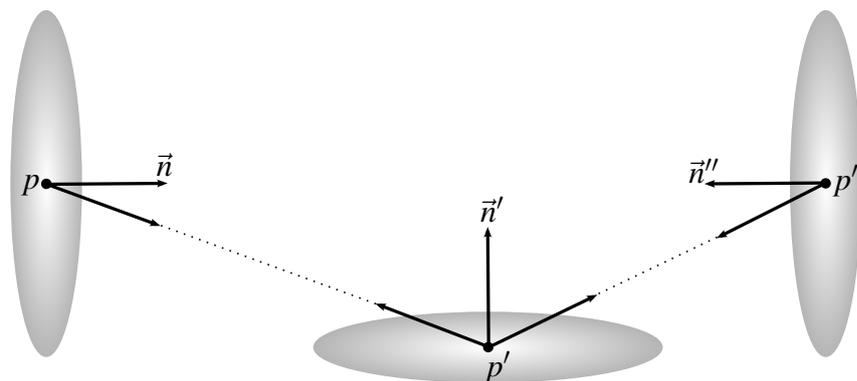


Figura 2.2: Variáveis utilizadas na segunda versão da Equação de Renderização.

2.3 Integração de Monte Carlo

Para estimar a integral da equação de renderização, o método de integração de Monte Carlo é o mais utilizado na prática devido à naturalidade com que trata a alta ordem do integrando,

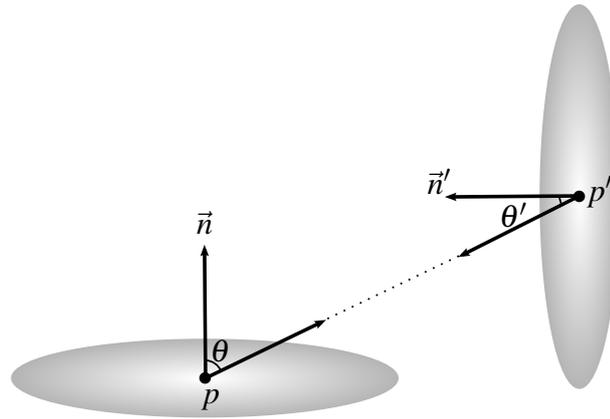


Figura 2.3: Variáveis utilizadas na definição do fator de forma.

à sua simplicidade e flexibilidade. Trata-se de um método estatístico que, dados um conjunto de pontos escolhidos aleatoriamente no domínio de integração, a distribuição usada para gerar esses pontos e o valor da função nesses pontos, é capaz de estimar corretamente (na média) a integral.

Da estatística, tem-se a seguinte definição de esperança de uma função $f(x)$ avaliada em pontos distribuídos segundo a PDF (*probability distribution function*) $p(x)$:

$$E_p[f(x)] = \int_D f(x)p(x) dx. \quad (2.8)$$

Uma aproximação para $E_p[f(x)]$ utilizando-se n pontos aleatórios gerados segundo a distribuição p é dada por

$$E_p[f(x)] = \int_D f(x)p(x) dx \approx \frac{1}{n} \sum_{i=1}^n f(x_i). \quad (2.9)$$

Observe que isso já é quase o que se quer (uma aproximação para uma integral), mas nesse caso o integrando é o produto de duas funções. Então, fazendo $fp = g$, tem-se

$$\int_D g(x) dx \approx \frac{1}{n} \sum_{i=1}^n \frac{g(x_i)}{p(x_i)}, \quad (2.10)$$

que é a estimativa de Monte Carlo para a integral de $g(x)$ no domínio D .

2.3.1 Variância

A solução aproximada dada pela Equação 2.10 é correta na média, mas apresenta uma certa variância que, em renderização, se manifesta como ruído na imagem gerada. Essa variância pode ser reduzida aumentando-se a quantidade de amostras, mas essa estratégia não é satisfatória pois sabe-se que a variância da estimativa de Monte Carlo cai numa taxa de $O(\sqrt{n})$ no número de amostras (VEACH, 1998), ou seja, é necessário quadruplicar a quantidade de amostras para que a variância caia pela metade.

2.3.2 Amostragem de Importância

Segundo a Equação 2.10, a estimativa de Monte Carlo para a integral de $f(x)$ é dada por

$$\frac{1}{n} \sum_{i=1}^n \frac{f(X_i)}{p(X_i)}, \quad (2.11)$$

onde $X \sim p$, ou seja, X_i são amostras distribuídas conforme a PDF $p(x)$. No caso mais simples, uma distribuição uniforme pode ser usada, ou seja,

$$p(x) = \frac{1}{|D|}, \quad (2.12)$$

onde $|D|$ é o n -volume do domínio de integração. Na prática, uma amostragem uniforme pode resultar em uma estimativa pobre, pois regiões do domínio onde a função $f(x)$ possui um valor elevado podem apresentar baixa densidade de amostras (Figura 2.4a).

Em vez de simplesmente aumentar o número de amostras, uma técnica de redução de variância mais efetiva consiste em escolher uma PDF $p(x)$ proporcional à função $f(x)$ (Figura 2.4b), ou seja,

$$p(x) = cf(x), \quad (2.13)$$

onde c é uma constante de normalização. Para que $p(x)$ seja uma PDF válida, deve obedecer à seguinte restrição:

$$\int_D p(x) dx = 1, \quad (2.14)$$

resultando na constante de normalização

$$c = \frac{1}{\int_D f(x) dx}. \quad (2.15)$$

Portanto, a PDF ideal para integrar $f(x)$ utilizando Monte Carlo e amostragem de importância (*importance sampling*) é dada por

$$p(x) = \frac{f(x)}{\int_D f(x) dx}. \quad (2.16)$$

Utilizando essa função de distribuição ideal, a estimativa de Monte Carlo obtém o valor desejado sem variância:

$$\frac{1}{n} \sum_{i=1}^n \frac{f(X_i)}{p(X_i)} = \frac{1}{n} \sum_{i=1}^n \frac{f(X_i)}{f(X_i)} \int_D f(x) dx = \int_D f(x) dx. \quad (2.17)$$

Na prática, a integral de $f(x)$ é justamente aquilo que se deseja obter e, portanto, não se dispõe da função de distribuição ideal. Entretanto, tendo-se algum conhecimento sobre $f(x)$ em certas porções do domínio, é possível obter distribuições melhores que a distribuição uniforme.

Caso o integrando seja o produto de vários termos, pode ser difícil escolher uma PDF adequada para todas as regiões do domínio, pois cada termo pode ter uma importância maior

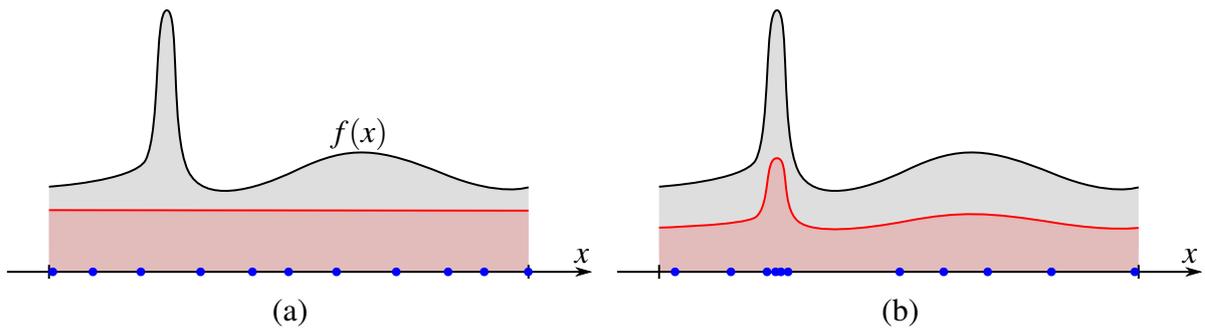


Figura 2.4: Amostragem de Importância: A linha vermelha representa a PDF usada para gerar amostras (pontos azuis). (a) Distribuição uniforme: regiões de maior importância da função são pobremente amostradas. (b) Distribuição proporcional a $f(x)$: a densidade de amostras em cada região do domínio é proporcional à importância da função na região.

dependendo da região. Amostragem de Importância Múltipla (*Multiple Importance Sampling*) (VEACH; GUIBAS, 1995) fornece uma solução robusta para esses casos. A ideia é calcular a contribuição de algumas amostras geradas segundo uma PDF específica para cada termo e combinar as contribuições de forma a preservar a maior importância.

2.4 Solução da Equação de Renderização por *Ray Tracing* e Monte Carlo

Com o objetivo de gerar imagens mais realísticas, a técnica de *Ray tracing* foi proposta por Whitted (1980). Essa técnica é capaz de gerar imagens com reflexos e refrações especulares mas não simula reflexões difusas — a mistura de cor entre objetos não especulares (*color bleeding*).

Para resolver esse problema, foi proposto um método conhecido como Radiosidade (GORAL et al., 1984). Ao contrário do *ray tracing* — que é um método de renderização baseado na imagem — radiosidade é um método baseado na cena e tem a vantagem de ser independente do observador, sendo adequado para aplicações interativas. Entretanto, existem algumas restrições: não consegue lidar facilmente com materiais especulares e a qualidade da solução é proporcional ao nível de discretização da cena em elementos (*patches*), o que torna a complexidade do método dependente da geometria da cena.

Atualmente, a maioria dos sistemas de renderização foto-realísticos são baseados em algoritmos de *ray tracing*. Um *ray tracer* precisa basicamente encontrar todos os tipos de caminhos que transportam luz entre as fontes de luz e o sensor (câmera). Um caminho é formado por vários vértices, em que o vértice de uma extremidade pertence a uma fonte de luz, e o vértice da outra extremidade pertence ao sensor. Os vértices internos (que no caso mais simples de *ray tracing* é apenas um) correspondem a interações (reflexão ou refração) entre a luz e os objetos da cena. A forma de encontrar esses caminhos, a variedade de caminhos que podem ser encontrados e como, a partir deles, computar a imagem final, é o que diferencia os diversos métodos de renderização baseados em *ray tracing* entre si.

Uma notação bastante útil para descrever caminhos de luz foi introduzida por Heck-

bert (1990). Essa notação descreve caminhos de luz por meio de expressões regulares do tipo

$$L(S|D)^*E,$$

onde as letras representam os vértices do caminho. Assim, L é um vértice em uma fonte de luz, S e D são vértice em superfícies onde ocorreram interação especular (que pode ser reflexão ou refração) e difusa, respectivamente e E é um vértice no sensor.

Quando o sistema calcula caminhos com no máximo um vértice interno, que geralmente são do tipo $L(D|\epsilon)E$, ele é dito algoritmo de iluminação local (ou direta), já que considera apenas a primeira interação luz/objeto. Esse tipo de técnica é geralmente conhecido como *ray casting* e tem a vantagem de ser bastante rápido, embora não possua muito realismo e rigor físico.

Uma extensão, que permite incluir efeitos de reflexão e refração especulares, consiste em traçar raios recursivamente a partir de objetos especulares. Nesse caso, quando o raio atinge um objeto com propriedade de reflexão (refração) especular, emite-se um raio na direção refletida (refratada), calculando essa contribuição recursivamente. A técnica resultante é conhecida como *ray tracing* clássico (WHITTED, 1980). Apesar desse algoritmo ser capaz de computar caminhos com vários vértices internos, ele não é considerado um algoritmo de iluminação global, já que considera apenas caminhos do tipo $L(D|\epsilon)S^*E$.

Apesar do *ray tracing* clássico não considerar todos os tipos caminhos, boa parte dos algoritmos de iluminação global baseados em *ray tracing* atuais também apresentam pequenas restrições sobre os tipos de caminhos computados devido a limitações intrínsecas ao processo de amostragem de caminhos (VEACH, 1998).

2.4.1 Monte Carlo Ray Tracing

Utilizando o método de integração de Monte Carlo dado pela Equação 2.5, pode-se solucionar a equação de renderização. Embora essa solução seja simples e teoricamente correta, sua implementação ingênua não apresenta bons resultados, pois requer muito tempo para convergir a um resultado aceitável. A ideia consiste em calcular $L(p, \omega_o)$ para todos os pontos p que são visíveis por um observador e direções ω_o que apontam de p para o observador.

Para decidir quais pontos são visíveis pelo observador, utiliza-se um teste de visibilidade baseado em traçamento de raios: gera-se um raio (O, \mathbf{d}) — O é a origem do raio e \mathbf{d} é a direção dada pelo sistema de câmera — e calcula-se a interseção entre o raio e os objetos da cena, obtendo-se p como o ponto de interseção mais próximo de O .

2.4.2 Path Tracing

Em seu artigo seminal, Kajiya (1986) propôs uma solução para a equação de renderização. Ela supõe que um observador ou sensor encontra-se na posição o , e está orientado segundo a direção $-\omega_o$. Nessas condições, para que o observador obtenha a radiância que o

atinge segundo a direção de observação, ele deve obter $L(t(o, -\omega_o), \omega_o)$, onde $t(p, \omega)$ retorna o primeiro ponto obtido traçando-se um raio a partir de um ponto p na direção ω .

Observando a Equação 2.5, percebe-se que L aparece nos dois lados da equação. Nesse caso, é necessário expandir L no lado direito, resultando em uma série de Neumann. Usando a notação do operador integral T , a Equação 2.5 é representada da seguinte forma:

$$L = L_e + TL, \quad (2.18)$$

onde tem-se a luz que atinge diretamente o observador representada por L_e e a iluminação indireta representada por TL . Expandindo o termo TL uma vez, obtém-se

$$L = L_e + TL_e + T^2L, \quad (2.19)$$

onde TL_e representa a iluminação indireta de primeira ordem (que sofreu exatamente uma reflexão)² e T^2L a iluminação indireta restante (de segunda ordem em diante).

A expansão completa é dada por:

$$L = \sum_{m=0}^{\infty} T^m L_e, \quad (2.20)$$

que deve ser interpretada como sendo a iluminação direta (L_e), mais a luz refletida uma vez (TL_e), mais a luz refletida duas vezes (T^2L_e), e assim por diante. Portanto, precisa-se de um algoritmo que consiga calcular cada um desses termos de forma eficiente.

Considere o termo TL_e na sua versão completa:

$$\int_{\Omega} f_r(p, \omega_o, \omega_i) L_e(p, \omega_i) |\cos \theta_i| d\omega_i, \quad (2.21)$$

que pode ser calculado por Monte Carlo da seguinte forma:

$$\frac{1}{N} \sum_{i=1}^N \frac{f_r(p, \omega_o, \omega_i) L_e(p, \omega_i) |\cos \theta_i|}{p(\omega_i)}, \quad (2.22)$$

onde $p = t(o, -\omega_o)$ e ω_i são direções aleatórias geradas com distribuição $p(\omega)$. Procedimento similar pode ser usado para calcular qualquer termo $T^m L_e$, sendo necessário traçar um caminho com m vértices internos.

O algoritmo *path-tracing* consiste, basicamente, em calcular o termo $T^{k+1}L_e$ reaproveitando o caminho traçado para o cálculo do termo $T^k L_e$, ou seja, um caminho é traçado incrementalmente. No primeiro ponto de interseção com a cena obtido traçando-se um raio a partir da câmera, é calculada a iluminação direta (TL_e). Depois, a partir desse ponto, é traçado outro raio, obtendo-se outro ponto de interseção. Nesse ponto, é calculada a iluminação direta, e, em seguida, é determinado quanto dessa iluminação é transportada para o ponto anterior (T^2L_e). O processo segue até que se decida terminar a soma. A decisão de parar a soma é feita usando-se

²Em alguns textos, o termo “iluminação indireta” é usado para designar apenas a iluminação de segunda ordem em diante.

uma técnica estatística chamada *roleta russa*.

Como o algoritmo gera os caminhos apenas a partir do observador, a convergência é muito lenta em vários casos de interesse prático. Uma técnica mais robusta consiste em traçar caminhos a partir do observador, das fontes de luz e combinar estes caminhos. Essa é a ideia básica do método *Bidirectional Path-tracing* (LAFORTUNE; WILLEMS, 1993).

2.5 Amostragem e Reconstrução

2.5.1 Amostragem

Conforme visto na seção anterior, o processo de solução da equação de renderização utilizando Monte Carlo requer amostras aleatórias. Para traçar o primeiro raio na cena, é necessária, no mínimo, uma amostra 2D (x, y) para a escolha de um ponto no sensor. Se forem incluídos efeitos de borrão de movimento (*motion blur*) e profundidade de campo (*depth of field*), são necessárias mais uma amostra 2D (u, v) para escolher um ponto na lente da câmera e uma amostra 1D (t) que é um instante de tempo no intervalo de abertura do obturador. Neste caso, para gerar o primeiro raio a partir da câmera, são necessários 5 valores aleatórios (x, y, u, v, t) .

Quando o primeiro raio é traçado, ele intersecta um ponto na cena onde será calculada a iluminação direta, e para isso é necessário pelo menos mais uma amostra 2D para a escolha de um ponto na fonte de luz. Então, agora são sete valores aleatórios, e apenas a iluminação direta calculada. Conforme os outros termos $(T^m L_e, m \geq 1)$ da Equação 2.20 são calculados, mais amostras são necessárias, aumentando a dimensionalidade do problema.

Portanto, para calcular uma contribuição no ponto (x, y) do sensor, são necessários (em teoria) infinitos valores aleatórios (u_1, u_2, u_3, \dots) , onde $u_1 = x$, $u_2 = y$ e assim por diante. Na prática, com ajuda da roleta russa, consegue-se limitar essa dimensionalidade e, ainda assim, obter um resultado estatisticamente correto. Isso equivale a utilizar pontos aleatórios em um hipercubo cuja dimensionalidade é finita, mas potencialmente grande.

Considerando então uma dimensionalidade limitada por n , o domínio de f passa a ser um hipercubo de alta dimensionalidade \mathcal{U} e a contribuição no ponto (x, y) é dada por

$$F(x, y) = \underbrace{\int_0^1 \dots \int_0^1}_{n-2} f(x, y, u_3, \dots, u_n) du_3 \dots du_n, \quad (2.23)$$

onde f é a função de cena.

A rigor, o domínio da função f é originalmente composto pelo plano da imagem, pelo tempo de abertura do obturador, pelos hemisférios em alguns pontos de interseção, pelas superfícies das fontes de luz, etc. Portanto, o que ocorre é um mapeamento do domínio canônico \mathcal{U} para esse domínio composto dependente da cena. Chamando esse mapeamento de $S : \mathcal{U} \rightarrow \mathcal{S}$,

o valor $F(x, y)$ no plano da imagem é dado por

$$F(x, y) = \int_u f(S(u)) \left| \frac{dS(u)}{du} \right| du. \quad (2.24)$$

2.5.2 Reconstrução

O valor final de um pixel k é reconstruído a partir de valores amostrais de $F(x, y)$, fazendo-se uma convolução com o filtro de reconstrução h :

$$I_k = \int_{D_I} F(\mathbf{x}) h_k(\mathbf{x}) dx, \quad (2.25)$$

onde I_k é o valor final do pixel k , h_k é o filtro de reconstrução centrado no pixel k e D_I é o domínio do plano da imagem.

Portanto, uma tarefa importante é como gerar esses pontos no hipercubo n -dimensional de forma a privilegiar regiões importantes do domínio. Isso é o que se busca fazer com amostragem adaptativa. Algumas técnicas se dedicam a adaptar as amostras apenas no plano da imagem, ou seja, considerando apenas as duas primeiras dimensões. Já outras tentam adaptar as amostras considerando múltiplas dimensões. Ambos os tipos de estratégias resultam em amostras distribuídas não uniformemente e precisam lidar com problemas na etapa de reconstrução. As técnicas de reconstrução tradicionais (derivadas diretamente da Equação 2.25) consideram amostras distribuídas uniformemente no plano da imagem e, portanto, não funcionam bem com distribuições não uniformes, resultando em artefatos na imagem final. Algumas técnicas de reconstrução, como em Mitchell (1987), visam minimizar esses problemas.

2.6 Considerações Finais

Este capítulo apresentou os principais conceitos necessários para o entendimento do problema de renderização como um problema de amostragem e reconstrução no domínio multidimensional. Os conceitos radiométricos fundamentais utilizados nos sistemas de *physically based rendering* foram apresentados, bem como a equação de renderização, que serve como base para os principais métodos de renderização modernos.

Em seguida, foram apresentados conceitos de integração de Monte Carlo e amostragem de importância como método de redução de variância. A motivação por trás da utilização deste método probabilístico para solução da equação de renderização, e como os métodos baseados em *ray tracing* são utilizados nesse contexto, também são apresentados. Por fim, o problema de solução da equação de renderização por Monte Carlo é reformulado como um problema de amostragem e reconstrução.

No próximo capítulo, são apresentados os trabalhos relacionados que servem de base para a técnica proposta nesta dissertação. Além disso, uma breve visão geral das técnicas de amostragem adaptativa é apresentada.

3 TRABALHOS RELACIONADOS

Neste capítulo é apresentada uma breve retrospectiva dos trabalhos sobre amostragem adaptativa em renderização, seguindo a classificação a seguir:

Adaptatividade no Plano da Imagem. Técnicas que se dedicam a reduzir o *aliasing* em regiões de alta frequência (rápida variação) no plano da imagem, sem considerar a possível natureza multidimensional desse *aliasing*.

Adaptatividade Orientada a Efeitos Específicos. Técnicas que buscam melhorar a qualidade das imagens com relação a efeitos específicos (*motion blur*, *depth-of-field*, etc). Nesta categoria, as abordagens são variadas e algumas se preocupam com a natureza multidimensional do problema.

Adaptatividade Multidimensional. Técnicas que procuram atacar o problema do ruído de forma mais geral, considerando a natureza multidimensional do problema.

Em seguida, duas técnicas são apresentadas com mais detalhes: *Multidimensional Adaptive Sampling and Reconstruction* (HACHISUKA et al., 2008) e *Metropolis Light Transport* (VEACH; GUIBAS, 1997). Esses dois trabalhos servem de base para a técnica proposta nessa dissertação.

3.1 Adaptatividade no Plano da Imagem

A ideia de amostragem adaptativa no plano da imagem não é nova. Em Whitted (1980) já se utiliza um mecanismo adaptativo com o objetivo de reduzir *aliasing*, tratando o plano da imagem como uma grade retangular, e obtendo amostras nos vértices de cada retângulo. De posse das quatro amostras de um retângulo, verifica-se a ocorrência de uma das seguintes situações: (1) diferenças bruscas de intensidade das amostras, (2) objetos muito pequenos entre as amostras e (3) mapeamento de textura; e, em caso afirmativo, o retângulo é subdividido em quatro e o processo continua até a resolução máxima do monitor ser atingida. Caso contrário, o valor médio das quatro amostras é considerado como uma boa aproximação para todo o retângulo. Apesar de adaptativa, essa técnica não é estocástica e, portanto, sofre de *aliasing* estruturado.

A técnica conhecida como *Distributed Ray Tracing* (COOK; PORTER; CARPENTER, 1984) propõe utilizar as amostras geradas no processo de super amostragem para integrar efeitos em outras dimensões, como *depth-of-field*, *motion blur* e *glossy reflection*. Baseado nessa técnica, foi proposta uma métrica estatística (variância aproximada) para decidir a quantidade de amostras usadas em cada pixel (LEE; REDNER; USELTON, 1985). A técnica utiliza uma distribuição estratificada uniforme inicial e, depois, é calculada a variância dessas amostras e decidido quais pixels devem receber mais amostras. O processo continua até que a variância atinja um certo limite calculado por meio de aproximações estatísticas. O trabalho também

mostra que a quantidade de amostras necessárias não depende da dimensionalidade utilizada, mas sim da variância das amostras.

Em Purgathofer (1987), é utilizada amostragem estocástica em cada pixel, eliminando o problema de *aliasing* estruturado. O critério de adaptação é baseado em dois parâmetros estatísticos: intervalo de confiança d e probabilidade α . Assim, calculado o desvio padrão das amostras, se o valor exato do pixel encontrar-se no intervalo $2d$ com probabilidade α (GEERTSEMA, 1970), não há necessidade de calcular novas amostras. A quantidade mínima necessária de amostras, dados d e α , também é apresentada no trabalho. Apesar de uma métrica adaptativa ser apresentada, nenhuma adaptação da distribuição usada na geração das amostras é feita.

Regiões de alta frequência na imagem são as principais candidatas a receberem uma quantidade maior de amostras. Entretanto, quando a quantidade de amostras obtida em cada pixel é variável, a etapa de reconstrução tradicional provoca alguns artefatos indesejados nas bordas dos objetos. Em Mitchell (1987) uma solução para esse problema é apresentada, aplicando-se o filtro de reconstrução em vários estágios, sendo que cada estágio utiliza um núcleo (*kernel*) de tamanho diferente.

Diversas outras técnicas adaptativas baseadas no plano da imagem foram propostas, mas o fato de serem baseadas na imagem, não adaptando outras dimensões, e por utilizarem valores com muito ruído para os cálculos estatísticos usados como critérios de adaptação, as tornam pouco efetivas em lidar com efeitos distribuídos oriundos de integração em outras dimensões.

3.2 Adaptatividade Orientada a Efeitos Específicos

Uma técnica de amostragem espaço-temporal focada em *motion blur* e um novo algoritmo de reconstrução são apresentados em Egan et al. (2009). Analisando-se o espectro de frequências de imagens com *motion blur*, verificou-se um cisalhamento no espectro. O trabalho propõe uma aproximação para esse efeito bem como um novo filtro cisalhado que explora esse efeito e permite bons resultados com uma quantidade ainda menor de amostras.

Em Chen et al. (2011), um método adaptativo focado em *depth-of-field* é apresentado. O método é dividido em três etapas principais: inicialização, amostragem e reconstrução. Na etapa de inicialização são construídos três mapas que serão usados nas etapas seguintes: mapa de profundidade, mapa de borrão e mapa de variância por pixel. Os mapas são construídos simulando-se uma câmera do tipo *pinhole* e traçando-se quatro raios por pixel. Na etapa de amostragem, a imagem é amostrada adaptativamente por Monte Carlo, utilizando-se os três mapas para guiar a distribuição de amostras, resultando em uma imagem intermediária chamada *pixel-mean image*. Finalmente, na etapa de reconstrução é aplicado um filtro multiescala na *pixel-mean image*. Essa filtragem utiliza o mapa de borrão para aplicar uma maior suavização em regiões mais borradas, e menos suavização em regiões menos borradas. Os resultados mostram que essa técnica é mais eficiente do que aquelas propostas por Hachisuka et al. (2008) e Overbeck, Donner e Ramamoorthi (2009) para simulação de *depth-of-field*.

3.3 Adaptatividade Multidimensional

Ao discutir técnicas de redução de variância para a integração de Monte Carlo, Kajiya (1986) propôs uma técnica hierárquica de amostragem e integração. Conforme as amostras são geradas, elas são guardadas em uma kd-tree. Para se gerar uma nova amostra, uma folha da kd-tree é escolhida, gera-se uma amostra em seu interior, e a folha é dividida. Uma estratégia adaptativa de escolha das folhas é proposta e alguns critérios de escolha são comentados, mas nenhum deles foi bem sucedido nos testes feitos pelo autor: “*So far our experiments in finding adaptive criteria have not been terribly successful.*” (KAJIYA, 1986).

Kelemen et al. (2002) propuseram uma abordagem mais simples ao *Metropolis Light Transport* de Veach e Guibas (1997). No *Metropolis Light Transport* original, as mutações eram feitas no espaço de caminhos da cena. A simplificação veio ao se considerar um domínio abstrato de amostragem (o hipercubo) de onde as amostras são mapeadas em caminhos da cena. Essa visão é a mesma que foi apresentada na Seção 2.5. As mutações, nesse caso, são feitas no domínio abstrato, utilizando algumas melhorias que aumentam a taxa de aceitação das mutações propostas, tornando o método mais eficiente.

Como *Metropolis Sampling* necessita de uma função $f : D \rightarrow \mathbb{R}$, ele não pode ser usado diretamente na equação de renderização, pois esta retorna valores de radiância, que é uma grandeza espectral. Portanto, no *Metropolis Light Transport* clássico utiliza-se uma *função de importância* $I(x)$, que geralmente é a luminância calculada a partir da radiância. Com essa função de importância, as amostras são adaptadas a regiões do domínio que representam caminhos de luz importantes, ou seja, que são de grande contribuição (alta luminância) à imagem. Hoberock e Hart (2010) apresentaram uma generalização do *Metropolis Light Transport* de forma a utilizar outros tipos de função de importância, não apenas baseado em luminância, mas em ruído.

O critério de adaptatividade é uma parte importante nos trabalhos de amostragem adaptativa. Muitos dos critérios são baseados em dados estatísticos (variância) e contraste, que nem sempre traduzem fielmente a qualidade perceptível da imagem. Bolin e Meyer (1998) apresentam um critério de adaptação baseado em percepção visual, utilizando um modelo de visão simplificado.

Inspirados no uso de *wavelets* em *importance sampling* (CLARBERG et al., 2005), Overbeck, Donner e Ramamoorthi (2009) apresentaram uma abordagem adaptativa baseada em *wavelets* capaz de lidar com efeitos multidimensionais, estimando a imagem diretamente no domínio *wavelet*. A técnica funciona em duas etapas: amostragem adaptativa e reconstrução da imagem. Na primeira etapa, a variância dos coeficientes de escala das *wavelets* é medida iterativamente. Sabendo-se que regiões de alta frequência na imagem (tais como arestas) geram coeficientes de alta amplitude em *wavelets*, e que mesmo regiões de baixa frequência podem apresentar alta variância nestes coeficientes devido à variância em outras dimensões, a geração de amostras é guiada pelos coeficientes e adaptada a essas duas situações por meio de *importance sampling*. Na segunda etapa, todas as *wavelets* são usadas para a reconstrução da imagem, subtraindo a variância medida nos coeficientes. A técnica gera resultados com baixo ruído mesmo com baixa quantidade de amostras (entre 16 e 32) e, apesar da representação *wavelet*

ser apenas no domínio 2D da imagem, bons resultados são obtidos em cenas com efeitos multidimensionais.

Rousselle, Knaus e Zwicker (2011) apresentam uma técnica gulosa de redução de erro (*mean squared error*), que é dividida em duas etapas. Primeiro, dada uma distribuição inicial de amostras, é feita uma otimização em cada pixel para a escolha do filtro que mais reduz o erro, sendo este filtro escolhido dentre um conjunto discreto de opções. Depois, dada a escolha de filtros, é feita uma segunda distribuição de amostras para reduzir ainda mais o erro. Os resultados mostram que a técnica apresenta vantagens sobre as técnicas propostas em Overbeck, Donner e Ramamoorthi (2009) e Hachisuka et al. (2008).

3.4 *Metropolis Light Transport*

Conforme apresentado no Capítulo 2, a estimativa de Monte Carlo para a integral de $f(x)$ é dada por:

$$\int_D f(x) dx \approx \frac{1}{n} \sum_{i=1}^n \frac{f(X_i)}{p(X_i)}, \quad (3.1)$$

onde $X \sim p$, ou seja, X_i são amostras distribuídas conforme a função de distribuição $p(x)$. Também foi mostrado que a distribuição $p(x)$ ideal é dada por

$$p(x) = \frac{f(x)}{\int_D f(x) dx}, \quad (3.2)$$

apesar de, geralmente, não se dispor de tal distribuição.

Metropolis Sampling oferece um procedimento para geração de amostras segundo a distribuição ideal para uma certa função $f : D \rightarrow \mathbb{R}$ escolhida arbitrariamente, e sem a necessidade de calcular a PDF explicitamente. O procedimento básico é apresentado no Algoritmo 3.1 .

Algorithm 3.1 *Metropolis Sampling* Básico

```

1:  $V \leftarrow \emptyset$ 
2:  $X \leftarrow$  amostra inicial
3: for  $i \leftarrow 1 \dots N$  do
4:    $Y \leftarrow$  muta( $X$ ) com distribuição  $T(X \rightarrow Y)$ 
5:    $a \leftarrow \min \left( 1, \frac{f(Y)T(Y \rightarrow X)}{f(X)T(X \rightarrow Y)} \right)$ 
6:   if rand() <  $a$  then
7:      $X \leftarrow Y$ 
8:    $V \leftarrow V \cup X$ 

```

A partir de uma amostra X , outra amostra Y é proposta por meio de uma mutação da amostra X . A mutação, em teoria, pode ser de qualquer tipo (inclusive uma que não leva em consideração a amostra anterior X), desde que seja possível calcular as PDFs $T(X \rightarrow Y)$ e $T(Y \rightarrow X)$, que dão a probabilidade de obter Y dado X e X dado Y , respectivamente. Então, de acordo com a probabilidade de aceitação a , essa amostra proposta é aceita ou descartada. No caso de a amostra proposta ser aceita, ela torna-se a amostra corrente, caso contrário, a própria

amostra X continua sendo a amostra corrente. No final do procedimento, o vetor V possui uma sequência de amostras cuja distribuição, no limite, atinge um estado de equilíbrio que é justamente a distribuição ideal para $f(x)$.

Uma observação importante é que, mesmo dispondo-se de amostras geradas segundo uma distribuição ideal, tal distribuição ($p(x)$) não é calculada explicitamente para ser usada na integração de Monte Carlo. Para solucionar esse problema, utiliza-se uma função de contribuição $I : D \rightarrow \mathbb{R}$ escolhida adequadamente e cuja integral possa ser calculada com menos precisão por um processo auxiliar. Nesse caso, a integral de $f(x)$ é obtida da seguinte forma:

$$\int_D f(x) dx \approx \left(\frac{1}{N} \sum_{i=1}^N \frac{f(X_i)}{I(X_i)} \right) \int_D I(x) dx, \quad (3.3)$$

onde as amostras X_i são geradas por meio de *Metropolis Sampling* na função de contribuição $I(x)$.

A utilização de *Metropolis Sampling* em renderização foi proposta por Veach e Guibas (1997) e baseia-se na seguinte formulação para o valor de um pixel da imagem:

$$\Phi_j = \int h_j(x) F(x) dx, \quad (3.4)$$

onde $h_j(x)$ é o filtro de reconstrução centrado no pixel j e $F(x)$ é a função da imagem.

Utilizando *Metropolis Sampling* e uma função de contribuição $I(x)$ adequada, obtém-se a seguinte estimativa para o valor do pixel j :

$$\Phi_j \approx \left(\frac{1}{N} \sum_{i=1}^N \frac{h_j(x) F(X_i)}{I(X_i)} \right) \int_D I(x) dx, \quad (3.5)$$

onde tradicionalmente utiliza-se a luminância de $F(x)$ como função de contribuição.

Como na prática a extensão do filtro de reconstrução é limitada — cada amostra contribui com poucos pixels — o seguinte procedimento pode ser usado para computar uma imagem usando *Metropolis Sampling*, dando origem ao *Metropolis Light Transport* básico:

Algorithm 3.2 *Metropolis Light Transport* básico

- 1: $b \leftarrow \int I(x) dx$
 - 2: $X \leftarrow$ amostra inicial
 - 3: **for** $i \leftarrow 1 \dots N$ **do**
 - 4: $Y \leftarrow$ muta(X) com distribuição $T(X \rightarrow Y)$
 - 5: $a \leftarrow \min \left(1, \frac{f(Y)T(Y \rightarrow X)}{f(X)T(X \rightarrow Y)} \right)$
 - 6: **if** $\text{rand}() < a$ **then**
 - 7: $X \leftarrow Y$
 - 8: $j \leftarrow$ pixel obtido a partir de X
 - 9: $\Phi_j \leftarrow \Phi_j + \frac{b}{N} \frac{h_j(X)F(X)}{I(X)}$
-

Uma simples modificação do procedimento básico permite utilizar a informação

obtida com as amostras rejeitadas, resultando no Algoritmo 3.3 .

Algorithm 3.3 *Metropolis Light Transport* com Uso de Valores Esperados

```

1:  $b \leftarrow \int I(x) dx$ 
2:  $X \leftarrow$  amostra inicial
3: for  $i \leftarrow 1 \dots N$  do
4:    $Y \leftarrow$  muta( $X$ ) com distribuição  $T(X \rightarrow Y)$ 
5:    $a \leftarrow \min \left( 1, \frac{f(Y)T(Y \rightarrow X)}{f(X)T(X \rightarrow Y)} \right)$ 
6:    $\Phi_X \leftarrow \Phi_X + (1 - a) \frac{b}{N} \frac{h_j(X)F(X)}{I(X)}$ 
7:    $\Phi_Y \leftarrow \Phi_Y + a \frac{b}{N} \frac{h_j(Y)F(Y)}{I(Y)}$ 
8:   if  $\text{rand}() < a$  then
9:      $X \leftarrow Y$ 

```

A principal vantagem da técnica *Metropolis Light Transport* reside na sua capacidade de explorar regiões importantes do domínio da cena sem sofrer da “maldição da dimensionalidade” (*curse of dimensionality*). Entretanto, regiões que normalmente apresentam pouco ruído (como regiões com predominância de iluminação direta) sofrem um acréscimo no ruído devido à dificuldade em estratificar as amostras.

3.5 Multidimensional Adaptive Sampling and Reconstruction

Multidimensional Adaptive Sampling and Reconstruction (MDAS) (HACHISUKA et al., 2008) é uma técnica que permite reconstruir a imagem final a partir de poucas amostras distribuídas adaptativamente no domínio multidimensional da função de cena. A técnica é dividida em duas etapas principais: Amostragem Adaptativa Multidimensional e Reconstrução. A primeira etapa é responsável por gerar as amostras e armazená-las em uma kd-tree, em que a própria kd-tree é utilizada para gerar amostras de forma adaptativa. A segunda etapa consiste em reconstruir a imagem final por meio de um processo de integração similar à soma de Riemann. A Figura 3.1 apresenta uma visão geral da etapas da técnica e em seguida, as etapas são apresentadas em mais detalhes.

3.5.1 Amostragem Adaptativa Multidimensional

A técnica começa gerando uma certa quantidade N_i de amostras distribuídas uniformemente no domínio multidimensional. Essas amostras são armazenadas nas folhas de uma kd-tree de dimensionalidade k — as folhas da kd-tree possuem uma capacidade máxima C_{max} .

Após a amostragem inicial, a kd-tree é refinada adaptativamente de acordo com as folhas que apresentam maior erro (contraste), ou seja, as folhas cujas amostras diferem mais entre si. O procedimento consiste em escolher iterativamente a folha de maior erro utilizando uma fila de prioridades. Para cada folha escolhida, gera-se uma nova amostra por um processo de melhor candidato (N_c tentativas) dentro de sua esfera envoltória e insere-se a nova amostra na

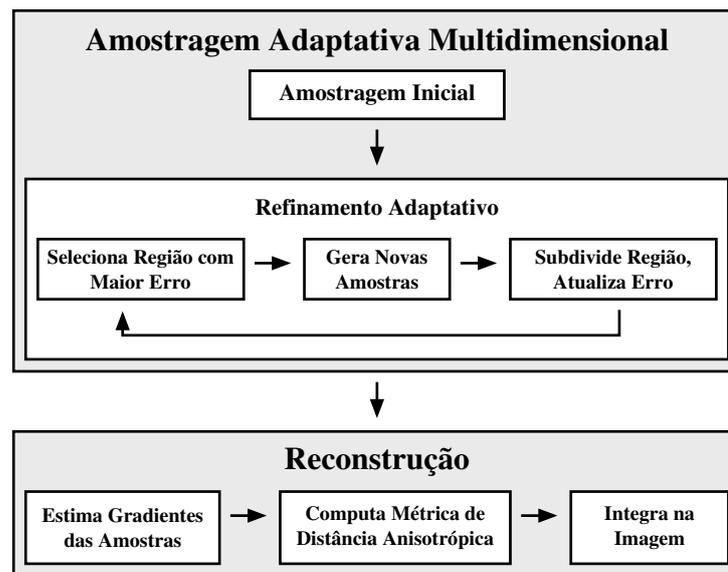


Figura 3.1: Visão geral das etapas da técnica MDAS. (Imagem traduzida de Hachisuka et al. (2008))

kd-tree. Sempre que a capacidade de uma folha for ultrapassada, a folha é subdividida segundo o maior eixo. A Figura 3.2 exemplifica esse procedimento.

O domínio da kd-tree consiste em um hiper-retângulo de dimensões $[d_1, \dots, d_k]$. As dimensões d_1 e d_2 correspondem às dimensões w e h do plano da imagem e são fixadas de forma a respeitar sua razão de aspecto (*aspect ratio*): se $h < w$, tem-se $d_1 = w/h$ e $d_2 = 1$, caso contrário, tem-se $d_1 = 1$ e $d_2 = h/w$. As demais dimensões são dadas pelo usuário por meio de fatores de escala. Tais parâmetros acabam determinando o grau de refinamento em cada dimensão, pois as folhas são divididas pelo maior eixo, resultando em maior refinamento nas dimensões que receberem maiores fatores de escala.

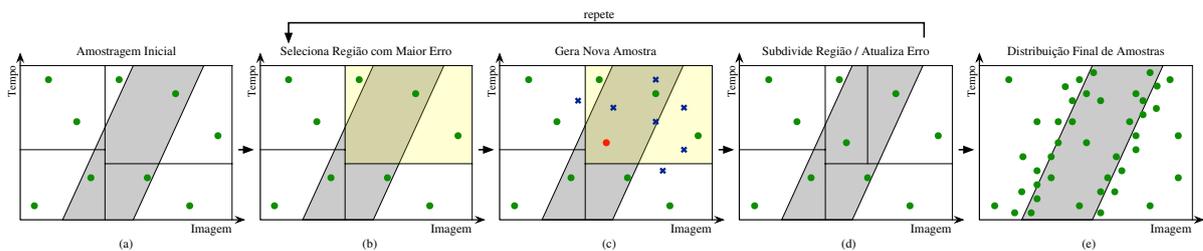


Figura 3.2: Visão geral do processo de amostragem adaptativa da técnica MDAS. (Imagem traduzida de Hachisuka et al. (2008))

3.5.2 Reconstrução

Conforme apresentado na Seção 2.5, o valor $F(x, y)$ na posição (x, y) no plano da imagem é dado por

$$F(x, y) = \int \dots \int f(x, y, u_1, \dots, u_n) du_1 \dots du_n, \quad (3.6)$$

onde f é a contribuição do caminho formado pela amostra (x, y, u_1, \dots, u_n) . Definindo-se, para cada posição (x, y) do plano da imagem, uma partição $P(x, y)$ das n dimensões restantes, o valor de $F(x, y)$ pode ser calculado da seguinte forma:

$$F(x, y) = \sum_{\Omega \in P(x, y)} \int_{\Omega} f(x, y, u_1, \dots, u_n) du_1 \dots du_n. \quad (3.7)$$

Considerando-se a função f constante em cada subdomínio Ω , pode-se obter o valor de $F(x, y)$ por meio da seguinte simplificação:

$$F(x, y) \approx \sum_{\Omega \in P(x, y)} V_{\Omega} f_{\Omega}. \quad (3.8)$$

Em MDAS, a kd-tree é utilizada como a partição P e o valor de f em cada subdomínio é obtido por meio de um processo de interpolação de vizinhos mais próximos, conforme explicado a seguir.

A etapa de reconstrução inicia com a subdivisão de todas as folhas da kd-tree até que haja apenas uma amostra por folha. Em seguida, cada amostra \mathbf{s} tem seu gradiente calculado a partir de N_v vizinhos mais próximos, utilizando o seguinte sistema:

$$\begin{bmatrix} \mathbf{s}_1 - \mathbf{s} \\ \vdots \\ \mathbf{s}_{N_v} - \mathbf{s} \end{bmatrix} \nabla f(\mathbf{s}) = \begin{bmatrix} f(\mathbf{s}_1) - f(\mathbf{s}) \\ \vdots \\ f(\mathbf{s}_{N_v}) - f(\mathbf{s}) \end{bmatrix}. \quad (3.9)$$

Em seguida, com o objetivo de preservar bordas, calcula-se uma métrica de distância distorcida que faz com que amostras cujos valores variam mais entre si pareçam mais distantes. Essa métrica consiste na distância de Mahalanobis definida pelo seguinte tensor que é calculado para cada amostra \mathbf{s} :

$$G(\mathbf{s}) = \frac{1}{N_v} \sum_{i=1}^{N_v} \nabla f(\mathbf{s}_i) \nabla f(\mathbf{s}_i)^T. \quad (3.10)$$

A distância de Mahalanobis entre dois pontos \mathbf{a} e \mathbf{b} é tal que $|\mathbf{v}|^2 = \mathbf{v}^T G \mathbf{v}$, onde $\mathbf{v} = \mathbf{a} - \mathbf{b}$. Por meio dessa métrica de distância pode-se calcular o valor de $f(\mathbf{x})$ para qualquer ponto \mathbf{x} , para isso, busca-se a amostra \mathbf{s} mais próxima de \mathbf{x} por meio da distância de Mahalanobis e considera-se $f(\mathbf{x}) = f(\mathbf{s})$.

Tendo-se a kd-tree definindo uma partição para do mínimo de f , e a capacidade de se calcular o valor de $f(\mathbf{x})$ em qualquer ponto, o valor final de cada pixel é obtido por meio da Equação 3.8. Para cada pixel (x, y) , obtém-se as folhas da kd-tree que cruzam esse ponto nas outras dimensões, calcula-se \mathbf{x} para cada ponto $\mathbf{x} = (x, y, c_3, \dots, c_k)$, onde $c = [c_1, \dots, c_k]$ é o centro de cada uma dessas folhas, multiplicam-se esses valores pelos volumes das respectivas folhas (em todas as dimensões exceto (x, y)) e acumula-se o resultado no pixel. A Figura 3.3 mostra o processo de reconstrução.

A técnica MDAS é capaz de obter bons resultados em cenas que demandam poucas dimensões no domínio de amostragem. Entretanto, devido ao fato de todas as amostras geradas

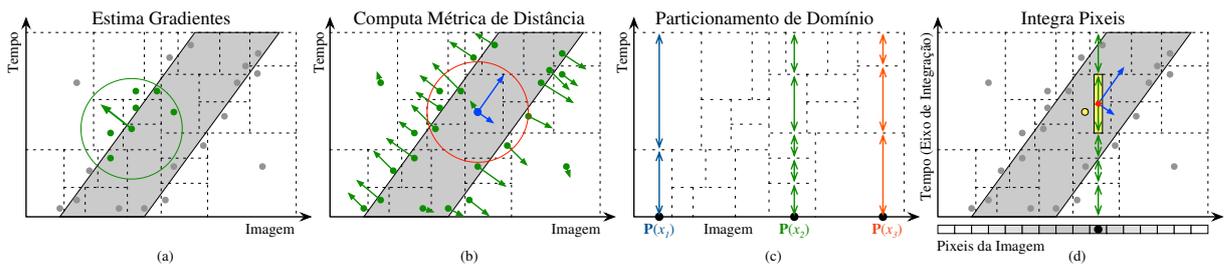


Figura 3.3: Visão geral da etapa de reconstrução da técnica MDAS. (Imagem traduzida de Hachisuka et al. (2008))

serem armazenadas em uma kd-tree, o custo de manutenção (armazenamento e busca) cresce à medida que mais amostras são utilizadas. O problema é agravado ainda mais pelo aumento da dimensionalidade, forçando, na prática, a utilização de uma quantidade reduzida de dimensões adaptáveis. Outra característica que pode ser citada como desvantagem é a grande quantidade de parâmetros dados pelo usuário, muitos deles implicam sensivelmente na qualidade das imagens obtidas e requerem uma fase de tentativa e erro para serem corretamente ajustados. A Tabela 3.1 sumariza os parâmetro utilizados em MDAS.

Tabela 3.1: Parâmetros da técnica MDAS

Parâmetro	Descrição
N_i	Número de amostras utilizadas na etapa de amostragem inicial
N_a	Número de amostras utilizadas na etapa de refinamento adaptativo da kd-tree
S_h	Fatores de escala das dimensões do hipercubo
C_{max}	Capacidade máxima das folhas
N_c	Número de candidatos na geração de amostras dentro das folhas
N_v	Quantidade de vizinhos mais próximos utilizados no cálculo dos gradientes, tensores, e na etapa de reconstrução
G_{max}	Tamanho máximo dos vetores gradiente

Liu, Wu e Zheng (2012) propuseram uma versão em paralelo de MDAS. A ideia consiste em utilizar as células da kd-tree — geradas a partir das amostras iniciais — para subdividir o domínio e gerar novas kd-trees em paralelo para cada subdomínio. Assim como MDAS, primeiro gera-se uma distribuição uniforme inicial de amostras no domínio multidimensional e constrói-se uma kd-tree a partir dessas amostras. A quantidade de amostras usadas na primeira etapa é obtida por meio de parâmetros controlados pelo usuário. Em seguida, calcula-se a quantidade máxima de amostras adaptativas que será investida na etapa seguinte em cada subdomínio. Para cada célula da kd-tree inicial gera-se, em paralelo, uma nova kd-tree por meio de amostragem adaptativa.

3.6 Considerações Finais

Este capítulo apresentou uma visão geral dos principais trabalhos sobre amostragem adaptativa em renderização, e apresentou em detalhe as duas técnicas que servem de base para o método proposto que será apresentado no próximo capítulo: *MDAS* e *Metropolis Light Transport*.

Metropolis Light Transport possui a capacidade de concentrar o esforço de amostragem em regiões importantes do domínio, sem sofrer do problema da dimensionalidade. Porém, acrescenta ruído em regiões suaves da imagem.

A técnica MDAS, por sua vez, é capaz de concentrar o esforço de amostragem em regiões de rápida variação do domínio multidimensional, gerando imagens suaves com uma quantidade relativamente baixa de amostras. Entretanto, devido aos custos de manutenção e utilização da kd-tree, cenas com maior dimensionalidade não podem ser tratadas. No próximo capítulo são apresentadas algumas melhorias que visam amenizar esses problemas.

4 TÉCNICA PROPOSTA

O método proposto neste trabalho consiste na adição de algumas etapas ao método *Multidimensional Adaptive Sampling and Reconstruction* (HACHISUKA et al., 2008): substituição de amostras, integração auxiliar e reconstrução por nó. A etapa de substituição de amostras visa reduzir o viés (*bias*) da etapa de amostragem inicial, ou seja, reduzir as chances de que características importantes da imagem sejam desconsideradas. A etapa de integração auxiliar visa agregar mais informação aos nós da kd-tree, permitindo a redução do ruído causado pela omissão de dimensões entre aquelas que são tratadas pela kd-tree. A etapa de reconstrução por nó visa reduzir o impacto da reconstrução no tempo de execução do método.

Com essas modificações, procura-se dotar o método MDAS de maior capacidade em lidar com cenas de maior dimensionalidade, sem que haja necessidade de aumentar a dimensionalidade da kd-tree utilizada no método.

4.1 Visão Geral

Conforme visto no Capítulo 3, a técnica de amostragem adaptativa multidimensional MDAS apresenta algumas limitações práticas quanto à dimensionalidade tratada e à quantidade de amostras utilizadas — elevado custo de armazenamento da kd-tree bem como do tempo de busca por vizinhos mais próximos. Essas limitações determinam o tipo de cena que pode ser tratada pelo método, que geralmente possuem uma variedade de efeitos que são limitados a priori.

O objetivo do método proposto é permitir a utilização de MDAS em cenas com maior dimensionalidade, permitindo a renderização de cenas com uma maior gama de efeitos. Para isso, acrescentamos duas etapas ao método MDAS: substituição de amostras e integração auxiliar (Figura 4.1). A etapa de substituição de amostras visa reduzir o viés, ou seja, aumentar as chances de características sutis da cena serem capturadas. A etapa de integração auxiliar visa melhorar a integração de efeitos oriundos de dimensões não tratadas pela kd-tree. As seções seguintes apresentam em detalhes as etapas que constituem o método.

4.2 Amostragem Inicial

Esta etapa é similar à etapa de amostragem inicial do MDAS, em que um conjunto inicial de amostras distribuídas uniformemente é gerado e utilizado para construir a kd-tree. A diferença é que a dimensionalidade da kd-tree não é definida como sendo igual à dimensionalidade da cena, ou seja, sendo k a dimensionalidade da kd-tree, e n a dimensionalidade da cena, pode ocorrer de $k \leq n$.

A kd-tree armazena amostras apenas em suas folhas. Durante o procedimento de construção da kd-tree a partir das amostras iniciais, as folhas são subdivididas recursivamente pela mediana do maior eixo até que a quantidade de amostras em cada folha seja menor ou igual à sua capacidade. Ao final desse processo, a kd-tree representa uma partição aproximadamente

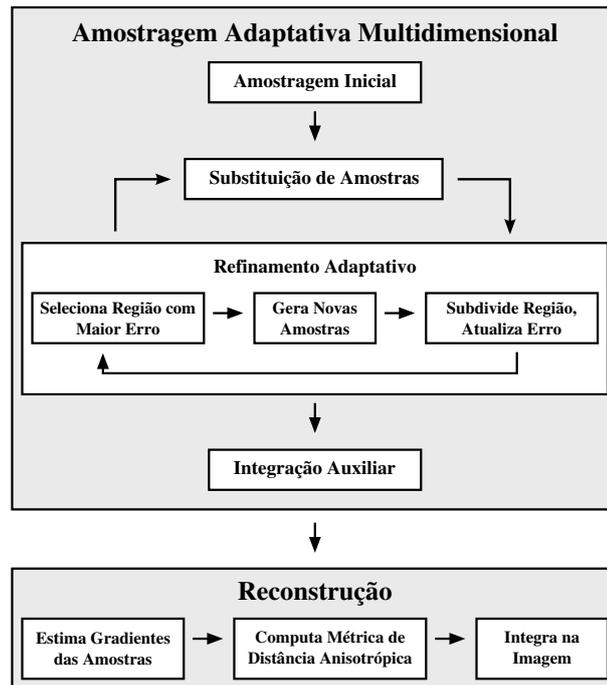


Figura 4.1: Visão geral da técnica proposta.

uniforme do domínio k -dimensional. A Figura 4.2 mostra o processo de divisão de uma célula da kd-tree para o caso 2D.

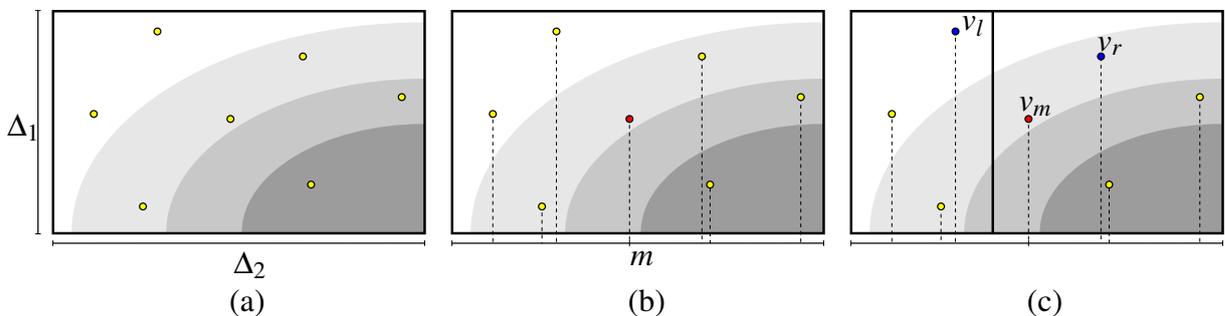


Figura 4.2: Divisão de uma célula da kd-tree (caso 2D): (a) Comparando-se as dimensões da célula, verifica-se que $\Delta_2 > \Delta_1$, portanto, escolhe-se esse eixo para subdivisão. (b) Comparando-se as coordenadas de cada amostra com relação ao eixo escolhido, encontra-se a mediana m . (c) Como ponto de divisão, escolhe-se a média entre a amostra mediana (v_m) e a amostra adjacente de maior contraste com relação à mediana (v_l).

Como essa etapa é responsável por capturar as informações que serão utilizadas para guiar a etapa de refinamento adaptativo, a quantidade de amostras utilizadas é de fundamental importância para que a imagem obtida tenha boa qualidade — quando a quantidade de amostras é muito pequena, a representação inicial da função de cena fica muito grosseira, tornando a etapa de refinamento adaptativo muito suscetível a viés, além de gerar artefatos na imagem final devido ao baixo refinamento da kd-tree em algumas regiões. Por outro lado, uma quantidade muito grande se torna inviável devido aos custos de manutenção e busca na kd-tree, além de torná-la mais propensa a instabilidades numéricas.

4.3 Substituição das Amostras Iniciais

Sabendo-se que as amostras iniciais são geradas segundo uma distribuição uniforme, a kd-tree construída a partir delas representa uma partição aproximadamente uniforme do domínio k -dimensional, ou seja, cada folha corresponde a um subdomínio próprio de D e cada subdomínio possui aproximadamente o mesmo volume. Entretanto, pode-se verificar que muitas folhas capturam pouca informação de contraste da função de cena f na região do domínio correspondente. Como a etapa de refinamento adaptativo se baseia nessa informação de contraste para decidir quais folhas devem ser refinadas, características importantes da cena acabam sendo desconsideradas. A Figura 4.3 exemplifica essa situação.

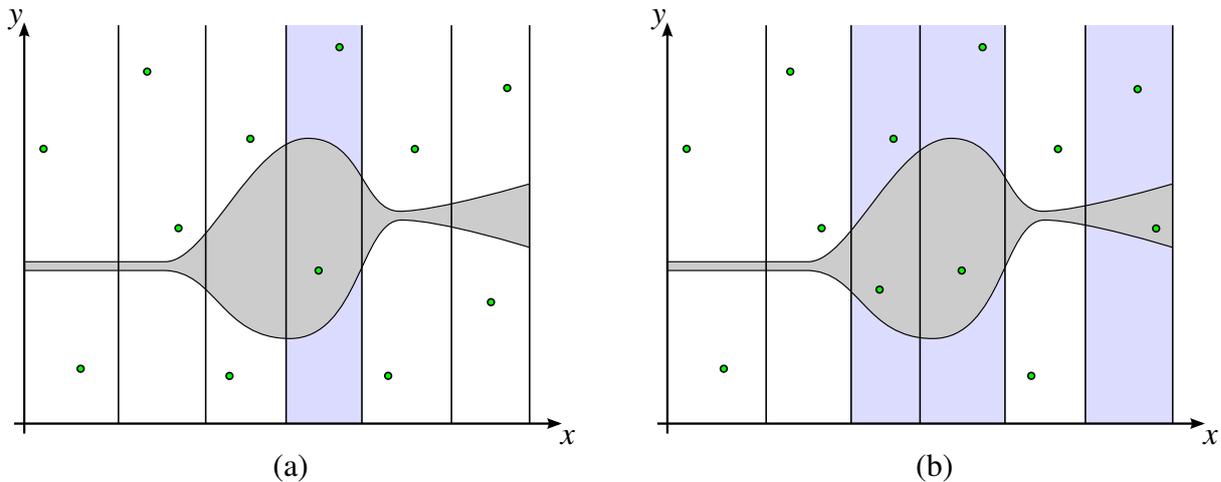


Figura 4.3: Etapa de substituição de amostras iniciais: (a) Distribuição inicial de amostras — apenas uma célula (em destaque) é candidata a subdivisão. (b) Nova distribuição inicial após algumas substituições — captura-se mais informação da função subjacente, melhorando-se o processo de subdivisão adaptativa.

Para tentar melhorar as amostras iniciais de forma a obter uma distribuição que captura mais informação da função subjacente, propõe-se uma etapa de substituição das amostras. O procedimento consiste em gerar um novo conjunto de amostras propostas S_U que será utilizado para substituir as amostras iniciais S_I de acordo com o contraste. Para cada amostra proposta, verifica-se em qual folha da kd-tree ela deve ser inserida. Uma vez encontrada essa folha, uma de suas amostras pode ser substituída de acordo com algum critério que permita a permanência de amostras mais concentradas em regiões de rápida variação da função de cena.

O critério de substituição utilizado baseia-se em uma métrica de distância que considera tanto a posição espacial da amostra, quanto sua contribuição (Algoritmo 4.4). Note que a posição das amostras é dada por k componentes (dimensionalidade da kd-tree), portanto, essa métrica de distância ignora as $n - k$ últimas dimensões espaciais. Dadas a amostra proposta (s), a folha onde se pretende inseri-la (B), e a métrica de distância, o procedimento de substituição fica definido de acordo com o Algoritmo 4.5.

Ao final da etapa de substituição, mesmo que a topologia da kd-tree não tenha sido alterada, as amostras armazenadas podem possuir mais informação de contraste da função de cena, permitindo uma etapa de refinamento adaptativo menos suscetível a viés.

Algorithm 4.4 Métrica de distância utilizada na etapa de substituição

```

1: procedure DIST(p, q)
2:    $ds = |\mathbf{p} - \mathbf{q}|^2$ 
3:    $vs = |f(\mathbf{p}) - f(\mathbf{q})|^2$ 
4:   return  $ds * dv$ 

```

Algorithm 4.5 Etapa de substituição

```

1: procedure SUBSTITUI( $B$ , s)
2:   (i, j) = par de amostras mais próximas em  $B$ 
3:    $d_{min} = \text{DIST}(\mathbf{i}, \mathbf{j})$ 
4:    $s_{min} =$  distância entre s e sua amostra mais próxima em  $B$ 
5:   if  $d_{min} < s_{min}$  then
6:     dentre i e j, substitui por s aquela de menor distância às amostras de  $B$ 

```

4.4 Refinamento Adaptativo

Assim como a etapa de amostragem inicial, a etapa de refinamento adaptativo proposta é quase idêntica à etapa de refinamento adaptativo de MDAS e a diferença é a mesma — as amostras geradas podem possuir uma dimensionalidade maior que a dimensionalidade da kd-tree. As k primeiras componentes de cada amostra são geradas por meio da kd-tree pelo procedimento apresentado na Seção 3.5, enquanto que as $n - k$ componentes restantes são geradas por meio de uma distribuição uniforme.

Como as amostras geradas possuem uma quantidade maior de componentes que a kd-tree, uma vez que a função de cena é avaliada para uma certa amostra, sua contribuição é armazenada na kd-tree junto com as k primeiras componentes da amostra, ou seja, aquelas componentes que foram geradas pela própria kd-tree.

Ao final do processo de refinamento adaptativo, subdividem-se todas as folhas da kd-tree até que haja apenas uma amostra por folha. A kd-tree resultante é utilizada para integração e reconstrução da imagem final nas etapas seguintes.

4.5 Integração Auxiliar

A etapa de refinamento adaptativo da seção anterior pode não ser suficiente para capturar características importantes da função de cena em certas regiões do domínio. Portanto, uma etapa de integração auxiliar é proposta, utilizando o domínio adaptativo representado pela kd-tree para reamostrar a função de cena. Duas alternativas são apresentadas a seguir: Integração Auxiliar Uniforme e Integração Auxiliar por Metropolis. Apesar da etapa de integração auxiliar utilizar uma quantidade maior de amostras, ambas as alternativas apresentadas não exigem refinamento adicional da kd-tree.

4.5.1 Integração Auxiliar Uniforme

A kd-tree construída na etapa anterior representa uma partição adaptativa do domínio da função de cena — ela é mais refinada em regiões de maior contraste —, mas, devido ao fato de $n - k$ dimensões não terem sido amostradas adaptativamente, pode ser que características da função de cena oriundas dessas dimensões não sejam corretamente integradas. O objetivo dessa etapa é utilizar a kd-tree como domínio de integração de Monte Carlo, de forma a obter mais informação acerca da função de cena sem a necessidade de refinar ainda mais a kd-tree. Para isso, pode-se reformular a integração por soma de Riemann efetuada em MDAS como integração de Monte Carlo.

Segundo a Equação 2.23, o valor do ponto (x, y) no plano da imagem é dado por

$$F(x, y) = \int \cdots \int f(x, y, u_2, \dots, u_n) du_2 \dots du_n, \quad (4.1)$$

onde n é a dimensionalidade da função de cena. Assumindo-se que a kd-tree possui dimensionalidade k ($2 \leq k \leq n$), o valor aproximado de $F(x, y)$ é dado por

$$F(x, y) \approx \sum_{\Omega \in P(x, y)} V_{\Omega} f_{\Omega}. \quad (4.2)$$

Para integrar f utilizando Monte Carlo e a kd-tree, utiliza-se um procedimento de amostragem que consiste em escolher aleatoriamente uma folha j da kd-tree (probabilidade $1/N$, onde N é o número total de folhas) e depois gerar um ponto aleatório, com distribuição uniforme, em seu interior (probabilidade $1/V_j$, onde V_j é o volume da folha escolhida). Esse procedimento resulta na seguinte PDF:

$$p(\mathbf{x}) = \frac{1}{NV(\mathbf{x})}, \quad (4.3)$$

onde $V(\mathbf{x})$ é o n -volume da folha que contém o ponto \mathbf{x} , e N é o número de folhas da kd-tree. Verifica-se que, no caso em que cada folha é amostrada exatamente uma vez, a integração de Monte Carlo por meio dessa PDF equivale à soma indicada na Equação 4.2:

$$\frac{1}{N} \sum_{i=1}^N \frac{f(\mathbf{X}_i)}{1/NV(\mathbf{X}_i)} = \sum_{i=1}^N f(\mathbf{X}_i)V(\mathbf{X}_i). \quad (4.4)$$

Para cada amostra \mathbf{x} gerada utilizando a PDF anterior (Equação 4.3), o valor de $f(\mathbf{x})$ é acumulado na única amostra que já está armazenada na folha correspondente. O valor final, portanto, é a média dessas amostras. A Figura 4.4 resume esse procedimento.

4.5.2 Integração Auxiliar por Metropolis

O procedimento de integração auxiliar uniforme permite agregar informação das $n - k$ dimensões que não são contempladas pela kd-tree. Porém, seria interessante tentar utilizar algum procedimento adaptativo nessas $n - k$ dimensões, pois efeitos de iluminação indireta de

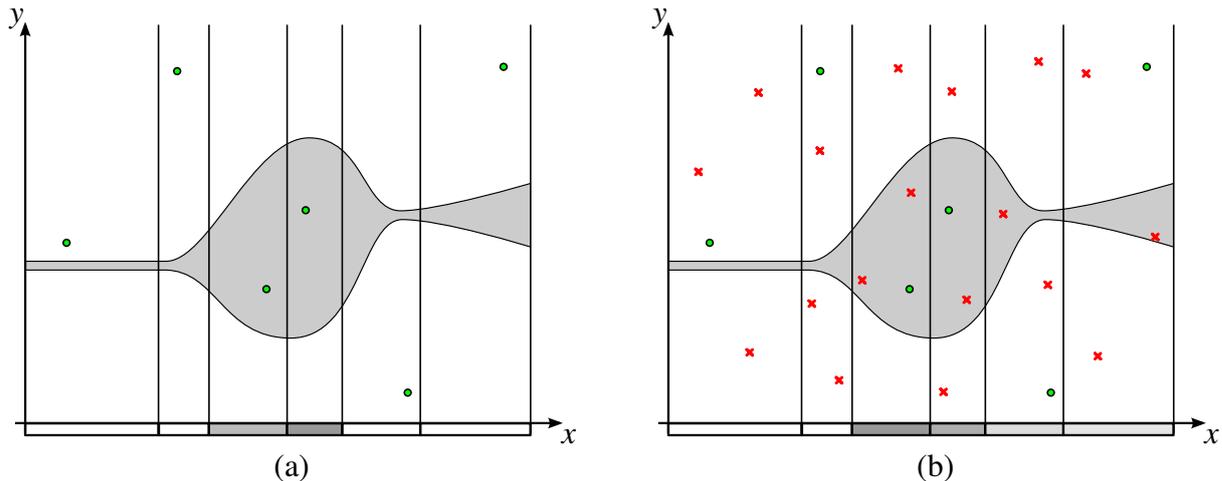


Figura 4.4: Integração Auxiliar: (a) Amostragem adaptativa pode não contabilizar certas características da função de cena. (b) Integração auxiliar permite melhorar as estimativas utilizando mais amostras em todas as regiões (cruzes vermelhas), sem necessidade de refinar a kd-tree.

difícil integração são geralmente oriundos dessas dimensões. Uma alternativa é utilizar MLT (*Metropolis Light Transport*), pois o seu procedimento de amostragem não sofre da “maldição da dimensionalidade” e não requer armazenamento extra de amostras.

O procedimento consiste em utilizar MLT conforme o Algoritmo 3.3 apresentado no Capítulo 3, porém, utilizando a kd-tree para gerar as k primeiras componentes da cada amostra e para armazenar os valores obtidos. O Algoritmo 4.6 sumariza o método. O procedimento `MDAS::geraNovaAmostra()` é o mesmo utilizado para gerar as amostras na etapa de refinamento adaptativo — retorna k componentes de uma amostra. Já o procedimento `MDAS::adicionaAmostra()` é responsável por armazenar os valores nas folhas da kd-tree — os valores são acumulados em uma variável auxiliar armazenada em cada folha da kd-tree. Cada chamada a `MDAS::adicionaAmostra()`, além de acumular o valor na variável auxiliar na folha correspondente, incrementa uma variável que conta quantas amostras foram acumuladas naquela folha.

Algorithm 4.6 MMDAS

- 1: computa amostras iniciais
 - 2: inicializa kd-tree com amostras iniciais
 - 3: calcula $b = \int I(x) dx$ com amostras iniciais
 - 4: $X \leftarrow$ amostra inicial para Metropolis
 - 5: **for** $i \leftarrow 1 \dots N$ **do**
 - 6: $Y[0 \dots k - 1] \leftarrow$ `MDAS::geraNovaAmostra()`
 - 7: $Y \leftarrow$ `muta`($X[k \dots n - 1]$)
 - 8: $a \leftarrow \min\{1, f(Y)/f(X)\}$
 - 9: `MDAS::adicionaAmostra`($X[0 \dots k - 1]$, $f(X)$, $I(X)$, b , $1 - a$)
 - 10: `MDAS::adicionaAmostra`($Y[0 \dots k - 1]$, $f(Y)$, $I(X)$, b , a)
 - 11: **if** `rand()` $< a$ **then**
 - 12: $X \leftarrow Y$
-

O Algoritmo 4.6 pode ser visto como um MLT em que as primeiras k componentes

das amostras sofrem a seguinte mutação: desconsidera o valor anterior e utiliza a kd-tree para gerar a nova amostra por meio da distribuição dada na Equação 4.3. Além disso, em vez de acumular os valores em pixels de uma imagem, os valores são acumulados nas folhas da kd-tree.

Ao fim dessa etapa, cada folha possui (além do valor já calculado na etapa de refinamento adaptativo) os valores acumulados por meio das chamadas a `MDAS::adicionaAmostra()`. Esses valores acumulados são usados na etapa de reconstrução para obter-se o valor final de cada pixel da imagem.

4.6 Reconstrução por nó

Conforme visto na Subseção 3.5.2, o algoritmo de reconstrução proposto em Hachisuka et al. (2008) consiste em percorrer cada pixel (x, y) e calcular seu valor somando-se os valores obtidos em cada célula da kd-tree que cruza o ponto (x, y) nas outras dimensões. Uma otimização para esse processo consiste em percorrer as folhas da kd-tree (em vez de pixels) e acumular o valor de cada folha em todos os pixels compreendidos por ela. O Algoritmo 4.7 resume esse processo.

Algorithm 4.7 Reconstrução

- 1: **for** cada folha l da kd-tree **do**
 - 2: $\mathbf{c} =$ centroide de l
 - 3: $v = f(\mathbf{c})$
 - 4: $v = v V_3(l)$ ▷ multiplica v pelo volume de l a partir da terceira dimensão

 - 5: **for** cada pixel p dentro dos limites de l **do**
 - 6: acumula v em p
-

Quando a etapa de integração auxiliar tiver sido utilizada, os valores acumulados em cada amostra devem ser corretamente contabilizados. No caso da integração auxiliar uniforme, o valor final de cada amostra é simplesmente a média de todas as amostras acumuladas. Já no caso da integração auxiliar por Metropolis, os valores finais são obtidos utilizando MLT em uma variável auxiliar armazenada em cada folha. O resultado é então calculado fazendo-se a média entre essa variável auxiliar e o valor da amostra original obtido por MDAS.

4.7 Considerações Finais

Este capítulo apresentou a técnica proposta neste trabalho, que consiste em uma variação de MDAS. O método visa superar algumas deficiências de MDAS quanto à acomodação de efeitos que requerem uma maior dimensionalidade. Para tanto, três etapas podem ser adicionadas ao método original: substituição de amostras, integração auxiliar e reconstrução por nó. Na etapa de integração auxiliar, por sua vez, duas alternativas foram apresentadas: integração auxiliar uniforme e integração auxiliar por Metropolis.

No próximo capítulo serão apresentados alguns resultados obtidos por meio da técnica proposta bem como algumas comparações com MDAS original.

5 RESULTADOS

Neste capítulo são apresentados alguns resultados obtidos com a técnica de renderização adaptativa proposta no Capítulo 4.

5.1 Considerações sobre implementação

O programa utilizado nos testes consiste em um sistema de renderização implementado em C++ e baseado na arquitetura do sistema PBRT (PHARR; HUMPHREYS, 2010). As classes de interesse para este trabalho são: *Sampler*, *Renderer* e *Integrator*. A classe *Sampler* é responsável por gerar amostras (pontos no hipercubo) que são usadas pelas demais classes. Dada uma amostra gerada por *Sampler*, as demais classes necessitam de certas componentes dessa amostra para executarem suas funções (por exemplo, *Camera* utiliza de duas a cinco componentes para gerar um raio para o algoritmo de *ray tracing*). A classe *Renderer* gerencia o processo de renderização, repassando amostras para outras classes e obtendo os resultados. *Integrator* implementa o cálculo da iluminação (por exemplo, *Ray-tracing*, *Path-tracing*, etc.). As técnicas de amostragem utilizadas nos testes foram implementadas como subclasses de *Sampler*.

A Tabela 5.1 descreve os parâmetros utilizados na implementação de MDAS com as modificações propostas. Os valores entre parênteses são os valores *default* utilizados em todos os testes, a menos que seja dito o contrário.

Tabela 5.1: Descrição dos parâmetros de MDAS com as modificações propostas

Parâmetro	Descrição
N_i (1)	Número de amostras utilizadas na etapa de amostragem inicial
N_s	Número de amostras utilizadas na etapa de substituição de amostras
N_a	Número de amostras utilizadas na etapa de refinamento adaptativo da kd-tree
N_x	Número de amostras utilizadas na etapa de integração auxiliar
S_h	Fatores de escala das dimensões do hipercubo a partir da terceira (as duas primeiras são calculadas conforme a Subseção 3.5.1)
C_{max} (6)	Capacidade máxima das folhas
N_c (4)	Número de candidatos na geração de amostras dentro das folhas
N_v (15)	Quantidade de vizinhos mais próximos utilizados para o cálculo dos gradientes e tensores
G_{max} (5)	Tamanho máximo dos vetores gradiente

5.2 Cenas

A seguir, são descritas as cenas utilizadas nos testes.

5.2.1 Fatia no hipercubo

Esta cena não utiliza um *Integrator*, ou seja, não requer cálculo de iluminação — as amostras geradas pelo *Sampler* (pontos no hipercubo) são utilizadas diretamente para computar uma função. A função corresponde a uma fatia plana perpendicular a um dos eixos do hipercubo, de acordo com a seguinte definição:

$$f(\mathbf{x}) = \begin{cases} 10 & l_{min} \leq x_i \leq l_{max} \\ 0 & \text{caso contrário,} \end{cases} \quad (5.1)$$

onde $l_{max} - l_{min}$ é a espessura da fatia, i é o eixo perpendicular à fatia e x_i é a componente i do vetor \mathbf{x} . A imagem ideal obtida com tal função deve apresentar apenas uma cor: um tom de cinza de luminância $10(l_{max} - l_{min})$. Todos os testes feitos com essa cena utilizaram os parâmetros $l_{min} = 0,2$; $l_{max} = 0,21$; $i = 4$ e $n = 4$. Essa configuração representa um desafio para o método MDAS pois apenas uma pequena porção do domínio possui contribuição não nula.

5.2.2 Esferas

Esta cena também consiste em uma função calculada diretamente a partir das amostras geradas pelo *Sampler*. A função consiste em quatro hiperesferas de raios R_i ($1 \leq i \leq 4$), centradas nos pontos \mathbf{c}_i , dada pela seguinte definição:

$$f(\mathbf{x}) = \begin{cases} v_1 & |\mathbf{x} - \mathbf{c}_1| \leq R_1 \text{ ou} \\ v_2 & |\mathbf{x} - \mathbf{c}_2| \leq R_2 \text{ ou} \\ v_3 & |\mathbf{x} - \mathbf{c}_3| \leq R_3 \text{ ou} \\ v_4 & |\mathbf{x} - \mathbf{c}_4| \leq R_4 \\ 0 & \text{se nenhum dos casos anteriores.} \end{cases} \quad (5.2)$$

Cada esfera provê uma característica a ser capturada pela técnica de amostragem, portanto, a técnica MDAS terá mais dificuldade em capturar informação sobre as esferas, conforme seus raios diminuem. Em todos os testes feitos com essa cena, utilizaram-se os seguintes parâmetros:

$c_1 = (0,26; 0,26; 0,26)$	$R_1 = 0,25$	$v_1 = 5$
$c_2 = (0,75; 0,75; 0,75)$	$R_2 = 0,125$	$v_2 = 20$
$c_3 = (0,75; 0,75; 0,75)$	$R_3 = 0,0625$	$v_3 = 80$
$c_4 = (0,25; 0,25; 0,25)$	$R_4 = 0,03125$	$v_4 = 320$

Com esses parâmetros, pode-se avaliar até que ponto a técnica de amostragem consegue capturar características da cena. A Figura 5.1 mostra a imagem de referência para esta cena.

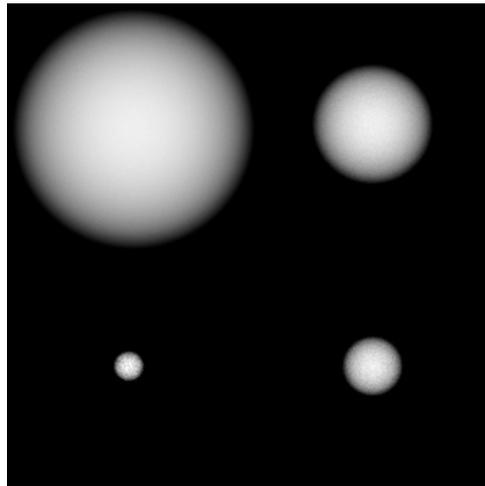


Figura 5.1: Imagem de referência da cena “esferas multidimensionais”.

5.2.3 Fendas

Esta cena consiste em um plano iluminado lateralmente por uma luminária parcialmente bloqueada por quatro pilares, conforme mostra o esquema da Figura 5.2a. A cena fornece um simples caso de iluminação direta, necessitando portanto de quatro dimensões (pixel + ponto na luminária). A técnica de amostragem adaptativa deve ser capaz de capturar luz das partes expostas da luminária.

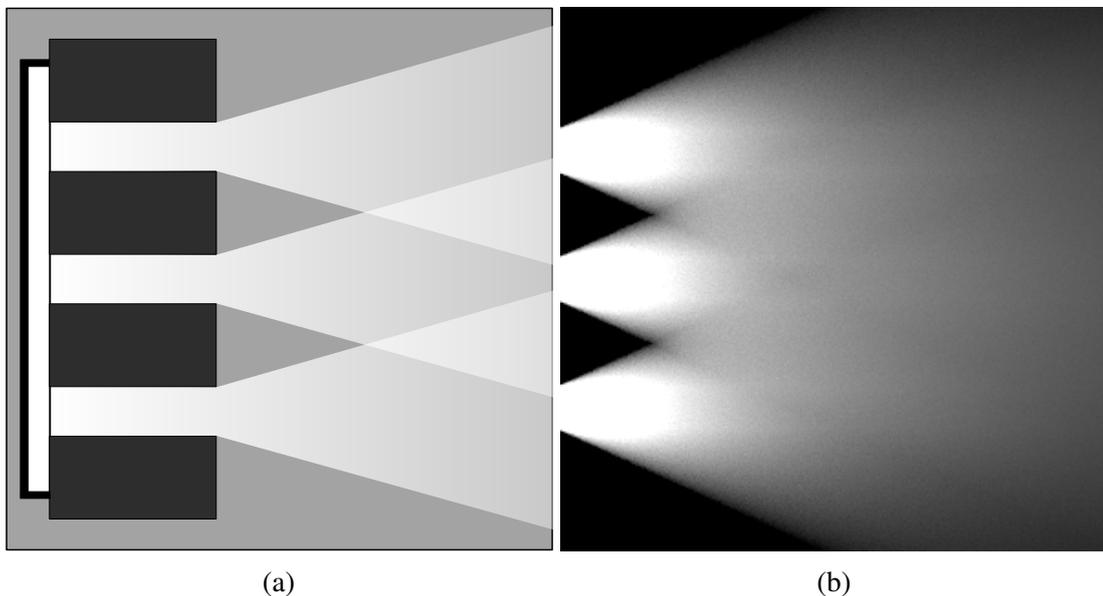


Figura 5.2: Cena “Fendas”: (a) Configuração da cena. (b) Imagem de referência.

5.2.4 Macaco

Esta cena utiliza iluminação direta com área (*area light*), profundidade de campo (*depth-of-field*) e borrão de movimento (*motion blur*), necessitando de sete dimensões em cada

amostra (Figura 5.3). O borrão encima do macaco consiste em uma esfera ligeiramente fora de foco se movimentando. Deseja-se que a técnica de amostragem seja capaz de capturar corretamente tanto o desfocagem quando o borrão de movimento, enquanto preserva detalhes do tabuleiro e do macaco na região focada.

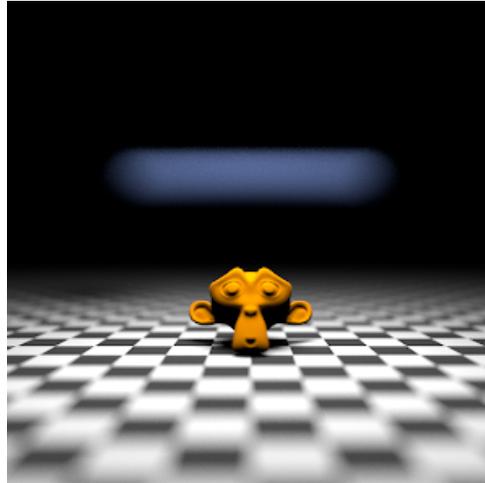


Figura 5.3: Imagem de referência da cena “macaco”.

5.2.5 *Cornell box*

A cena consiste no *Cornell box* clássico, renderizado com *Path-tracing* utilizando-se no máximo caminhos de luz de profundidade dez (Figura 5.5). Um domínio de alta dimensionalidade como este torna difícil a utilização de MDAS no caso em que $k = n$.

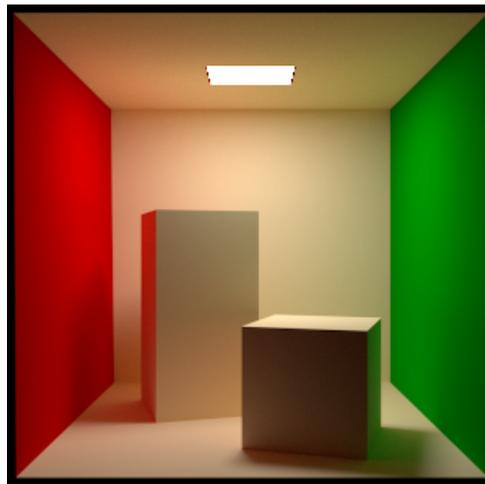


Figura 5.4: Imagem de referência da cena *Cornell box*.

5.2.6 *Cornell box com esfera*

Esta cena possui a mesma configuração do *Cornell box* clássico anterior, com exceção de uma esfera de vidro que foi adicionada próximo à face verde. A esfera produz *caustics* nas

superfícies próximas a ela, e, por isso, exige grande concentração de amostras e representa uma significativa fonte de ruído proveniente de uma dimensão fora do alcance da kd-tree.

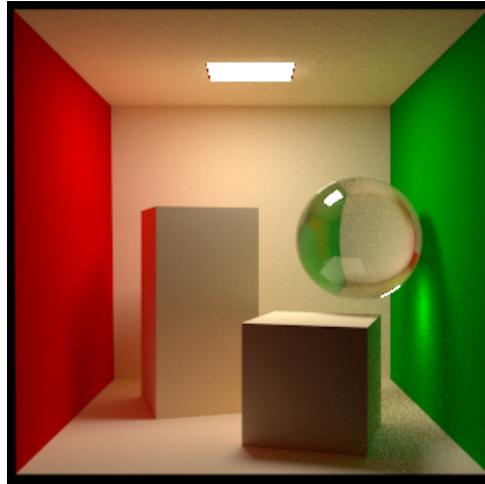


Figura 5.5: Imagem de referência da cena “*Cornell box* com esfera”.

A seguir são descritos os resultados obtidos com a adição de cada uma das etapas propostas ao método MDAS.

5.3 Substituição de amostras

5.3.1 Fatia no hipercubo

Dadas as características da cena, o processo de amostragem adaptativa precisa detectar as fronteiras representadas pela fatia na quarta dimensão, depositando mais amostra nessa região. No primeiro teste (Figura 5.6) foi utilizado MDAS e substituição de amostras, no qual a dimensionalidade da kd-tree (k) foi definida como sendo igual à dimensionalidade da função de cena (n). A técnica consegue, embora de forma grosseira, detectar a região de importância do domínio. Esperava-se que a substituição de algumas amostras tornasse essa detecção mais robusta nesse caso, mas não houve variação consistente na qualidade das imagens.

Para avaliar o procedimento de substituição de amostras em dimensões não tratadas pela kd-tree, a cena “Fatia no hipercubo” foi renderizada com $k = 3$, ou seja, deixando a dimensão que delimita a região de contribuição não nula do domínio fora do alcance da kd-tree. Conforme mostra a Figura 5.7a, MDAS obteve quase nenhuma informação sobre a região de contribuição não nula da cena, resultando em uma imagem quase inteiramente preta. A adição da etapa de substituição conseguiu fornecer mais informação sobre a dimensão não tratada pela kd-tree, entretanto, os resultados não parecem convergir para o valor correto esperado para essa cena, conforme pode ser verificado no gráfico de erro quadrático médio (Figura 5.8b). O motivo desse erro deve ser a alteração brusca na distribuição das amostras em dimensões não tratadas pela kd-tree, e que, por isso, não têm sua contribuição corrigida adequadamente.

Uma outra característica observada foi a baixa taxa de aceitação de amostras de substituição. Conforme visto no Algoritmo 4.5, uma amostra de substituição só substitui uma

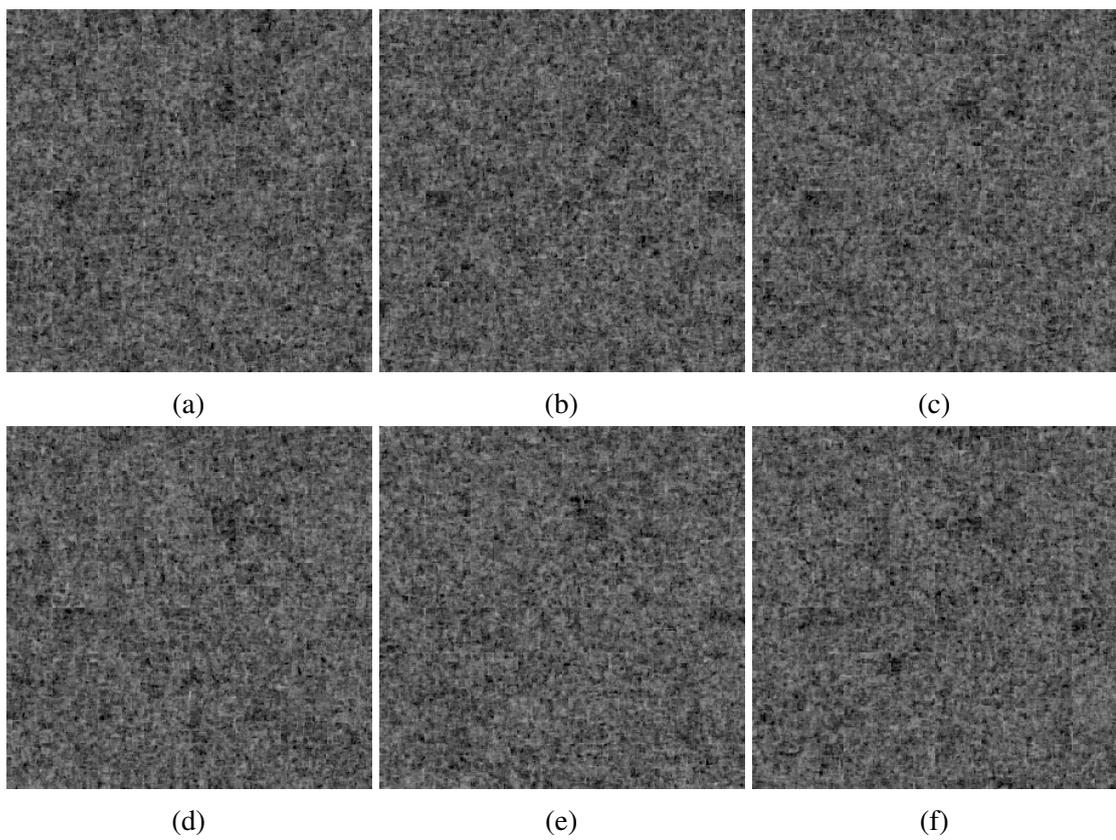


Figura 5.6: Substituição de amostras na cena “Fatia no hipercubo” com $k = n$: (a) MDAS comum. (b) Substituição de amostras com uma amostra por pixel em média (1 app). (c) Substituição com 5 app. (d) Substituição com 10 app. (e) Substituição com 100 app.

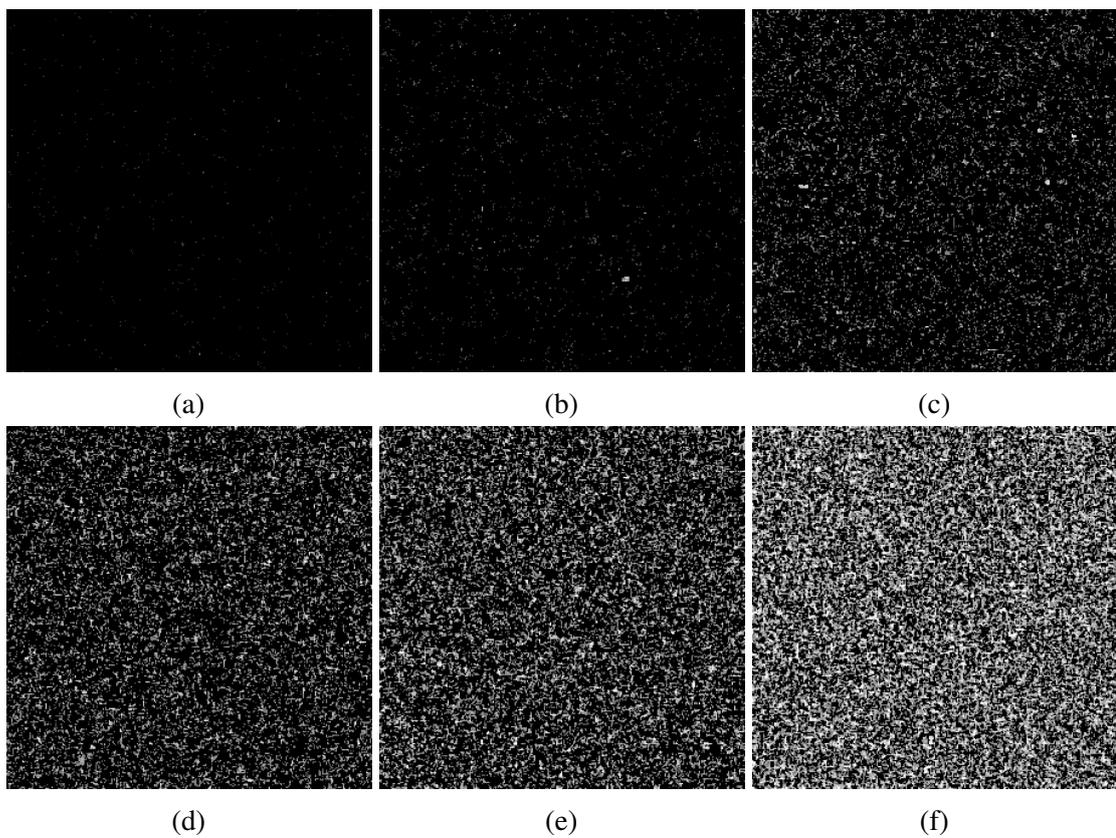


Figura 5.7: Substituição de amostras na cena “Fatia no hipercubo” com $k < n$: (a) MDAS comum. (b) Substituição de amostras com uma amostra por pixel em média (1 app). (c) Substituição com 5 app. (d) Substituição com 10 app. (e) Substituição com 15 app. (f) 100 app.

amostra da kd-tree caso passe no teste da linha 5 do algoritmo. A Figura 5.8a mostra a evolução da taxa de aceitação das substituições nos casos $k = n$ e $k < n$. Os parâmetros de MDAS utilizados tanto no caso $k = n$ quanto no caso $k < n$ foram: $N_a = 10$ amostras por pixel (app) e $\mathbf{S}_h = [1/4; 1/4]$.

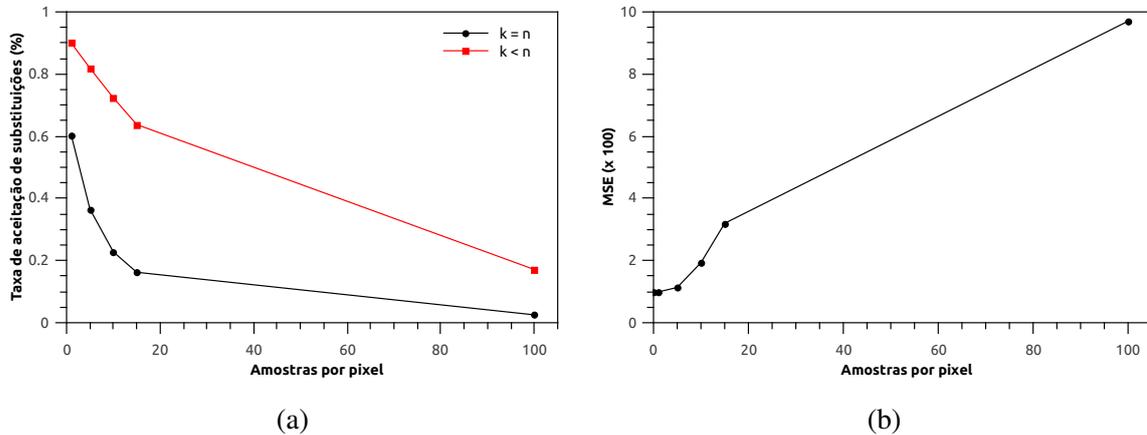


Figura 5.8: Substituição de amostras na cena “Fatia no hiper cubo”: (a) Taxa de aceitação das amostras de substituição. (b) Erro quadrático médio.

5.3.2 Esferas

A Figura 5.11 mostra os resultados com substituição de amostras na cena “Esferas” com $k = n$. Observou-se que mesmo com a kd-tree contemplando todas as dimensões da cena, MDAS não foi capaz de detectar a esfera de menor raio com os parâmetros utilizados. A adição da etapa de substituição permitiu a detecção de todas as esferas, embora com bastante ruído.

A taxa de aceitação das amostras de substituição apresentou comportamento parecido com o caso da cena “Fatia no hiper cubo”, mantendo-se abaixo de 1%. A Figura 5.12 exibe a taxa de aceitação bem como a evolução do erro quadrático médio.

Para o caso $k = 3$, as esferas se tornam ainda mais difíceis de serem detectadas, o que aumenta o ruído nas imagens (Figura 5.11). Da mesma forma que no caso anterior, a adição da etapa de substituição permitiu a detecção da esfera de menor raio, embora não tenha havido melhora consistente na qualidade da imagem como um todo. A Figura 5.12 mostra a evolução da taxa de aceitação das substituições e do erro quadrático médio. Os parâmetros de MDAS utilizados para os dois casos ($k = n$ e $k < n$) foram: $N_a = 10$ app e $\mathbf{S}_h = [1/4; 1/4]$.

5.3.3 Fendas

Este é mais um caso onde a dimensionalidade da kd-tree é igual à dimensionalidade da cena ($k = n$), mas desta vez é utilizado *ray tracing* para calcular a iluminação. Os resultados seguem o mesmo padrão do teste com a cena “Fatia no hiper cubo” com $k = n$, ou seja, sem variação consistente na qualidade da imagem. A Figura 5.13 exibe os resultados e a Figura 5.14

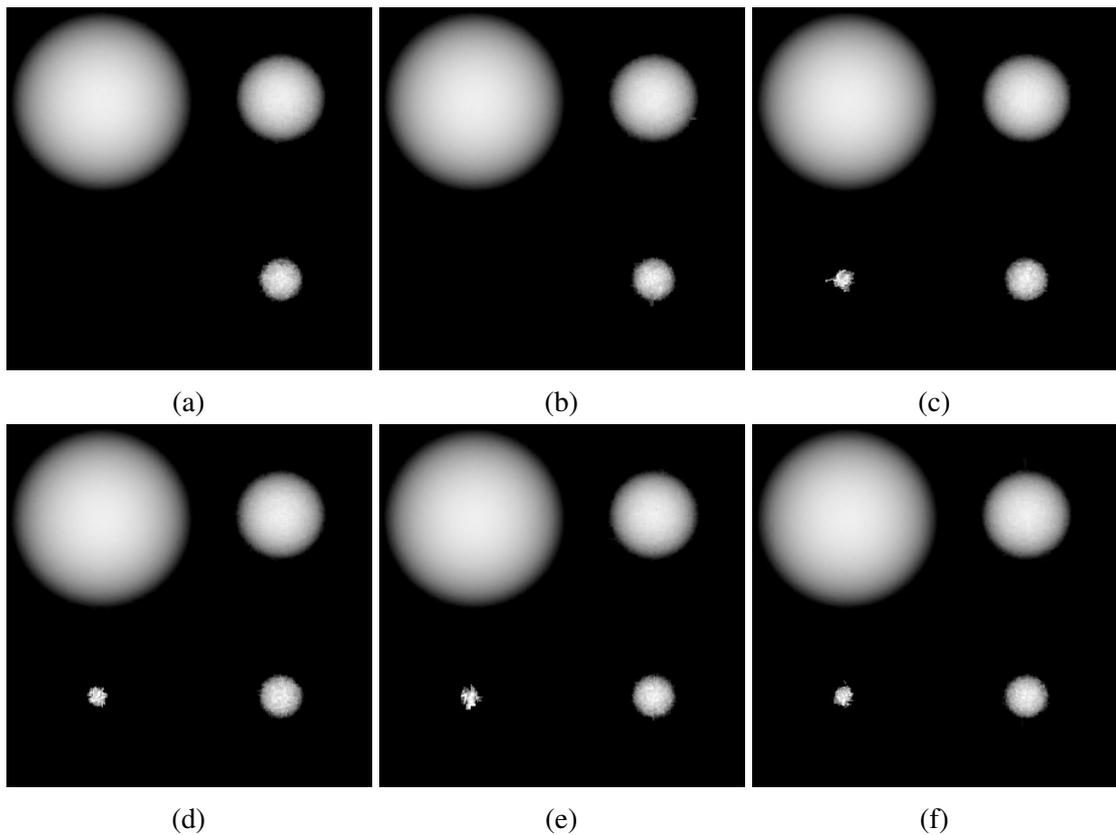


Figura 5.9: Substituição de amostras na cena “Esferas” com $k = n$: (a) MDAS comum. (b) Substituição com 1 app. (c) Substituição com 5 app. (d) Substituição com 10 app. (e) Substituição com 15 app. (f) Substituição com 100 app.

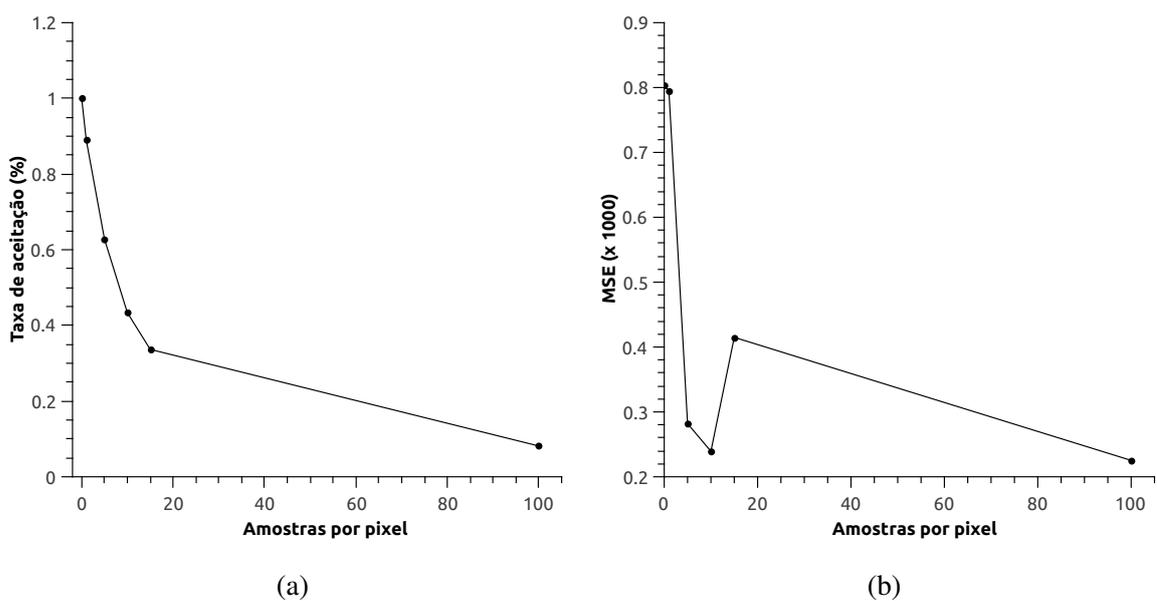


Figura 5.10: Substituição de amostras na cena “Esferas” com $k = n$: (a) Taxa de aceitação das amostras de substituição. (b) Erro quadrático médio (EQM) das imagens obtidas.

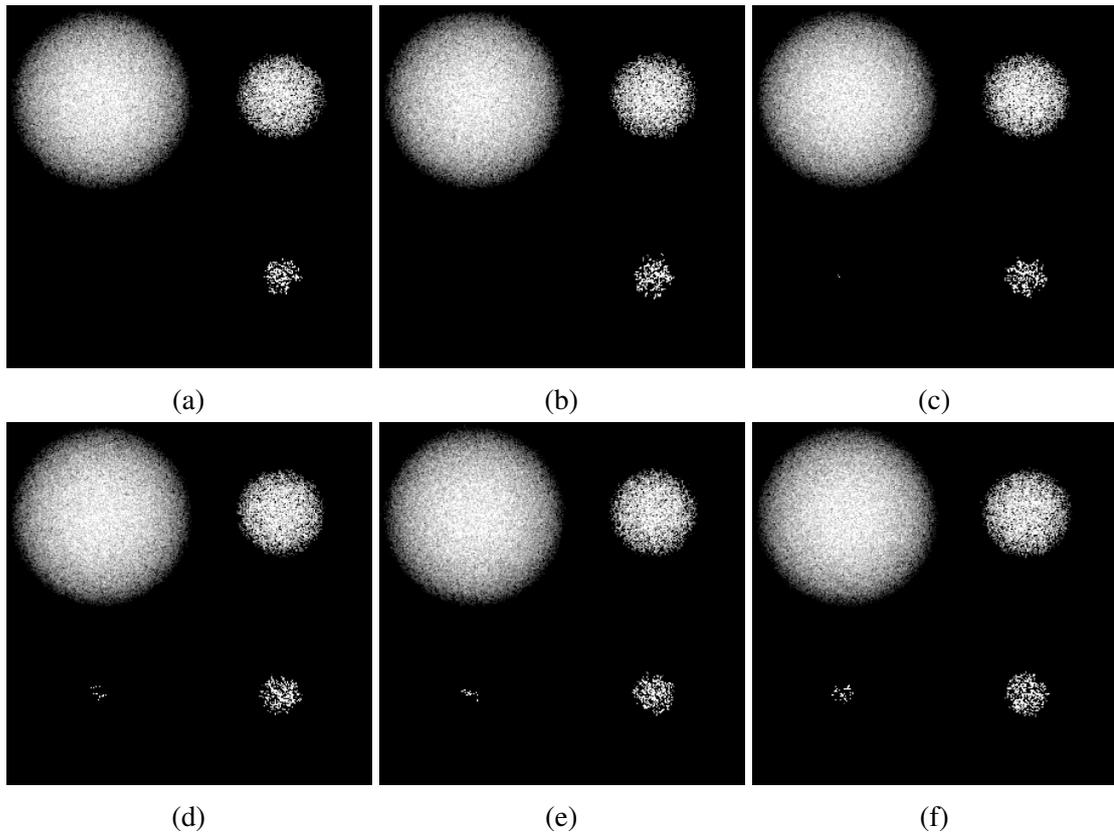


Figura 5.11: Substituição de amostras na cena “Esferas” com $k < n$: (a) MDAS comum. (b) Substituição com 1 app. (c) Substituição com 5 app. (d) Substituição com 10 app. (e) Substituição com 15 app. (f) Substituição com 100 app.

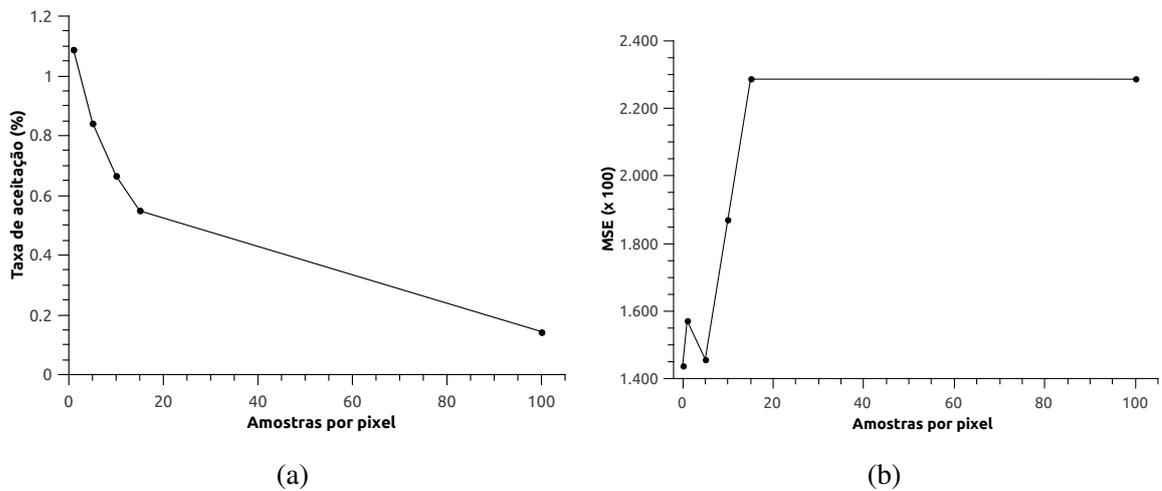


Figura 5.12: Substituição de amostras na cena “Esferas” com $k < n$: (a) Taxa de aceitação das amostras de substituição. (b) Erro quadrático médio (EQM) das imagens obtidas.

mostra a evolução da taxa de aceitação de mutações e do erro quadrático médio. Os parâmetros de MDAS foram utilizados: $N_a = 20$ app e $\mathbf{S}_h = [1/4; 1/4]$.

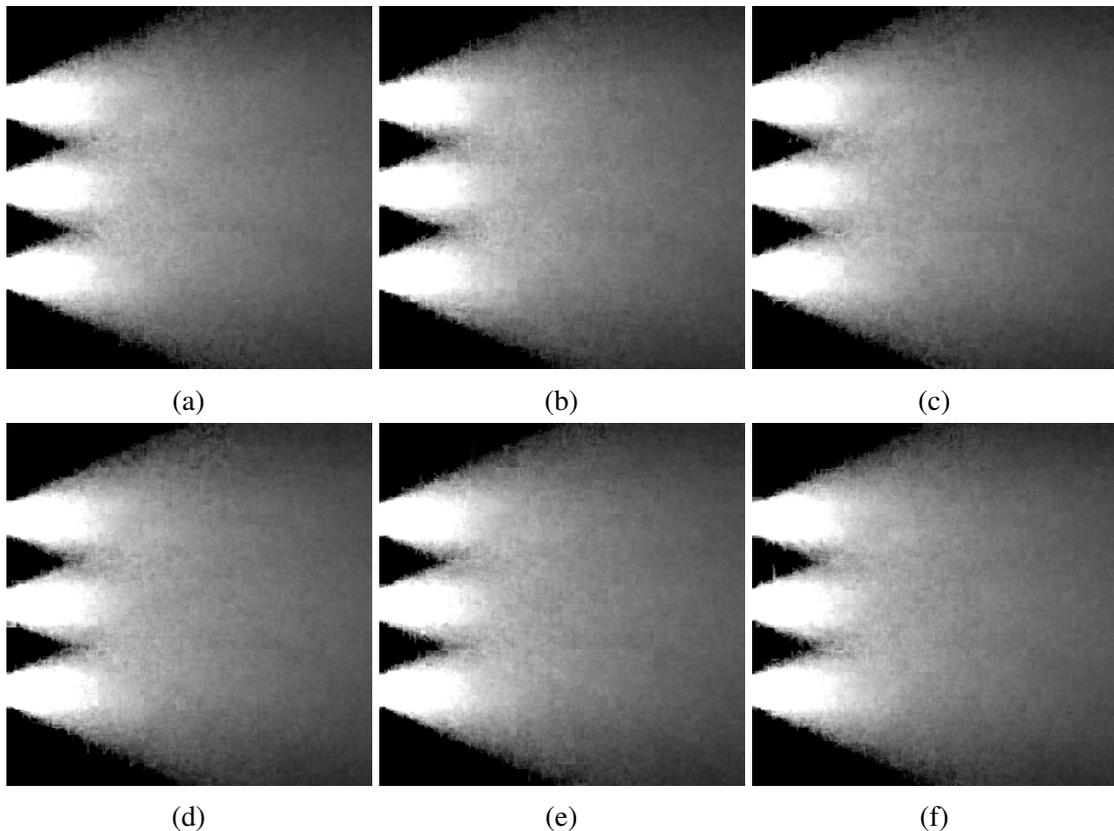


Figura 5.13: Substituição de amostras na cena “Fendas”: (a) MDAS comum. (b) Substituição com 10 app. (c) 20 app (d) 30 app. (e) 40 app. (f) 50 app.

5.4 Integração Auxiliar Uniforme

A seguir são exibidos alguns resultados obtidos com a etapa de integração auxiliar uniforme. Os testes seguem o mesmo padrão da seção anterior — compara-se MDAS comum com MDAS modificado. A integração auxiliar por Metropolis não chegou a ser implementada a tempo, portanto, esta alternativa fica como trabalho futuro.

5.4.1 Fatia

Ao contrário dos resultados obtidos com a etapa de substituição de amostras para o caso $k = n$, a etapa de integração auxiliar conseguiu melhorar a qualidade da imagem de forma consistente para a mesma cena. A Figura 5.15 exhibe as imagens obtidas e a Figura 5.16 mostra a variação no erro médio quadrático de acordo com a quantidade de amostras utilizadas.

No caso $k < n$, mesmo a etapa de integração auxiliar não interferindo na construção da kd-tree, as amostras extras acumuladas em cada uma das folhas obtiveram mais informação sobre a região de contribuição não nula da quarta dimensão. A Figura 5.17 exhibe as imagens

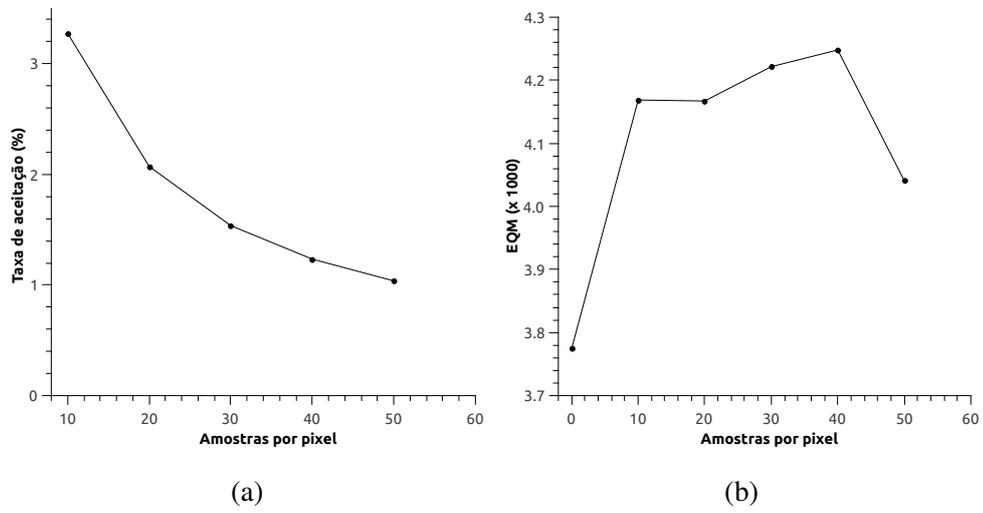


Figura 5.14: Substituição de amostras na cena “Fendas”: (a) Taxa de aceitação das amostras de substituição. (b) Erro quadrático médio (EQM) das imagens obtidas.

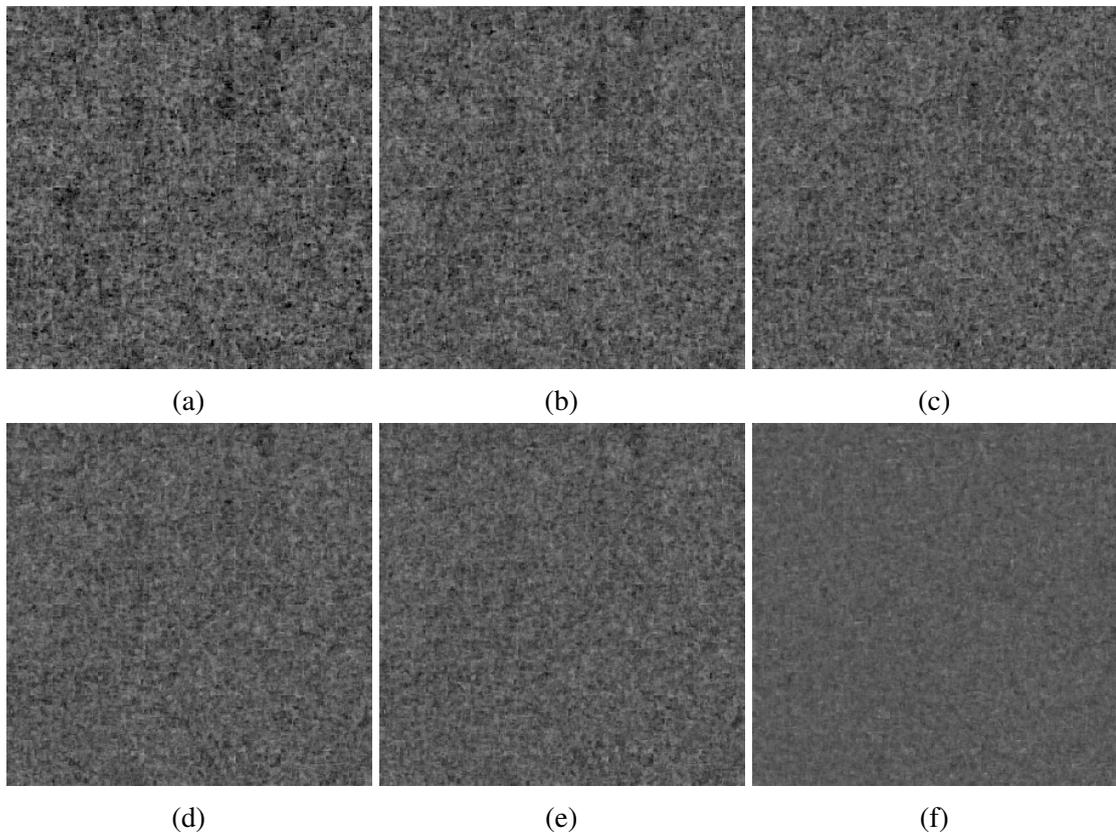


Figura 5.15: Integração auxiliar uniforme na cena “Fatia no hipercubo” com $k = n$: (a) MDAS comum. (b) 5 app. (c) 10 app. (d) 15 app. (e) 20 app. (f) 100 app.

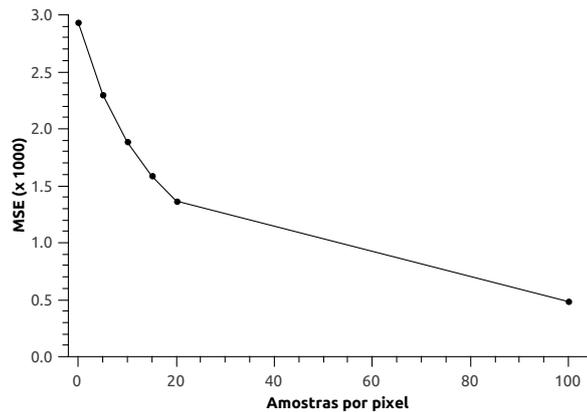


Figura 5.16: Erro quadrático médio (EQM) das imagens obtidas com integração auxiliar uniforme na cena “Fatia no hipercubo” com $k < n$.

obtidas e a Figura 5.18 mostra a variação no erro médio quadrático de acordo com a quantidade de amostras utilizada. Observa-se também que, nesta cena, o erro quadrático médio aumenta nas primeiras imagens. Isso ocorre porque as primeiras imagens, embora mostrem que mais informação está sendo obtida, apresentam mais ruído quando comparadas com MDAS comum.

A utilização da etapa de integração auxiliar em conjunto com a etapa de substituição de amostras também foi testada para esta cena. A Figura 5.19 mostra cortes das imagens obtidas (corte 3) em comparação com as etapas isoladas (cortes 1 e 2). Observa-se que a etapa de integração auxiliar uniforme ganha maior poder de convergência quando precedida da etapa de substituição de amostras. Os parâmetros de MDAS utilizados em todos os casos ($k = n$ e $k < n$) foram: $N_a = 10$ amostras por pixel (app) e $S_h = [1/4; 1/4]$.

5.4.2 Esferas

No caso da cena “Esfera” com integração auxiliar uniforme e $k = n$, apenas no caso extremo de 100 amostras por pixel observou-se alguma informação da esfera de menor raio (Figura 5.20). A evolução do erro quadrático médio neste caso também mostrou algum aumento de ruído nos primeiros testes (Figura 5.21).

Para o caso com $k < n$, o mesmo aconteceu com relação à esfera de menor raio, ou seja, para obter informação significativa desta região, uma grande quantidade de amostras foi necessária (Figura 5.22). Por outro lado, conforme pode ser visto nas imagens e no gráfico de erro quadrático médio (Figura 5.23), ouve redução consistente de ruído nas regiões bem detectadas (esferas maiores). Os parâmetros de MDAS utilizados em ambos os casos foram: $N_a = 10$ app e $S_h = [1/4; 1/4]$.

5.4.3 Fendas

Para a cena “Fendas”, a integração auxiliar uniforme promoveu uma diminuição sutil no ruído (Figura 5.24). O gráfico do erro quadrático médio ressalta esse resultado (Figura 5.25).

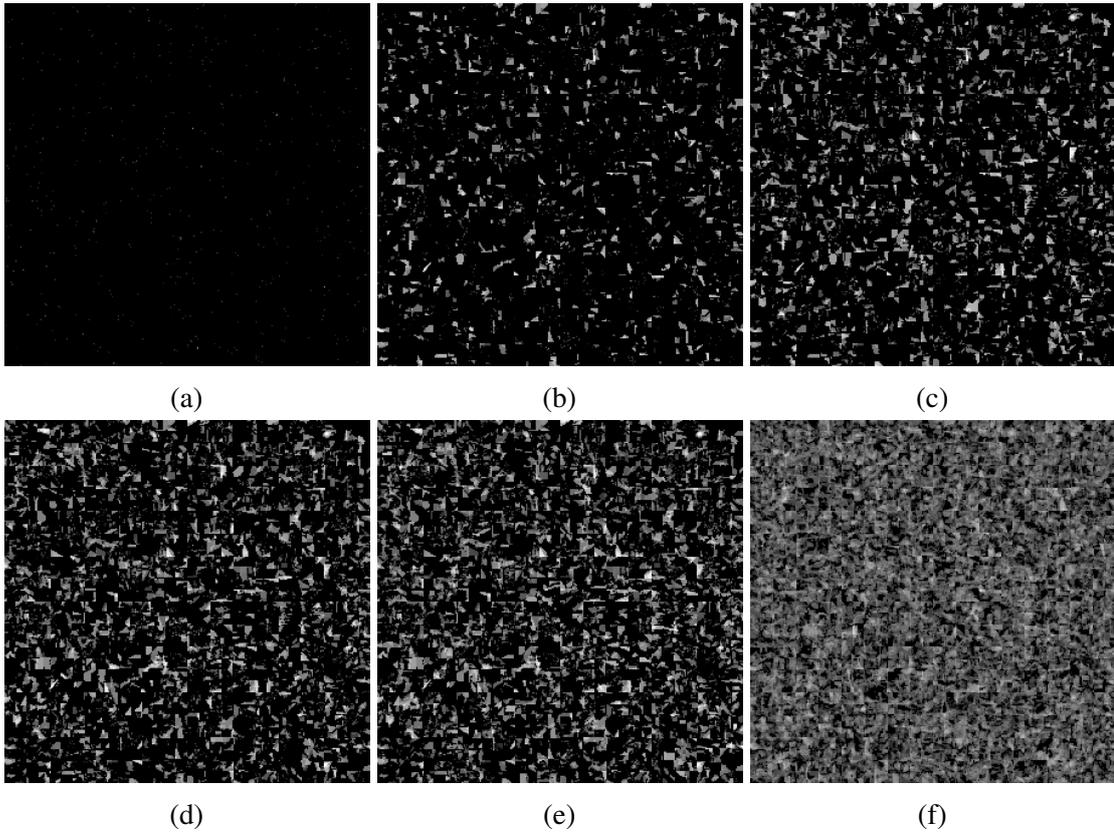


Figura 5.17: Integração auxiliar uniforme na cena “Fatia no hipercubo” com $k < n$: (a) MDAS comum. (b) 5 app. (c) 10 app. (d) 15 app. (e) 20 app. (f) 100 app.

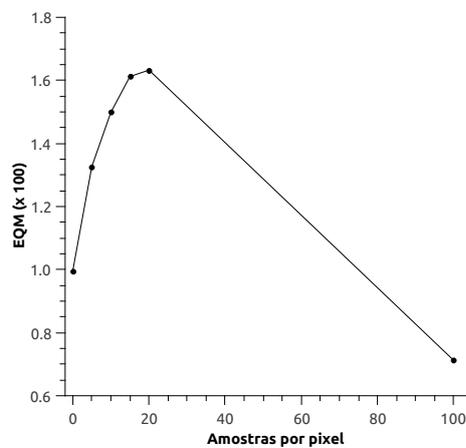


Figura 5.18: Erro quadrático médio (EQM) das imagens obtidas com integração auxiliar uniforme na cena “Fatia no hipercubo” com $k < n$.

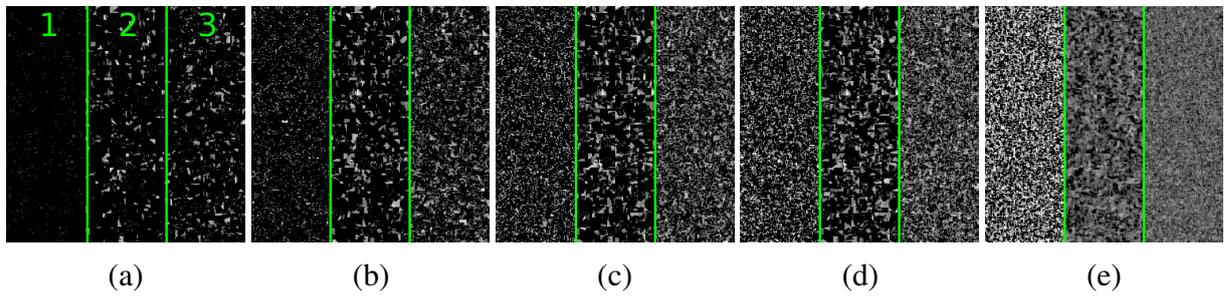


Figura 5.19: Comparação entre as imagens obtidas da cena “Fatia no hipercubo” utilizando-se substituição de amostras (1), integração auxiliar uniforme (2) e ambos (3). Foram utilizados os seguintes quantidades de amostras na utilização em conjunto das duas técnicas: (a) 1 app (substituição) + 5 app (integração auxiliar). (b) 5 + 10 app. (c) 10 + 15 app. (d) 15 + 20 app. (e) 100 + 100 app.(1 + 5)

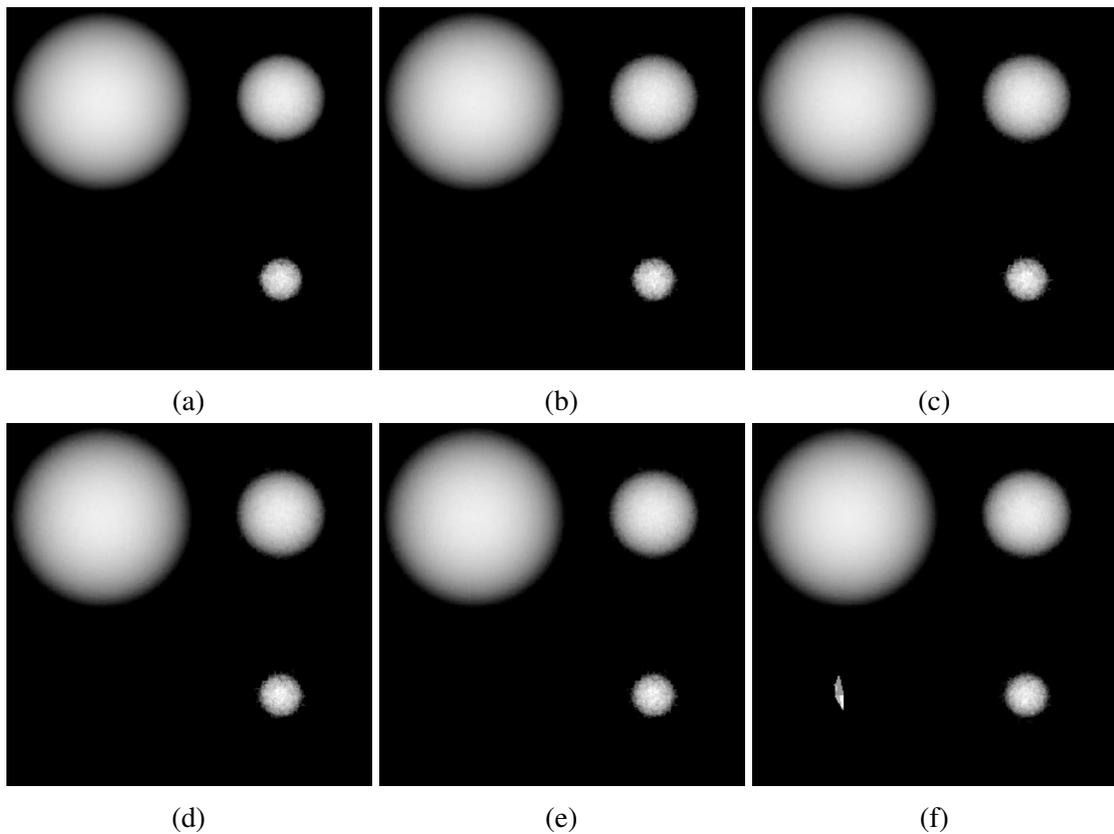


Figura 5.20: Integração auxiliar uniforme na cena “Esferas” com $k = n$: (a) MDAS comum. (b) 5 app. (c) 10 app. (d) 15 app. (e) 20 app. (f) 100 app.

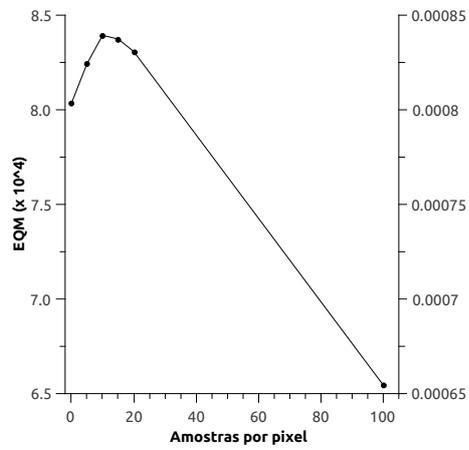


Figura 5.21: Erro quadrático médio (EQM) das imagens obtidas com integração auxiliar uniforme na cena “Esferas” com $k = n$.

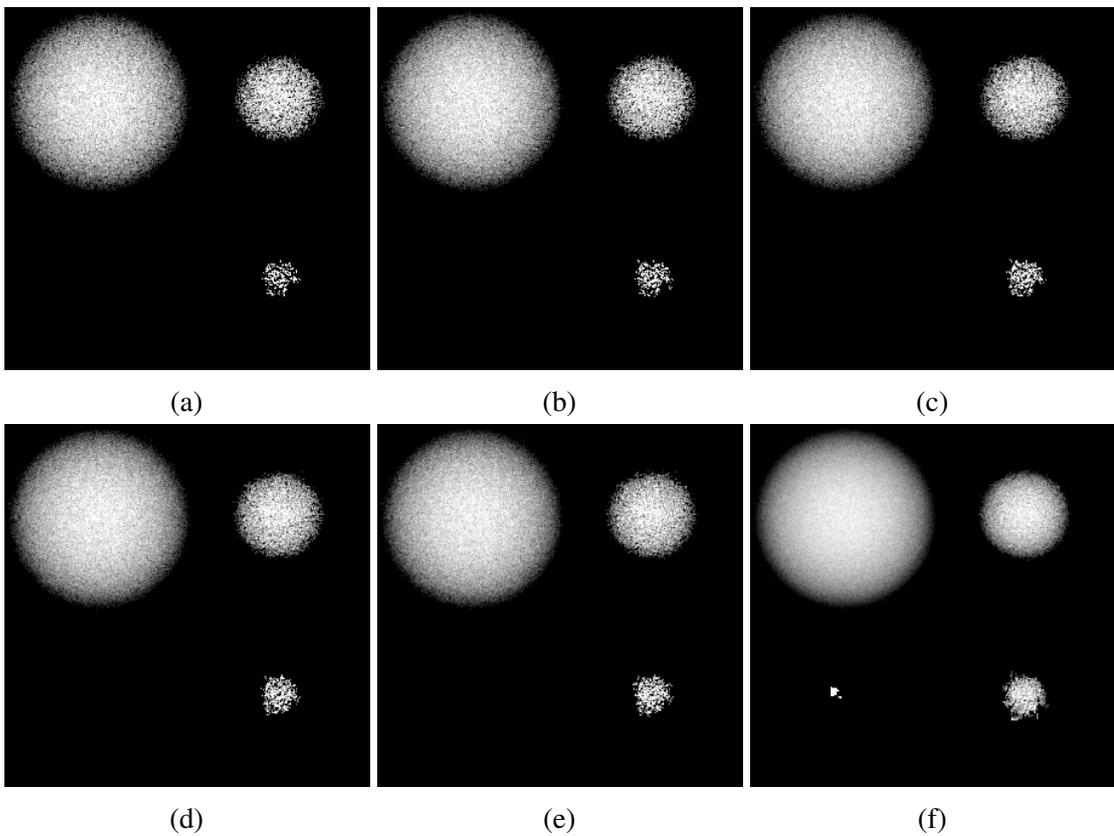


Figura 5.22: Integração auxiliar uniforme na cena “Esferas” com $k < n$: (a) MDAS comum. (b) 5 app. (c) 10 app. (d) 15 app. (e) 20 app. (f) 100 app.

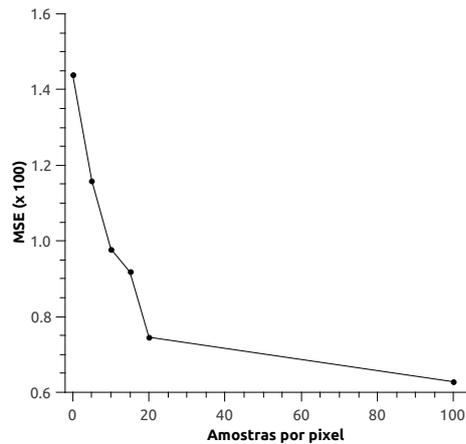


Figura 5.23: Erro quadrático médio (EQM) das imagens obtidas com integração auxiliar uniforme na cena “Esferas” com $k < n$.

Os parâmetros de MDAS utilizados foram: $N_a = 20$ amostras por pixel (app) e $\mathbf{S}_h = [1/4; 1/4]$.

5.4.4 Cornell box

Na cena Cornell box, a etapa de integração auxiliar uniforme promoveu uma diminuição do ruído nas regiões mais suaves da imagem (Figura 5.26). Entretanto, conforme aumenta-se a quantidade de amostras, alguns artefatos próximos às arestas começam a ser ressaltados, o que interfere no gráfico de erro quadrático médio (Figura 5.25). Os parâmetros de MDAS utilizados foram: $N_a = 30$ app, $C_{max} = 4$, $N_c = 3$ e $\mathbf{S}_h = [1/26; 1/26]$.

5.4.5 Cornell box com esfera

Na cena “Cornell box com esfera“, a etapa de integração auxiliar uniforme obteve resultados similares à cena “Cornell box“ tradicional — há uma ligeira melhora no ruído certas regiões, mas há também o surgimento de artefatos que se tornam ressaltados conforme aumenta-se a quantidade de amostras (Figura 5.28), o que pode ser verificado no gráfico de erro quadrático médio (Figura 5.29). Os parâmetros de MDAS utilizados foram os mesmos da cena “Cornell box“.

5.5 Reconstrução por nó

Para avaliar a etapa de reconstrução por nó, as cenas “Cornell box” e “Macaco” foram renderizadas, e o tempo gasto em reconstrução foi comparado com o de MDAS comum, que utiliza reconstrução por pixel (Figuras 5.30 e 5.31). Observou-se uma redução no tempo de reconstrução, entretanto, houve uma ligeira piora na qualidade das imagens — as imagens obtidas com reconstrução por nó são ligeiramente mais borradas e apresentam artefatos retangulares provenientes da projeção de nós que cobrem grandes blocos de pixels. Os parâmetros de MDAS

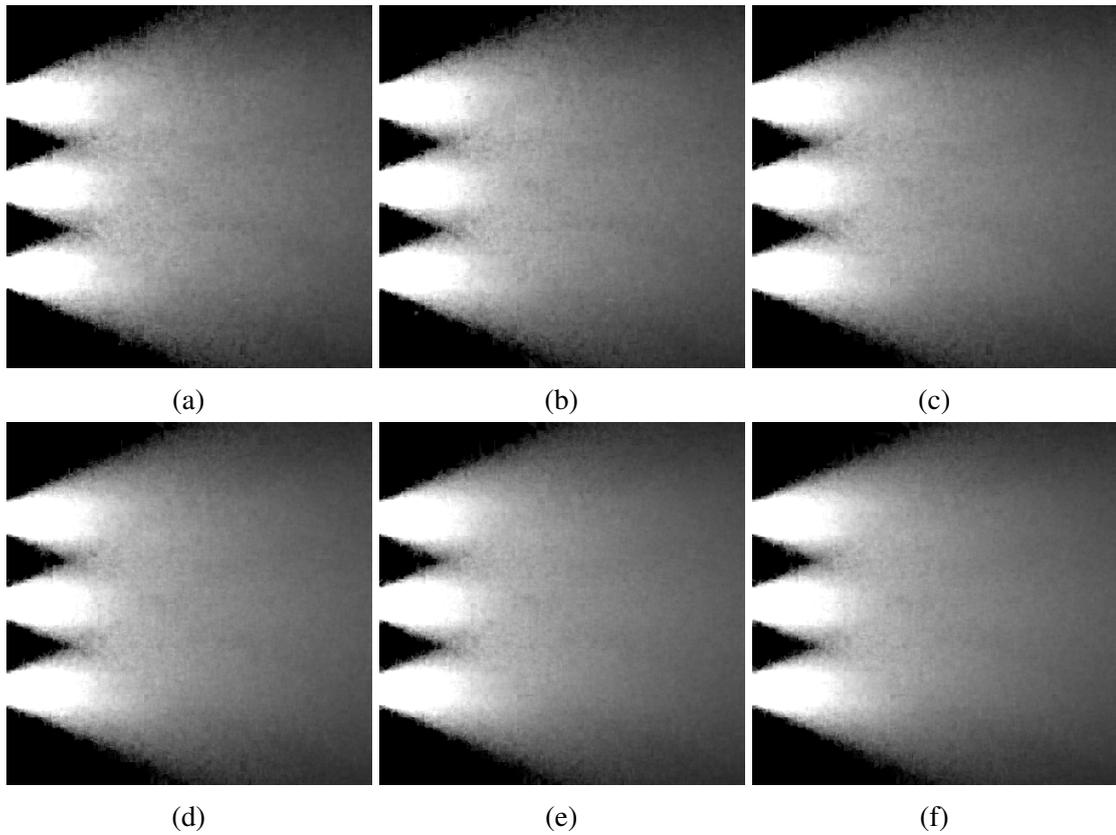


Figura 5.24: Integração auxiliar uniforme na cena “Fendas”: (a) MDAS comum. (b) 10 app. (c) 20 app. (d) 30 app. (e) 40 app. (f) 50 app.

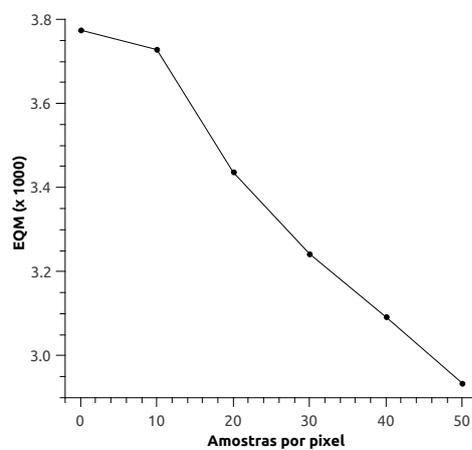


Figura 5.25: Erro quadrático médio das imagens obtidas da cena “Fendas” com integração auxiliar uniforme.

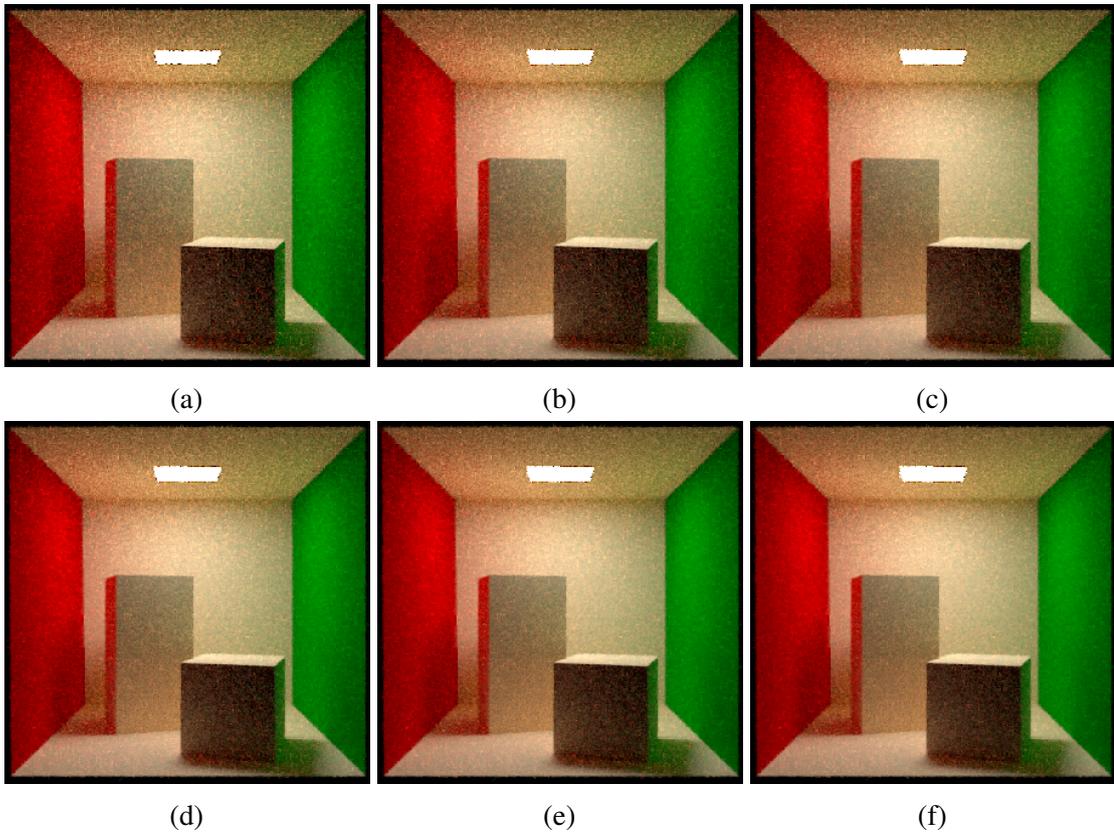


Figura 5.26: Integração auxiliar uniforme na cena Cornell box: (a) MDAS comum. (b) 10 app. (c) 20 app. (d) 30 app. (e) 40 app. (f) 50 app.

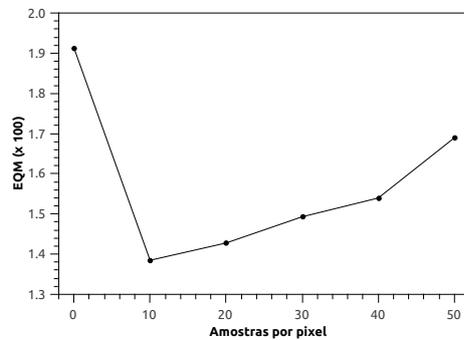


Figura 5.27: Erro quadrático médio das imagens obtidas da cena Cornell box com integração auxiliar uniforme.

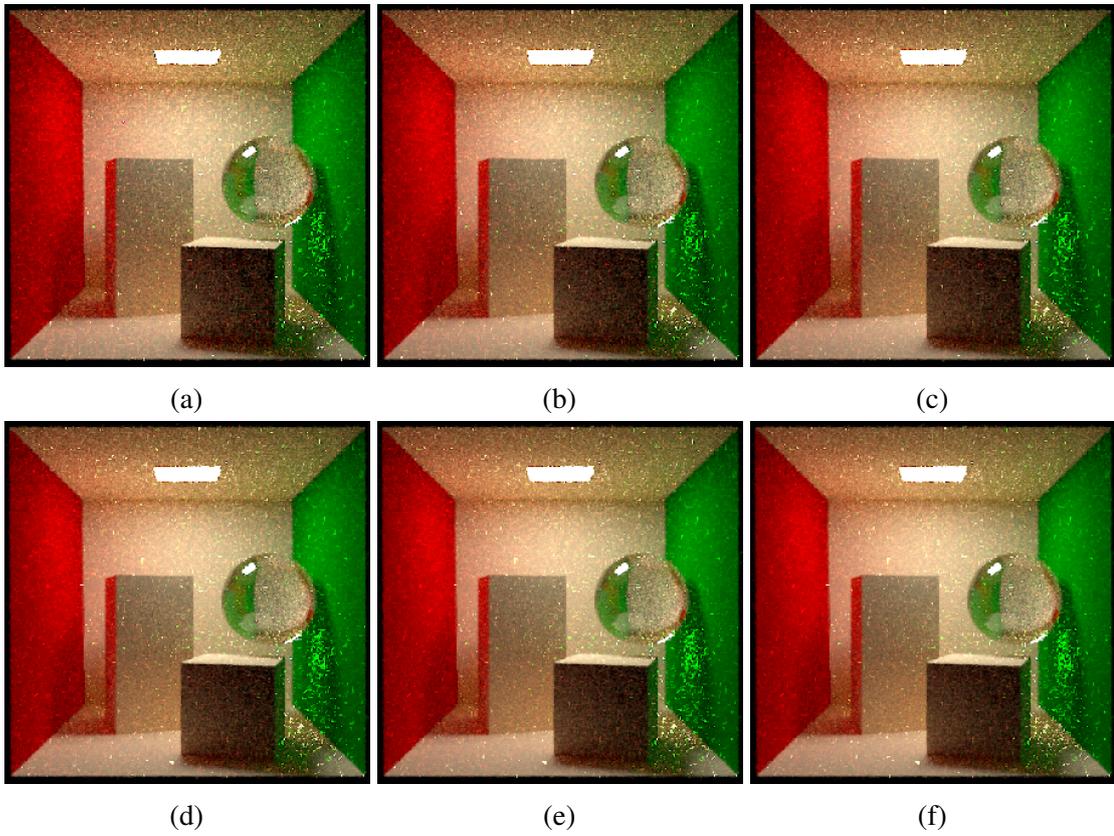


Figura 5.28: Integração auxiliar uniforme na cena “Cornell box com esfera”: (a) MDAS comum. (b) 10 app. (c) 20 app. (d) 30 app. (e) 40 app. (f) 50 app.

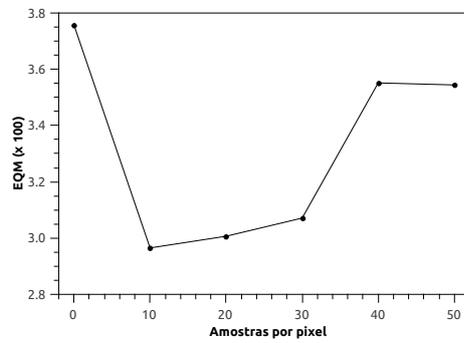
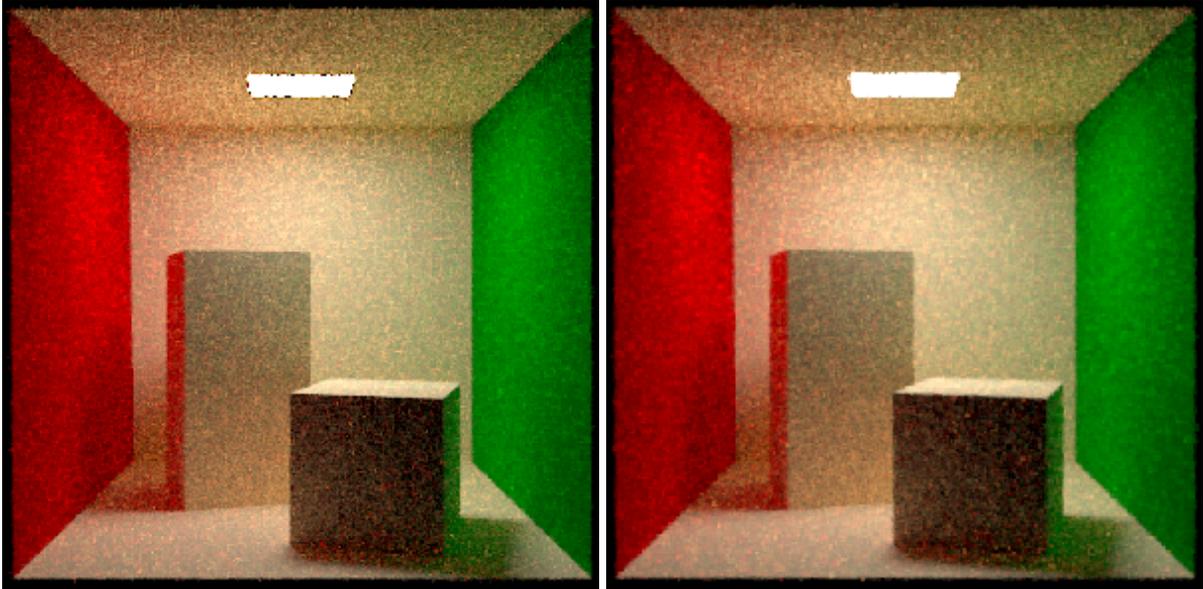


Figura 5.29: Erro quadrático médio das imagens obtidas da cena “Cornell box com esfera” com integração auxiliar uniforme.

utilizados na cena “Cornell box” foram os mesmos utilizados na integração auxiliar uniforme, já para a cena “Macaco”, os parâmetros utilizados foram: $N_i = 2$ app, $N_a = 30$ app, $C_{max} = 4$, $N_c = 3$, $k = 6$ e $S_h = [1/26; 1/26; 1/44; 1/64]$.



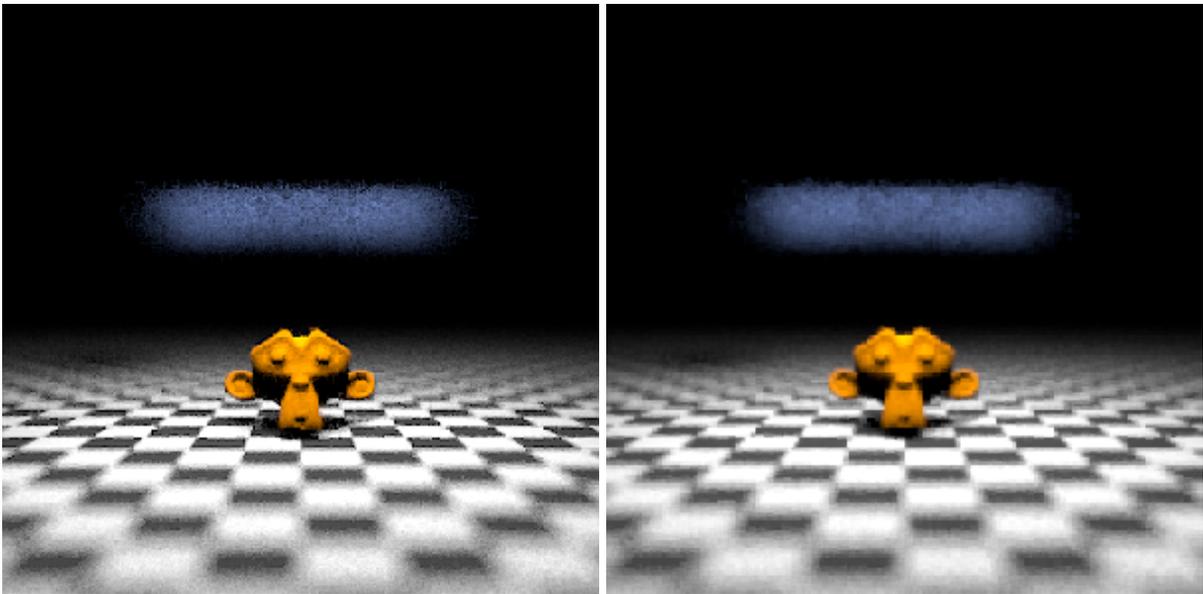
(a) Tempo de reconstrução ≈ 27 s.

(b) Tempo de reconstrução ≈ 16 s.

Figura 5.30: Reconstrução por nó da cena “Cornell box”: (a) MDAS comum. (b) Reconstrução por nó. Houve redução de 39,8 % no tempo de reconstrução.

5.6 Considerações Finais

Este capítulo apresentou diversos testes feitos com a técnica de renderização adaptativa proposta neste trabalho. A técnica de substituição de amostras, embora tenha se mostrado capaz de induzir alguma adaptação em regiões de difícil detecção do domínio multidimensional, ainda produz muito ruído em casos onde $k < n$. Já a etapa de integração auxiliar uniforme mostrou-se capaz de reduzir ligeiramente o ruído em todos os casos testados, embora ainda haja problemas em cenas com significativa fonte de ruído em dimensões não tratadas pela kd-tree. Por fim, foram feitos alguns testes com a técnica de reconstrução por nó, os resultados mostram que há significativa redução no tempo de execução da etapa de reconstrução, entretanto, as imagens obtidas sofrem com perda de definição e artefatos retangulares em regiões de baixa discretização da kd-tree. No próximo capítulo são apresentadas as conclusões finais e algumas possíveis linhas para trabalhos futuros.



(a) Tempo de reconstrução $\approx 79,4$ s.

(b) Tempo de reconstrução ≈ 22 s.

Figura 5.31: Reconstrução por nó da cena “Macaco”: (a) MDAS comum. (b) Reconstrução por nó. Houve redução de 72,3 % no tempo de reconstrução.

6 CONCLUSÃO E TRABALHOS FUTUROS

Este trabalho apresentou uma visão geral da área de amostragem adaptativa em renderização, com especial foco em amostragem adaptativa multidimensional. Foram propostas algumas modificações à técnica de amostragem adaptativa MDAS (*Multidimensional Adaptive Sampling and Reconstruction* (HACHISUKA et al., 2008)), visando amenizar algumas de suas deficiências. A primeira modificação proposta consiste em tentar melhorar o processo de amostragem inicial por meio de uma etapa de substituição de amostras, aumentando sua capacidade de detectar características pequenas com relação ao domínio espacial, mas que podem ter grande contribuição na imagem final. A segunda modificação visa permitir um meio de reduzir o ruído em cenas de alta dimensionalidade ($n \gg k$), como no caso das cenas com iluminação global. Essas duas modificações foram propostas de modo a não necessitarem de armazenamento extra de amostras, ou seja, de refinamento extra da kd-tree. A última modificação consiste em um algoritmo de reconstrução, visando reduzir a parcela de tempo gasta na fase de reconstrução.

Conforme pode-se observar nos resultados, a etapa de substituição de amostras, permitiu a captura de características que escapam à técnica MDAS original. Entretanto, essa maior robustez veio acompanhada de aumento de ruído nos casos em que $k < n$, possivelmente devido à mudança na distribuição das amostras sem a devida compensação no valor de suas contribuições. Além disso, observou-se que a taxa de aceitação das amostras de substituição é muito baixa, resultando em desperdício de amostras. Algumas possíveis linhas de investigação para esses problemas são: formulação de novas estratégias de substituição; aproveitar a contribuição das amostras de substituição rejeitadas; encontrar alguma representação barata de PDF que represente a mudança de distribuição nas dimensões fora do alcance da kd-tree, permitindo a correção da contribuição das amostras devido a mudanças nesta distribuição.

Quanto à etapa de integração auxiliar, foi implementada a versão uniforme. Os resultados mostraram que a técnica possibilita uma ligeira redução de ruído tanto nos casos onde $k = n$ quanto nos casos em que $k < n$. Porém, como essa técnica utiliza a kd-tree construída e refinada nas etapas anteriores, possíveis falhas podem tornar-se mais destacadas à medida que mais amostra são agregadas nessas regiões. Uma possibilidade de trabalho futuro é investigar outras formas de acumular os valores amostrais em cada folha da kd-tree por meio um filtro de reconstrução anisotrópico baseado nas informações de gradiente em cada folha. Além disso, pretende-se implementar integração auxiliar por Metropolis.

Por fim, foi proposto um algoritmo de reconstrução por nó. Esse algoritmo percorre iterativamente todas as folhas da kd-tree e acumula sua contribuição nos pixels subjacentes. Os resultados mostraram que o algoritmo proposto é mais rápido que a reconstrução por pixel, porém, as imagens obtidas apresentam alguma piora: imagens mais borradas e com artefatos retangulares. Uma possível melhoria a ser investigada consiste em utilizar um processo de superamostragem em cada folha da kd-tree durante a etapa de reconstrução, neste caso, pode-se também utilizar uma versão mais otimizada do algoritmo de busca, visando reduzir o impacto da superamostragem no custo total da reconstrução.

REFERÊNCIAS BIBLIOGRÁFICAS

- ARVO, J. The role of functional analysis in global illumination. In: *Rendering Techniques '95*. [S.l.]: Springer-Verlag, 1995. p. 115–126.
- BOLIN, M. R.; MEYER, G. W. A perceptually based adaptive sampling algorithm. In: *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*. New York, NY, USA: ACM, 1998. (SIGGRAPH '98), p. 299–309. ISBN 0-89791-999-8. Disponível em: <<http://doi.acm.org/10.1145/280814.280924>>.
- CHEN, J. et al. Efficient depth-of-field rendering with adaptive sampling and multiscale reconstruction. *Computer Graphics Forum*, Blackwell Publishing Ltd, v. 30, n. 6, p. 1667–1680, 2011. ISSN 1467-8659. Disponível em: <<http://dx.doi.org/10.1111/j.1467-8659%-.2011.01854.x>>.
- CLARBERG, P. et al. Wavelet importance sampling: efficiently evaluating products of complex functions. In: *ACM SIGGRAPH 2005 Papers*. New York, NY, USA: ACM, 2005. (SIGGRAPH '05), p. 1166–1175. Disponível em: <<http://doi.acm.org/10.1145/1186822.1073328>>.
- COOK, R. L.; PORTER, T.; CARPENTER, L. Distributed ray tracing. *SIGGRAPH Comput. Graph.*, ACM, New York, NY, USA, v. 18, n. 3, p. 137–145, jan. 1984. ISSN 0097-8930. Disponível em: <<http://doi.acm.org/10.1145/964965.808590>>.
- DUTRE, P. et al. *Advanced Global Illumination*. [S.l.]: AK Peters Ltd, 2006. ISBN 1568813074.
- EGAN, K. et al. Frequency analysis and sheared reconstruction for rendering motion blur. *ACM Trans. Graph.*, ACM, New York, NY, USA, v. 28, n. 3, p. 93:1–93:13, jul. 2009. ISSN 0730-0301. Disponível em: <<http://doi.acm.org/10.1145/1531326.1531399>>.
- GEERTSEMA, J. C. Sequential confidence intervals based on rank tests. *The Annals of Mathematical Statistics*, Institute of Mathematical Statistics, v. 41, n. 3, p. pp. 1016–1026, 1970. ISSN 00034851. Disponível em: <<http://www.jstor.org/stable/2239254>>.
- GORAL, C. M. et al. Modeling the interaction of light between diffuse surfaces. In: *Proceedings of the 11th annual conference on Computer graphics and interactive techniques*. New York, NY, USA: ACM, 1984. (SIGGRAPH '84), p. 213–222. ISBN 0-89791-138-5. Disponível em: <<http://doi.acm.org/10.1145/800031.808601>>.
- HACHISUKA, T. et al. Multidimensional adaptive sampling and reconstruction for ray tracing. *ACM Trans. Graph.*, ACM, New York, NY, USA, v. 27, n. 3, p. 33:1–33:10, ago. 2008. ISSN 0730-0301. Disponível em: <<http://doi.acm.org/10.1145/1360612.1360632>>.
- HECKBERT, P. S. Adaptive radiosity textures for bidirectional ray tracing. *SIGGRAPH Comput. Graph.*, ACM, New York, NY, USA, v. 24, n. 4, p. 145–154, set. 1990. ISSN 0097-8930. Disponível em: <<http://doi.acm.org/10.1145/97880.97895>>.
- HOBEROCK, J.; HART, J. C. Arbitrary importance functions for metropolis light transport. *Computer Graphics Forum*, Blackwell Publishing Ltd, v. 29, n. 6, p. 1993–2003, 2010. ISSN 1467-8659. Disponível em: <<http://dx.doi.org/10.1111/j.1467-8659%-.2010.01713.x>>.

KAJIYA, J. T. The rendering equation. *SIGGRAPH Comput. Graph.*, ACM, New York, NY, USA, v. 20, n. 4, p. 143–150, ago. 1986. ISSN 0097-8930. Disponível em: <<http://doi.acm.org/10.1145/15886.15902>>.

KELEMEN, C. et al. A simple and robust mutation strategy for the metropolis light transport algorithm. *Computer Graphics Forum*, Blackwell Publishing, Inc, v. 21, n. 3, p. 531–540, 2002. ISSN 1467-8659. Disponível em: <<http://dx.doi.org/10.1111/1467-8659.t01-1-00703>>.

LAFORTUNE, E. P.; WILLEMS, Y. D. Bi-Directional Path Tracing. In: *Proceedings of the 3rd international conference on Computational graphics and Visualization techniques (COMPUGRAPHICS '93)*. [S.l.: s.n.], 1993. p. 145–153.

LEE, M. E.; REDNER, R. A.; USELTON, S. P. Statistically optimized sampling for distributed ray tracing. *SIGGRAPH Comput. Graph.*, ACM, New York, NY, USA, v. 19, n. 3, p. 61–68, jul. 1985. ISSN 0097-8930. Disponível em: <<http://doi.acm.org/10.1145/325165.325179>>.

LIU, X.-D.; WU, J.-Z.; ZHENG, C.-W. Kd-tree based parallel adaptive rendering. *The Visual Computer*, Springer-Verlag, v. 28, p. 613–623, 2012. ISSN 0178-2789. Disponível em: <<http://dx.doi.org/10.1007/s00371-012-0709-9>>.

MITCHELL, D. P. Generating antialiased images at low sampling densities. *SIGGRAPH Comput. Graph.*, ACM, New York, NY, USA, v. 21, n. 4, p. 65–72, ago. 1987. ISSN 0097-8930. Disponível em: <<http://doi.acm.org/10.1145/37402.37410>>.

OVERBECK, R. S.; DONNER, C.; RAMAMOORTHY, R. Adaptive wavelet rendering. *ACM Trans. Graph.*, ACM, New York, NY, USA, v. 28, n. 5, p. 140:1–140:12, dez. 2009. ISSN 0730-0301. Disponível em: <<http://doi.acm.org/10.1145/1618452.1618486>>.

PHARR, M.; HUMPHREYS, G. *Physically Based Rendering, Second Edition: From Theory To Implementation*. 2nd. ed. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2010. ISBN 0123750792, 9780123750792.

PURGATHOFER, W. A statistical method for adaptive stochastic sampling. *Computers & Graphics*, v. 11, n. 2, p. 157 – 162, 1987. ISSN 0097-8493. Disponível em: <<http://www.sciencedirect.com/science/article/pii/009784938790029X>>.

ROUSSELLE, F.; KNAUS, C.; ZWICKER, M. Adaptive sampling and reconstruction using greedy error minimization. *ACM Trans. Graph.*, ACM, New York, NY, USA, v. 30, n. 6, p. 159:1–159:12, dez. 2011. ISSN 0730-0301. Disponível em: <<http://doi.acm.org/10.1145/2070781.2024193>>.

VEACH, E. *Robust monte carlo methods for light transport simulation*. Tese (Doutorado), Stanford, CA, USA, 1998. AAI9837162.

VEACH, E.; GUIBAS, L. J. Optimally combining sampling techniques for monte carlo rendering. In: *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*. New York, NY, USA: ACM, 1995. (SIGGRAPH '95), p. 419–428. ISBN 0-89791-701-4. Disponível em: <<http://doi.acm.org/10.1145/218380.218498>>.

VEACH, E.; GUIBAS, L. J. Metropolis light transport. In: *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*. New York, NY, USA: ACM Press/Addison-Wesley Publishing Co., 1997. (SIGGRAPH '97), p. 65–76. ISBN 0-89791-896-7. Disponível em: <<http://dx.doi.org/10.1145/258734.258775>>.

WHITTED, T. An improved illumination model for shaded display. *Commun. ACM*, ACM, New York, NY, USA, v. 23, n. 6, p. 343–349, jun. 1980. ISSN 0001-0782. Disponível em: <<http://doi.acm.org/10.1145/358876.358882>>.