

GUSTAVO PICHORIM BOIKO

**Análise Comparativa de Técnicas de Iluminação Global  
para Síntese de Imagens com Realismo**

Dissertação apresentada como requisito parcial à  
obtenção do grau de Mestre. Programa de Pós-  
Graduação em Informática, Setor de Ciências Exatas,  
Universidade Federal do Paraná.  
Orientador: Prof. Dr. Hélio Pedrini

CURITIBA

2009

## AGRADECIMENTOS

Agradeço à minha esposa, Elenise, pela dedicação, apoio, incentivo, carinho e compreensão.

Aos meus pais, Elisabete e José, por seus sábios ensinamentos e por sua paciência e à minha irmã, Karina, por sua amizade e apoio.

Ao meu orientador Prof. Dr. Hélio Pedrini agradeço a imensa confiança, o apoio e todo o conhecimento que me foi passado.

À empresa Conectiva S.A, que cedeu parte do meu horário de trabalho para a realização desta pesquisa, tempo este que foi realmente precioso.

Gostaria também de agradecer aos artistas Dabrovic e Gawel, que gentilmente autorizaram o uso de suas cenas para os testes realizados, e à equipe de desenvolvimento do YafaRay, cujo apoio foi de grande importância para a execução das técnicas propostas.

Finalmente, agradeço a todos que contribuíram para a concretização deste trabalho. A conclusão de uma etapa importante como esta não é mérito de uma pessoa só, e só foi possível com a ajuda, cooperação e compreensão de parentes, amigos e colegas.



## SUMÁRIO

|                                                           |             |
|-----------------------------------------------------------|-------------|
| <b>LISTA DE FIGURAS</b>                                   | <b>vii</b>  |
| <b>LISTA DE TABELAS</b>                                   | <b>viii</b> |
| <b>ABSTRACT</b>                                           | <b>ix</b>   |
| <b>RESUMO</b>                                             | <b>x</b>    |
| <b>1 INTRODUÇÃO</b>                                       | <b>1</b>    |
| 1.1 Caracterização do Problema . . . . .                  | 1           |
| 1.2 Objetivos e Contribuições . . . . .                   | 2           |
| 1.3 Organização da Dissertação . . . . .                  | 2           |
| <b>2 REVISÃO BIBLIOGRÁFICA</b>                            | <b>3</b>    |
| 2.1 Os Primórdios da Renderização Fotorrealista . . . . . | 3           |
| 2.2 Transporte da Luz . . . . .                           | 4           |
| 2.3 Métodos de Monte Carlo . . . . .                      | 5           |
| 2.3.1 Integração de Monte Carlo . . . . .                 | 5           |
| 2.3.2 Roleta Russa . . . . .                              | 7           |
| 2.3.3 Amostragem . . . . .                                | 8           |
| 2.3.3.1 Amostragem Estratificada . . . . .                | 9           |
| 2.3.3.2 Amostragem por Importância . . . . .              | 9           |
| 2.3.3.3 Inversão de Função . . . . .                      | 11          |
| 2.3.3.4 Rejeição . . . . .                                | 11          |
| 2.3.3.5 Metropolis . . . . .                              | 12          |
| 2.4 Iluminação Global . . . . .                           | 13          |
| 2.4.1 Ray Tracing . . . . .                               | 14          |
| 2.4.2 Radiosidade . . . . .                               | 18          |

|          |                                                                |           |
|----------|----------------------------------------------------------------|-----------|
| 2.4.3    | Equação da Renderização . . . . .                              | 21        |
| 2.4.4    | Path Tracing . . . . .                                         | 21        |
| 2.4.5    | Irradiance Caching . . . . .                                   | 25        |
| 2.4.6    | Algoritmos de Múltiplos Passos . . . . .                       | 33        |
| 2.4.7    | Photon Mapping . . . . .                                       | 37        |
| 2.4.8    | Metropolis Light Transport . . . . .                           | 42        |
| <b>3</b> | <b>METODOLOGIA</b>                                             | <b>45</b> |
| 3.1      | Componentes de um Renderizador . . . . .                       | 46        |
| 3.1.1    | Câmera . . . . .                                               | 46        |
| 3.1.1.1  | Câmera Ortográfica . . . . .                                   | 48        |
| 3.1.1.2  | Câmera Perspectiva . . . . .                                   | 49        |
| 3.1.1.3  | Outros Tipos de Câmera . . . . .                               | 49        |
| 3.1.1.4  | Profundidade de Campo . . . . .                                | 50        |
| 3.1.2    | Amostrador . . . . .                                           | 52        |
| 3.1.3    | Integradores . . . . .                                         | 52        |
| 3.1.3.1  | Integradores de Superfície . . . . .                           | 53        |
| 3.1.3.2  | Integradores de Volume . . . . .                               | 54        |
| 3.1.4    | Película . . . . .                                             | 55        |
| 3.1.5    | Renderizador . . . . .                                         | 56        |
| 3.2      | Ambiente para Implementação e Avaliação das Técnicas . . . . . | 56        |
| 3.3      | Técnicas Avaliadas . . . . .                                   | 57        |
| 3.3.1    | Irradiance Caching . . . . .                                   | 58        |
| 3.3.2    | Irradiance Caching com Refinamento Adaptativo . . . . .        | 59        |
| 3.3.3    | Path Tracing . . . . .                                         | 60        |
| 3.3.4    | Photon Mapping com Irradiance Caching . . . . .                | 60        |
| 3.3.5    | Photon Mapping com Final Gathering . . . . .                   | 61        |
| 3.4      | Avaliação das Técnicas . . . . .                               | 61        |

|          |                                             |           |
|----------|---------------------------------------------|-----------|
| <b>4</b> | <b>RESULTADOS EXPERIMENTAIS</b>             | <b>63</b> |
| 4.1      | Plataforma de Hardware e Software . . . . . | 63        |
| 4.2      | Conjunto de Teste . . . . .                 | 63        |
| 4.2.1    | Caixa . . . . .                             | 64        |
| 4.2.2    | Sala Vermelha . . . . .                     | 65        |
| 4.2.3    | Catedral de São Jacó . . . . .              | 65        |
| 4.2.4    | Jardim Botânico . . . . .                   | 66        |
| 4.3      | Resultados Obtidos . . . . .                | 66        |
| 4.3.1    | Caixa . . . . .                             | 67        |
| 4.3.2    | Sala Vermelha . . . . .                     | 71        |
| 4.3.3    | Catedral de São Jacó . . . . .              | 75        |
| 4.3.4    | Jardim Botânico . . . . .                   | 78        |
| 4.4      | Discussão dos Resultados . . . . .          | 82        |
| <b>5</b> | <b>CONCLUSÕES E TRABALHOS FUTUROS</b>       | <b>84</b> |
|          | <b>REFERÊNCIAS BIBLIOGRÁFICAS</b>           | <b>86</b> |

## LISTA DE FIGURAS

|      |                                                                                                                                                                  |    |
|------|------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 2.1  | Para gerar o sombreamento da esfera, a intensidade da cor é modificada de acordo com o ângulo de incidência da luz. . . . .                                      | 3  |
| 2.2  | Exemplo de reflexão especular simulada por sombreamento de Phong [49]. . .                                                                                       | 4  |
| 2.3  | Diferença entre os resultados produzidos por iluminações local e global. Modelo de Marko Dabrovic [13]. . . . .                                                  | 13 |
| 2.4  | Cena de exemplo sintetizada com o uso de <i>ray tracing</i> . . . . .                                                                                            | 14 |
| 2.5  | Novo raio gerado a partir da interseção do raio original com uma superfície que possui certa especularidade. . . . .                                             | 15 |
| 2.6  | Novo raio gerado a partir da interseção do raio original com uma superfície que possui certa transparência, levando em conta o índice de refração desta. . . . . | 15 |
| 2.7  | Para superfícies que possuem reflexão difusa, raios são gerados em direção às fontes de luz para identificar se o objeto está iluminado ou em sombras. . . . .   | 16 |
| 2.8  | Para cada interseção, novos raios são gerados recursivamente de acordo com as características da superfície. . . . .                                             | 16 |
| 2.9  | Detalhe da Figura 2.4 mostrando o <i>aliasing</i> causado na discretização do espaço feita pela técnica de <i>ray tracing</i> . . . . .                          | 17 |
| 2.10 | O uso de raios extras ( <i>stochastic ray tracing</i> ) possibilita a remoção de defeitos causados por <i>aliasing</i> . . . . .                                 | 18 |
| 2.11 | A luminosidade de cada <i>patch</i> é calculada através de todos os <i>form factors</i> que o contêm. . . . .                                                    | 19 |
| 2.12 | Caminhos são escolhidos por amostragem para resolver a integral da iluminação do ponto intersectado no algoritmo de <i>path tracing</i> . . . . .                | 22 |
| 2.13 | Um caminho gerado pelo algoritmo de <i>path tracing</i> . Para o último vértice do caminho, sua contribuição é definida através de iluminação direta. . . . .    | 23 |
| 2.14 | Cena que gera resultados com uma variância bastante alta com o uso de <i>path tracing</i> . . . . .                                                              | 24 |

|      |                                                                                                                                                                                                                                                                                                                                                                                  |    |
|------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 2.15 | O caminho gerado por <i>path tracing bidirecional</i> se inicia nos dois extremos: no visualizador e em uma fonte de luz. $p_i$ representa um vértice gerado a partir do visualizador e $q_i$ indica um vértice gerado a partir da fonte de luz. A junção dos caminhos nos vértices $q_n$ e $p_n$ requer um <i>shadow ray</i> para determinar a visibilidade entre eles. . . . . | 24 |
| 2.16 | O método de <i>path tracing</i> bidirecional descrito por Lafortune [38] realiza conexões entre todos os vértices de cada uma das partes do caminho. . . . .                                                                                                                                                                                                                     | 25 |
| 2.17 | Cena complexa para o algoritmo de <i>path tracing</i> gera resultados com uma variância bem menor com o uso de <i>path tracing</i> bidirecional. . . . .                                                                                                                                                                                                                         | 25 |
| 2.18 | Diferença de resultados obtidos com o uso de <i>path tracing</i> e <i>irradiance caching</i> . . . . .                                                                                                                                                                                                                                                                           | 28 |
| 2.19 | Para o cálculo dos gradientes de irradiância é necessário que o hemisfério seja amostrado de forma estratificada. Adaptado de [37]. . . . .                                                                                                                                                                                                                                      | 29 |
| 2.20 | O uso de gradientes de irradiância diminui a presença de descontinuidades no resultado gerado. . . . .                                                                                                                                                                                                                                                                           | 30 |
| 2.21 | A amostra $P_3$ é discernível do conjunto e portanto, seu raio de influência ( $a_3$ ) precisa ser reduzido para que ela não mais contribua para a irradiância no ponto $P$ . Adaptado de [36]. . . . .                                                                                                                                                                          | 33 |
| 2.22 | O efeito de <i>caustics</i> . . . . .                                                                                                                                                                                                                                                                                                                                            | 34 |
| 2.23 | Os caminhos possíveis no método progressivo de Chen, Rushmeier, Miller e Turner [6]. Adaptado de [6]. . . . .                                                                                                                                                                                                                                                                    | 37 |
| 2.24 | Emissão de fótons a partir de diferentes tipos de fontes de luz. Adaptado de [32]. . . . .                                                                                                                                                                                                                                                                                       | 38 |
| 2.25 | O caminho percorrido pelos fótons na cena. Adaptado de [32]. . . . .                                                                                                                                                                                                                                                                                                             | 39 |
| 2.26 | Mapa de <i>caustics</i> e o mapa global de fótons. Adaptado de [16]. . . . .                                                                                                                                                                                                                                                                                                     | 40 |
| 2.27 | Cena renderizada com o uso de <i>photon mapping</i> , mostrando também o conteúdo do mapa de fótons. Modelo de Krzysztof Gawel [19]. . . . .                                                                                                                                                                                                                                     | 41 |
| 3.1  | A estrutura básica de um renderizador. Adaptado de [48]. . . . .                                                                                                                                                                                                                                                                                                                 | 46 |
| 3.2  | Funcionamento da câmera <i>pinhole</i> . A imagem é projetada no plano de fundo (película) de forma invertida e reduzida. . . . .                                                                                                                                                                                                                                                | 47 |

|      |                                                                                                                                                  |    |
|------|--------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 3.3  | Exemplos de imagens geradas com câmeras de projeção. . . . .                                                                                     | 48 |
| 3.4  | Região retangular projetada na cena pela câmera ortográfica. O retângulo frontal representa o plano da imagem (película). . . . .                | 48 |
| 3.5  | Resultado gerado por câmeras com o uso de perspectiva. . . . .                                                                                   | 49 |
| 3.6  | Raios são lançados a partir de um ponto central da câmera passando pelo plano da imagem em direção à cena. . . . .                               | 50 |
| 3.7  | Resultado gerado por câmeras do tipo olho de peixe ( <i>fish eye</i> ). . . . .                                                                  | 50 |
| 3.8  | A projeção de pontos que estão fora do plano focal não são projetados de forma definida no plano da imagem (película). Adaptado de [48]. . . . . | 51 |
| 3.9  | Círculos de confusão gerados em pontos fora da área de foco gerada pela profundidade de campo. . . . .                                           | 51 |
| 3.10 | Exemplos de imagens simulando profundidade de campo. . . . .                                                                                     | 51 |
| 3.11 | Exemplos de imagens comparando o efeito de neblina simulado por um integrador de volume. . . . .                                                 | 54 |
| 3.12 | Escala de valores usada para representar erros na imagem de diferença. . . . .                                                                   | 62 |
| 4.1  | Geometria da cena Caixa. . . . .                                                                                                                 | 64 |
| 4.2  | Geometria da cena Sala Vermelha ( <i>red room</i> ). . . . .                                                                                     | 65 |
| 4.3  | Geometria da cena Catedral de São Jacó. . . . .                                                                                                  | 66 |
| 4.4  | Geometria da cena Jardim Botânico. . . . .                                                                                                       | 67 |
| 4.5  | Resultados para a cena de teste Caixa. . . . .                                                                                                   | 68 |
| 4.6  | Locais onde ocorreram erros nos testes da cena Caixa. . . . .                                                                                    | 69 |
| 4.7  | Resultados para a cena de teste Sala Vermelha. . . . .                                                                                           | 72 |
| 4.8  | Locais onde ocorreram erros nos testes da cena Sala Vermelha. . . . .                                                                            | 73 |
| 4.9  | Resultados para a cena de teste Catedral de São Jacó. . . . .                                                                                    | 76 |
| 4.10 | Locais onde ocorreram erros nos testes da cena Catedral de São Jacó. . . . .                                                                     | 77 |
| 4.11 | Resultados para a cena de teste Jardim Botânico. . . . .                                                                                         | 79 |
| 4.12 | Locais onde ocorreram erros nos testes da cena Jardim Botânico. . . . .                                                                          | 80 |

## LISTA DE TABELAS

|      |                                                                                         |    |
|------|-----------------------------------------------------------------------------------------|----|
| 4.1  | Parâmetros usados para <i>irradiance caching</i> na cena Caixa . . . . .                | 70 |
| 4.2  | Parâmetros usados para <i>path tracing</i> na cena Caixa . . . . .                      | 70 |
| 4.3  | Parâmetros usados para <i>photon mapping</i> na cena Caixa . . . . .                    | 70 |
| 4.4  | Resultados RMSE para a cena Caixa . . . . .                                             | 71 |
| 4.5  | Parâmetros usados para <i>irradiance caching</i> na cena Sala Vermelha . . . . .        | 71 |
| 4.6  | Parâmetros usados para <i>path tracing</i> na cena Sala Vermelha . . . . .              | 74 |
| 4.7  | Parâmetros usados para <i>photon mapping</i> na cena Sala Vermelha . . . . .            | 74 |
| 4.8  | Resultados RMSE para a cena Sala Vermelha . . . . .                                     | 74 |
| 4.9  | Parâmetros usados para <i>irradiance caching</i> na cena Catedral de São Jacó . . . . . | 75 |
| 4.10 | Parâmetros usados para <i>path tracing</i> na cena Catedral de São Jacó . . . . .       | 75 |
| 4.11 | Parâmetros usados para <i>photon mapping</i> na cena Catedral de São Jacó . . . . .     | 78 |
| 4.12 | Resultados RMSE para a cena Catedral de São Jacó . . . . .                              | 78 |
| 4.13 | Parâmetros usados para <i>irradiance caching</i> na cena Jardim Botânico . . . . .      | 81 |
| 4.14 | Parâmetros usados para <i>path tracing</i> na cena Jardim Botânico . . . . .            | 81 |
| 4.15 | Parâmetros usados para <i>photon mapping</i> na cena Jardim Botânico . . . . .          | 81 |
| 4.16 | Resultados RMSE para a cena Jardim Botânico . . . . .                                   | 82 |

## ABSTRACT

Realistic image synthesis is a subject that has already been studied and improved for a long time already. However, the existent global illumination techniques are not yet robust enough to handle all the different tridimensional scene features. The usage of these techniques requires an evaluation of the scene to be rendered in order to identify the approach that better matches the available parameters to obtain images having a high level of realism. This research presents a comparative evaluation of irradiance caching, path tracing and photon mapping techniques, describing some variations, highlighting their qualities and discussing their application in different types of scenes. The results are analysed in a qualitative way through the visual inspection of the generated images, and in a quantitative way by using an error metric based on a reference image. The techniques discussed are implemented and made available as open source software, making the experiments easier to reproduce, as well as the development of new researches based on what was implemented.



## RESUMO

A síntese de imagens com realismo é um assunto que vem sendo estudado e aprimorado ao longo de vários anos. Entretanto, as técnicas de iluminação global existentes ainda não são robustas o suficiente para lidar com as diferentes características das cenas tridimensionais. A aplicação dessas técnicas requer uma avaliação da cena que se deseja renderizar, a fim de identificar a abordagem mais adequada e ajustar os parâmetros disponíveis para a obtenção de imagens com alto grau de realismo. Este trabalho apresenta uma avaliação comparativa das técnicas de *irradiance caching*, *path tracing* e *photon mapping*, descrevendo algumas variações, ressaltando suas qualidades e discutindo a sua aplicação em cenas com diferentes características. Os resultados são analisados de forma qualitativa, por meio de inspeção visual das imagens geradas, e de forma quantitativa, por meio de uma métrica de erro com base em uma imagem de referência. As técnicas discutidas são implementadas e disponibilizadas como código livre e aberto, possibilitando a fácil reprodução dos experimentos realizados, bem como o desenvolvimento de novos trabalhos a partir do que foi implementado.

# CAPÍTULO 1

## INTRODUÇÃO

Este capítulo visa descrever o contexto no qual o trabalho de pesquisa está inserido, destacando-se alguns aspectos sobre a síntese de imagens com realismo, objetivos, justificativa e contribuições do trabalho, bem como a organização do texto.

### 1.1 Caracterização do Problema

A geração de imagens com realismo com o uso do computador (também chamada de renderização) é um objetivo que vem sendo perseguido e aprimorado durante anos por pesquisadores da área de computação gráfica [2, 22, 32, 35, 68]. A maioria dos estudos e técnicas desenvolvidos teve início na observação do mundo à sua volta e da tentativa de reproduzir computacionalmente comportamentos e fenômenos encontrados na natureza.

Para obter resultados relevantes e produzir imagens que sejam tão realistas a ponto de serem comparáveis a fotografias, faz-se necessária a simulação do comportamento da luz e sua interação com objetos, variados tipos de superfície e meios. Essa interação não é fácil de ser representada e os efeitos produzidos por ela são, em sua maioria, computacionalmente caros de serem simulados.

Outro aspecto relevante na síntese de imagens realistas (apesar de não ser o foco primário deste trabalho) é o estudo do aspecto psicológico envolvido na nossa percepção de realismo. A maneira como os olhos recebem os estímulos luminosos e como estes são interpretados são aspectos relevante na criação de imagens realistas, principalmente por permitir algumas simplificações e a omissão de informações que não sejam visualmente perceptíveis das imagens produzidas.

A possibilidade de gerar imagens com um grau de realismo elevado possui várias aplicações em diversas áreas do conhecimento. Na arquitetura e *design*, ela possibilita a prototipação, avaliação e estudo de ambientes e produtos que eventualmente possam ser construídos. Nas

artes, ela possibilita a imersão de efeitos especiais realistas em filmagens tradicionais, além de possibilitar a criação de mundos, criaturas e objetos imaginários dando a estes um realismo convincente.

## **1.2 Objetivos e Contribuições**

Este trabalho tem como objetivo realizar uma avaliação comparativa entre as diferentes técnicas de síntese de imagens realistas baseadas em iluminação global através da implementação destas em um pacote de software livre, visando identificar suas principais vantagens e aspectos positivos, bem como identificar suas limitações.

Para as técnicas já implementadas no software de síntese de imagens, modificações são efetuadas de forma que seja possível comparar corretamente as técnicas discutidas para verificar a qualidade dos resultados obtidos e para avaliar a relação custo-benefício que cada uma apresenta.

A identificação de cenários em que cada uma das técnicas avaliadas é mais indicada, bem como a possibilidade de combinar características e processos de algumas das técnicas são contribuições deste trabalho, colaborando para uma melhor aplicação do conhecimento já existente sobre o tema.

## **1.3 Organização da Dissertação**

O restante do trabalho é dividido como segue. O Capítulo 2 apresenta uma revisão bibliográfica sobre as principais técnicas de sintetização de imagens realistas, sua evolução e os principais desafios. A metodologia de pesquisa é descrita no Capítulo 3 e os resultados são apresentados e discutidos no Capítulo 4. Por fim, o Capítulo 5 apresenta as conclusões do trabalho e algumas propostas para trabalhos futuros.

## CAPÍTULO 2

### REVISÃO BIBLIOGRÁFICA

Este capítulo apresenta um breve histórico das técnicas utilizadas para a síntese de imagens com realismo, mostrando sua evolução, motivações e seus principais desafios.

#### 2.1 Os Primórdios da Renderização Fotorrealista

A busca por técnicas de síntese de imagens fotorrealistas tem sido objeto de estudo de diversos pesquisadores desde muito tempo. Entretanto, as limitações computacionais existentes e os algoritmos até então conhecidos não permitiam a obtenção de resultados que realmente se aproximassem a fotografias. Vários fenômenos naturais não podiam ser simulados, o que tornava a aparência das imagens ainda menos realista.

As primeiras técnicas que pretendiam gerar imagens com essa característica realista eram bastante simplistas: atribuíam um valor de cor fixo para cada objeto e a intensidade era modificada conforme o ângulo incidente da luz [18]. Apesar de os resultados conseguirem representar visualmente uma noção de profundidade, claramente as imagens estavam longe de ser fotorrealistas (Figura 2.1).

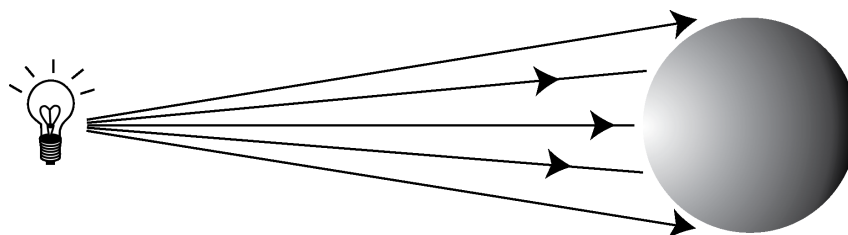


Figura 2.1: Para gerar o sombreamento da esfera, a intensidade da cor é modificada de acordo com o ângulo de incidência da luz.

Uma das primeiras técnicas que tentou realmente simular a incidência de luz em objetos de uma cena foi conhecida como sombreamento de Gouraud [22]. A técnica calcula a intensidade da incidência de luz nos vértices de cada objeto. Estas intensidades são, posteriormente,

interpoladas para obter a intensidade da iluminação em outros pontos da superfície do objeto. Com isso, esta técnica elimina descontinuidades na intensidade de luz observada nas superfícies.

Outra técnica que contribuiu para os avanços nesta área de pesquisa foi a técnica de sombreado de Phong [49], que interpola o vetor normal da superfície ao invés de interpolar a intensidade. Em superfícies com reflexão especular, esta técnica possibilita a simulação de pontos de destaque de luz no reflexo (Figura 2.2).



Figura 2.2: Exemplo de reflexão especular simulada por sombreado de Phong [49].

## 2.2 Transporte da Luz

Para produzir imagens sintetizadas com realismo, é importante modelar a interação da luz com os objetos tentando simular o comportamento físico desta interação conforme observamos no mundo real. Para que isso seja possível, é necessário que existam formas de medir a quantidade de energia que uma determinada fonte de luz emite, a capacidade de uma superfície refletir, transmitir ou absorver essa energia.

A luz é capaz de produzir diversos fenômenos na natureza. Pesquisadores sempre tentaram explicar a natureza e o comportamento desses fenômenos e diversos deles puderam ser devidamente explicados. No entanto, para cada grupo de fenômenos, o modelo utilizado para representar a luz e descrevê-los precisou ser adaptado. Para alguns desses fenômenos, a luz precisou ser modelada como uma onda, para outros a luz precisou ser modelada como partículas. Em alguns casos é necessário que a luz seja modelada como um dual de onda e

partícula.

As técnicas de síntese de imagens com realismo fazem uso da luz modelada através de conceitos geométricos [16], e algumas simplificações de comportamento são assumidas para simplificar os cálculos (sem comprometer o resultado final):

- a luz viaja em linhas retas desconsiderando possíveis efeitos causados por difração;
- a luz viaja instantaneamente em diferentes meios, assumindo com isso que a sua velocidade é infinita;
- a luz não é influenciada por fatores externos como gravidade e campos magnéticos.

A medição das quantidades de energia de fontes luminosas é chamada de radiometria [43].

## 2.3 Métodos de Monte Carlo

Muitas das técnicas apresentadas na seção 2.4 fazem uso de abordagens estatísticas para auxiliar na obtenção dos resultados desejados. Dentre essas abordagens, os métodos de Monte Carlo são os mais comumente utilizados. Métodos de Monte Carlo [24, 25, 45] consistem em um conjunto de técnicas probabilísticas [46, 50] que fazem uso de amostragem aleatória para obtenção de resultados. Seu objetivo principal é o de aproximar o resultado de problemas cuja obtenção da solução real é muito dispendiosa ou inviável. Este objetivo é atingido através da observação de variáveis aleatórias que reproduzam o comportamento do problema original, utilizando-as para inferir o resultado desejado. As próximas seções descrevem o funcionamento de alguns desses métodos.

### 2.3.1 Integração de Monte Carlo

As técnicas de síntese de imagens realistas geralmente fazem uso de integrais. Muitas dessas não possuem solução analítica, e por apresentarem uma dimensão mais alta, também não são facilmente resolvidas por técnicas de integração numérica tradicionais, cuja convergência nestes casos é muito lenta. Um exemplo de integral que se enquadra nessas características é a integral da radiância em um ponto de uma superfície. Para determinar a solução desta

integral, faz-se necessário calcular a radiância vinda de todos os pontos da esfera unitária ao redor do ponto de interesse. Como a solução é computacionalmente inviável, faz-se necessário o uso de alguma técnica que consiga solucionar, ou pelo menos aproximar o resultado da integral.

Para ilustrar o funcionamento da integração de Monte Carlo, suponha que o problema a ser resolvido seja solucionar a integral

$$I = \int_a^b f(x) dx \quad (2.1)$$

Uma solução para esta integral seria o cálculo do valor médio de  $f(x)$  no intervalo de  $a$  até  $b$  e então multiplicá-lo pelo tamanho do intervalo  $b - a$ . A idéia da integração de Monte Carlo consiste na aproximação desse valor médio através do uso de variáveis aleatórias. Para tanto, são escolhidos  $N$  valores aleatórios  $\xi_1, \xi_2, \dots, \xi_N$  uniformemente distribuídos no intervalo de integração. Com isso, podemos estimar o valor da integral ( $I_M$ ) como

$$I_M = \frac{(b - a)}{N} \sum_{i=1}^N f(\xi_i) \quad (2.2)$$

O conjunto de variáveis aleatórias necessárias para determinar o valor da função é chamado de amostra. No exemplo apresentado, cada amostra consiste em uma variável aleatória,  $\xi$ , para a obtenção do valor de  $f(x)$ .

A qualidade do resultado estimado por  $I_M$  depende da quantidade de amostras utilizadas, sendo que no limite  $I_M = I$ . A convergência da integração de Monte Carlo para a solução final do problema é lenta. Entretanto, para integrais de dimensões mais altas o método ainda é melhor do que outros métodos disponíveis [51]. A variância [25] da estimativa  $I_M$  é

$$\sigma^2 = \frac{1}{N} \int_a^b (f(x) dx - I)^2 \quad (2.3)$$

Conforme aumenta o número de amostras ( $N$ ), a variância decresce linearmente. O erro da estimativa é proporcional ao desvio padrão [25] e reduz na escala de  $\sqrt{N}$ , ou seja, para reduzir o erro pela metade são necessárias quatro vezes mais amostras.

A escolha das amostras é outro fator importante para redução da variância. O uso de uma função de densidade de probabilidade adequada pode contribuir bastante para a redução da variância. Com uma generalização da integração de Monte Carlo, podemos usar variáveis aleatórias obedecendo uma função de densidade de probabilidade arbitrária  $p$ , resultando na estimativa

$$I_M = \frac{(b-a)}{N} \int_a^b \frac{f(x_i)}{p(x_i)} \quad (2.4)$$

A extensão da integração de Monte Carlo para múltiplas dimensões é bastante direta. Suponha a integral tridimensional

$$\int_{x_0}^{x_1} \int_{y_0}^{y_1} \int_{z_0}^{z_1} f(x, y, z) dx dy dz \quad (2.5)$$

Para amostrar este domínio, suponha o uso de um paralelepípedo de  $(x_0, y_0, z_0)$  até  $(x_1, y_1, z_1)$  amostrado de maneira uniforme. A função de densidade de probabilidade é constante

$$p = \frac{1}{(x_1 - x_0)} \frac{1}{(y_1 - y_0)} \frac{1}{(z_1 - z_0)} \quad (2.6)$$

A integração de Monte Carlo para esta função é definida da seguinte maneira

$$I_M = \frac{(x_1 - x_0)(y_1 - y_0)(z_1 - z_0)}{N} \sum_{i=1}^N f(x_i, y_i, z_i) \quad (2.7)$$

É importante observar que o número de amostras a ser utilizado independe da dimensão da integral sendo usada.

### 2.3.2 Roleta Russa

A seção 2.4.1 descreve o algoritmo de *raytracing* que possui uma natureza recursiva. Essa recursão normalmente é limitada a uma certa profundidade, o que reduz a quantidade de raios gerados, mas por outro lado pode prejudicar a qualidade da imagem gerada em algumas cenas. Em cenas com muitas superfícies especulares é provável que um raio precise percorrer um caminho bem maior para conseguir um bom resultado. Por outro lado, em cenas com



superfícies predominantemente difusas, os raios precisarão de pouca recursão para obter um bom resultado.

Roleta Russa [5] é um método de Monte Carlo que pode ser utilizado em casos como o descrito acima, permitindo que a escolha da profundidade da recursão possa ser feita de forma imparcial e automática. Considere a integral

$$I = \int_0^1 f(x) dx \quad (2.8)$$

Se calcular o resultado de  $f(x)$  para cada uma das amostras do integrador de Monte Carlo for inviável, pode ser necessário estimar o resultado tentando calcular  $f(x)$  o menor número de vezes possível. Escalando a integral original horizontalmente por um fator  $P$  e verticalmente por  $1/P$ , e sendo  $P \leq 1$ , podemos expressar a integral  $I$  como

$$I_{RR} = \int_0^P \frac{1}{P} f\left(\frac{x}{P}\right) dx \quad (2.9)$$

Assumindo a função de densidade de probabilidade [50]  $p(x) = 1$  e aplicando a integração de Monte Carlo a  $I_{RR}$ , temos a estimativa

$$I_{RRM} = \begin{cases} \frac{1}{P} f\left(\frac{x}{P}\right) & \text{se } x \leq P \\ 0 & \text{se } x > P \end{cases} \quad (2.10)$$

No exemplo do algoritmo de *raytracing*, a recursão seria interrompida quando a probabilidade se igualasse a  $\alpha = 1 - P$ , sendo  $\alpha$  chamado de probabilidade de absorção. Se  $\alpha$  é um valor alto, a recursão é interrompida mais rapidamente, mas o resultado possui uma variância maior. Para um valor  $\alpha$  menor, a recursão demora mais para terminar, mas o resultado será mais preciso. O valor esperado de  $I_{RRM}$  se iguala a  $I$ .

### 2.3.3 Amostragem

A qualidade dos resultados obtidos pelos métodos de Monte Carlo é diretamente proporcional à qualidade das amostras escolhidas. Apesar de aleatórias, é desejável que os valores

estejam distribuídos respeitando o máximo possível a distribuição de probabilidade do problema de interesse. Como visto na seção 2.3.1, para reduzir a variância dos resultados é necessário um aumento significativo no número de amostras, portanto, se escolhermos as amostras de modo bem distribuído, evitamos a necessidade de aumentar muito o número de amostras. As subseções a seguir descrevem alguns métodos de amostragem que podem ser utilizados.

### 2.3.3.1 Amostragem Estratificada

Ao amostrar um determinado domínio, a estratégia de amostragem aplicada pode gerar uma distribuição de amostras não uniforme, que priorize algumas regiões do domínio e ignore outras. A técnica de amostragem estratificada [50] parte do princípio de que, se o domínio for dividido em subdomínios menores não sobrepostos e que, se um mesmo número de amostras for utilizado em cada uma dessas subregiões, a variância será reduzida. Dado um domínio  $S$ , podemos subdividi-lo em subdomínios  $S_1, S_2, \dots, S_n$ , tal que

$$\bigcup_{i=1}^n S_i = S \quad (2.11)$$

Dada a integração de uma função  $f$  sobre esse domínio, temos que esta integral é equivalente à soma da integração de  $f$  em todos os subdomínios  $S_i$

$$\int_S f(x) dx = \sum_{i=1}^n \int_{S_i} f(x) dx \quad (2.12)$$

É possível provar que a aplicação da integração de Monte Carlo às integrais nos subdomínios  $S_i$  gera resultados com uma variância menor ou igual à variância encontrada na aplicação da integração de Monte Carlo à integral original [48].

### 2.3.3.2 Amostragem por Importância

A técnica de amostragem por importância [15] é uma técnica de redução de variância em métodos de Monte Carlo. Seu objetivo principal é o de melhorar a convergência da integração de Monte Carlo através do uso de uma função de densidade de probabilidade  $p(x)$ , cuja

distribuição seja similar à da função original  $f(x)$  sendo integrada.

Escolher  $p(x)$  de modo ideal requer conhecer o valor da integração de  $f(x)$  e, portanto, não é factível. Assim, esta distribuição é escolhida de modo a se aproximar o máximo possível da função levando em conta as informações conhecidas. A escolha de uma função de densidade de probabilidade ruim pode causar um aumento da variância (ao invés da redução da mesma, que é o propósito sendo perseguido).

Normalmente, a integração de Monte Carlo é utilizada para estimar o valor da integração de uma única função, ou seja, estimar integrais na forma

$$I = \int f(x) dx \quad (2.13)$$

Entretanto, muitas das técnicas utilizadas requerem a integração da multiplicação de duas ou mais funções, tais como:

$$I = \int f(x)g(x) dx \quad (2.14)$$

Definir uma função de densidade de probabilidade  $p(x)$  para este caso é uma decisão complicada, pois envolve decidir se  $p(x)$  terá uma curva semelhante a  $f(x)$  ou a  $g(x)$  e, como dito anteriormente, se esta escolha for feita de modo errôneo, a variância do resultado tende a aumentar. Para solucionar este problema, faz-se uso de uma técnica chamada de amostragem por importância múltipla [60, 62]. Esta técnica propõe o uso de múltiplas funções de densidade de probabilidade. Para a integral mostrada anteriormente, poderiam ser utilizadas duas funções de densidade de probabilidade,  $p_f(x)$  e  $p_g(x)$ , cujas distribuições se assemelham às de  $f(x)$  e  $g(x)$  respectivamente. A integração de Monte Carlo para este caso seria

$$I_M = \frac{1}{n_f + n_g} \left( \sum_{i=1}^{n_f} \frac{f(x_i)g(x_i)w_f(x_i)}{p_f(x_i)} + \sum_{j=1}^{n_g} \frac{f(y_j)g(y_j)w_g(y_j)}{p_g(y_j)} \right) \quad (2.15)$$

em que  $n_f$  é o número de amostras escolhidas de acordo com a distribuição de  $p_f(x)$  e  $n_g$  é o número de amostras escolhidas de acordo com a distribuição de  $p_g(x)$ . As funções  $w_f(x)$  e  $w_g(x)$  são pesos utilizados para balancear a soma, de modo que o valor esperado da estimativa seja o valor da integral original.

### 2.3.3.3 Inversão de Função

O método da inversão [33, 48] faz uso de uma ou mais variáveis aleatórias uniformes que são mapeadas para variáveis aleatórias no domínio do problema. Dada uma função de densidade de probabilidade  $p(x)$  em uma distribuição contínua, para aplicar o método da inversão é necessário encontrar a função cumulativa de densidade  $P(x)$

$$P(x) = \int_{-\infty}^{\infty} p(x') dx' \quad (2.16)$$

Para o mapeamento de uma variável aleatória uniforme  $\xi$ , é necessário computar o inverso de  $P(X)$ ,  $P^{-1}(x)$ , e utilizá-lo para obter a variável aleatória no domínio original

$$X_i = P^{-1}(\xi) \quad (2.17)$$

### 2.3.3.4 Rejeição

Em muitos casos, encontrar a função de densidade de probabilidade de uma determinada função não é factível, ou não é possível inverter a função cumulativa de densidade para a aplicação do método da inversão. Nesses casos, o método de rejeição [33, 48] pode ser utilizado. Dada uma função  $f(x)$ , da qual queremos gerar amostras, assume-se que a função de densidade da probabilidade  $p(x)$  é tal que  $f(x) < cp(X)$  para alguma constante  $c$ , e que sabemos como amostrar  $p$ .

Um valor  $X$  é amostrado de acordo com a distribuição  $p$ , e um valor  $\xi$  de distribuição uniforme. Se os valores amostrados obedecem à seguinte restrição

$$\xi < \frac{f(X)}{cp(X)} \quad (2.18)$$

a amostra é aceita. Caso contrário, ela é rejeitada. Isto garante que as amostras escolhidas sempre estejam abaixo da função  $f(x)$ . A eficiência desta técnica está diretamente ligada a quanto  $cp(x)$  se aproxima de  $f(x)$ .

### 2.3.3.5 Metropolis

O método de Metropolis [27, 44] é uma cadeia de Markov [1, 39] que faz uso de mutações aleatórias para produzir uma série de amostras distribuídas com uma densidade proporcional à da função original  $f$ . Para cada ponto  $x$  do domínio  $S$  da função original, deve ser possível gerar um novo ponto  $y$  cuja densidade é definida por uma função  $p(x \rightarrow y)$ . Dado um ponto aleatório  $x_0$  em  $S$ , é possível gerar um novo ponto  $x_1$  com densidade  $p(x_0 \rightarrow x_1)$ . A partir de  $x_1$ , pode-se gerar um novo ponto  $x_2$  com densidade  $p(x_1 \rightarrow x_2)$ , e assim por diante. No limite, esta distribuição terá densidade  $p$  independente do ponto inicial  $x_0$ .

A escolha de  $p$  deve ser feita de modo que a iteração convirja para uma distribuição proporcional a  $f$ . Tendo um ponto aleatório  $x_i$ , a escolha de um candidato para o ponto  $x_{i+1}$  é feita através do uso de uma função de densidade  $t(x_i \rightarrow x_{i+1})$ , sendo que a probabilidade de  $x_i$  transitar para si próprio é potencialmente não nula. Um possível candidato  $y \sim t(x_i \rightarrow y)$  tem a probabilidade  $a(x_i \rightarrow y)$  de ser aceito (de se tornar  $x_{i+1}$ ), e probabilidade  $1 - a(x_i \rightarrow y)$  de ser rejeitado (de  $x_i$  se tornar  $x_{i+1}$ ).

Para que seja possível obter  $p$  de modo a aproximar a função  $f$ , sua distribuição precisa ser estacionária, o que significa que o fluxo de amostras entre dois pontos precisa ser o mesmo

$$\text{fluxo}(x \rightarrow y) = kf(x)t(x \rightarrow y)a(x \rightarrow y) \quad (2.19)$$

$$\text{fluxo}(y \rightarrow x) = kf(y)t(y \rightarrow x)a(y \rightarrow x) \quad (2.20)$$

para uma constante  $k$ . Igualando esses dois fluxos, obtemos

$$\frac{a(x \rightarrow y)}{a(y \rightarrow x)} = \frac{f(x)t(x \rightarrow y)}{f(y)t(y \rightarrow x)} \quad (2.21)$$

De modo a atingir o equilíbrio o mais rápido possível, a melhor estratégia é fazer com que  $a(x \rightarrow y)$  e  $a(y \rightarrow x)$  assumam valores tão grandes quanto possível. Para tanto, temos que

$$a(x \rightarrow y) = \min \left\{ 1, \frac{f(x)t(x \rightarrow y)}{f(y)t(y \rightarrow x)} \right\} \quad (2.22)$$

Com isso, transições em uma direção são sempre aceitas, enquanto que, em outra direção,

algumas transições são rejeitadas, mantendo o equilíbrio da distribuição.

## 2.4 Iluminação Global

Duas abordagens foram importantes para o avanço das técnicas de síntese de imagens com realismo: *ray tracing* e *radiosidade*. A primeira foi importante por possibilitar a simulação de alguns efeitos naturais, como reflexão especular cuja reprodução por computador até então era inviável. A segunda por possibilitar a simulação da iluminação indireta de superfícies.

A iluminação indireta é aquela onde a luz que incidiu em uma superfície difusa reflexiva será refletida em outras superfícies próximas. Aos métodos de iluminação que, além de levarem em conta a iluminação direta de luzes sobre as superfícies, consideram a iluminação indireta que ocorre entre as superfícies é dado o nome de Iluminação Global [16]. A Figura 2.3 ilustra a diferença entre os resultados obtidos quando se usa iluminação local e iluminação global. As regiões da imagem que não são visíveis às fontes de luz aparecem totalmente escuras na Figura 2.3(a). Entretanto, em uma cena real estas regiões não se apresentam escuras, pois são iluminadas indiretamente através da luz refletida por superfícies que estão diretamente iluminadas. O resultado da iluminação indireta pode ser visto na Figura 2.3(b).

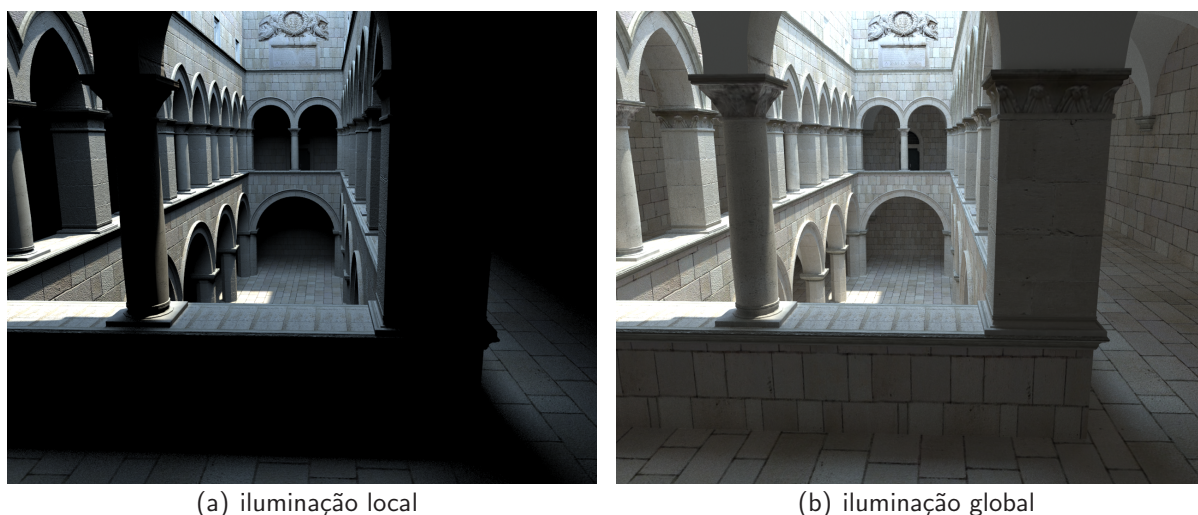


Figura 2.3: Diferença entre os resultados produzidos por iluminações local e global. Modelo de Marko Dabrovic [13].

## 2.4.1 Ray Tracing

A técnica de Ray Tracing [68], apesar de não ser uma técnica de iluminação global completa, possibilitou uma evolução muito rápida das técnicas de síntese de imagens. Este algoritmo traça raios de luz a partir de um observador em direção à cena, tentando encontrar objetos que possuam interseção com estes raios. Uma vez que um raio encontre um objeto em seu caminho, as propriedades da superfície intersectada serão avaliadas. A Figura 2.4 exemplifica uma cena renderizada com o uso desta técnica, em que se pode observar a simulação de reflexos (visíveis na esfera), superfícies transparentes (visível no cone) e sombras.

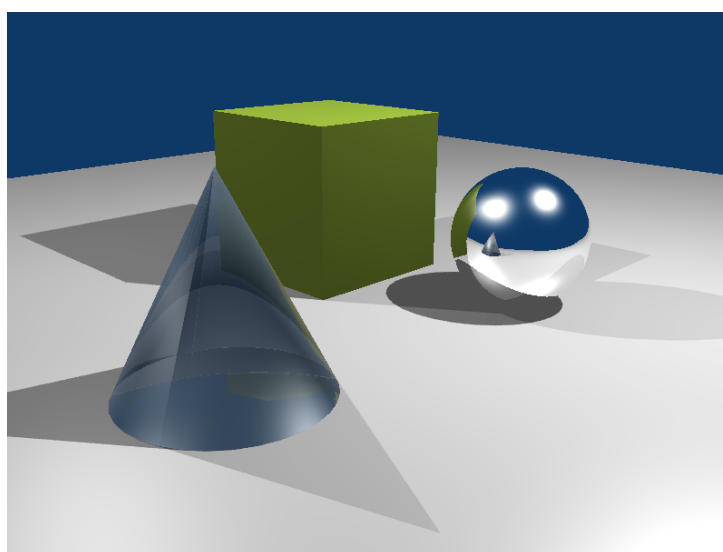


Figura 2.4: Cena de exemplo sintetizada com o uso de *ray tracing*.

Se a superfície tiver um componente especular, um novo raio será lançado na direção refletida (Figura 2.5) procurando pela interseção que irá retornar o valor deste componente. Se a superfície possuir certa transparência, um novo raio será lançado para o lado da superfície oposto ao que foi intersectado inicialmente, levando em conta o índice de refração da superfície intersectada para calcular a direção do novo raio (Figura 2.6).

Se o raio incidente não for totalmente tratado por reflexão e transmissão (ou seja, se os índices de especularidade e de transparência da superfície não totalizarem a radiância desta), significa que a superfície também possui um parcela de reflexão difusa (reflexão Lambertiana). Para calcular a quantidade de luz que incide na superfície, raios são gerados a partir do ponto intersectado em direção a todas as fontes de luz presentes na cena (Figura 2.7). Se o raio

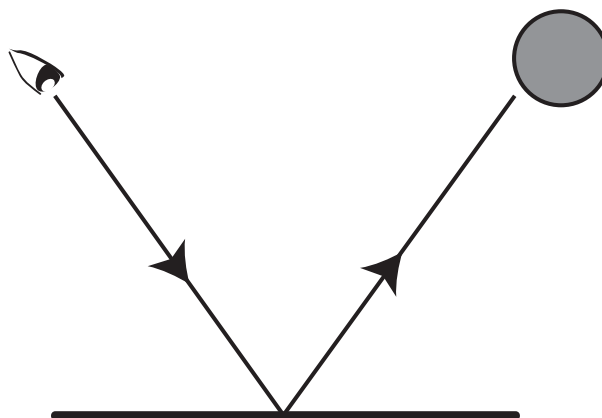


Figura 2.5: Novo raio gerado a partir da interseção do raio original com uma superfície que possui certa especularidade.

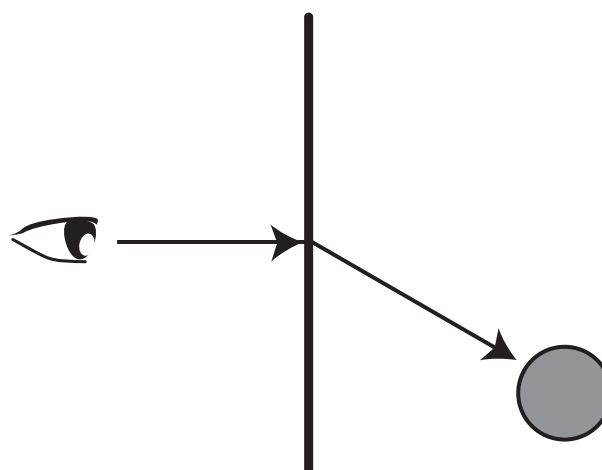


Figura 2.6: Novo raio gerado a partir da interseção do raio original com uma superfície que possui certa transparência, levando em conta o índice de refração desta.

lançado consegue ir do ponto da superfície até a fonte de luz sem intersectar outros objetos, ou se este intersecta apenas superfícies translúcidas, isto significa que existe visibilidade entre estes (este raio é chamado de raio de iluminação ou *illumination ray*) e, portanto, o ponto específico da superfície não está obstruído e a quantidade de luz sendo emitida pela fonte na direção do ponto de interesse é adicionada à luminosidade deste. Se o raio lançado encontra algum objeto entre o ponto e a fonte de luz, isto significa que aquele ponto da superfície está em sombra em relação àquela fonte de luz (chamado de raio de sombra ou *shadow ray*).

Este processo é repetido recursivamente para cada um dos novos raios gerados (com exceção dos raios de iluminação e de sombra) até que uma superfície totalmente difusa seja alcançada ou até que um limite máximo de níveis de recursão seja atingido (Figura 2.8). Devido



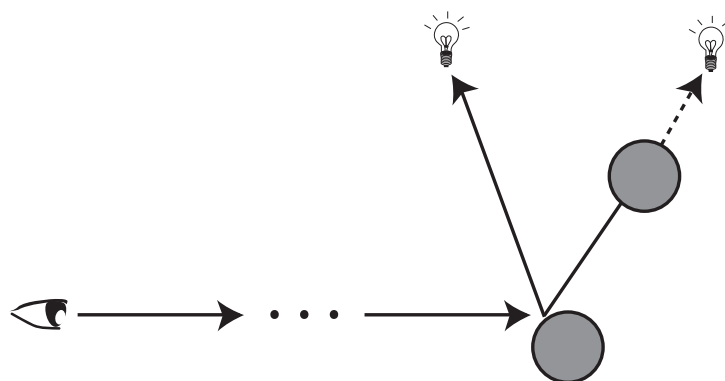


Figura 2.7: Para superfícies que possuem reflexão difusa, raios são gerados em direção às fontes de luz para identificar se o objeto está iluminado ou em sombras.

à essa característica recursiva, o cálculo do valor de cada pixel da imagem sendo sintetizada (cada pixel requer pelo menos um raio a partir do visualizador) pode gerar uma quantidade extremamente grande de novos raios, potencialmente percorrendo toda a cena.

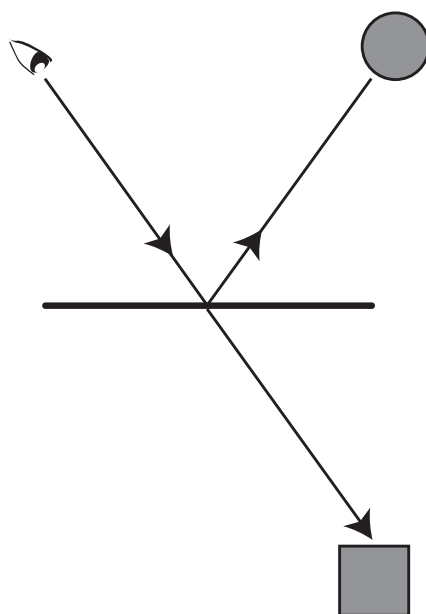


Figura 2.8: Para cada interseção, novos raios são gerados recursivamente de acordo com as características da superfície.

A principal inovação desta técnica foi permitir que fenômenos como reflexo entre objetos, sombras e superfícies translúcidas pudessem ser facilmente simuláveis. O cálculo da interseção dos raios com objetos e superfícies é geometricamente simples [14, 20, 40, 56], o que contribuiu para que esta técnica se tornasse bastante popular.

Por esta técnica realizar uma discretização do espaço da cena (que é de natureza contínua),

é possível a observação de ruídos na imagem sintetizada que a deixam com uma aparência serrilhada (*aliasing*), conforme pode ser observado na Figura 2.9.

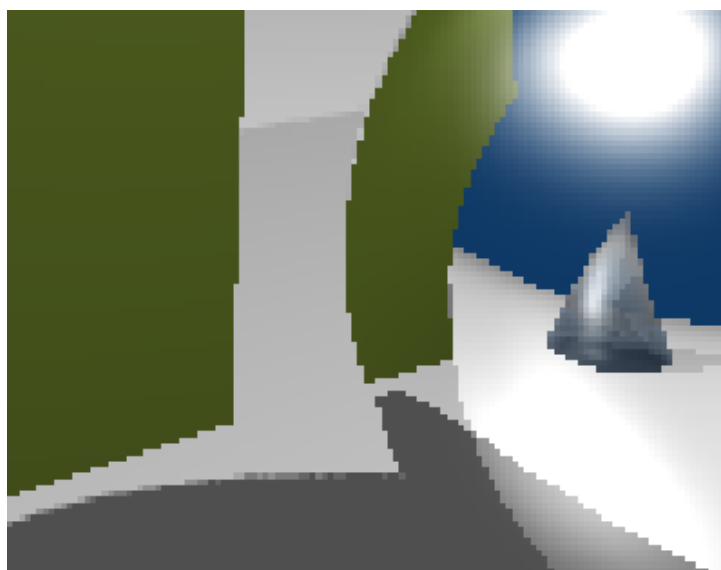


Figura 2.9: Detalhe da Figura 2.4 mostrando o *aliasing* causado na discretização do espaço feita pela técnica de *ray tracing*.

Outra característica perceptível é a presença de sombras e reflexos bem definidos (diferente do que ocorre na natureza, onde são observados sombras com uma região de penumbra e reflexos com uma aparência borrada). Através da distribuição e amostragem de raios extras (*stochastic ray tracing* [10, 11]) é possível resolver o problema de *aliasing*, das sombras e dos reflexos bem definidos e ainda simular alguns outros fenômenos (Figura 2.10). Se estes raios forem amostrados num determinado intervalo de tempo (considerando que cada raio foi obtido num instante de tempo diferente), podemos simular o efeito de desfocagem por movimento (*motion blur*). Através da amostragem da área da lente da câmera, podemos obter o efeito de profundidade de campo (o mesmo produzido pelas lentes de câmeras fotográficas).

A técnica de *ray tracing* não suporta inter-reflexão difusa, mas algumas melhorias (seção 2.4.5) foram feitas para suprir essa necessidade. Baseadas nessas tentativas, outras técnicas surgiram posteriormente combinando as características positivas do *ray tracing* e eliminando suas principais limitações.



Figura 2.10: O uso de raios extras (*stochastic ray tracing*) possibilita a remoção de defeitos causados por *aliasing*.

## 2.4.2 Radiosidade

Com uma abordagem um pouco diferente da utilizada pela técnica de *ray tracing*, a técnica de Radiosidade [21] foi proposta. A técnica consiste em subdividir as superfícies da cena em uma ou mais superfícies menores (*patches*) para as quais um fator de visibilidade é calculado aos pares (chamados de *form factors*), indicando o quanto cada par da superfície é visível entre si. Estes fatores são usados como coeficientes em um sistema de equações lineares que, quando resolvidas, resultam no valor da radiosidade (ou brilho) de cada *patch* (Figura 2.11).

A equação para o cálculo da radiosidade pode ser resumida da seguinte forma:

$$B_j = E_j + \rho_j H_j \quad (2.23)$$

em que  $B_j$  é a radiosidade da superfície  $j$ , ou seja, a quantidade total de energia saindo da superfície por unidade de tempo por unidade de área (*watts/metro<sup>2</sup>*),  $E_j$  é quantidade de energia emitida pela superfície por unidade de tempo por unidade de área (*watts/metro<sup>2</sup>*),  $\rho_j$  indica a reflexividade da superfície  $j$  e representa a fração da luz incidente que será refletida

e, por fim,  $H_j$  é o total de luz incidindo na superfície  $j$ , e pode ser calculado como

$$H_j = \sum_{i=1}^N B_i F_{ij} \quad (2.24)$$

em que  $B_i$  é a radiosidade da superfície  $i$ , e  $F_{ij}$  representa a fração da energia saindo da superfície  $i$  que incide na superfície  $j$ .

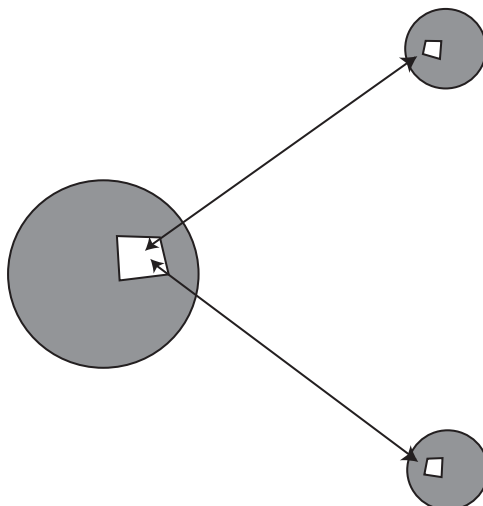


Figura 2.11: A luminosidade de cada *patch* é calculada através de todos os *form factors* que o contêm.

O método de Radiosidade é independente de visualização, o que possibilita que a iluminação de uma cena possa ser pré-calculada e utilizada em diversos quadros com a câmera posicionada em locais diferentes da cena.

Cohen e Greenberg [9] estenderam a técnica de radiosidade inicialmente proposta possibilitando a sua utilização em diversos casos não cobertos originalmente. Dentre os casos, destaca-se a possibilidade de aplicação em cenas mais complexas que possuam superfícies obstruídas, através do uso de um algoritmo que adiciona testes de visibilidade no cálculo dos *form factors*. Este algoritmo foi chamado de *hemi-cube*.

Cohen et al. [8] também propuseram modificações que tornavam possível uma abordagem de refinamento progressivo do resultado da renderização, também reduzindo o uso de memória através do cálculo dinâmico dos *patches*. Nesta técnica, a cada passo, o *patch* de maior brilho é escolhido e sua radiosidade é lançada aos *patches* que são visíveis a este. Este procedimento

é repetido até que a solução tenha convergido. Ao contrário da técnica original, em cada passo da técnica, o valor da radiosidade de cada *patch* ainda não é o valor final. Com essas melhorias, a técnica de radiosidade tornou-se uma opção viável para uso em renderizações interativas, nas quais o retorno para o usuário deve ser apresentado no menor tempo possível.

Outra abordagem de refinamento progressivo para radiosidade foi desenvolvida com o objetivo de acelerar a convergência do método progressivo proposto anteriormente [54]. Este método combina aspectos da solução de radiosidade original com aspectos do método de refinamento progressivo. Em cada passo, além de lançar a radiosidade do *patch* de maior brilho, as radiosidades parciais dos *patches* que ainda não foram processados são coletadas e lançadas ao *patch*, sendo somadas e lançadas juntamente com a radiosidade deste.

Hanrahan, Salzman e Aupperle [26] apresentaram uma abordagem hierárquica para a matriz de *form factors* que, junto com o uso de técnicas de *ray tracing*, propõe heurísticas de otimização para o teste de visibilidade, tornando o algoritmo mais eficiente. Esta abordagem foi posteriormente melhorada através do uso de funções de importância [57], aproximando o cálculo de *form factors* que contribuem pouco para imagem final. Apesar de tornar a técnica de radiosidade dependente da visualização, esta abordagem baseada em importância apresenta uma otimização considerável sem comprometer muito o resultado final.

As técnicas baseadas em radiosidade apresentam uma limitação bastante severa: seu uso somente é aplicável a superfícies reflexivas difusas (Lambertianas), nas quais a luz incidente é dispersada de forma homogênea em todo o hemisfério da superfície. A justificativa apresentada para a não inclusão de superfícies especulares é a de que estas contribuem pouco para a iluminação global. Entretanto, a contribuição destas não se limita à iluminação global, sendo também responsáveis por efeitos causados por reflexão especular.

As aproximações poligonais usadas para calcular os *patches* causam erros de cálculo das sombras e erros de desenho dos polígonos. Para reduzir a visibilidade de tais erros, um passo extra é utilizado para reavaliar o valor de cada pixel baseado nas informações já calculadas (*final gathering*) [41]. Este passo é bem custoso e aumenta consideravelmente o tempo da renderização. Scheel, Stamminger e Seidel [52] propuseram algumas otimizações para o cálculo desse passo final através da classificação dos emissores pela sua importância [53] e também o

uso de uma grade (*grid*) para interpolar os valores de iluminação em superfícies que possuam uma geometria detalhada (nas quais o valor de iluminação varia pouco sobre a superfície).

### 2.4.3 Equação da Renderização

Com a finalidade de facilitar a categorização, comparação e identificação dos elementos das diferentes técnicas de renderização, uma equação integral foi proposta por Immel et al. [29] e Kajiya [35]. Esta equação segue uma abordagem semelhante à equação do método de radiosidade (apresentado na seção 2.4.2), balanceando o fluxo de energia entre pontos da cena, mas não faz uso explícito de nenhuma característica de reflexão das superfícies.

A Equação da Renderização apresenta a forma

$$I(x, x') = g(x, x') \left[ \epsilon(x, x') + \int_S \rho(x, x', x'') I(x', x'') dx'' \right] \quad (2.25)$$

em que  $I(x, x')$  indica a intensidade de luz passando de  $x'$  para  $x$ ,  $g(x, x')$  é um termo geométrico representando a visibilidade entre  $x$  e  $x'$  e  $\epsilon(x, x')$  indica a intensidade de luz emitida de  $x'$  para  $x$ .  $S$  representa a união de todas as superfícies de todos os objetos da cena e, portanto, a integral em  $S$  cobre todos os pontos da cena. Nesta integral,  $\rho(x, x', x'')$  é a intensidade da luz transmitida de  $x''$  para  $x$  passando pela superfície em  $x'$ .

O segundo componente da integral em  $S$ ,  $I(x', x'')$  mostra claramente que a solução desta equação não é simples, pois para o cálculo da intensidade de luz que passa de  $x'$  para  $x$ , faz-se necessário conhecer a quantidade de luz que é transmitida de todos os pontos  $x''$  para  $x'$ .

A maioria dos algoritmos apresentados até o surgimento da equação da renderização pode ser facilmente mapeado para os componentes desta, o que fez com que a partir deste ponto, a equação da renderização e suas variantes passassem a ser utilizadas como referência para a elaboração, modelagem e avaliação de novas técnicas de renderização.

### 2.4.4 Path Tracing

Junto com a equação da renderização, Kajiya [35] propôs uma solução para ela, chamada de *path tracing*. Esta técnica foi uma das primeiras de uso geral a usar métodos de Monte

Carlo (seção 2.3) para estimar o valor da integral da radiância em pontos de interesse.

Seu funcionamento é muito semelhante ao algoritmo de *ray tracing* (seção 2.4.1), mas este não se limita a calcular a iluminação direta incidente em objetos. Um raio é traçado a partir da câmera na direção que se deseja calcular. Se uma interseção é encontrada, torna-se necessário o cálculo da radiância do ponto encontrado, ou seja, a luz incidindo no ponto encontrado vinda a partir de fontes de luz ou de outras superfícies (sejam elas difusas ou especulares). O cálculo correto deste valor exigiria a solução da integral vista na seção 2.4.3 na esfera unitária ao redor do ponto de interseção, identificando toda a radiância incidente. Como este cálculo é computacionalmente inviável, a técnica utiliza integração de Monte Carlo (seção 2.3.1) para aproximar o resultado desta integral.

Esta amostragem é feita através da escolha de caminhos a partir do ponto da primeira interseção (Figura 2.12). Cada um dos caminhos é construído incrementalmente. Em cada vértice do caminho, uma nova direção é amostrada (de acordo com as propriedades da superfície) e o novo vértice é encontrado traçando um raio na direção amostrada para encontrar a interseção mais próxima.

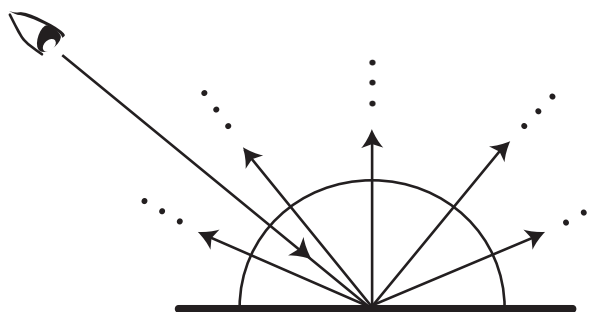


Figura 2.12: Caminhos são escolhidos por amostragem para resolver a integral da iluminação do ponto intersectado no algoritmo de *path tracing*.

O caminho gerado é possivelmente infinito, visto que a inter-reflexão difusa pode ocorrer infinitamente na cena. Entretanto, conforme o caminho se torna mais longo, a contribuição de novos vértices para a radiância final diminui. Como seria computacionalmente inviável calcular caminhos infinitos, a técnica de Roleta Russa (seção 2.3.2) é utilizada para decidir quando um caminho é interrompido. A geração de novos vértices para o caminho também pode ser terminada pela imposição de uma profundidade limite.

Quando o caminho é interrompido, a radiância do último vértice encontrado é calculada baseando-se apenas na iluminação direta incidindo no ponto em relação às fontes de luz da cena. Este cálculo é feito utilizando os mesmos conceitos apresentados na seção 2.4.1, traçando raios a partir do ponto em direção às fontes de luz e com o uso de *shadow rays* para determinar se o caminho entre a fonte de luz e o ponto está obstruído por outro objeto. A Figura 2.13 ilustra a geração de um caminho completo de *path tracing*.

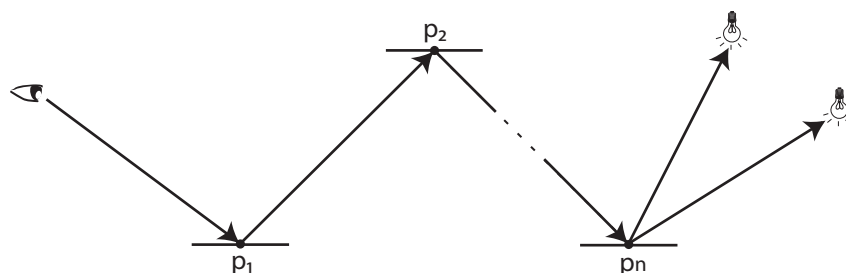


Figura 2.13: Um caminho gerado pelo algoritmo de *path tracing*. Para o último vértice do caminho, sua contribuição é definida através de iluminação direta.

## Path Tracing Bidirecional

Apesar de produzir bons resultados em geral, existem algumas cenas particulares que fazem com que o resultado produzido pela técnica de *path tracing* possua uma aparência bastante granulada. A Figura 2.14 ilustra uma cena com essa característica. A iluminação direta da cena ocorre apenas em uma pequena região do teto, enquanto que toda iluminação restante se dá indiretamente.

A probabilidade de a técnica de *path tracing* gerar caminhos que alcancem a fonte de luz é muito pequena e, portanto, temos vários caminhos que contribuem pouco para a iluminação, e alguns caminhos que terão uma radiância maior. O efeito visual causado pela variância pode ser visto claramente na Figura 2.17(a).

A fim de melhorar os resultados, uma variação do algoritmo de *path tracing* original foi desenvolvida. Esta variação, chamada de *path tracing bidirecional* [38, 60, 61] segue o mesmo princípio do algoritmo original: traçar caminhos incrementais para estimar a radiância. Entretanto, ele se difere por dividir o caminho em dois para a construção. A primeira parte do caminho é construída a partir do visualizador e a segunda parte a partir de uma fonte de luz.



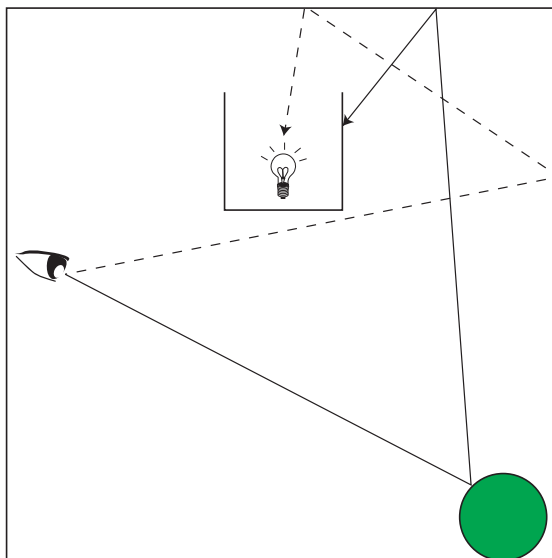


Figura 2.14: Cena que gera resultados com uma variância bastante alta com o uso de *path tracing*.

A conexão destas duas partes pode ser feita de duas maneiras diferentes. A maneira proposta por Veach e Guibas [61] é a junção destes caminhos nos vértices centrais. Para tornar possível essa conexão, torna-se necessário traçar um *shadow ray* entre o último vértice de cada um dos caminhos para garantir que estes sejam conectáveis (Figura 2.15).

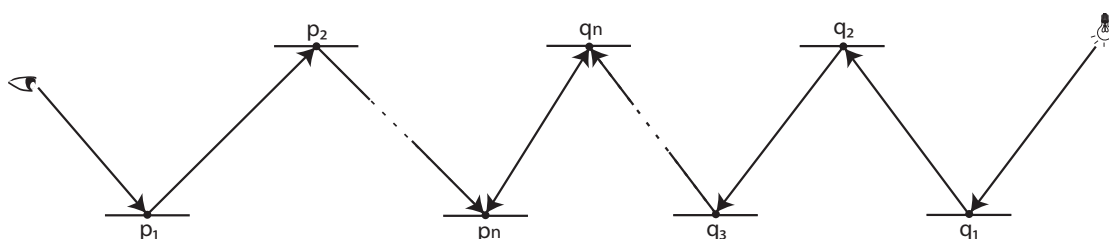


Figura 2.15: O caminho gerado por *path tracing bidirecional* se inicia nos dois extremos: no visualizador e em uma fonte de luz.  $p_i$  representa um vértice gerado a partir do visualizador e  $q_i$  indica um vértice gerado a partir da fonte de luz. A junção dos caminhos nos vértices  $q_n$  e  $p_n$  requer um *shadow ray* para determinar a visibilidade entre eles.

A segunda maneira, proposta por Lafortune e Willems [38], é a de conectar todos os vértices do caminho do visualizador a todos os vértices do caminho da fonte de luz (Figura 2.16). Para cada conexão, um *shadow ray* é traçado para testar a visibilidade entre os dois vértices, e a radiância é calculada. Cada caminho gerado por estas conexões recebe um determinado peso (de acordo com sua contribuição para o resultado final) para o balanceamento da radiância sendo estimada.

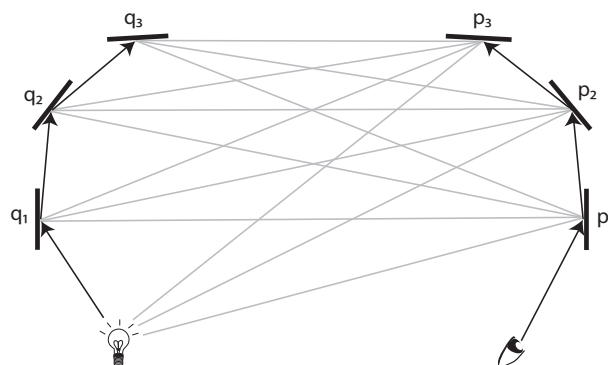


Figura 2.16: O método de *path tracing* bidirecional descrito por Lafortune [38] realiza conexões entre todos os vértices de cada uma das partes do caminho.

Com o uso desta técnica para renderizar a cena mostrada na Figura 2.14, é possível perceber que mais raios conseguirão alcançar a fonte de luz e, portanto, a variância será reduzida, conforme podemos observar na Figura 2.17(b).

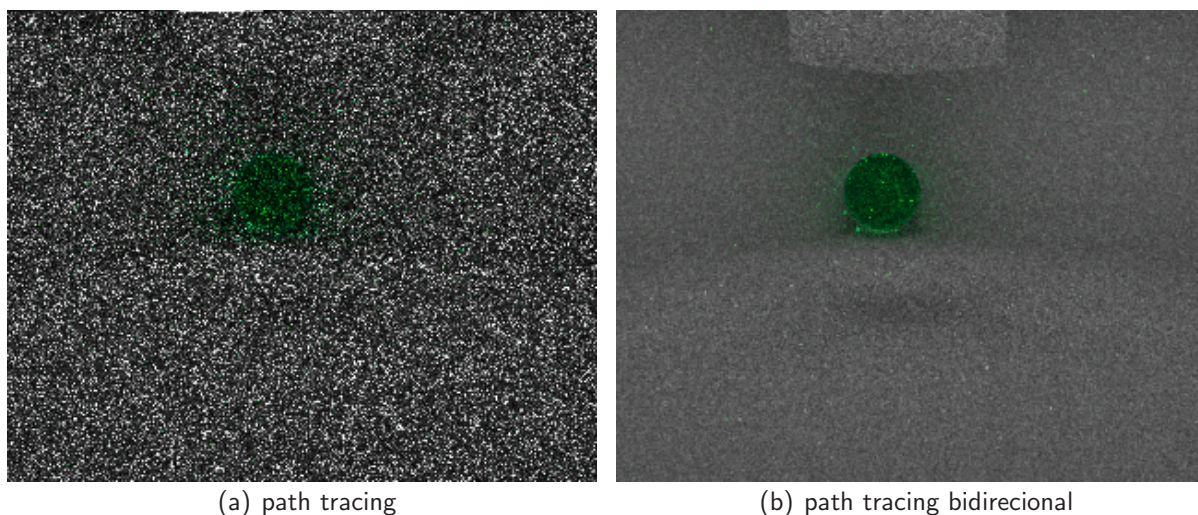


Figura 2.17: Cena complexa para o algoritmo de *path tracing* gera resultados com uma variância bem menor com o uso de *path tracing* bidirecional.

## 2.4.5 Irradiance Caching

As técnicas baseadas em integração de Monte Carlo apresentadas anteriormente apresentam uma característica indesejável: a variância característica destes métodos é vista no resultado final na forma de ruídos de alta frequência [47]. Como visto na seção 2.3.3, a convergência dos métodos de integração de Monte Carlo exigem o uso de uma quantidade de amostras relativamente alta. A técnica de *irradiance caching* [65, 67] explora o fato de que

a radiância causada por iluminação indireta em superfícies difusas, apesar de computacionalmente cara de ser calculada, varia de forma bastante suave na maioria das cenas.

Para cada ponto em que seja necessário o cálculo da irradiância, um *cache* é consultado em busca de valores calculados para pontos próximos. Se o resultado encontrado for suficiente para estimar a irradiância do ponto, os valores encontrados são interpolados e utilizados. Caso não sejam encontrados elementos suficientes no *cache*, a irradiância do ponto é calculada utilizando alguma técnica de iluminação global (*path tracing* normalmente é uma boa escolha - seção 2.4.4) e o valor encontrado é inserido no *cache*. Em áreas mais planas e abertas (onde a variação da irradiância ocorre de forma mais suave), mais pontos serão calculados utilizando o *cache*. Em superfícies com uma curvatura maior ou próximas a outros objetos, a variação da irradiância ocorre de forma mais brusca e, portanto, poucos pontos poderão ser calculados através do uso do *cache*. Apesar de esta técnica ser tendenciosa (*biased*) e os resultados produzidos não refletirem exatamente a irradiância da superfície, os resultados gerados não apresentam os ruídos característicos da integração de Monte Carlo.

Para estimar a irradiância de um novo ponto  $P$ , os pontos  $P_i$  presentes no *cache* são consultados. Para cada ponto encontrado, um peso é calculado para decidir se o ponto deve ser ou não utilizado. Este peso é calculado como

$$w_i(P) = \frac{1}{\epsilon_i(P)} \quad (2.26)$$

em que  $\epsilon_i(P)$  representa uma estimativa de quanto a irradiância mudou de  $P_i$  para  $P$ . Esta estimativa é calculada através de modelo de esfera dividida [67]. Este modelo assume um hemisfério acima de  $P_i$ , sendo que metade deste hemisfério é preta e a outra metade é branca. Com isto, este hemisfério representa um limite bem definido para a iluminação incidente, provendo uma boa estimativa das mudanças mais relevantes na irradiância conforme o ponto  $P$  se afasta de  $P_i$ . As duas mudanças possíveis que são detectadas por esta esfera dividida são a mudança na localização e na orientação da superfície. A partir deste modelo, a mudança na

irradiância (erro) pode ser estimada como

$$\epsilon_i(P) = \frac{\|P_i - P\|}{R_i} + \sqrt{1 - N_P \bullet N_{P_i}} \quad (2.27)$$

em que  $R_i$  é a média harmônica das distâncias às superfícies encontradas quando do cálculo da amostra de irradiância,  $N_P$  é o vetor normal da superfície no ponto cuja irradiância está sendo estimada e  $N_{P_i}$  é o vetor normal da superfície no ponto em que a amostra de irradiância foi calculada.

Se o ponto possui um valor de erro alto (grande mudança), seu peso é pequeno e vice-versa. Um limiar deve ser especificado manualmente para eliminar amostras cujo peso seja pequeno demais. A irradiância interpolada no ponto  $P$  é calculada como

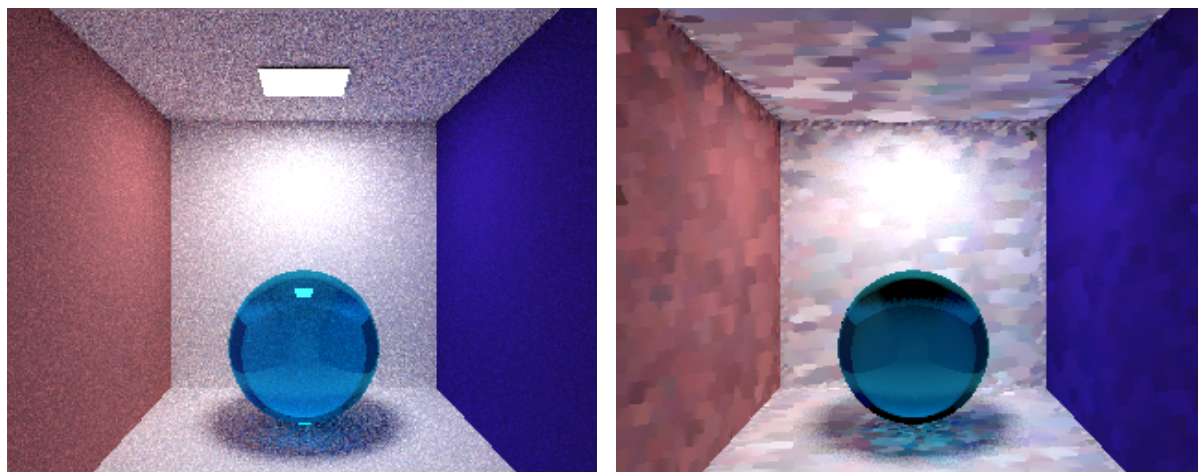
$$E(P) = \frac{\sum_{i=1}^N w_i(P) E_i(P)}{\sum_{i=1}^N w_i(P)} \quad (2.28)$$

em que  $E_i$  é a irradiância no ponto  $P_i$  extrapolada para o ponto  $P$ .

O *cache* da irradiância é armazenado em uma *octree* [14] com a dimensão da cena. Quando a irradiância em um ponto precisa ser calculada, uma busca é feita nessa *octree* para encontrar pontos próximos que possam ser utilizados para interpolar a irradiância. O grau de proximidade a ser utilizado é baseado no valor fornecido manualmente para o limiar do peso da amostra. A utilização ou não de uma amostra é determinada pelo cálculo do erro descrito anteriormente.

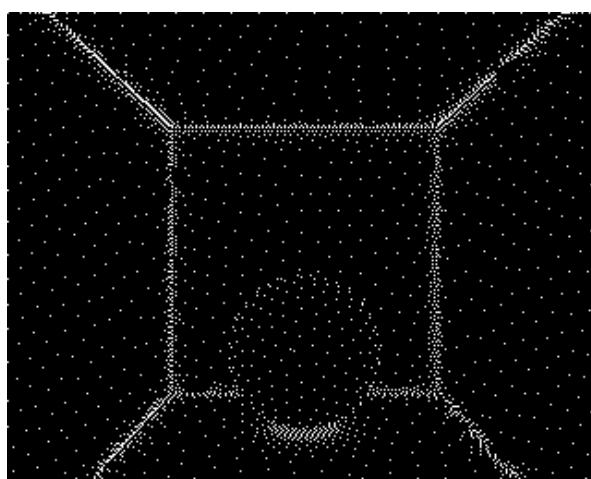
A Figura 2.18 mostra a diferença do resultado produzido com o uso das técnicas de *path tracing* e *irradiance caching*. É possível observar que a iluminação das superfícies da figura renderizada com *path tracing* (Figura 2.18(a)) possui uma variância bastante grande que pode ser vista pelo aspecto granuloso. Na imagem gerada com o uso de *irradiance caching* (Figura 2.18(b)), este ruído não é observado, porém, o resultado não é totalmente preciso. Também é possível observar na Figura 2.18(c) que, em regiões cuja orientação da superfície muda de forma mais abrupta, mais pontos precisam ter sua irradiância calculada para que a estimativa da irradiância seja feita corretamente. Para as duas imagens foram utilizadas oito

amostras para o cálculo da iluminação indireta.



(a) *path tracing*

(b) *irradiance caching*



(c) pontos em que o cálculo da irradiância foi necessário

Figura 2.18: Diferença de resultados obtidos com o uso de *path tracing* e *irradiance caching*.

Apesar de não apresentar os ruídos relacionados à variância, o resultado mostrado na Figura 2.18(b) apresenta várias discontinuidades. O uso de vetores gradiente [37, 66] foi proposto como forma de amenizar estas discontinuidades. As amostras do hemisfério utilizadas para gerar uma nova entrada no cache de irradiância têm sua direção e orientação interpretadas e utilizadas como forma de prever mudanças na irradiância em função da posição e da orientação desta.

Dois gradientes foram propostos: um gradiente para a rotação e outro para a translação. Para tornar possível o cálculo desses gradientes, a amostragem do hemisfério deve ser feita de forma estratificada, dividindo-o em células cuja projeção no plano tangente gera um disco

dividido em partes de igual área projetada (Figura 2.19). O gradiente de rotação é definido como

$$\nabla_r E = \frac{\pi}{MN} \sum_{k=0}^{N-1} \left( v_k \sum_{j=0}^{M-1} \tan \theta_j L_{j,k} \right) \quad (2.29)$$

em que  $M$  e  $N$  definem o número de células geradas na divisão do hemisfério,  $\theta_j$  é o ângulo polar do centro da célula atual,  $v_k$  é um vetor unitário no plano na direção  $\Phi_k - \frac{\pi}{2}$ ,  $\Phi_k$  é o ângulo azimutal do centro da célula atual e  $L_{j,k}$  é a radiância incidente na direção  $(\theta_j, \Phi_k)$ .

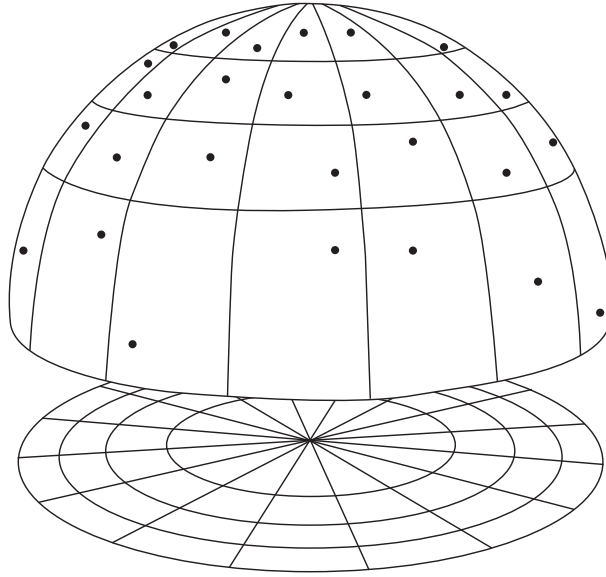


Figura 2.19: Para o cálculo dos gradientes de irradiância é necessário que o hemisfério seja amostrado de forma estratificada. Adaptado de [37].

O gradiente de translação, por sua vez, é definido como

$$\nabla_t E = \sum_{k=0}^{N-1} \left\{ u_k \frac{2\pi}{N} \sum_{j=1}^{M-1} \frac{\sin \theta_{j-} \cos^2 \theta_{j-}}{\min(R_{j,k}, R_{j-1,k})} (L_{j,k} - L_{j-1,k}) + v_{k-} \sum_{j=0}^{M-1} \frac{\sin \theta_{j+} - \sin \theta_{j-}}{\min(R_{j,k}, R_{j,k-1})} (L_{j,k} - L_{j,k-1}) \right\} \quad (2.30)$$

em que  $u_k$  é um vetor unitário na direção de  $\Phi_k$ ,  $v_{k-}$  é um vetor unitário na direção  $\Phi_{k-} + \frac{\pi}{2}$ ,  $\theta_{j-}$  é o ângulo polar na divisa da célula atual com a célula anterior,  $\theta_{j+}$  é o ângulo polar na divisa com a próxima célula,  $\Phi_{k-}$  é o ângulo azimutal na divisa com a célula anterior e  $R_{j,k}$  é a distância de interseção encontrada pelo caminho gerado na célula  $(j, k)$ .

Estes gradientes são então utilizados na interpolação das entradas do *cache* para um



determinado ponto como

$$E(P) = \frac{\sum_S w_i(P) \left[ E_i(P) + (N_{P_i} \times N_P) \bullet \nabla_r E_i + (P - P_i) \bullet \nabla_t E_i \right]}{\sum_S w_i(P)} \quad (2.31)$$

A diferença nos resultados obtidos com o uso dos gradientes de rotação e translação pode ser visualizada na Figura 2.20.

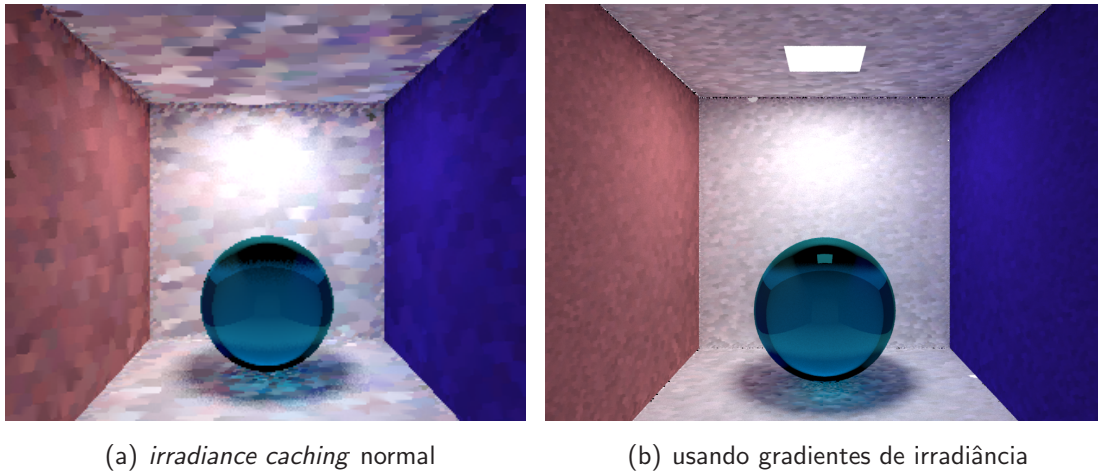


Figura 2.20: O uso de gradientes de irradiância diminui a presença de discontinuidades no resultado gerado.

Outra maneira de amenizar as discontinuidades apresentadas nos resultados foi proposta por Tabellion e Lamorlette [59] através de uma nova maneira de calcular o peso e o erro das amostras em relação ao ponto sendo interpolado, fazendo com que, mesmo em cenas com um nível de detalhes mais elevado, o resultado seja mais preciso. O erro de uma amostra é calculado como

$$\epsilon_i(P) = k \max \left( \epsilon_{pi}(P), \epsilon_{ni}(N_P) \right) \quad (2.32)$$

sendo que  $\epsilon_{pi}(P)$  é definido como

$$\epsilon_{pi}(P) = \frac{\|P - P_i\|}{\max \left( \min \left( \frac{R_i}{2}, R_+ \right), R_- \right)} \quad (2.33)$$

e  $\epsilon_{ni}(N_P)$  como

$$\epsilon_{ni}(N_P) = \frac{\sqrt{1 - N_P \cdot N_{P_i}}}{\sqrt{1 - \cos(\alpha_+)}} \quad (2.34)$$

em que  $R_i$  é a menor distância às superfícies encontradas quando do cálculo da amostra de irradiância,  $R_+$  e  $R_-$  são 10 e 1.5 vezes o valor de  $R_i$  e  $\alpha_+$  é um ângulo fixo em  $10^\circ$  no algoritmo. Esses valores foram derivados a partir do ângulo a partir do qual o ponto  $P$  pode ser visto.  $k$  é um parâmetro configurável pelo usuário para controlar o nível de aceitação do erro.

O peso de cada amostra é então definido como

$$w_i(P) = 1 - \epsilon_i(P) \quad (2.35)$$

## Irradiance Cache com Refinamento Adaptativo

Apesar de modificações nos cálculos do peso e do erro melhorarem os resultados, estes ainda apresentam descontinuidades. Isto se deve ao fato de estas métricas levarem em conta somente a distância do ponto à amostra e os vetores normais. Nenhuma verificação é feita em relação à homogeneidade das amostras utilizadas para interpolar um determinado ponto.

Krivánek et. al. [36] propuseram uma variação da técnica de *irradiance cache* que controla a densidade das amostras necessárias para cada região analisando as mudanças na iluminação indireta. O algoritmo se difere da técnica tradicional pelo fato de que o cálculo da iluminação indireta não é feito imediatamente junto com o cálculo da iluminação direta. Em vez disso, os pontos cuja iluminação indireta precisam ser calculados são salvos em uma lista para avaliação posterior.

Após o término do cálculo da iluminação direta de todos os pontos, um segundo passo é iniciado. Neste passo, todos os pontos são inseridos em uma fila e processados um a um. Junto com cada ponto, uma lista (inicialmente vazia) das amostras que contribuem para sua iluminação indireta é armazenada. No início do segundo passo, um ponto é desenfileirado e as amostras cuja contribuição para o ponto não são mais válidas, de acordo com critérios que serão descritos posteriormente, são removidas da lista do ponto.

Se todas as amostras da lista forem removidas, uma nova amostra de irradiância precisa ser calculada e é inserida na lista desse ponto. O algoritmo identifica, então, todos os pontos



que se encontram dentro do raio de influência desta amostra, determinado como

$$a_i R_i \quad (2.36)$$

em que  $a_i$  é o fator de influência da amostra (que será modificado ao longo do refinamento), e  $R_i$  é a média harmônica das distâncias encontradas no cálculo desta amostra. Estes pontos são inseridos na fila para que a contribuição desta nova amostra possa ser avaliada.

Se o ponto sendo processado possui mais de uma amostra em sua lista de contribuições, é feito um refinamento adaptativo das amostras do ponto. Neste refinamento, cada uma das amostras é testada para garantir que esta não cause nenhuma descontinuidade na irradiância do ponto. Este teste é feito de acordo com uma métrica de discernibilidade. Esta métrica faz uso de dois valores de irradiância,  $E_1(P)$  e  $E_2(P)$ , definidos como

$$E_1 = E(P) + L_d(P) \quad E_2 = E_{P_i}(P) + L_d(P) \quad (2.37)$$

em que  $L_d(P)$  é o valor da iluminação direta do ponto e  $E_{P_i}(P)$  é a interpolação da irradiância das amostras para o ponto  $P$  excluindo a amostra  $P_i$  que está sendo testada.

A amostra  $P_i$  é dita discernível do conjunto de amostras de  $P$  se

$$|Y(E_1) - Y(E_2)| > 0,02 Y(E_1) + \max(\sigma(E_1), \sigma(E_2)) \quad (2.38)$$

em que  $Y$  denota o canal de luminância de um valor e  $\sigma$  é o desvio padrão da irradiância, adicionado para diminuir a aleatoriedade do cálculo, visto que estas amostras foram calculadas através do uso de integração de Monte Carlo sobre seu hemisfério. O valor 0,02 vem do fato de que uma alteração de até 2% de diferença na luminância não é perceptível ao olho humano.

Se uma amostra  $P_i$  é discernível do conjunto, sua contribuição para o ponto  $P$  é removida. Isto é feito através do ajuste do valor de  $a_i$  para

$$a_i = \frac{\|P - P_i\|}{R_i} - \epsilon \quad (2.39)$$

em que  $\epsilon$  é um número muito pequeno (o suficiente para que  $P$  não esteja mais no raio de influência da amostra  $P_i$ ). A Figura 2.21 ilustra essa idéia.

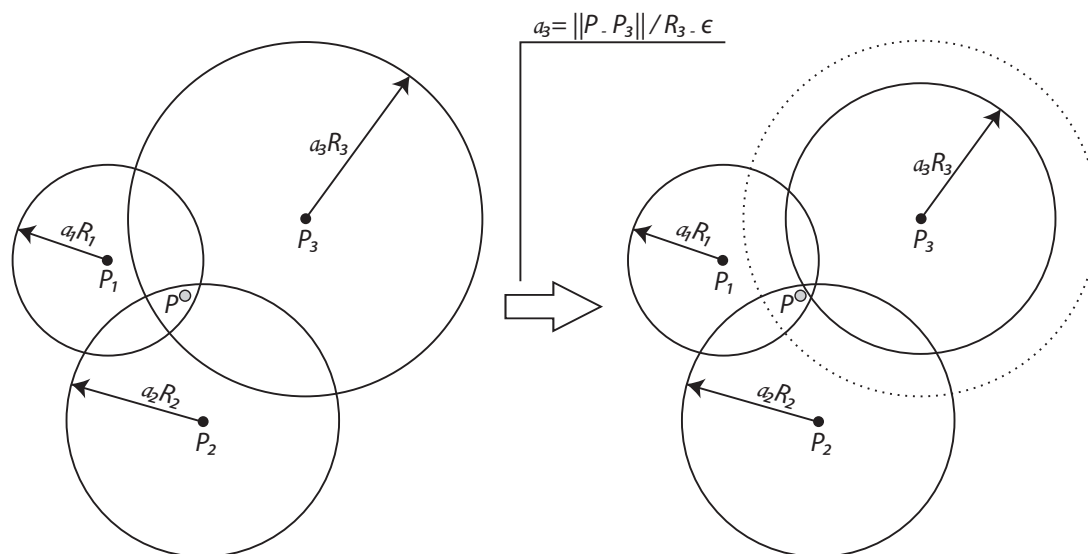


Figura 2.21: A amostra  $P_3$  é discernível do conjunto e portanto, seu raio de influência ( $a_3$ ) precisa ser reduzido para que ela não mais contribua para a irradiância no ponto  $P$ . Adaptado de [36].

Após o processamento de todos os pontos na fila, para cada uma das amostras que foram afetadas pelo refinamento adaptativo, os pontos que fazem uso desta amostra para a interpolação da irradiância são novamente enfileirados e um novo passo do algoritmo se inicia. Ao final do processamento, o algoritmo garante que todos os pontos são gerados com o uso de amostras de aparência homogênea, evitando descontinuidades na imagem.

## 2.4.6 Algoritmos de Múltiplos Passos

As técnicas apresentadas possuem aspectos positivos e algumas limitações. A técnica de radiosidade, por exemplo, é muito boa para uso em cenas com superfícies difusas, mas não é capaz de tratar superfícies especulares. A técnica de *ray tracing*, por sua vez, é muito recomendada para uso em superfícies especulares, mas não é capaz de simular reflexão difusa, ou o efeito de *caustics* (no qual a luz é refratada em uma superfície translúcida e focalizada em alguns pontos específicos, conforme a Figura 2.22). Essas particularidades de cada uma das técnicas causaram o surgimento de técnicas híbridas que operam em múltiplos passos combinando as características positivas e eliminando algumas limitações.

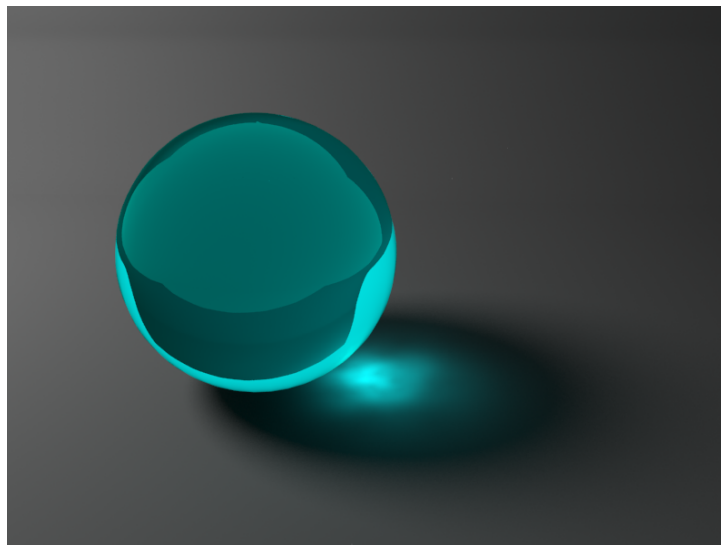


Figura 2.22: O efeito de *caustics*.

A fim de simular corretamente os efeitos de *caustics* usando o algoritmo de *ray tracing*, a técnica de *backward ray tracing* [2] utilizava um passo de pré-processamento no qual emitia diversos raios a partir de fontes de luz em direções diversas, cada um carregando uma certa quantidade de energia. Para armazenar a informação da energia que foi depositada em cada superfície da cena, um mapa de iluminação é utilizado. Este mapa de iluminação é bastante semelhante a um mapa de textura, com a diferença de carregar informações sobre iluminação em cada ponto da superfície. Apesar de resolver o problema da simulação de *caustics*, o algoritmo de *backward ray tracing* ainda possuía a limitação de não suportar inter-reflexão difusa nas superfícies.

Outra técnica, proposta por Wallace et al. [64], combina aspectos do algoritmo de *ray tracing* e do algoritmo de radiosidade. Esta solução utiliza um primeiro passo de pré-processamento que usa todos os princípios da técnica de radiosidade para determinar a inter-reflexão difusa e a iluminação indireta, e um segundo passo de pós-processamento baseado em *ray tracing* que, além de levar em conta a iluminação direta, também considera os valores calculados no pré-processamento para gerar o resultado final.

Apesar de melhorar significativamente os resultados, estas técnicas ainda não suportam iluminação indireta ocasionada por reflexão especular. O uso de um passo extra resolve este problema. Na técnica proposta por Shirley [55], um primeiro passo é executado apenas para

armazenar essa iluminação indireta, guardando estas informações em um mapa de iluminação. Em um segundo passo, uma variação da técnica básica de radiosidade é utilizada para calcular a inter-reflexão difusa, diferenciando-se do método original por omitir a iluminação direta (que é calculada em passo separado). O último passo é uma variação do algoritmo *stochastic ray tracing*, substituindo a constante de ambiente pela radiosidade calculada no segundo passo e pelo mapa de iluminação obtido no primeiro passo.

Uma limitação visível destas técnicas é o fato de estas só exibirem algum resultado no último passo de execução. Para uso no processo de criação das cenas, isto é bastante crítico pois para observar se um modelo está correto é necessário aguardar toda a execução até o último passo para só então corrigir eventuais problemas ou para ajustar parâmetros da cena (iluminação, etc). Para tanto, fazia-se necessária uma técnica que fosse tão completa quanto as apresentadas anteriormente, mas que fosse capaz de retornar resultados preliminares para avaliação e possível interrupção da execução.

A técnica de refinamento progressivo, proposta por Bergman et al. [3], tem como objetivo fundamental a produção rápida de resultados e, para tanto, a parcela de informação mais relevante deve ser produzida nos primeiros passos do algoritmo. Este resultado pode ser melhorado indefinidamente com o passar do tempo, convergindo para a solução exata. Seguindo estes princípios básicos, Chen et al. [6] modificaram as técnicas anteriormente propostas para que pudessem ser aplicadas de forma progressiva. Todos os caminhos de luz são divididos em quatro categorias:

- caminhos de iluminação direta: caminhos que vão da luz, passam por exatamente uma superfície difusa e por zero ou mais superfícies especulares até chegarem no visualizador (Figura 2.23(a));
- caminhos de *caustics*: caminhos que vão da luz, passando por uma ou mais superfícies especulares e terminando em uma superfície difusa. Este caminho gera o efeito de *caustics*, mas ainda é necessário o resto do caminho até chegar ao visualizador (a partir da superfície difusa, passando por zero ou mais superfícies especulares chegando no visualizador (Figura 2.23(b));

- caminhos de destaque (*highlight*): caminhos que vão da luz passando por zero ou mais superfícies especulares até chegarem no visualizador. Note que este tipo de caminho é chamado de destaque, pois não há nenhuma superfície difusa no percurso, ou seja, a luz vai diretamente ao visualizador sendo refletida em superfícies especulares (Figura 2.23(c));
- caminhos de radiosidade: caminhos que causam inter-reflexão difusa, ou seja, caminhos que se iniciam na luz e passam por pelo menos duas superfícies difusas até chegarem no visualizador (Figura 2.23(d)).

A fim de facilitar a separação dos caminhos nas categorias descritas, estes foram representados com o uso de expressões regulares[31], utilizando a seguinte notação:

- L: representa uma fonte de luz;
- E: representa um visualizador;
- D: representa uma superfície difusa;
- S: representa uma superfície especular;
- \*: indica que zero ou mais instâncias do elemento que precede serão utilizadas (por exemplo, S\* indica que o caminho passa por zero ou mais superfícies especulares);
- +: indica que uma ou mais instâncias do elemento que precede este sinal serão utilizadas (por exemplo, D+ indica que o caminho passa por uma ou mais superfícies difusas).

Para os caminhos de iluminação direta e de destaque, o uso de uma técnica de *path tracing* tradicional é suficiente. Para os caminhos de radiosidade, um algoritmo de radiosidade por refinamento progressivo é utilizado em conjunto com *path tracing*. Finalmente, para os caminhos de *caustics*, é usada a técnica de *backward ray tracing* em conjunto com *path tracing*. O refinamento progressivo é feito através de estimativas para alguns desses caminhos, que são aos poucos refinados para a obtenção de resultados mais próximos do real.

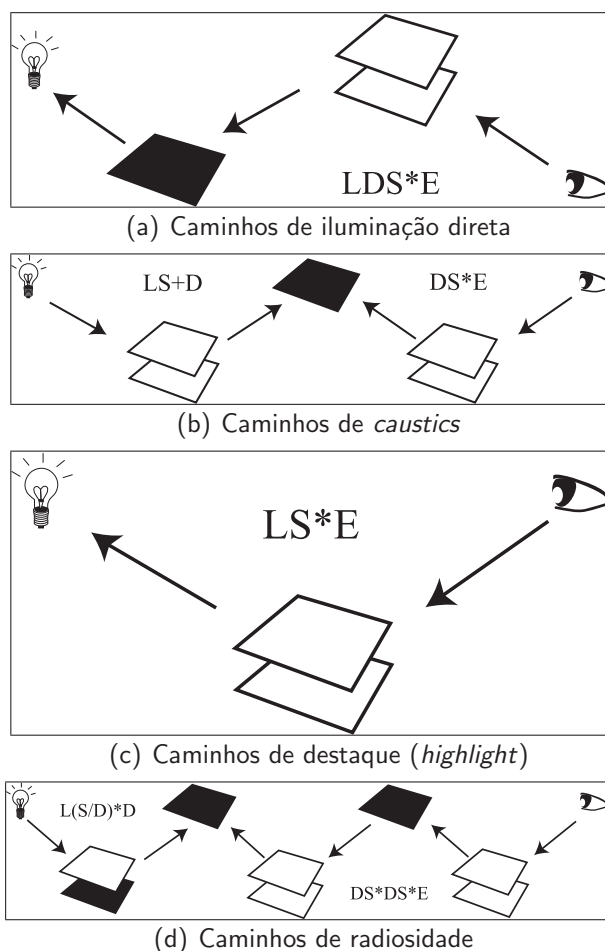


Figura 2.23: Os caminhos possíveis no método progressivo de Chen, Rushmeier, Miller e Turner [6]. Adaptado de [6].

## 2.4.7 Photon Mapping

Com o objetivo de sintetizar de forma eficiente imagens de qualidade elevada a partir de cenas com geometria complexa, a técnica de *photon mapping* [32, 34] foi criada. Outro requisito definido para esta técnica é a capacidade de tratar qualquer tipo de geometria e qualquer função de reflexão.

Métodos de radiosidade (seção 2.4.2) geram imagens com diversos defeitos (artefatos) e, em geral, não são muito adequados para uso em superfícies especulares, tendo um custo computacional extremamente elevado quando usado em cenas com geometria mais complexa. Técnicas de multi-passo (seção 2.4.6) também possuem sérias limitações em relação à geometria complexa (uma característica desse tipo de algoritmo), que são contornáveis através do uso de simplificações das malhas poligonais. Entretanto, esse processo de simplificação

geométrica em geral é um processo que requer intervenção manual e que resulta em erros difíceis de serem filtrados. Na técnica de *photon mapping*, a iluminação não está acoplada à geometria da cena. Com isso, a restrição da complexidade da cena é eliminada.

Ao contrário de técnicas como *ray tracing* (seção 2.4.1), a técnica de *photon mapping* não lança raios em direção à cena. Em lugar de raios, fótons são emitidos a partir das fontes de luz da cena. Um ponto importante do algoritmo é a escolha dos pontos na fonte de luz a partir de onde os fótons serão emitidos. Para cada tipo de fonte de luz, uma abordagem é utilizada para decidir a partir de quais pontos os fótons serão lançados (a Figura 2.24 ilustra o lançamento de fótons em alguns tipos de fontes de luz). Em fontes de luz pontuais (aquelas representadas apenas por um único ponto), esta decisão é bastante simples: os pontos serão lançados a partir do único ponto que a fonte possui e direções aleatórias são escolhidas (Figura 2.24(a)). Já no caso de fontes de luz quadradas, primeiro é escolhido um ponto aleatório na superfície desta e, logo em seguida, uma direção aleatória é escolhida.

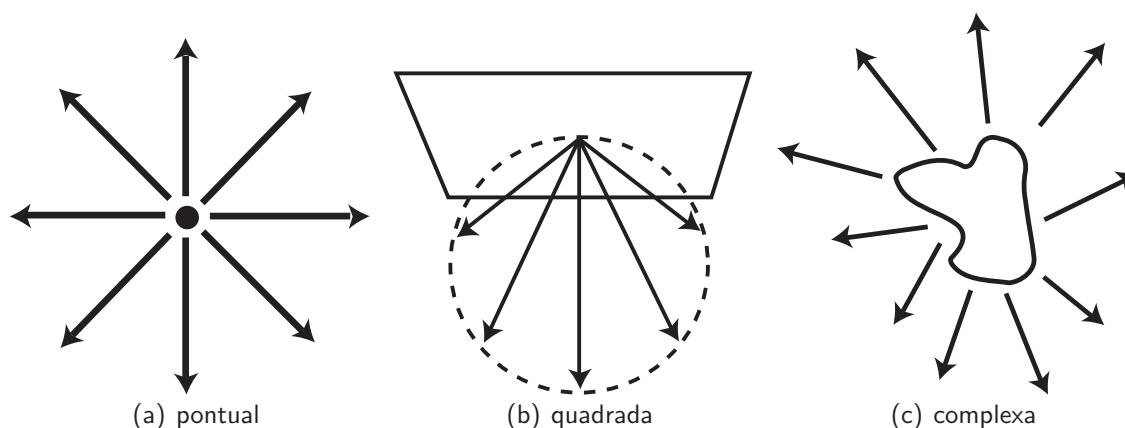


Figura 2.24: Emissão de fótons a partir de diferentes tipos de fontes de luz. Adaptado de [32].

Os fótons lançados são, então, traçados em direção à cena procurando por interseções com objetos nesta. Quando uma interseção é encontrada, decide-se o destino que este fóton terá (Figura 2.25). Ele pode ser refletido ou absorvido e este destino é decidido através de roleta russa (seção 2.3.2). Dada a probabilidade de reflexão difusa  $p_d$ , a probabilidade de reflexão

especular  $p_s$ , e uma variável aleatória uniforme  $\xi$ , tem-se

$$\xi \in [0, p_d] \rightarrow \text{reflexão difusa} \quad (2.40)$$

$$\xi \in [p_d, p_d + p_s] \rightarrow \text{reflexão especular} \quad (2.41)$$

$$\xi \in [p_d + p_s, 1] \rightarrow \text{absorção} \quad (2.42)$$

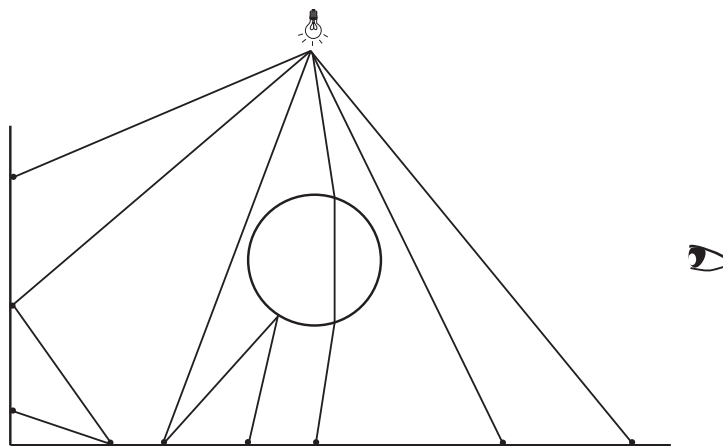


Figura 2.25: O caminho percorrido pelos fótons na cena. Adaptado de [32].

A absorção de fótons só ocorre em superfícies que possuem reflexão difusa, pois em superfícies que só possuem reflexão especular a radiância é calculada através de raios refletidos. Quando um fóton chega ao final de seu percurso, se este for absorvido por uma superfície difusa, a informação do fluxo de energia que ele possui e o ponto no qual a interseção foi detectada são salvos em um mapa de fótons que será integrado posteriormente para determinar a imagem final.

Em caminhos de fótons que geram *caustics*, descritos na seção 2.4.6 através da expressão regular LS+D, ou seja, caminhos que se iniciam na fonte de luz, passam por uma ou mais superfícies especulares e terminam em uma superfície difusa, são necessários mais fótons para chegar a um resultado com pouco ruído. Para tanto, é utilizado um mapa de fótons especializado para este tipo de caminho. Quando um caminho deste tipo é detectado, assim que o caminho encontrar a primeira superfície difusa após ter passado por uma ou mais superfícies especulares, o fóton é armazenado no mapa de fótons de *caustics*. A geração do caminho continua da forma descrita anteriormente até que o fóton seja adicionado ao mapa



de fótons global (chamado apenas de mapa de fótons). A Figura 2.26 ilustra a diferença entre caminhos de *caustics* e os caminhos normais.

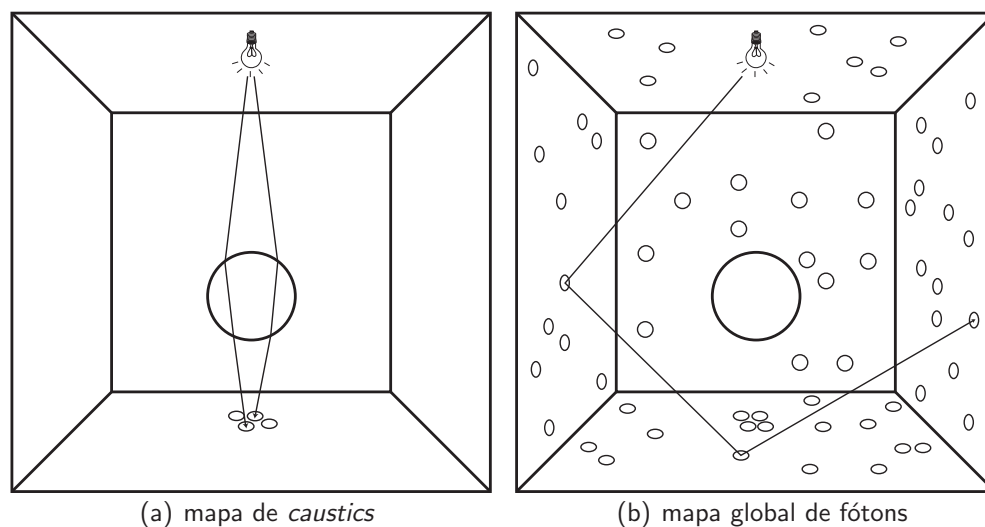


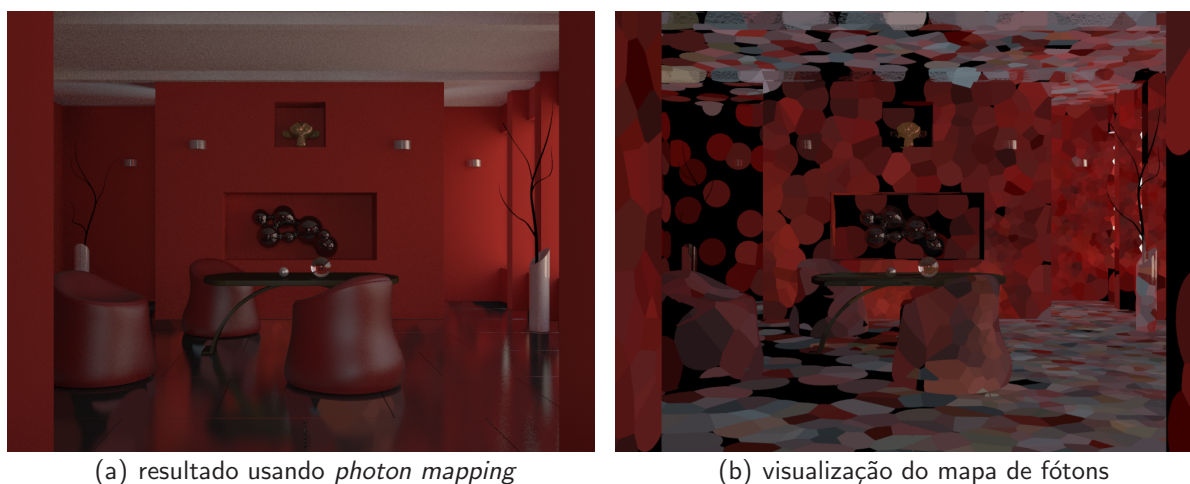
Figura 2.26: Mapa de *caustics* e o mapa global de fótons. Adaptado de [16].

Este mapa de fótons de *caustics* é importante para reduzir o ruído e suavizar a aparência do efeito de *caustics* visualizado, visto que este é composto basicamente de pontos cuja luminância é muito maior do que a dos pontos fora da região de *caustics*.

Para a geração da imagem final é utilizado um *ray-tracer* tradicional, com uma única modificação no cálculo da radiância difusa de superfícies: além de lançar raios em direção às fontes de luz para calcular a iluminação direta o mapa de fótons é utilizado para determinar a iluminação indireta no ponto. Isto é feito através da localização no mapa dos fótons que tiveram interseções detectadas em pontos próximos ao ponto cuja radiância está sendo calculada.

A visualização direta do mapa de fótons pode gerar resultados com ruídos se o mapa de fótons possuir poucos fótons próximos ao ponto cuja radiância está sendo calculada. Uma forma de conseguir resultados melhores é através do uso de *final gathering* [16, 48] para a busca por fótons. Esta técnica consiste em amostrar raios no hemisfério do ponto e buscar fótons nas interseções encontradas. A Figura 2.27 mostra o resultado do uso de *photon mapping* com *final gather* (Figura 2.27(a)) e uma visualização do conteúdo do mapa de fótons (Figura 2.27(b)).

Com esses dois passos básicos, a técnica de *photon mapping* pode ser considerada uma

(a) resultado usando *photon mapping*

(b) visualização do mapa de fótons

Figura 2.27: Cena renderizada com o uso de *photon mapping*, mostrando também o conteúdo do mapa de fótons. Modelo de Krzysztof Gawęł [19].

solução completa para a equação da renderização (apresentada na seção 2.4.3).

A técnica de *irradiance caching* (seção 2.4.5) pode ser integrada à técnica de *photon mapping* para o cálculo da iluminação indireta [32]. Neste caso, o mapa de fótons é utilizado no lugar do *cache* de irradiância, e somente se não houver fótons suficientes para estimar a irradiância esta é calculada (com o uso de *path tracing* ou alguma técnica similar).

Entre as melhorias propostas para a técnica de *photon mapping* está uma abordagem de refinamento progressivo [23]. Esta abordagem de múltiplos passos é iniciada com um primeiro passo de *ray tracing* (seção 2.4.1), com uma pequena modificação: para cada raio traçado são salvos os pontos encontrados que possuam componentes de reflexão difusa. Este ponto (chamado de *hitpoint*) é armazenado juntamente com outras informações relacionadas ao acerto (vetor normal da superfície, direção, localização do ponto no plano de imagem, entre outros).

Neste primeiro passo, o mapa de fótons ainda não foi calculado e, portanto, não pode ser utilizado para estimar a iluminação indireta. Este cálculo é feito em vários passos subsequentes. A partir do segundo passo, apenas a iluminação indireta será afetada e progressivamente refinada. Em cada passo, fótons são lançados na cena da mesma maneira feita na técnica de *photon mapping* original, com a diferença de que um número menor de fótons é utilizado a cada passo. Ao final de cada passo, a radiância dos pontos salvos no primeiro passo é recalculada levando em conta os fótons próximos e o mapa de fótons é descartado para ser

recalculado no passo seguinte.

Por se tratar de uma abordagem de refinamento progressivo, não existe um número limite de passos, sendo que, a cada passo, ruídos são removidos e o resultado torna-se mais próximo do ideal. Ao final de cada passo, uma imagem pode ser gerada para que o usuário tenha controle sobre o nível de refinamento desejado.

Algumas cenas são difíceis de serem renderizadas com *photon mapping*. Entre elas, um exemplo particularmente complexo são as cenas onde apenas uma pequena parte de todos os fótons lançados irá realmente contribuir para o resultado. Uma variação do algoritmo de *photon mapping* foi desenvolvida como forma de tentar solucionar esses casos.

A técnica foi chamada de *Metropolis photon sampling* [17]. O princípio utilizado é bastante semelhante ao encontrado na técnica de *Metropolis light transport* (descrita na seção 2.4.8). Ao invés de lançar fótons e armazená-los no mapa de fótons sem considerar a posição do visualizador na cena, esta abordagem gera um caminho completo entre a fonte de luz e o visualizador e extrai fótons a partir desse caminho, armazenando-os em pontos que tenham maior chance de serem utilizados para a geração do resultado final. Para a geração de novas amostras, mutações são feitas no caminho inicial, com a possibilidade de serem rejeitadas, e o algoritmo prossegue até que o número limite de mutações seja alcançado. Concluída a geração do mapa de fótons, o algoritmo se comporta da mesma maneira que a técnica original, utilizando um segundo passo para gerar o resultado final.

Outra proposta interessante desta mesma variação é a possibilidade de o usuário guiar o algoritmo através do fornecimento de alguns caminhos que indiquem as regiões mais importantes para a geração de amostras. Isto faz com que a convergência do algoritmo seja mais rápida, mas ainda assim sem tornar o resultado tendencioso.

## 2.4.8 Metropolis Light Transport

Técnicas como *path tracing* e *path tracing* bidirecional tendem a apresentar uma variância alta, visível na imagem gerada na forma de ruído. Este comportamento é característico de métodos baseados em integração de Monte Carlo (seção 2.3.1). Isto se deve ao fato de estes utilizarem amostras aleatórias para tentar aproximar o resultado desejado. Esta variância é

particularmente visível em cenas onde os caminhos até as fontes de luz são difíceis de serem encontrados através de amostragem aleatória.

A técnica de *Metropolis light transport* [7, 60, 63] tenta amenizar a variância aplicando conceitos da amostragem de Metropolis (seção 2.3.3.5) para solucionar a equação da renderização. Este algoritmo utiliza um caminho inicial, gerado por *path tracing* ou *path tracing* bidirecional e aplica mutações neste caminho a fim de gerar uma sequência de novos caminhos que exploram melhor o espaço de amostragem.

Estas mutações podem ser aceitas ou rejeitadas, de acordo com a probabilidade de aceitação. Com isso, após a aplicação de um certo número de mutações o método começa a convergir para a solução. A velocidade com que essa convergência ocorre é diretamente dependente das estratégias de mutação utilizadas. Essas mutações devem respeitar a restrição de que a distribuição aproximada é estacionária, ou seja, a probabilidade de a partir de um caminho  $X$  um caminho  $Y$  ser gerado deve ser proporcional à probabilidade de o caminho  $X$  ser gerado a partir do caminho  $Y$ , fazendo com que exista um balanceamento no fluxo.

Veach e Guibas [63] descrevem três estratégias de mutação para o algoritmo:

- mutações bidirecionais: têm por objetivo substituir parte do caminho original. Esta mutação dá preferência para mudanças pequenas no caminho (que são computacionalmente mais baratas de serem feitas), removendo alguns poucos vértices e substituindo-os por outros. Com isso, o espaço próximo ao caminho original é localmente explorado. Entretanto, também há a possibilidade da remoção de mais vértices ou da inserção de novos, o que faz com que outras regiões também sejam exploradas;
- perturbações: em algumas cenas, as mutações bidirecionais têm grandes chances de serem rejeitadas. Um exemplo são cenas envolvendo *caustics*, nas quais um pequeno conjunto de caminhos tem contribuição muito acima da média de todos os outros caminhos. Nestes casos, alguns vértices do caminho são ligeiramente movidos a fim de melhor explorar regiões cuja contribuição para o resultado seja maior. Estas modificações são chamadas de perturbações;
- mutações no subcaminho até o visualizador: estas mutações têm por objetivo estratificar

as amostras que chegam até o visualizador e também reduzir o custo de amostragem através da reutilização de caminhos. Uma parte do caminho utilizado na amostra anterior é substituído por outro de mesmo tamanho, respeitando as mesmas decisões de geração do caminho anterior (por exemplo, se em um determinado vértice a escolha entre refletir ou refratar o raio foi feita, esta mesma escolha deve ser feita para o vértice correspondente do novo caminho). Esta parte do caminho deve obrigatoriamente incluir o vértice do visualizador.

Quando o algoritmo encontra algum caminho que contribui mais para o resultado, mutações neste caminho têm maiores chances de serem aceitas. Isto faz com que mais caminhos sejam usados em regiões de maior contribuição. Efeitos de *caustics* também são facilmente capturados e renderizados com uma variância pequena, pois uma das perturbações aplicadas ao caminho envolve explorar regiões próximas à parte do caminho que gerou *caustics* (caminhos do tipo  $DS+D$ , de acordo com a notação de expressões regulares mostrada na seção 2.4.6).

Outro benefício desta técnica é o fato de o cálculo da contribuição de um novo caminho ser computacionalmente barata de ser calculada, visto que apenas uma parte do caminho foi alterada e, portanto, apenas os vértices novos do caminho precisam ser avaliados.

## CAPÍTULO 3

### METODOLOGIA

Este capítulo descreve o processo utilizado para a geração de imagens realistas, bem como a estrutura de um renderizador projetado de forma modular, possibilitando extensões e melhorias com a simples substituição de alguns de seus componentes por outros equivalentes sem a necessidade de grandes alterações.

A síntese de imagens (com aparência realista ou não) é feita a partir de uma cena de entrada, na qual são descritos os objetos (que podem ser compostos de primitivas, superfícies, curvas, volumes, entre outros) que possuem materiais descrevendo o comportamento da luz em sua superfície (relevo, textura, especularidade, etc), fontes de luz, e ao menos uma câmera.

Estas informações são então processadas por um renderizador, que se encarrega de simular, por meio de métodos como os descritos na seção 2.4, a interação das fontes de luz com os outros elementos da cena a fim de gerar a imagem final, que é definida por uma das câmeras da cena.

Para a obtenção de imagens realistas, alguns cuidados com os vários elementos do processo precisam ser muito bem observados:

- a geometria dos objetos da cena deve ser bem detalhada para que a aparência destes seja bastante semelhante aos objetos encontrados no mundo real;
- os materiais das superfícies têm que descrever de forma precisa a maneira como a luz interage com estas superfícies;
- a iluminação da cena precisa ser cuidadosamente projetada de maneira a simular a iluminação encontrada em ambientes reais;
- as técnicas a serem utilizadas para a síntese destas imagens devem tentar reproduzir, com a maior fidelidade possível, o comportamento da luz e a interação desta com os elementos;

- os parâmetros das técnicas escolhidas devem ser ajustados cautelosamente a fim de melhorar a qualidade da imagem gerada.

### 3.1 Componentes de um Renderizador

A estrutura interna de um renderizador pode variar bastante de um software para outro. Pharr e Humphreys [48] descrevem a estrutura de um renderizador flexível baseado em plugins, que é organizado conforme a Figura 3.1. Esta estrutura foi adotada como referência para a implementação das técnicas avaliadas.

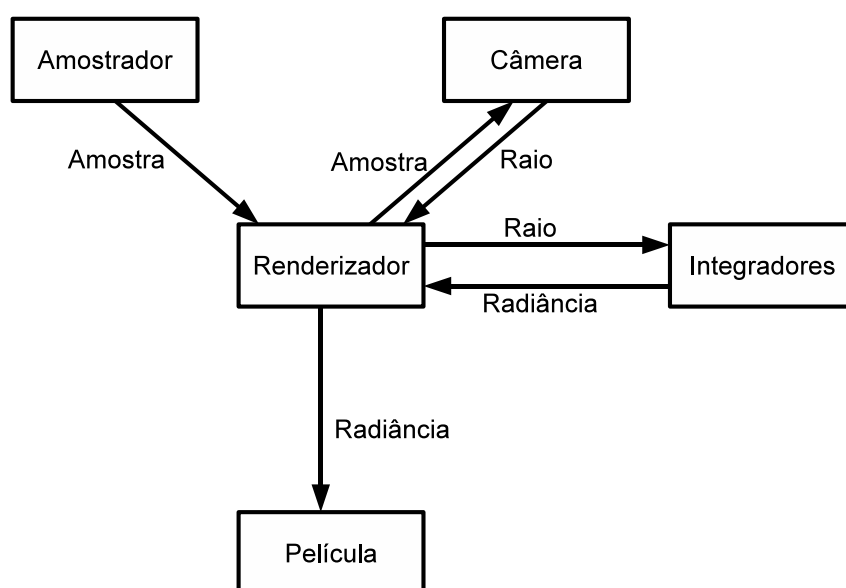


Figura 3.1: A estrutura básica de um renderizador. Adaptado de [48].

As seções que seguem descrevem cada um dos componentes apresentando suas características e responsabilidades.

#### 3.1.1 Câmera

A câmera é o componente do renderizador responsável por definir a maneira como a cena será visualizada. Seu principal papel é criar uma projeção bidimensional de uma cena representada de forma tridimensional. Seu funcionamento é bastante semelhante ao de uma câmera fotográfica, na qual a imagem final gerada vai ser definida pelos parâmetros da lente utilizada. O exemplo mais simples de câmera é a *pinhole* (Figura 3.2), que é composta de

uma caixa fechada, totalmente isolada de qualquer incidência de luz. Em sua face frontal, ela apresenta um pequeno orifício pelo qual a luz entra na caixa e é projetada em um filme foto-sensível que, após revelado, contém a imagem fotografada.

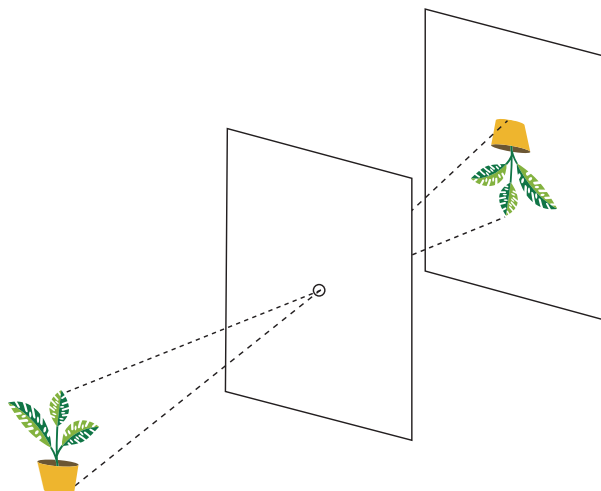


Figura 3.2: Funcionamento da câmera *pinhole*. A imagem é projetada no plano de fundo (película) de forma invertida e reduzida.

Este tipo de câmera é bastante utilizado em computação gráfica por sua simplicidade de simulação. Entretanto, para uso em imagens com aparência realista, ela não é recomendada, pois todas as imagens sintetizadas com este tipo de câmera terão suas bordas muito bem definidas, sendo facilmente reconhecidas como geradas por computador. Para produzir resultados mais convincentes são necessários modelos que simulem com mais fidelidade o comportamento de imagens no mundo real.

O modo como a câmera realiza a projeção da cena no plano de visualização (película) define a aparência da imagem gerada. As duas formas de projeção mais comuns são a projeção ortográfica e a projeção perspectiva (Figura 3.3).

O uso de uma câmera pelo renderizador é obrigatório. Entretanto, o renderizador não deve se restringir ao uso de um tipo de câmera específico. Para tanto, este deve ser feito através de uma interface abstrata que deverá ser implementada para cada uma das câmeras disponíveis. Essa interface deverá prover um meio de gerar raios que serão então lançados na cena retornando a radiância encontrada de acordo com os integradores utilizados, que implementam algumas das técnicas descritas na seção 2.4.



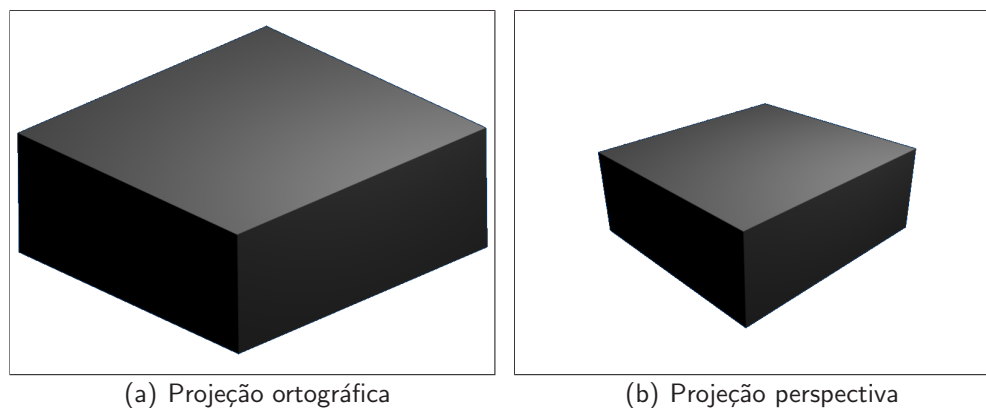


Figura 3.3: Exemplos de imagens geradas com câmeras de projeção.

### 3.1.1.1 Câmera Ortográfica

A câmera ortográfica faz uso de uma região retangular da cena identificando a porção desta que será visível, projetando esta parte da cena na face frontal da caixa que define essa região. A imagem gerada não possui a noção de profundidade percebida pelo sistema visual humano, na qual objetos mais distantes dão a impressão de serem menores do que objetos que estão mais próximos. Com essa transformação, linhas paralelas continuam paralelas na imagem gerada, e a distância relativa entre os objetos é preservada. A Figura 3.4 mostra como a região retangular é projetada na cena.

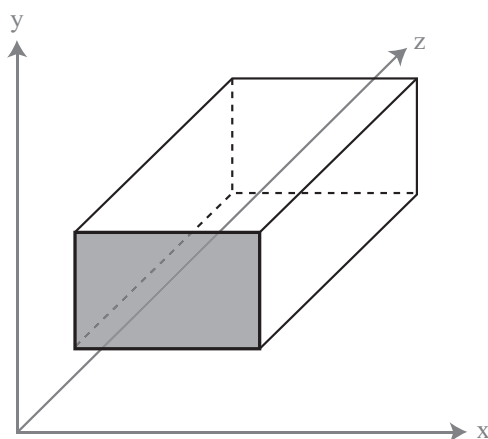


Figura 3.4: Região retangular projetada na cena pela câmera ortográfica. O retângulo frontal representa o plano da imagem (película).

### 3.1.1.2 Câmera Perspectiva

A câmera perspectiva é a que mais se aproxima da forma como o olho humano gera imagens de cenas tridimensionais. Neste tipo de câmera, objetos mais distantes aparentam ser menores que objetos próximos, e distâncias e ângulos não são preservados na imagem gerada. A Figura 3.5 exemplifica o efeito observado em imagens geradas por câmeras de perspectiva.

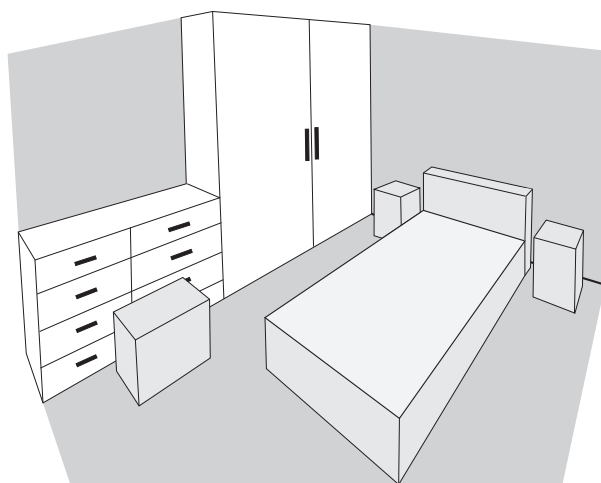


Figura 3.5: Resultado gerado por câmeras com o uso de perspectiva.

Para o cálculo da perspectiva, o plano da imagem é posicionado um pouco à frente do plano da câmera. Os raios são então lançados a partir de um ponto central no plano da câmera em direção a cada um dos pontos do plano da imagem, passando por estes e sendo lançados em direção à cena (Figura 3.6).

### 3.1.1.3 Outros Tipos de Câmera

As duas câmeras citadas anteriormente (ortogonal e perspectiva) são as mais usadas em aplicações de computação gráfica. Entretanto, existem outros modelos de câmera que também podem ser utilizados. Uma das vantagens proporcionadas por algoritmos baseados em *ray tracing* é a possibilidade da fácil implementação desses outros modelos. Uma câmera possível é a câmera olho de peixe (*fish eye*), que gera imagens como a mostrada na Figura 3.7.

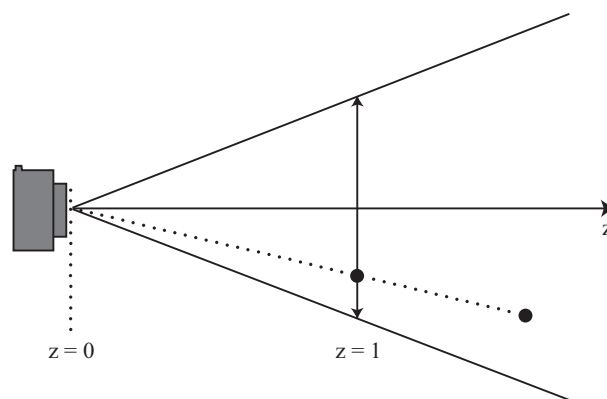


Figura 3.6: Raios são lançados a partir de um ponto central da câmera passando pelo plano da imagem em direção à cena.

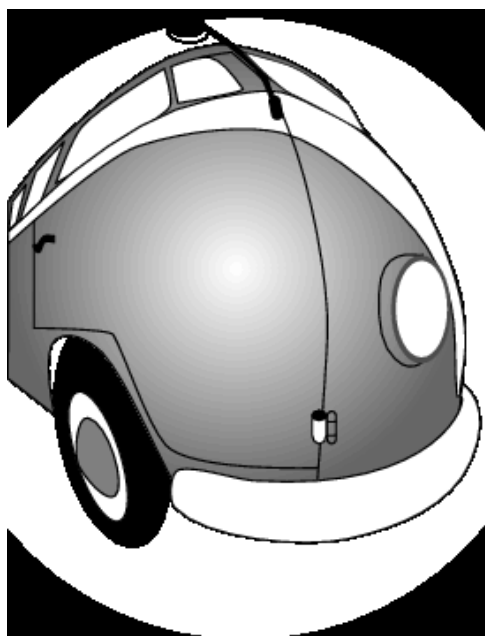


Figura 3.7: Resultado gerado por câmeras do tipo olho de peixe (*fish eye*).

#### 3.1.1.4 Profundidade de Campo

As câmeras reais possuem um sistema de lentes que focalizam a luz através de uma abertura finita. Isto faz com que apenas algumas partes da cena estejam focalizadas, outros pontos da cena estão localizados em regiões que terão uma aparência borrada (Figura 3.8). A transição de uma região focalizada para uma desfocada não é abrupta, é gradual e é medida através de círculos de confusão (Figura 3.9). Quanto mais próximo da região de foco, menor é o círculo de confusão. A esta característica de abertura da lente dá-se o nome de profundidade de campo (*depth of field*). A distância da lente ao plano dos objetos que estão em foco é

chamada de distância focal e junto com a abertura definem a região de foco da imagem. A Figura 3.10 ilustra o efeito causado por profundidade de campo.

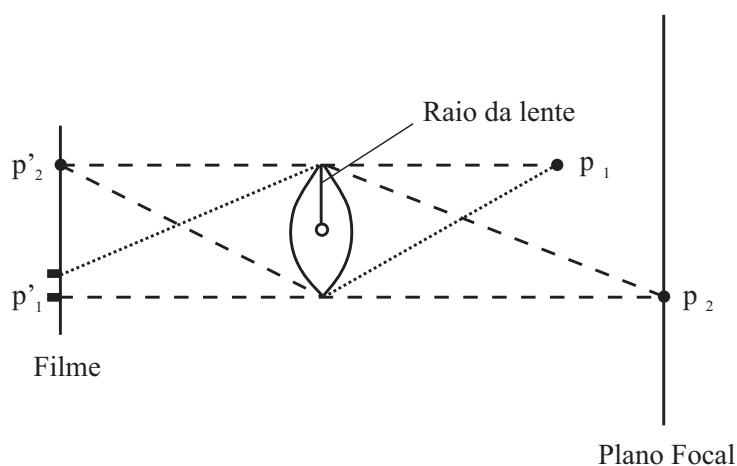


Figura 3.8: A projeção de pontos que estão fora do plano focal não são projetados de forma definida no plano da imagem (película). Adaptado de [48].

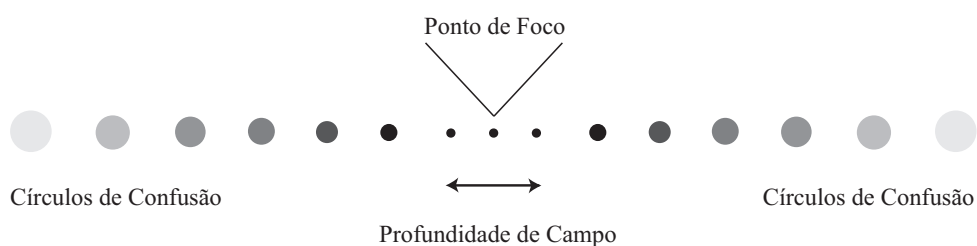
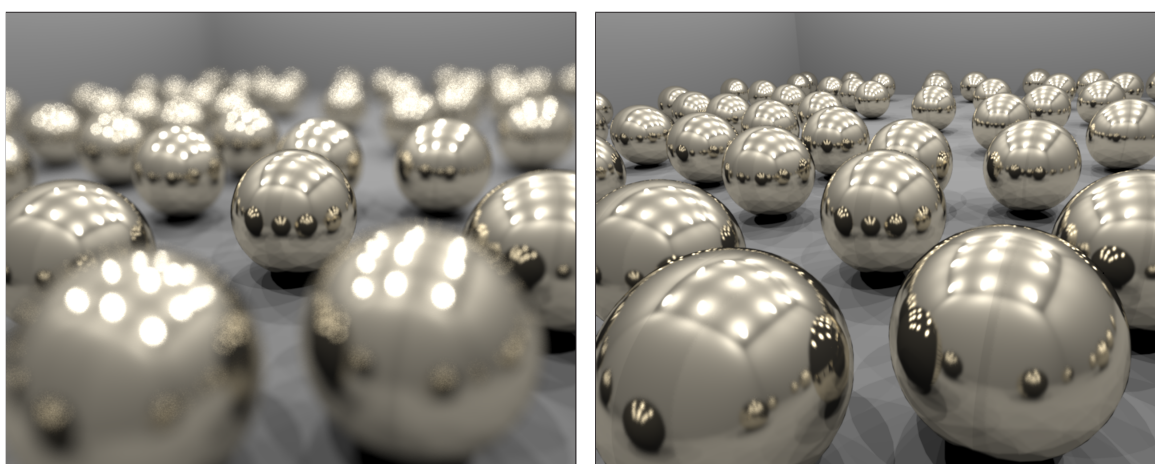


Figura 3.9: Círculos de confusão gerados em pontos fora da área de foco gerada pela profundidade de campo.



(a) com profundidade de campo pequena

(b) com profundidade de campo infinita

Figura 3.10: Exemplos de imagens simulando profundidade de campo.

A profundidade de campo é simulada através da amostragem de raios extras, de forma semelhante à abordagem do algoritmo de *stochastic ray tracing* [11] obedecendo aos parâmetros definidos para a câmera. A escolha das amostras é um fator importante para a qualidade da imagem gerada.

### 3.1.2 Amostrador

A imagem gerada a partir da síntese de uma cena tridimensional é uma função contínua da radiância incidente no plano da imagem (película). A representação desse plano é feita através de uma grade discreta de pontos coloridos. Portanto, uma discretização dessa função se faz necessária, e isso pode causar ruídos e defeitos na imagem gerada.

A teoria da amostragem descreve métodos para coletar amostras discretas em domínios de funções contínuas e utilizar estas amostras para reconstruir funções que sejam semelhantes à função original.

No contexto da síntese de imagens com realismo, estas amostras são usadas para gerar raios extras a partir do plano da imagem a fim de amenizar, se não eliminar, os problemas causados por essa discretização. No renderizador utilizado neste trabalho, este papel é desempenhado pelo amostrador.

O amostrador fornece ao renderizador amostras que são encaminhadas à câmera. A câmera, então, gera raios a partir dessas amostras e lança-os na cena para que sua radiância seja calculada. Assim como a câmera, o amostrador também é implementado através de uma interface abstrata que deve ser provida por todos os amostradores a serem usados. Isso possibilita a fácil substituição da estratégia de amostragem sem a necessidade de grandes mudanças estruturais no renderizador.

### 3.1.3 Integradores

Os integradores são responsáveis por calcular a radiância dos raios requisitados a partir dos elementos da cena. A câmera, de acordo com seu tipo, gera os raios a partir das amostras com sua projeção voltada para a cena. Estes raios são, então, repassados aos integradores para que

sejam intersectados. Ao final do processo, os integradores retornam a radiância encontrada para o raio sendo calculado.

Os integradores possuem esse nome pois seu papel é o de solucionar integrais de radiância. Os integradores utilizados são de duas categorias distintas:

- Integradores de Superfície: são responsáveis por avaliar a luz e sua interação com a superfície dos elementos da cena. Isso inclui simular reflexão difusa e especular, transmissão e absorção. A forma como esta integração é realizada fica a cargo do integrador em uso;
- Integradores de Volume: são responsáveis por determinar como a luz interage com volumes da cena. Volumes, em geral, são usados para representar efeitos tais como fumaça, atenuação atmosférica, neblina, entre outros.

No contexto do renderizador, os integradores devem prover uma interface abstrata que possibilite o cálculo do pré-processamento e, dado um certo raio, a integração dos valores no ponto de interesse. Todos os algoritmos de interesse deste trabalho podem ser implementados com o uso desta estrutura de dois passos para a integração e, portanto, este é um modelo adequado para ser usado com o propósito de comparação destas abordagens.

### 3.1.3.1 Integradores de Superfície

A seção 2.4 descreveu algumas técnicas de iluminação global cujo propósito é simular a interação da luz emitida pelas fontes de luz com as superfícies dos objetos da cena. No renderizador utilizado neste trabalho, essas estratégias são implementadas como integradores de superfície. O integrador de superfície funciona em dois passos.

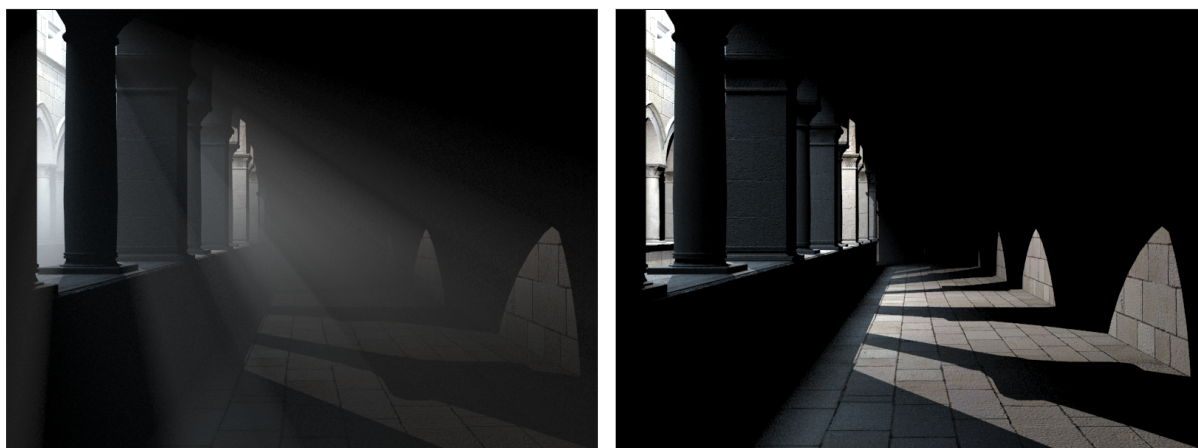
O primeiro realiza um pré-processamento dos elementos da cena (equivalente ao passo independente de visualização das técnicas multi-passo descritas na seção 2.4.6). Alguns integradores não necessitam desse pré-processamento (como por exemplo as técnicas de *ray tracing* e *path tracing*), e nestas esse passo simplesmente realiza algumas inicializações. Um exemplo de integrador que necessita de pré-processamento é o de *photon mapping*, que utiliza esse passo para lançar os fótons e com isso gerar o mapa de fótons.

O segundo passo realiza a integração dependente de visualização. O raio gerado pela câmera é passado ao integrador de superfície que, levando em conta as características da abordagem usada, irá calcular a radiância que este raio visualiza na cena. Para integradores que necessitam de pré-processamento, este passo irá fazer uso dos dados calculados no passo anterior.

### 3.1.3.2 Integradores de Volume

Tradicionalmente, as técnicas de iluminação global consideram que o meio não interfere nos raios sendo processados. Com isso, efeitos como atenuação atmosférica, neblina e fumaça são completamente ignorados. Entretanto, com uma extensão simples é possível adicionar suporte a este efeito ao renderizador de forma também flexível.

Após calcular a radiância de um determinado raio, o integrador de volume é chamado para este mesmo raio para avaliar e calcular a interferência do ambiente neste caminho. Volumes podem emitir luminosidade, refletir a luz que incide neles ou atenuar a luminosidade. A Figura 3.11 mostra a comparação do resultado de uma cena gerada sem considerar a interferência do meio na imagem gerada com outra simulando o efeito de neblina.



(a) neblina simulada por um integrador de volume

(b) a mesma cena sem a integração de volume

Figura 3.11: Exemplos de imagens comparando o efeito de neblina simulado por um integrador de volume.

Os integradores de volume são bastante semelhantes aos integradores de superfície. Sua estrutura também possui um passo de pré-processamento no qual os integradores têm a oportunidade de calcular informações que sejam independentes da visualização. No passo de

visualização, além de calcular a radiância do raio informado, o integrador de volume também precisa informar a transmissibilidade (*transmittance*) dos volumes encontrados pelo raio sendo processado. Esse valor de transmissibilidade é multiplicado pela radiância encontrada pelo integrador de volume, e depois somada à radiância retornada pela integração dos volumes.

### 3.1.4 Película

Quando uma câmera fotográfica é usada para capturar uma cena do mundo real, esta imagem é gravada em uma película química sensível à luz. Na maioria dos casos, quando um raio de luz incide em um ponto desta película (também chamada de filme), ele afeta também alguns pontos vizinhos. Em um renderizador, é importante que este comportamento possa ser simulado com a maior fidelidade possível.

Quando o raio lançado pela câmera já foi repassado aos integradores para o cálculo da radiância, ele é repassado à película para que esta possa ser sensibilizada pela radiância deste raio. A radiância é informada através de valores de ponto flutuante que, para serem exibidos em dispositivos de visualização, precisam ser convertidos para espaços de cores suportados pelo dispositivo. Essa conversão é uma discretização que pode causar erros. Portanto, é importante que esta película aplique algumas transformações à imagem gerada para atenuar possíveis erros.

Outro problema importante que precisa ser abordado é a incapacidade dos dispositivos de visualização de exibir imagens com extremos de brilho (valores muito altos ou muito baixos). Como a maioria das técnicas apresentadas produz resultados que possuem esses extremos, é necessário que se faça uma conversão nesses valores de tal forma que sua aparência, quando vista no dispositivo, seja o mais próxima possível dos valores presentes na película.

Se a implementação da película levar em conta os aspectos da percepção visual humana, é possível melhorar a qualidade dos resultados produzidos, fazendo com que os valores presentes na imagem final estejam em um intervalo mais sensível ao olho humano.



### 3.1.5 Renderizador

O componente central de um software para síntese de imagens com realismo, aqui chamado apenas de renderizador, é quem dá início ao processo, e também é responsável por configurar todos os parâmetros a serem utilizados. Quando a cena já está carregada e devidamente representada em estruturas de dados apropriadas, o renderizador inicia a síntese requisitando aos integradores que realizem o pré-processamento necessário aos passos seguintes.

O próximo passo é solicitar ao amostrador uma amostra a ser utilizada. Esta amostra é repassada à câmera, que gera um novo raio a partir dos valores informados. Este raio é então repassado aos integradores que irão calcular a radiância que é percebida por este.

Assim que o valor de radiância é encontrado, ele é repassado à película, que irá fazer o registro dos valores encontrados, e aplicar algumas transformações a fim de atenuar os ruídos e melhorar a percepção da imagem.

Após todos esses passos, a película é requisitada a salvar a imagem gerada, finalizando o processo. Com estes elementos, temos uma estrutura flexível e bastante coerente para a síntese de imagens com realismo.

## 3.2 Ambiente para Implementação e Avaliação das Técnicas

A estrutura descrita na seção 3.1 apresenta um renderizador flexível que se adequa aos requisitos deste trabalho que visa realizar uma comparação entre as diferentes abordagens para síntese de imagens realistas.

O software utilizado para a implementação das técnicas e geração dos resultados é o YafaRay [69], que segue o modelo descrito anteriormente e, portanto, é de fácil preparação para os recursos necessários. O software é escrito na linguagem de programação C++ [58] e é de livre distribuição. Apesar da existência de outros aplicativos [30, 42] com arquitetura semelhante à do software utilizado, a opção pelo YafaRay foi feita pela familiaridade do autor com o projeto e com outros desenvolvedores deste.

Os integradores implementados no software são completos e suficientes para a realização dos testes. O código necessário para o uso de *irradiance caching* (seção 2.4.5) foi implemen-

tado e os integradores de iluminação direta e de *photon mapping* foram modificados para que pudessem fazer uso de *irradiance caching*. Também foi totalmente implementada a abordagem adaptativa de *irradiance caching*, com algumas modificações descritas na seção 3.3.2.

O YafaRay possui dois modos de entrada de dados da cena. O primeiro deles é através de um arquivo texto que descreve a cena e seus elementos usando um formato baseado em XML [70]. O segundo é o uso direto do aplicativo para a síntese de modelos diretamente a partir do software de modelagem Blender [4]. A síntese a partir do Blender é indicada para quando se está preparando uma cena para ser usada para produzir resultados. A operação realizada através de um arquivo texto em formato XML é indicada a partir do momento em que a cena já está totalmente definida, pois permite a automatização dos testes a partir de parâmetros pré-configurados.

Conforme descrito na seção 3.1.3, o uso de diferentes abordagens é feito através da troca do integrador de superfície utilizado. Isto é válido para os integradores implementados no YafaRay e será utilizado para a geração de resultados de uma mesma cena sintetizada com cada uma das técnicas disponíveis a fim de possibilitar uma análise comparativa consistente entre as imagens geradas.

### 3.3 Técnicas Avaliadas

As técnicas avaliadas neste trabalho são todas implementadas como integradores de superfície (seção 3.1.3.1). A implementação de um integrador de superfície consiste em escrever uma função que calcula a radiância de um determinado raio que foi lançado a partir da câmera em direção à película, passando por esta e encontrando objetos na cena.

Opcionalmente, um integrador de superfície pode prover também uma função de pré-processamento, para processar um primeiro passo independente do visualizador (como por exemplo, a técnica de *photon mapping*, descrita na seção 2.4.7, poderia utilizar esse pré-processamento para lançar fótons na cena).

Os seguintes integradores de superfície foram utilizados:

- iluminação direta: este integrador implementa o método de *ray tracing* tradicional

(seção 2.4.1), com a possibilidade do uso de um mapa de fótons para *caustics*, *irradiance caching* tradicional e adaptativo;

- *photon mapping*: integrador que implementa a técnica de *photon mapping* (seção 2.4.7), com a possibilidade do uso de *final gathering* ou *irradiance caching* para geração do resultado final;
- *path tracing*: implementação padrão do algoritmo de *path tracing* (seção 2.4.4).

Alguns desses integradores foram utilizados para a geração de mais de um resultado. Por exemplo, o integrador de iluminação direta foi utilizado para a geração de resultados com o uso de *irradiance caching* tradicional e adaptativo. Os testes realizados, seus parâmetros e principais aspectos a serem observados são descritos nas próximas seções.

A eliminação do serrilhado das imagens (*aliasing*) foi feita através do uso de mais raios por pixel. Estes raios são definidos pela câmera e pelo amostrador e, portanto, a implementação usada é idêntica para todos os integradores.

### 3.3.1 Irradiance Caching

A implementação de *irradiance caching* utilizada faz uso da técnica de *path tracing* quando a irradiância de um determinado ponto não pôde ser interpolada com as amostras existentes no *cache*. Ela também faz uso do cálculo descrito por Tabellion e Lamorlette [59] para o peso e o erro da amostra em relação ao ponto sendo interpolado. Os gradientes de irradiância também são utilizados na interpolação dos pontos a fim de diminuir as discontinuidades encontradas em alguns resultados.

Os seguintes parâmetros são configuráveis para a geração dos resultados:

- número de caminhos: este parâmetro define o número de caminhos de *path tracing* que serão utilizados para geração de novas amostras de irradiância;
- profundidade do caminho: define a profundidade máxima das inter-reflexões do caminho gerado;

- profundidade do raio: número máximo de recursões de *ray tracing* a serem utilizadas até encontrar uma superfície com reflexão difusa.

Como a técnica não precisa calcular a irradiância para todos os pontos da imagem de resultado, para a execução dos testes, mais caminhos podem ser utilizados quando uma nova amostra de irradiância é gerada, tornando menor o erro encontrado nos pontos interpolados.

### 3.3.2 Irradiance Caching com Refinamento Adaptativo

Este método se difere dos outros utilizados por utilizar um passo dependente do visualizador (usando *ray tracing* recursivo) que calcula a iluminação direta dos pontos necessários e adiciona estes pontos a uma lista e, ao término deste passo, executa um passo de refinamento adaptativo que vai iterar a lista de pontos determinando locais onde a irradiância precisa ser estimada e pontos onde ela pode ser interpolada com o uso das amostras do *cache*.

Um detalhe importante a ser observado é o fato de o passo de *ray tracing* recursivo poder encontrar mais de uma superfície com componente de reflexão difusa ao longo do caminho percorrido pelo raio. Com isso, o valor final da radiância de um ponto no plano da imagem (película) será calculado com o uso de um ou mais pontos encontrados anteriormente. Quando a iluminação indireta de um determinado ponto do plano da imagem for calculada, todos os pontos de reflexão difusa encontrados ao longo do caminho devem ter sua iluminação indireta estimada e sua contribuição deve ser devidamente balanceada.

Esta abordagem faz uso elevado de memória pois para cada um dos pontos cuja radiância precisa ser calculada, informações sobre este são salvas em uma lista para serem usados posteriormente na fase de refinamento. Se mais de uma amostra por pixel for utilizada, isto causará um grande consumo de memória, limitando a aplicação do algoritmo.

Para contornar essa limitação, uma pequena modificação na forma como a técnica é aplicada foi realizada. Um primeiro passo de renderização é executado, utilizando apenas uma amostra por pixel. Os pontos encontrados nesse passo são então armazenados e o refinamento adaptativo é processado. O cálculo final da radiância incidente no ponto, entretanto, não é feito. Apenas as amostras resultantes do refinamento são inseridas em uma *octree*.

Após a execução desse passo inicial, o algoritmo segue utilizando a abordagem tradicional

de *irradiance caching*. A diferença é que ao início da execução do passo tradicional, o cache de irradiância já contém várias amostras dispostas de modo a tornar a interpolação mais precisa. Se para algum dos novos pontos calculados ainda assim as amostras do cache não forem suficientes, uma nova amostra de irradiância é calculada e inserida no cache. Como apenas poucas amostras serão calculadas a partir desse passo, o resultado não é muito afetado.

Os parâmetros a serem usados por testes envolvendo esta técnica são os mesmos descritos para testes com a técnica de *irradiance caching* tradicional, descritos na seção 3.3.1.

### 3.3.3 Path Tracing

Para os testes com a técnica de *path tracing*, os seguintes parâmetros podem ser definidos para a execução dos testes:

- número de caminhos: este parâmetro define a quantidade de caminhos a serem gerados para estimar a radiância de um determinado ponto;
- profundidade do caminho: define a profundidade máxima dos caminhos gerados. Este parâmetro visa delimitar o número máximo de inter-reflexões a serem utilizadas;
- profundidade do raio: número máximo de recursões a serem usadas no cálculo da iluminação gerada por reflexão especular e refração (via *ray tracing* recursivo).

### 3.3.4 Photon Mapping com Irradiance Caching

Como discutido na seção 2.4.7, a visualização direta do mapa de fótons da técnica de *photon mapping* gera resultados ruins dependendo da densidade deste. Portanto, esta técnica será usada em conjunto com as abordagens de *final gathering* e *irradiance caching*.

Para os testes com *irradiance caching*, os seguintes parâmetros podem ser especificados:

- número de fótons: define o número de fótons a serem lançados na cena;
- profundidade dos fótons: define o número máximo de vezes que um fóton pode ser refletido (de forma especular ou difusa) antes de ser absorvido;

- número de caminhos para *irradiance cache*: quando o número de fótons utilizados para interpolar a irradiância não é o suficiente, a irradiância será calculada através do uso de *path tracing*. Esta opção define o número de caminhos a serem utilizados;
- profundidade do caminho de *irradiance cache*: se o cálculo da radiância for necessário, esta opção determina o tamanho máximo dos caminhos gerados;
- profundidade do raio: define o nível máximo de recursão para refração e reflexão especular.

### 3.3.5 Photon Mapping com Final Gathering

Os testes com a técnica de *photon mapping* com *final gathering* podem ser modificados alterando os seguintes parâmetros:

- número de fótons: exatamente como na seção 3.3.4, esta opção define a quantidade de fótons que serão lançados na cena;
- profundidade dos fótons: define o número máximo de vezes que um fóton pode ser refletido (de forma especular ou difusa) antes de ser absorvido;
- número de amostras de *final gathering*: esta opção determina o número de raios que serão traçados para realizar o processo de *final gathering*;
- profundidade de *final gather*: define o tamanho máximo dos caminhos de *final gathering*;
- profundidade do raio: define o nível máximo de recursão para refração e reflexão especular.

## 3.4 Avaliação das Técnicas

Para cada uma das cenas de teste, uma imagem de referência será gerada usando um dos métodos de iluminação global baseados em integração de Monte Carlo com um número de amostras, tal que a variância seja tão pequena a ponto de não poder ser distinguida na

imagem resultante. O método escolhido deve ser não tendencioso, fazendo com que a imagem de referência não seja alterada de forma incorreta pelo algoritmo.

Os resultados obtidos serão avaliados de forma qualitativa e quantitativa. A avaliação qualitativa será feita através da inspeção visual da imagem e comparação com a imagem de referência. Além da exatidão do resultado, a presença e a perceptibilidade de ruídos também serão analisadas.

A avaliação quantitativa será feita através do cálculo da raiz do erro médio quadrático (RMSE - *root mean square error*) [47] do resultado em relação à imagem de referência. Esta métrica é definida como

$$\text{RMSE} = \sqrt{\frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} [f(x, y) - g(x, y)]^2} \quad (3.1)$$

em que  $M$  e  $N$  são as dimensões das imagens,  $f$  é a imagem de referência e  $g$  é a imagem de resultado sendo avaliada.

De acordo com este cálculo, imagens com menor RMSE têm semelhança maior com a imagem original. É importante mencionar que, apesar de esta métrica ressaltar a corretude do resultado, em diversas aplicações essa corretude pode ser negligenciada em função de resultados com uma aparência visual mais atraente.

Para identificar os pontos em que erros apareceram nos resultados, assim como para identificar o quanto estes interferem na imagem final, uma imagem será gerada através da diferença entre a imagem de teste e a imagem de referência. A diferença, calculada pixel a pixel, será visível na imagem através do uso da escala de cores mostrada na Figura 3.12. Pontos que não apresentaram erros serão exibidos na cor preta. Conforme a diferença entre a referência e o teste aumentam, a cor preta vai aos poucos assumindo a tonalidade verde, seguida pela cor amarela para indicar uma diferença média, chegando na tonalidade vermelha que é usada para indicar que os pontos são totalmente diferentes.



Figura 3.12: Escala de valores usada para representar erros na imagem de diferença.

## CAPÍTULO 4

### RESULTADOS EXPERIMENTAIS

Este capítulo apresenta os resultados experimentais que seguem a abordagem descrita no capítulo 3. A seção 4.1 descreve a plataforma de hardware e software utilizada para a implementação e execução das técnicas avaliadas. A seção 4.2 descreve o conjunto de cenas utilizadas para a avaliação das técnicas, ressaltando seus principais aspectos. Em seguida, a seção 4.3 apresenta os resultados obtidos e, por fim, a seção 4.4 discute os resultados obtidos.

#### 4.1 Plataforma de Hardware e Software

O software YafaRay [69], usado para a geração dos resultados, suporta diversas plataformas de software, entre elas, as mais comuns são Windows, Linux e Mac OSX. Seu código fonte é escrito em C++ de forma totalmente modular, tornando simples a adição e teste de novas técnicas. Ao contrário de jogos e outras aplicações que realizam processamento na placa de vídeo, todo o processamento realizado para a síntese das imagens no software utilizado é feito no processador de uso geral. O uso de processamento na placa de vídeo para aceleração das operações já foi estudado e proposto por outros pesquisadores, mas como foco deste trabalho está nas técnicas propriamente ditas, estas otimizações não serão utilizadas.

Os resultados foram gerados com o uso de várias máquinas com configurações de hardware e software distintas. Os sistemas operacionais utilizados foram o Mac OS X na versão 10.5.7 e Mandriva Linux na versão 2009.1.

#### 4.2 Conjunto de Teste

Nesta seção são descritas as cenas utilizadas como teste para as técnicas avaliadas. Cada uma delas visa simular diferentes ambientes, diferentes situações e efeitos de iluminação. As imagens utilizadas para mostrar a geometria das cenas foram renderizadas sem o uso de



texturas e materiais para facilitar a observação. As texturas e os materiais dos objetos podem ser visualizados nos testes que serão apresentados nas seções a seguir.

Todas as imagens foram renderizadas em tamanho de  $1024 \times 768$ , com o uso de 4 amostras por pixel para eliminar o efeito de *aliasing*. Para cada uma das cenas, uma imagem de referência foi renderizada utilizando a técnica de *path tracing* com o uso de um grande número de caminhos por amostra por pixel, de tal modo que o ruído da imagem seja pequeno o bastante para ser ignorado.

### 4.2.1 Caixa

Esta é uma cena sintética cujo propósito é avaliar a qualidade dos efeitos de iluminação (como por exemplo inter-reflexão difusa e *caustics*). Ela é bastante semelhante à tradicional cena Cornell Box [21], sendo composta de uma caixa com paredes laterais nas cores vermelho e azul, uma fonte de luz no teto e um objeto no chão. O objeto inserido é uma esfera cujos materiais simulam um vidro colorido. A geometria da cena pode ser vista na Figura 4.1.

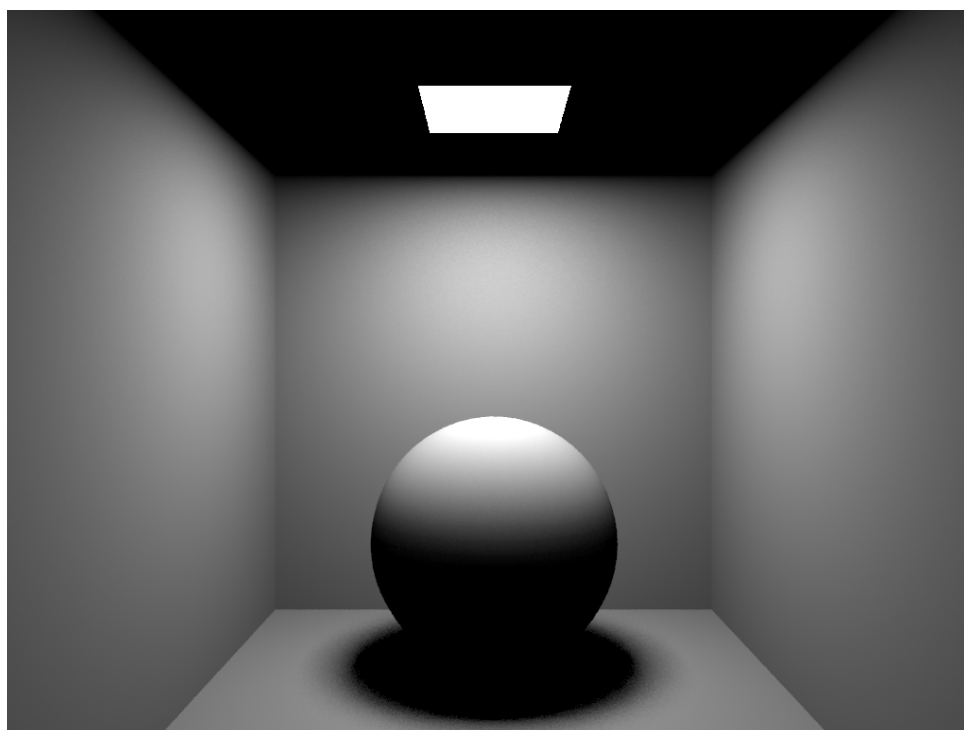


Figura 4.1: Geometria da cena Caixa.

### 4.2.2 Sala Vermelha

Este modelo mostra uma sala composta por diversos objetos e paredes vermelhas. A sala vermelha (*red room*) foi modelada por Krzysztof Gawel [19]. Assim como na cena anterior, a iluminação desta cena se dá através das janelas do ambiente. Ela se caracteriza por apresentar objetos com reflexão lustrosa (*glossy*), que são bastante sensíveis ao método de renderização utilizado. A Figura 4.2 mostra a geometria desta cena.

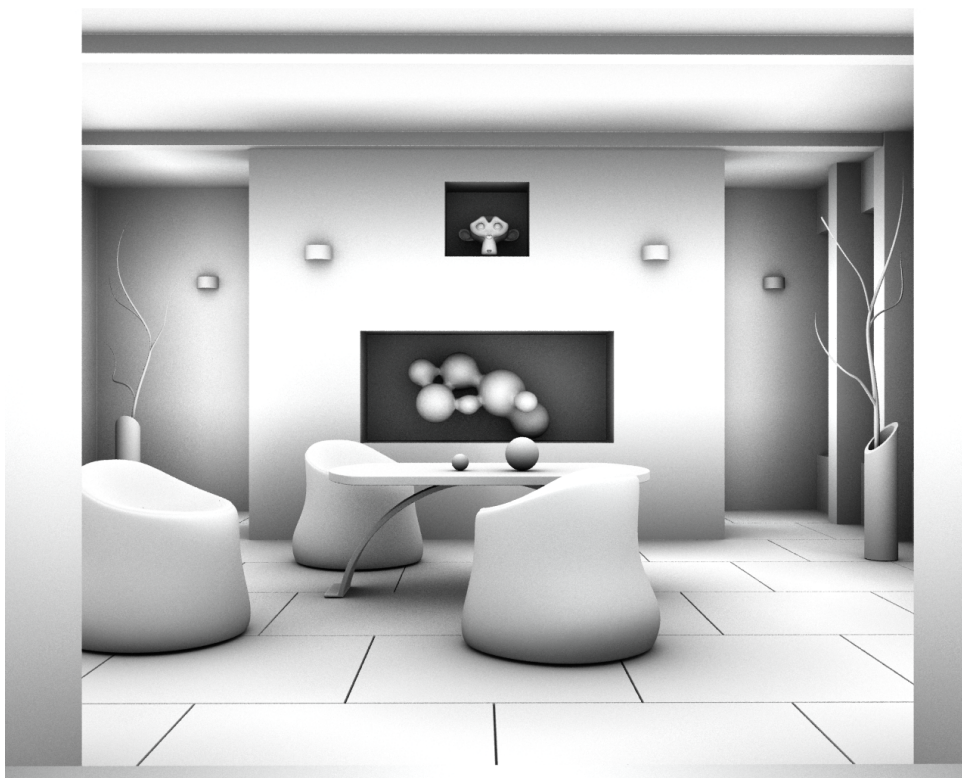


Figura 4.2: Geometria da cena Sala Vermelha (*red room*).

### 4.2.3 Catedral de São Jacó

Esta cena é um modelo da Catedral de São Jacó, em Sibenik - Croácia, criado por Marko Dabrovic [12]. O modelo não possui nenhuma fonte de luz em seu interior, utilizando apenas fontes de luz externas e a luz emitida pelo hemisfério (simulando o céu). Para melhorar um pouco a qualidade dos resultados, em vez de utilizar apenas uma luz para simular a iluminação do sol, fontes de luz foram adicionadas nas janelas do modelo. A dificuldade encontrada pelo renderizador nesta cena está no fato de a única maneira de caminhos encontrarem as fontes

de luz (sol e céu) ser através das janelas. A geometria desta cena pode ser vista na Figura 4.3.



Figura 4.3: Geometria da cena Catedral de São Jacó.

#### 4.2.4 Jardim Botânico

Esta cena representa um ambiente externo, iluminado pelo sol e pelo hemisfério ao redor da cena. A dificuldade nesta cena está nas plantas dentro da estufa, em que iluminação até elas deve chegar através de um vidro, cujo índice de refração desvia os raios gerados. Outra dificuldade está no fato de as plantas possuírem uma geometria bastante irregular e, portanto, a iluminação indireta varia de forma mais abrupta. As plantas utilizadas neste modelo fazem parte do projeto Blender Green House [28]. A Figura 4.4 mostra a geometria desta cena.

### 4.3 Resultados Obtidos

Esta seção descreve os resultados obtidos pela aplicação das técnicas avaliadas a cada uma das cenas apresentadas na seção 4.2.

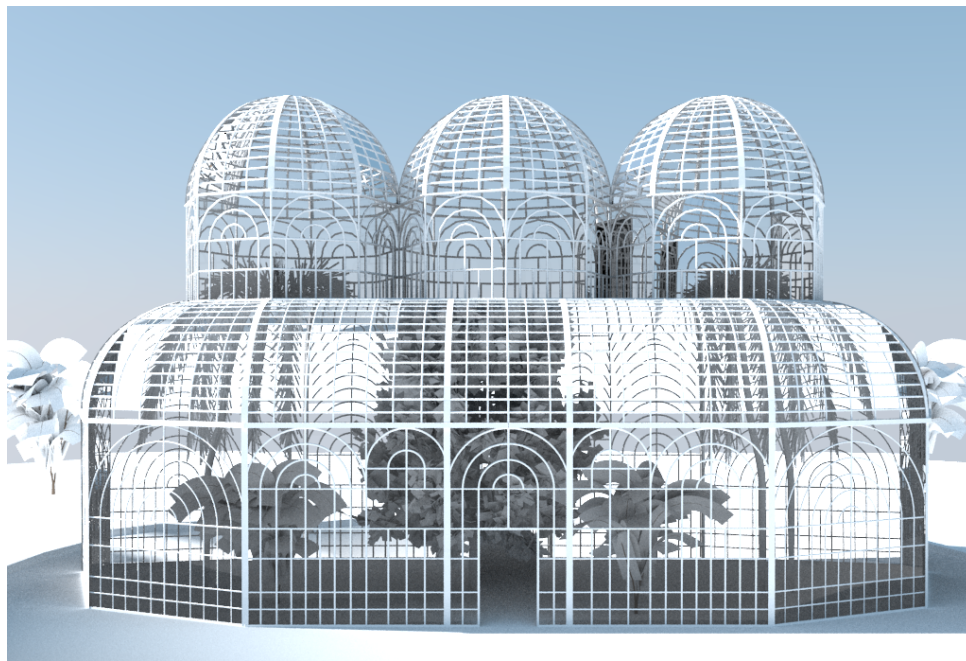


Figura 4.4: Geometria da cena Jardim Botânico.

### 4.3.1 Caixa

A imagem de referência desta cena (Figura 4.5(a)) foi renderizada com o uso de 160 caminhos por amostra e 4 amostras por pixel. Por se tratar de uma cena cuja iluminação é relativamente simples, o resultado gerado possui poucos ruídos e pode ser utilizado para compação com os resultados obtidos pelas outras técnicas. Por este mesmo motivo e por a cena também apresentar uma geometria bastante simples, este teste foi o que obteve os melhores resultados.

Os parâmetros utilizados nos testes envolvendo a técnica de *irradiance caching* são mostrados na Tabela 4.1. Como esperado, a abordagem de refinamento adaptativo gerou 510 amostras de irradiância a mais para a conclusão do processo, número relativamente pequeno em relação à quantidade total de amostras.

O resultado gerado pela abordagem tradicional apresentou descontinuidades na iluminação indireta. Estas descontinuidades foram amenizadas na imagem gerada com o uso da abordagem de refinamento adaptativo. Apesar de calcular a iluminação indireta da cena, a esfera no centro da cena ainda ficou com uma aparência escura que a técnica de *irradiance caching* não



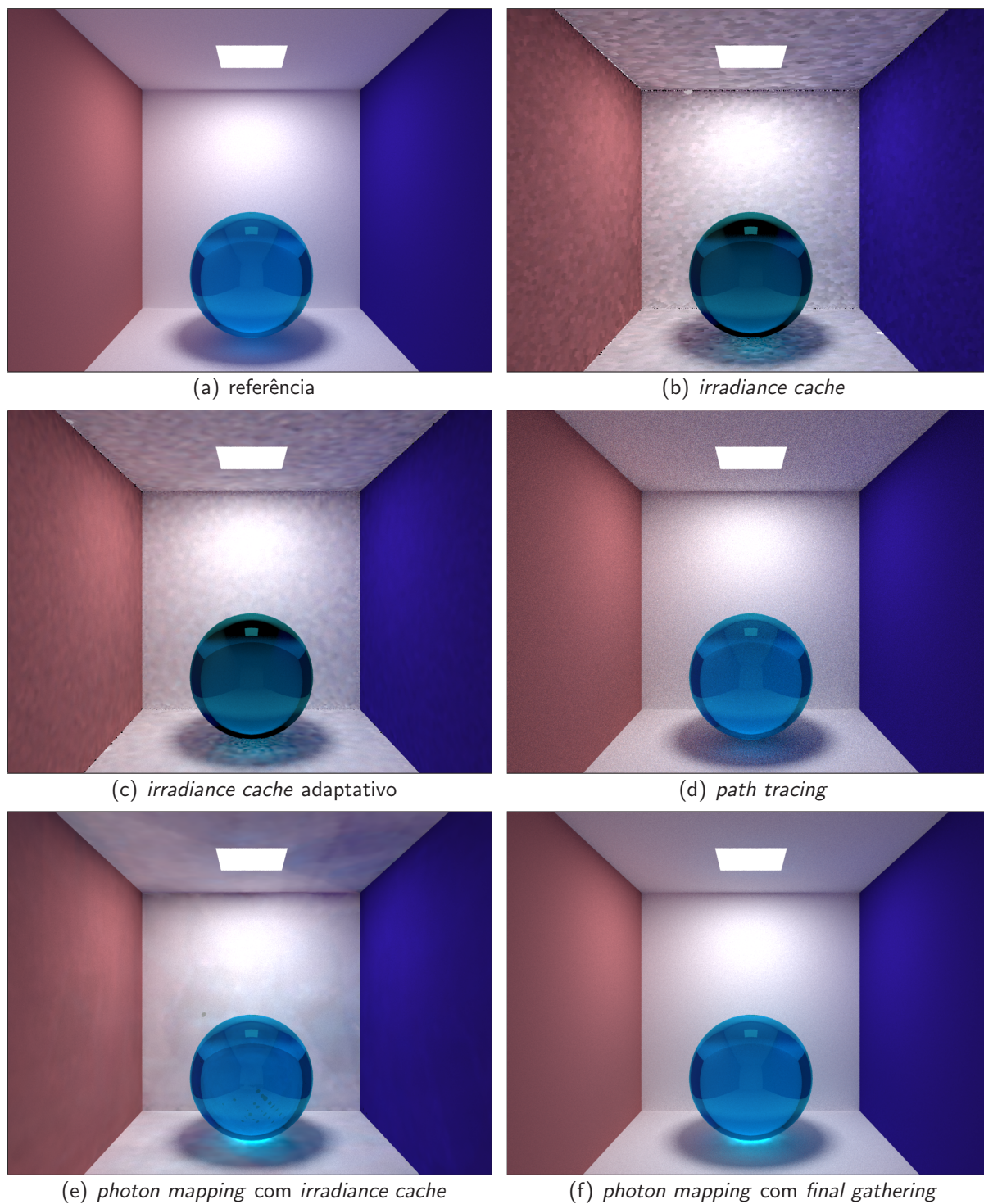


Figura 4.5: Resultados para a cena de teste Caixa.

conseguiu eliminar.

A Tabela 4.2 apresenta os parâmetros usados para a técnica de *path tracing*. O teste com esta técnica gerou uma imagem bastante semelhante à imagem de referência. A imagem resultante apresentou um aspecto levemente granuloso e com ruídos. Entretanto, devido à

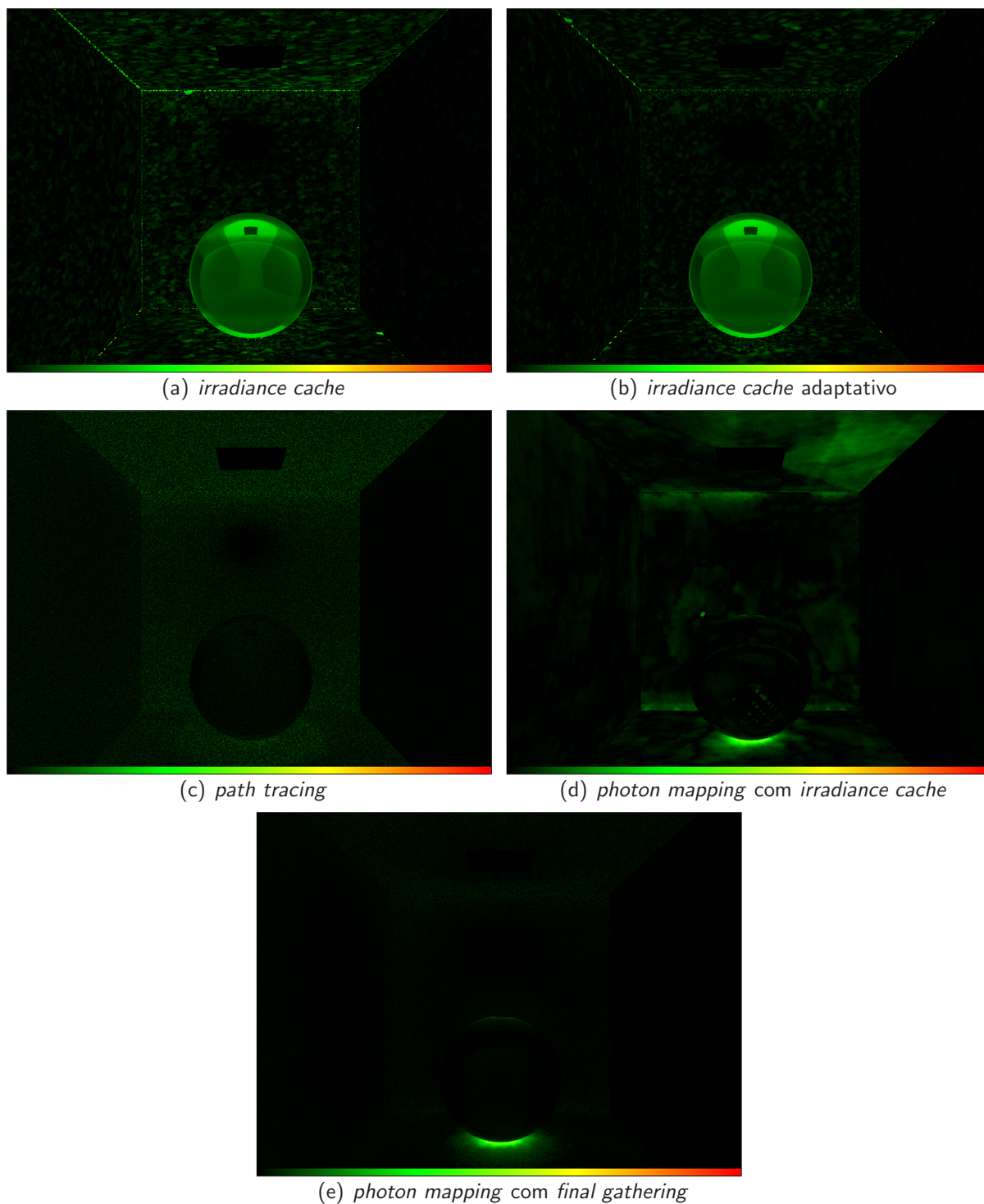


Figura 4.6: Locais onde ocorreram erros nos testes da cena Caixa.

iluminação mais simples e facilmente alcançável pelos caminhos, este ruído aparece em menor escala, chegando ao ponto de poder ser ignorado dependendo do caso de uso da imagem.

Os parâmetros usados nas imagens geradas com *photon mapping* são apresentados na Tabela 4.3. A imagem gerada com *photon mapping* e *irradiance cache* apresenta uma aparência

Tabela 4.1: Parâmetros usados para *irradiance caching* na cena Caixa

| Parâmetro                                           | Valor |
|-----------------------------------------------------|-------|
| número de caminhos                                  | 108   |
| profundidade dos caminhos                           | 5     |
| profundidade dos raios                              | 5     |
| número de amostras geradas (abordagem tradicional)  | 15397 |
| número de amostras geradas (refinamento adaptativo) | 15907 |

Tabela 4.2: Parâmetros usados para *path tracing* na cena Caixa

| Parâmetro                 | Valor |
|---------------------------|-------|
| número de caminhos        | 8     |
| profundidade dos caminhos | 5     |
| profundidade dos raios    | 5     |

mais homogênea nas regiões onde uma quantidade maior de fótons ficou concentrada. A imagem gerada por *photon mapping* com *final gathering* foi a que apresentou o melhor resultado para esta cena, sendo a que mais se aproximou da imagem de referência e a que menos apresentou erros. É interessante perceber que a técnica de *photon mapping*, tanto com o uso de *irradiance cache* quanto com o uso de *final gathering*, conseguiu simular com maior precisão o efeito de *caustics* causado pela esfera central da cena.

A cena foi a que apresentou os menores valores de RMSE entre todas as técnicas e os valores registrados são compatíveis com os obtidos através da inspeção visual das imagens. Os valores da métrica RMSE para esta técnica são mostrados na Tabela 4.4. Parte do erro apresentado nas imagens geradas por *photon mapping* se deve ao fato da diferença de iluminação na região de *caustics*, sendo que o restante da cena apresentou iluminação semelhante à da imagem de referência.

A técnica de *irradiance caching* foi a que apresentou os piores resultados, devido às des-

Tabela 4.3: Parâmetros usados para *photon mapping* na cena Caixa

| Parâmetro                                              | Valor  |
|--------------------------------------------------------|--------|
| número de fótons                                       | 500000 |
| profundidade dos fótons                                | 5      |
| número de caminhos para <i>irradiance caching</i>      | 120    |
| profundidade dos caminhos de <i>irradiance caching</i> | 5      |
| número de caminhos de <i>final gathering</i>           | 16     |
| profundidade dos caminhos de <i>final gathering</i>    | 5      |
| profundidade dos raios                                 | 5      |

Tabela 4.4: Resultados RMSE para a cena Caixa

| Técnica                                           | RMSE    |
|---------------------------------------------------|---------|
| <i>irradiance caching</i>                         | 10,5463 |
| <i>irradiance caching</i> adaptativo              | 9,26403 |
| <i>path tracing</i>                               | 7,69168 |
| <i>photon mapping</i> com <i>irradiance cache</i> | 7,88408 |
| <i>photon mapping</i> com <i>final gathering</i>  | 4,89016 |

Tabela 4.5: Parâmetros usados para *irradiance caching* na cena Sala Vermelha

| Parâmetro                                           | Valor |
|-----------------------------------------------------|-------|
| número de caminhos                                  | 108   |
| profundidade dos caminhos                           | 5     |
| profundidade dos raios                              | 5     |
| número de amostras geradas (abordagem tradicional)  | 34029 |
| número de amostras geradas (refinamento adaptativo) | 49912 |

continuidades percebidas na imagem.

### 4.3.2 Sala Vermelha

A imagem de referência desta cena precisou ser renderizada com o uso de 80 caminhos por amostras, e 32 amostras por pixel. Para garantir que ruídos fossem removidos da imagem, 5 passos extras de amostragem adaptativa foram utilizados em regiões de maior variância. A dificuldade desta cena está no fato de que a maior parte de sua iluminação vem de uma única região da cena, fazendo com que a iluminação predominante seja indireta.

O teste com a técnica de *irradiance caching*, cujos parâmetros são mostrados na Tabela 4.5, conseguiu aproximar bem a iluminação da cena. Entretanto é visível que a técnica não foi suficiente para conseguir um resultado aceitável nas superfícies. A presença de superfícies com reflexão lustrosa (*glossy*) e ausência de iluminação direta tornaram ainda mais visível que a aproximação da iluminação foi feita de forma tendenciosa (*biased*).

A abordagem adaptativa do algoritmo de *irradiance caching* reduziu bastante as discontinuidades visíveis no teste com a abordagem tradicional. Este fato é particularmente visível próximo à região da cena em que se encontra a iluminação (lado direito da imagem), que ficou com uma aparência bastante semelhante à da imagem de referência. Entretanto, este método apresentou os mesmos problemas que o teste anterior em relação às superfícies com reflexão



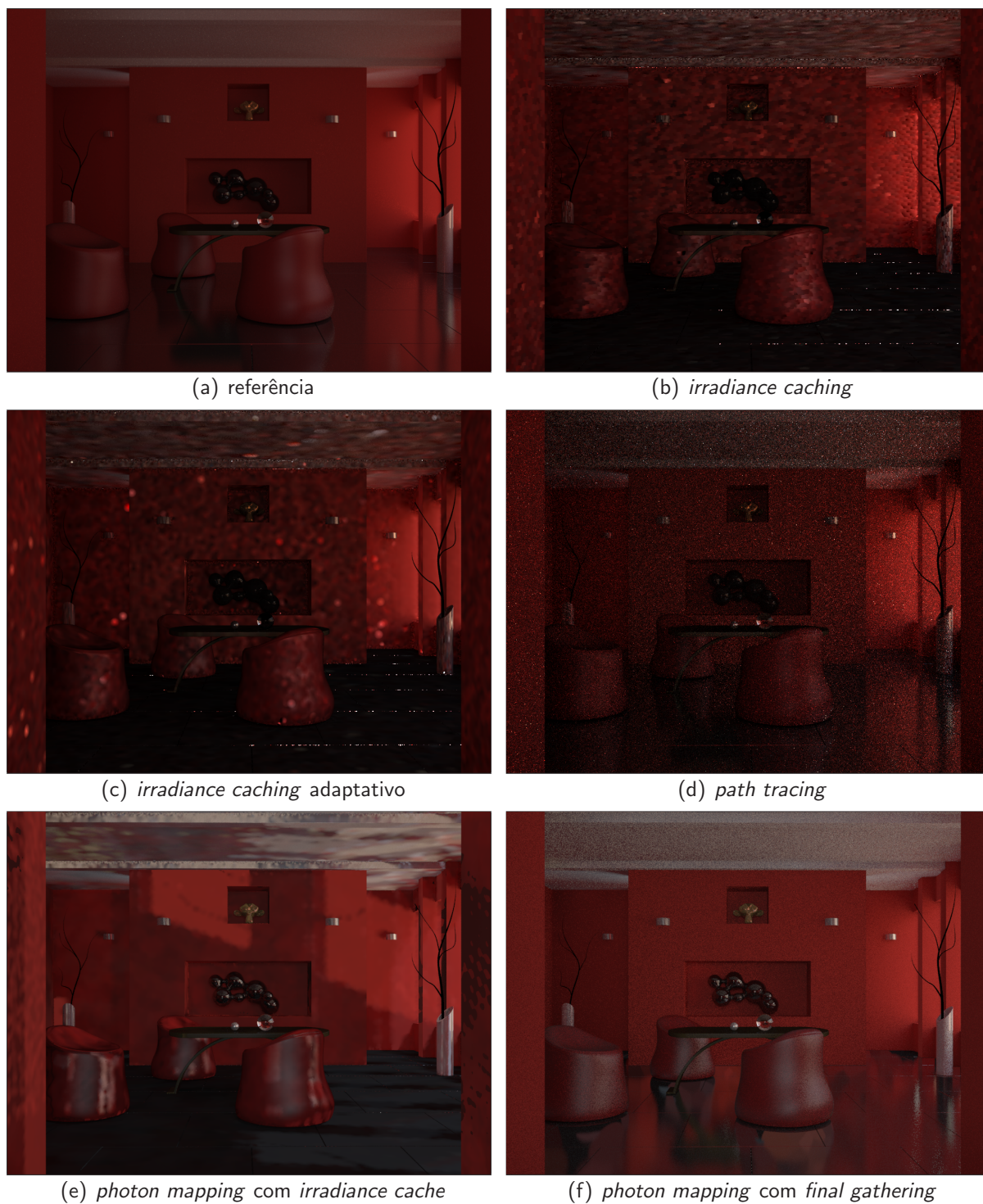


Figura 4.7: Resultados para a cena de teste Sala Vermelha.

lustrosa e com a pouca iluminação direta.

A técnica de *path tracing*, com parâmetros mostrados na Tabela 4.6, gerou resultados com uma aparência bastante ruidosa devido ao baixo número de caminhos utilizado. Apesar dessa aparência, é bastante visível que esta técnica conseguiu lidar melhor com as superfícies de

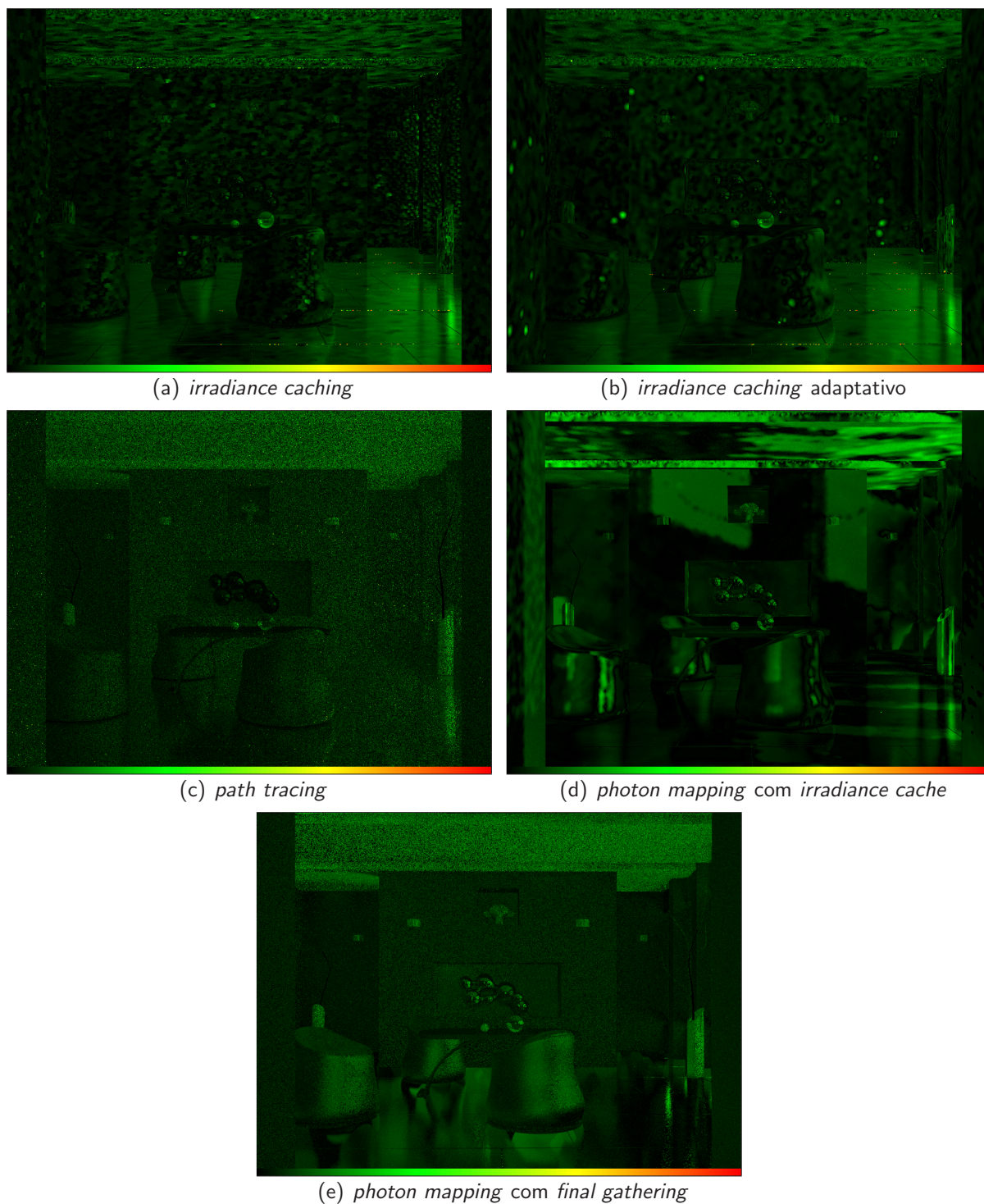


Figura 4.8: Locais onde ocorreram erros nos testes da cena Sala Vermelha.

reflexão lustrosa. A dificuldade de iluminação da cena continuou a ser um problema encontrado nos testes, sendo o principal fator causador dos ruídos vistos neste resultado.

No primeiro teste utilizando *photon mapping*, o uso de *irradiance cache* para visualização do mapa de fótons gerou um resultado com aparência ruim, principalmente na parede vista ao

Tabela 4.6: Parâmetros usados para *path tracing* na cena Sala Vermelha

| Parâmetro                 | Valor |
|---------------------------|-------|
| número de caminhos        | 8     |
| profundidade dos caminhos | 5     |
| profundidade dos raios    | 5     |

Tabela 4.7: Parâmetros usados para *photon mapping* na cena Sala Vermelha

| Parâmetro                                              | Valor  |
|--------------------------------------------------------|--------|
| número de fótons                                       | 500000 |
| profundidade dos fótons                                | 5      |
| número de caminhos para <i>irradiance caching</i>      | 32     |
| profundidade dos caminhos de <i>irradiance caching</i> | 5      |
| número de caminhos de <i>final gathering</i>           | 10     |
| profundidade dos caminhos de <i>final gathering</i>    | 5      |
| profundidade dos raios                                 | 5      |

fundo. Isto se deve ao fato de o cache de irradiância ter sido gerado como visualização direta do mapa de fótons, sem o uso de *final gathering*. Apesar dos problemas mencionados, o nível de iluminação da cena gerado pela técnica é bem melhor do que os resultados anteriores. Isto se deve ao fato de os fótons serem lançados a partir das fontes de luz em direção à cena.

O uso de *final gathering* para a visualização do mapa de fótons gerou o melhor resultado para esta cena. A reflexão lustrosa do piso apresentou alguns pequenos erros, mas no geral ficou com uma aparência correta. O resultado ficou um pouco mais claro do que a cena de referência, mas do ponto de vista de inspeção visual, é o teste que obteve melhor desempenho. Os parâmetros utilizados para os testes com *photon mapping* são mostrados na Tabela 4.7.

De acordo com a métrica RMSE, foram obtidos os valores mostrados na Tabela 4.8 que apontou os resultados gerados por *irradiance caching* como sendo os melhores. O resultado das cenas com *photon mapping* foram os que tiveram os piores resultados, devido principalmente ao fato de a iluminação estar um pouco mais clara.

Tabela 4.8: Resultados RMSE para a cena Sala Vermelha

| Técnica                                           | RMSE    |
|---------------------------------------------------|---------|
| <i>irradiance caching</i>                         | 13,5906 |
| <i>irradiance caching</i> adaptativo              | 14,2293 |
| <i>path tracing</i>                               | 16,3605 |
| <i>photon mapping</i> com <i>irradiance cache</i> | 17,8143 |
| <i>photon mapping</i> com <i>final gathering</i>  | 17,1709 |



Tabela 4.9: Parâmetros usados para *irradiance caching* na cena Catedral de São Jacó

| Parâmetro                                           | Valor  |
|-----------------------------------------------------|--------|
| número de caminhos                                  | 108    |
| profundidade dos caminhos                           | 5      |
| profundidade dos raios                              | 5      |
| número de amostras geradas (abordagem tradicional)  | 282575 |
| número de amostras geradas (refinamento adaptativo) | 306124 |

Tabela 4.10: Parâmetros usados para *path tracing* na cena Catedral de São Jacó

| Parâmetro                 | Valor |
|---------------------------|-------|
| número de caminhos        | 10    |
| profundidade dos caminhos | 5     |
| profundidade dos raios    | 5     |

### 4.3.3 Catedral de São Jacó

A imagem de referência desta cena foi a mais complexa de ser gerada. Foram utilizados 260 caminhos por amostra, 32 amostras por pixel e mais 4 passos de amostragem adaptativa para remover o serrilhado. Várias tentativas anteriores foram feitas com números menores de caminhos, mas os resultados apresentaram uma quantidade de ruído bastante elevada.

Os testes utilizando *irradiance caching*, cujos parâmetros são mostrados na Tabela 4.9, gerou uma imagem com poucos ruídos, apenas apresentando as discontinuidades características da técnica. A abordagem adaptativa diminuiu as discontinuidades da técnica tradicional. Entretanto, para algumas regiões da cena, a abordagem convergiu para resultados errados, gerando manchas escuras, visíveis principalmente no lado esquerdo da imagem.

A imagem gerada pela técnica de *path tracing* com os parâmetros descritos na Tabela 4.10 apresentou uma quantidade bastante elevada de ruído. Este tipo de comportamento é comum em técnicas baseadas em integração de Monte Carlo em situações onde as fontes de luz não são facilmente encontradas pelos raios. Este comportamento normalmente é amenizado aumentando-se o número de caminhos por amostra.

Devido à espessura das paredes do modelo utilizado, nos testes envolvendo a técnica de *photon mapping*, alguns fótons que haviam sido armazenados do lado de fora do modelo acabaram sendo incluídos para estimar a irradiância das paredes do lado de dentro. Isto fez com que o resultado ficasse muito mais claro do que deveria. Este vazamento pode ser visto

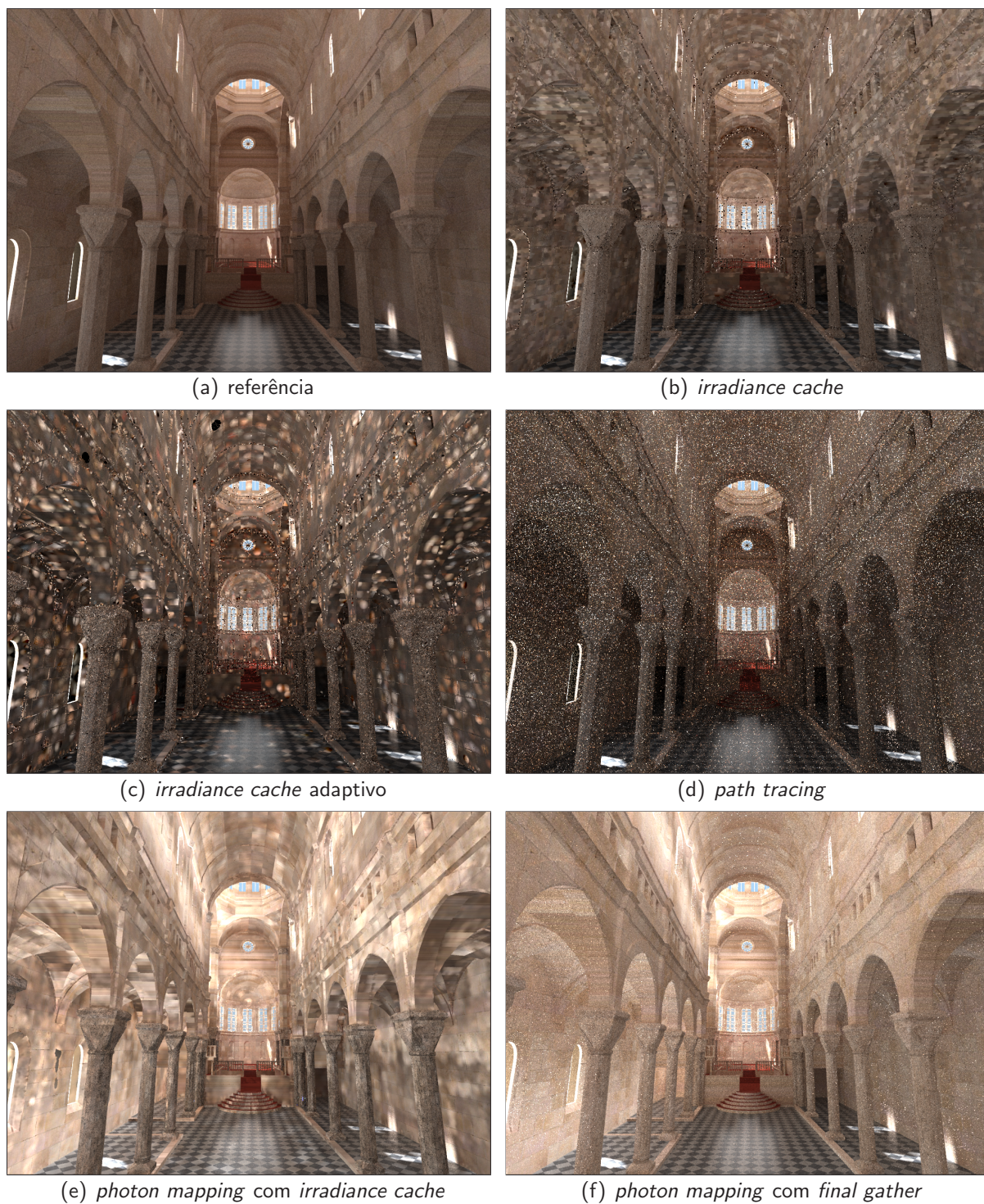


Figura 4.9: Resultados para a cena de teste Catedral de São Jacó.

claramente nas paredes do lado esquerdo (próximo ao rodapé) e na parte superior do modelo (próximo às janelas). Os parâmetros utilizados para esta técnica são mostrados na Tabela 4.11.

A imagem gerada por *photon mapping com irradiance cache* possui uma aparência com



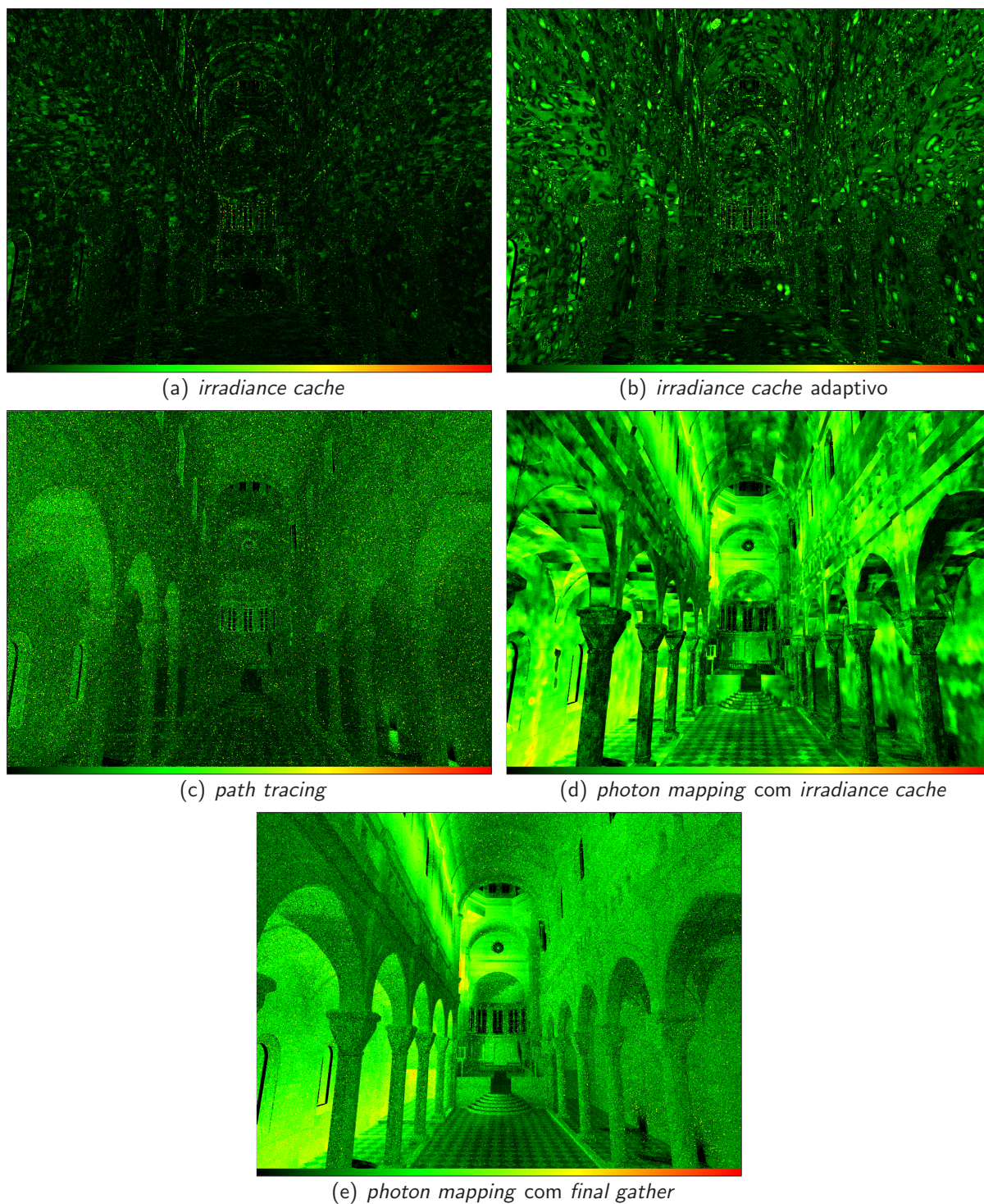


Figura 4.10: Locais onde ocorreram erros nos testes da cena Catedral de São Jacó.

poucos ruídos, apesar do vazamento que ocorreu. Entretanto, alguns pontos mais claros são observados em várias partes da construção. Isto se deve ao fato de o cache de irradiância ser usado para visualização direta do mapa de fótons, sem o uso de *final gathering*

A imagem gerada com o uso de *photon mapping com final gathering* gerou um resultado

Tabela 4.11: Parâmetros usados para *photon mapping* na cena Catedral de São Jacó

| Parâmetro                                              | Valor  |
|--------------------------------------------------------|--------|
| número de fótons                                       | 500000 |
| profundidade dos fótons                                | 5      |
| número de caminhos para <i>irradiance caching</i>      | 120    |
| profundidade dos caminhos de <i>irradiance caching</i> | 5      |
| número de caminhos de <i>final gathering</i>           | 20     |
| profundidade dos caminhos de <i>final gathering</i>    | 5      |
| profundidade dos raios                                 | 5      |

Tabela 4.12: Resultados RMSE para a cena Catedral de São Jacó

| Técnica                                           | RMSE    |
|---------------------------------------------------|---------|
| <i>irradiance caching</i>                         | 15,4329 |
| <i>irradiance caching</i> adaptativo              | 28,4363 |
| <i>path tracing</i>                               | 42,4436 |
| <i>photon mapping</i> com <i>irradiance cache</i> | 64,3173 |
| <i>photon mapping</i> com <i>final gathering</i>  | 68,5992 |

com uma quantidade de ruídos menor do que a encontrada na imagem gerada por *path tracing*, mas ainda assim presentes.

Os resultados da métrica RMSE para esta imagem são mostrados na Tabela 4.12 indicando que o menor erro foi encontrado na técnica de *irradiance caching*. Este resultado pode ser também observado por inspeção visual. A grande presença de ruídos no teste realizado com *path tracing* fez com que o valor da métrica ficasse bastante alto para este caso. A comparação dos testes envolvendo *photon mapping* apresentou os piores resultados devido ao vazamento de fótons do lado externo da parede para o lado interno.

#### 4.3.4 Jardim Botânico

A imagem de referência desta cena foi gerada com o uso de 120 caminhos por amostra e 8 amostras por pixel. A iluminação externa da cena é de simulação fácil, visto que se trata de um espaço aberto com uma fonte de luz, representando o sol, facilmente acessível. A parte interna da cena é um pouco mais complexa. Devido ao efeito de refração que ocorre quando a luz incide no vidro da estufa, a iluminação das plantas é complexa de ser simulada. Como a proposta do trabalho era a de usar técnicas não tendenciosas para a geração das imagens de referência, optou-se por manter o uso de *path tracing* para a geração da imagem de referência.



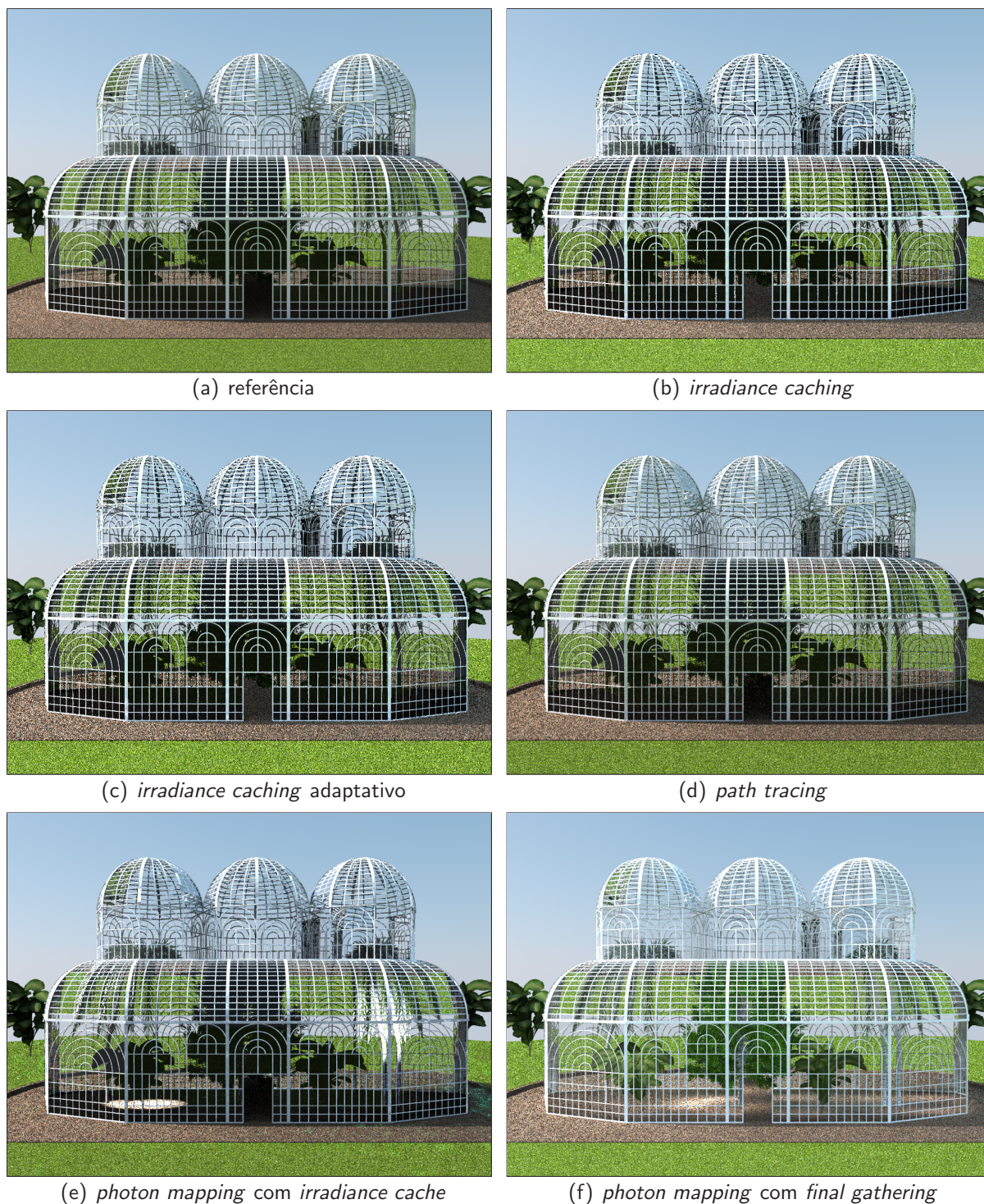


Figura 4.11: Resultados para a cena de teste Jardim Botânico.

Os testes com a técnica de *irradiance caching*, cujos parâmetros são mostrados na Tabela 4.13, geraram um número bastante alto de amostras de irradiância. Isto se deve ao fato de a geometria das plantas dentro da estufa ser bem detalhada e as superfícies intersectadas possuírem um vetor normal bem diferente umas das outras. O resultado apresentado pelos



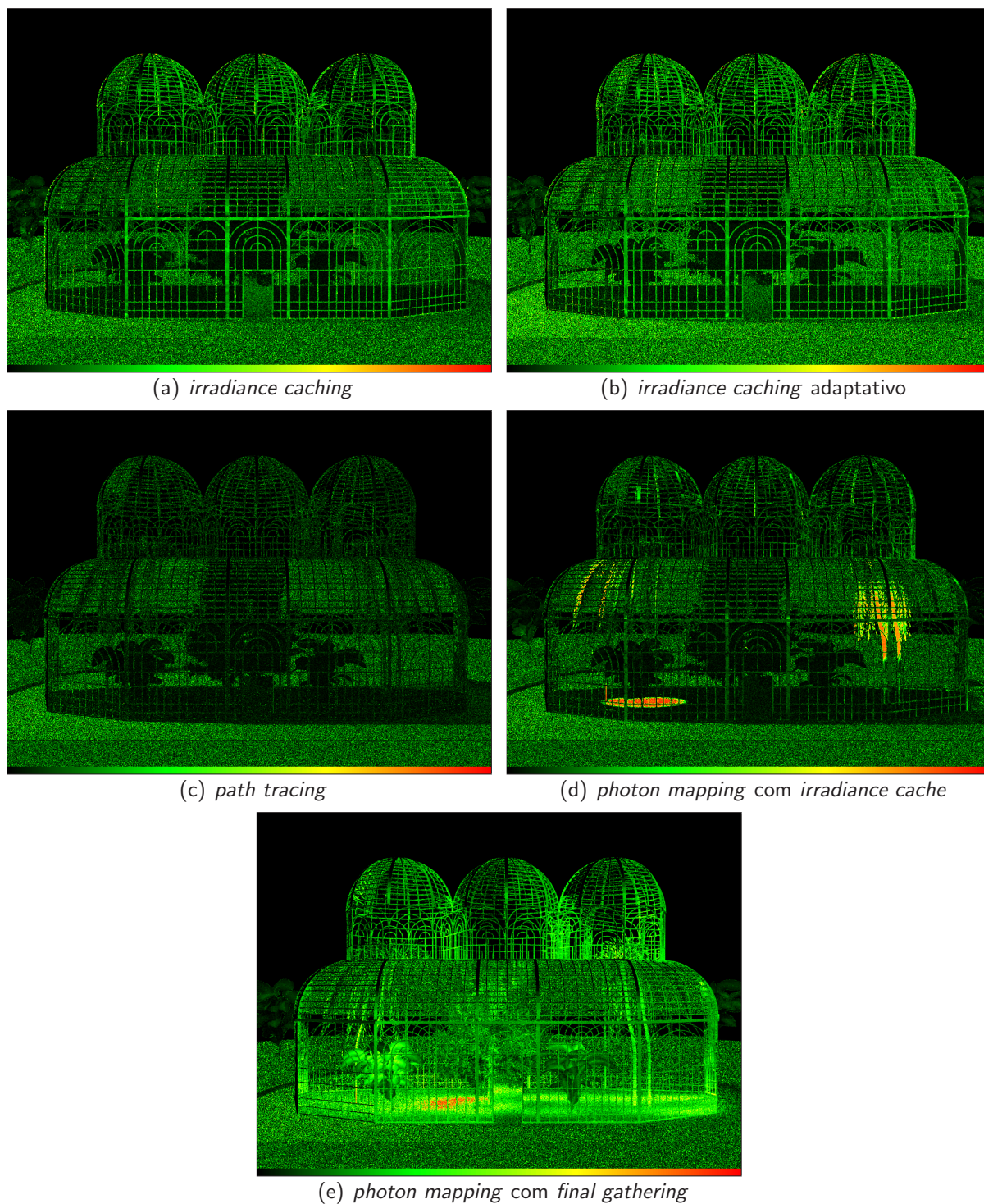


Figura 4.12: Locais onde ocorreram erros nos testes da cena Jardim Botânico.

dois testes foi bastante similar, sendo que algumas regiões das imagens ficaram um pouco mais claras.

A técnica de *path tracing* foi utilizada com poucos caminhos por amostra, e ainda assim gerou um resultado bem aceitável. O ruído apresentado por esta técnica nas imagens dos

Tabela 4.13: Parâmetros usados para *irradiance caching* na cena Jardim Botânico

| Parâmetro                                           | Valor  |
|-----------------------------------------------------|--------|
| número de caminhos                                  | 32     |
| profundidade dos caminhos                           | 5      |
| profundidade dos raios                              | 5      |
| número de amostras geradas (abordagem tradicional)  | 237293 |
| número de amostras geradas (refinamento adaptativo) | 285232 |

Tabela 4.14: Parâmetros usados para *path tracing* na cena Jardim Botânico

| Parâmetro                 | Valor |
|---------------------------|-------|
| número de caminhos        | 4     |
| profundidade dos caminhos | 4     |
| profundidade dos raios    | 5     |

testes anteriores quase não se manifestou neste teste. Este é perceptível nas plantas dentro da estufa, entretanto, como estas estão pouco iluminadas, o ruído acaba sendo ocultado. Os parâmetros usados para esta técnica são apresentados na Tabela 4.14.

Os resultados gerados com *photon mapping*, cujos parâmetros são apresentados na Tabela 4.15, conseguiram simular de forma mais precisa o efeito de *caustics*. Na imagem gerada com o auxílio de *irradiance cache*, entretanto, este efeito ficou restrito a uma pequena área interna da estufa, e a iluminação das plantas não foi muito bem representada. A imagem gerada com o uso de *final gathering* foi a que melhor conseguiu iluminar as plantas dentro da estufa e a que gerou melhores resultados de *caustics*.

Os valores de RMSE para esta cena são mostrados na Tabela 4.16, sendo que esta cena foi a que apresentou os maiores valores para a métrica. Mesmo a técnica de *path tracing*, que também foi utilizada para gerar a imagem de referência, apresentou valores altos. Entretanto, esta cena é a que menos apresenta influência visual deste erro. O erro da técnica de *photon*

Tabela 4.15: Parâmetros usados para *photon mapping* na cena Jardim Botânico

| Parâmetro                                              | Valor  |
|--------------------------------------------------------|--------|
| número de fótons                                       | 500000 |
| profundidade dos fótons                                | 5      |
| número de caminhos para <i>irradiance caching</i>      | 32     |
| profundidade dos caminhos de <i>irradiance caching</i> | 5      |
| número de caminhos de <i>final gathering</i>           | 10     |
| profundidade dos caminhos de <i>final gathering</i>    | 5      |
| profundidade dos raios                                 | 5      |

Tabela 4.16: Resultados RMSE para a cena Jardim Botânico

| Técnica                                           | RMSE    |
|---------------------------------------------------|---------|
| <i>irradiance caching</i>                         | 32,3114 |
| <i>irradiance caching</i> adaptativo              | 38,7004 |
| <i>path tracing</i>                               | 23,047  |
| <i>photon mapping</i> com <i>irradiance cache</i> | 30,9545 |
| <i>photon mapping</i> com <i>final gathering</i>  | 42,5679 |

*mapping* com *final gathering* se deve principalmente às mudanças de iluminação no interior da estufa.

#### 4.4 Discussão dos Resultados

A geração dos resultados envolveu a avaliação do comportamento das técnicas utilizadas em cada um dos ambientes descritos. A definição dos parâmetros a serem utilizados tentou aproximar ao máximo o custo computacional de cada abordagem. A técnica de *irradiance caching* e sua variante de refinamento adaptativo só tem seu custo computacional conhecido ao final da execução, visto que o número de amostras é dependente da geometria da cena e da visualização sendo feita. Baseado no número de amostras necessárias para cada teste, o número de caminhos da técnica de *path tracing* e o número de caminhos de *final gathering* para a técnica de *photon mapping* foram ajustados.

A utilização da técnica de *path tracing* para o cálculo das amostras usadas na técnica de *irradiance caching* apresentou bons resultados para a maioria dos testes. Entretanto, na cena de teste Catedral de São Jacó, a dificuldade em encontrar as fontes de luz fez com que as amostras tivessem aparência mais heterogênea. Esta diferença na iluminação das amostras foi propagada através do uso do *cache*, gerando uma aparência com grandes disparidades em regiões que deveriam ser contínuas. Isto é particularmente visível na Figura 4.10(b), na qual alguns pontos de erro maior são visíveis nas paredes e no teto.

Em ambientes com poucas fontes de luz e ambientes onde as fontes de luz não são facilmente acessíveis, como é o caso das cenas de teste Sala Vermelha e Catedral de São Jacó, a técnica de *path tracing* gera imagens com uma aparência bastante granulada, nas quais os erros não são vistos em regiões específicas da imagem, mas sim distribuídos por toda a

imagem. Isto pode ser facilmente observado nas imagens de diferença da Figura 4.8(c) e da Figura 4.10(c).

Na cena Jardim Botânico, que apresenta um ambiente aberto, apesar de as imagens de diferença da Figura 4.12 indicarem que uma quantidade significativa de erros ocorreram, estes não são visualmente tão perceptíveis quanto os erros ocorridos nos outros testes.

A técnica de *irradiance cache* com refinamento adaptativo requer o armazenamento de informações sobre todos os pontos cuja radiância incidente precisa ser calculada. Estes pontos são então posicionados em uma *kd-tree* para que possam ser facilmente localizados pela sua posição no espaço tridimensional. Estes pontos também precisam ser inseridos numa fila para serem processados individualmente. Isto faz com que o consumo de memória seja bastante elevado. Esta foi uma limitação que dificultou a renderização da cena de teste Jardim Botânico, visto que o carregamento da cena em memória por si só já ocupava bastante espaço.

## CAPÍTULO 5

### CONCLUSÕES E TRABALHOS FUTUROS

Este trabalho descreveu o problema de síntese de imagens com realismo revisando as técnicas existentes e comparando suas características e suas limitações. A implementação das técnicas e disponibilização destas como software livre permite a fácil reprodução dos experimentos realizados, bem como a execução de novos trabalhos baseados na infra-estrutura implementada.

A geração de imagens com realismo, apesar de ser um assunto bastante estudado, oferece diversos desafios, muitos deles ainda não solucionados. As técnicas existentes não são genéricas o suficiente para a obtenção de um desempenho ótimo independente da cena sendo renderizada, portanto, a escolha da melhor técnica ainda deve ser feita com base nas características da cena utilizada e nos resultados que se deseja obter. O desenvolvimento de um algoritmo capaz de se adaptar automaticamente às condições existentes na cena ainda é um objetivo a ser perseguido.

O uso de métricas quantitativas, apesar de fornecer uma base mais sólida para análise dos resultados, deve ser usado de forma ponderada, visto que às vezes o resultado esperado pode ser atendido através de aproximações às vezes tendenciosas da solução ideal, o que faz com que algumas métricas reportem resultados incorretos a respeito da qualidade de uma determinada cena. Uma proposta de trabalho futuro é o uso de métricas qualitativas baseadas no sistema visual humano para avaliação de resultados.

Técnicas baseadas em *photon mapping* são atualmente as que produzem os melhores resultados no geral. Entretanto, o uso de técnicas baseadas em *irradiance caching* se mostraram uma alternativa viável em situações onde métodos baseados em integração de Monte Carlo falham em gerar resultados aceitáveis.

A variação de refinamento adaptativo da técnica de *irradiance caching* consegue melhorar de forma considerável a qualidade da técnica tradicional. Entretanto, o custo do processo de refinamento e o consumo de memória podem ser fatores que limitam sua aplicação. Como pro-

posta de trabalho futuro está o desenvolvimento de uma abordagem que combine os aspectos do refinamento adaptativo sem um consumo excessivo de memória. Para a técnica existente, um proposta de trabalho futuro é a tentativa de paralelizar os passos de refinamento adaptativo a fim de reduzir seu impacto no tempo total de execução da técnica.

O uso de *irradiance caching* em conjunto com *photon mapping* ainda é bastante limitado. Trabalhos futuros sobre esse assunto poderiam incluir a combinação destas técnicas com *final gathering* e abordagens de refinamento progressivo, sejam elas do *cache* de irradiância ou do mapa de fótons.

Outras propostas de trabalhos futuros incluem a aplicação de *irradiance caching* em outras técnicas não discutidas neste trabalho, bem como a identificação de características adicionais que possam tornar a interpolação de amostras existentes e a identificação da necessidade de novas amostras mais precisas.

Finalmente, outro possível trabalho futuro está no uso do poder de processamento das placas de vídeo (GPU, *graphics processing unit*) para acelerar os processos de geração de imagens realistas.

## REFERÊNCIAS BIBLIOGRÁFICAS

- [1] ANDRADÓTTIR, S., HEYMAN, D. P. E OTT, T. J. Smoothing Methods for Variance Reduction in Simulation of Markov Chains. In *Winter Simulation Conference* (1992), pp. 453–457.
- [2] ARVO, J. Backward Ray Tracing. *ACMSIGGRAPH'86 Course Notes - Developments in Ray Tracing 12* (1986), 259–263.
- [3] BERGMAN, L., FUCHS, H., GRANT, E. E SPACH, S. Image Rendering by Adaptive Refinement. *Proceedings of the 13th Annual Conference on Computer Graphics and Interactive Techniques 20*, 4 (Agosto 1986), 29–37.
- [4] BLENDER. Blender 3D Content Creation Suite, 2008. <http://www.blender.org>.
- [5] CARTER, L. E CASHWELL, E. Particle-transport simulation with the monte carlo method. Tech. rep., TID–26607, Los Alamos Scientific Lab., N. Mex.(USA), 1975.
- [6] CHEN, S., RUSHMEIER, H., MILLER, G. E TURNER, D. A Progressive Multi-Pass Method for Global Illumination. *Proceedings of the 18th annual conference on Computer graphics and interactive techniques 25*, 4 (Julho 1991), 165–174.
- [7] CLINE, D. E EGBERT, P. A practical introduction to metropolis light transport. Tech. rep., Technical Report: Department of Computer Science, Brigham Young University, 2005.
- [8] COHEN, M., CHEN, S., WALLACE, J. E GREENBERG, D. A Progressive Refinement Approach to Fast Radiosity Image Generation. *Proceedings of the 15th Annual Conference on Computer Graphics and Interactive Techniques 22*, 4 (Agosto 1988), 75–84.
- [9] COHEN, M. E GREENBERG, D. The Hemi-Cube: A Radiosity Solution for Complex Environments. *Proceedings of the 12th Annual Conference on Computer Graphics and Interactive Techniques 19*, 3 (Julho 1985), 31–40.

- [10] COOK, R. Stochastic Sampling in Computer Graphics. *ACM Transactions on Graphics (TOG)* 5, 1 (Janeiro 1986), 51–72.
- [11] COOK, R., PORTER, T. E CARPENTER, L. Distributed Ray Tracing. *Proceedings of the 11th Annual Conference on Computer Graphics and Interactive Techniques* 18, 3 (Julho 1984), 137–145.
- [12] DABROVIC, M. Catedral de Sao Jacó, 2008. <http://hdri.cgtechniques.com/~sibenik2/>, acesso em: 30 de novembro de 2008.
- [13] DABROVIC, M. Sponza Atrium, 2008. <http://hdri.cgtechniques.com/~sponza/files/>, acesso em: 30 de novembro de 2008.
- [14] DE BERG, M., VAN KREVELD, M., OVERMARS, M. E SCHWARZKOPF, O. *Computational Geometry: Algorithms and Applications*. Springer, 2000.
- [15] DENNY, M. Introduction to importance sampling in rare-event simulations. *European Journal of Physics* 22, 4 (2001), 403–412.
- [16] DUTRÉ, P., BEKAERT, P. E BALA, K. *Advanced Global Illumination*. AK Peters, Ltd., 2003.
- [17] FAN, S., CHENNEY, S. E LAI, Y. Metropolis Photon Sampling with Optional User Guidance. *Eurographics Symposium on Rendering* (2005), 127–138.
- [18] FOLEY, J., ET AL. *Introduction to Computer Graphics*. Addison-Wesley Reading, Mass, 1994.
- [19] GAWEL, K. Red room, 2008. modelo gentilmente cedido por Krzysztof Gawel.
- [20] GLASSNER, A. *An Introduction to Ray Tracing*. Morgan Kaufmann, 1989.
- [21] GORAL, C., TORRANCE, K., GREENBERG, D. E BATTAILE, B. Modeling the Interaction of Light Between Diffuse Surfaces. *Proceedings of the 11th Annual Conference on Computer Graphics and interactive techniques* 18, 3 (Julho 1984), 213–222.



- [22] GOURAUD, H. Continuous Shading of Curved Surfaces. *IEEE Transactions on Computers* 20, 6 (Junho 1971), 623–629.
- [23] HACHISUKA, T., OGAKI, S. E JENSEN, H. Progressive photon mapping. *ACM Transactions on Graphics (TOG)* 27, 5 (2008).
- [24] HALTON, J. A retrospective and prospective survey of the monte carlo method. *Siam review* (1970), 1–63.
- [25] HAMMERSLEY, J. E HANDSCOMB, D. *Monte carlo methods*. Methuen young books, 1964.
- [26] HANRAHAN, P., SALZMAN, D. E AUPPERLE, L. A Rapid Hierarchical Radiosity Algorithm. *ACM SIGGRAPH Computer Graphics* 25, 4 (Julho 1991), 197–206.
- [27] HASTINGS, W. Monte carlo sampling methods using markov chains and their applications. *Biometrika* 57, 1 (1970), 97–109.
- [28] HOUSE, B. G. Blender Green House, 2009. <http://blender-archi.tuxfamily.org/Greenhouse>, acesso em: 13 de fevereiro de 2009.
- [29] IMMEL, D., COHEN, M. E GREENBERG, D. A radiosity method for non-diffuse environments. *ACM SIGGRAPH Computer Graphics* 20, 4 (1986), 133–142.
- [30] INDIGO. Indigo Renderer, 2008. <http://www.indigorenderer.com>, acesso em: 23 de abril de 2008.
- [31] JARGAS, A. *Expressões Regulares - Guia de Consulta Rápida*. Novatec, 2001.
- [32] JENSEN, H. *Realistic Image Synthesis Using Photon Mapping*. AK Peters, Ltd., 2001.
- [33] JENSEN, H., ARVO, J., DUTRE, P., KELLER, A., OWEN, A., PHARR, M. E SHIRLEY, P. Monte Carlo Ray Tracing. *ACM SIGGRAPH Course Notes* 44 (2003).
- [34] JENSEN, H. E CHRISTENSEN, N. Photon Maps in Bidirectional Monte Carlo Ray Tracing of Complex Objects. *Computers and Graphics* 19, 2 (Março 1995), 215–224.

- [35] KAJIYA, J. The Rendering Equation. *International Conference on Computer Graphics and Interactive Techniques: Proceedings of the 13th Annual Conference on Computer Graphics and Interactive Techniques 1986* (Agosto 1986), 143–150.
- [36] KRIVÁNEK, J., BOUATOUCH, K., PATTANAİK, S. E ZARA, J. Making radiance and irradiance caching practical: Adaptive caching and neighbor clamping. *Rendering Techniques (June 2006)*, Akenine-Möller T., Heidrich W.,(Eds.), Eurographics Association (2006).
- [37] KŘIVÁNEK, J., GAUTRON, P., WARD, G., ARIKAN, O. E JENSEN, H. Practical global illumination with irradiance caching. In *International Conference on Computer Graphics and Interactive Techniques* (2007), ACM New York, NY, USA.
- [38] LAFORTUNE, E. E WILLEMS, Y. Bi-directional path tracing. In *Proceedings of Third International Conference on Computational Graphics and Visualization Techniques* (1993).
- [39] LAI, Y.-C., LIU, F., ZHANG, L. E DYER, C. Efficient Schemes for Monte Carlo Markov Chain Algorithms in Global Illumination. In *Proceedings of the 4th International Symposium on Advances in Visual Computing* (Berlin, Heidelberg, 2008), Springer-Verlag, pp. 614–623.
- [40] LENGYEL, E. *Mathematics for 3D game programming and computer graphics*. Charles river media, 2003.
- [41] LISCHINSKI, D., TAMPIERI, F. E GREENBERG, D. Combining hierarchical radiosity and discontinuity meshing. In *Proceedings of the 20th annual conference on Computer graphics and interactive techniques* (1993), ACM New York, NY, USA, pp. 199–208.
- [42] LUX. Lux Render, 2008. <http://www.luxrender.net>, acesso em: 23 de abril de 2008.
- [43] MCLUNEY, W. *Introduction to Radiometry and Photometry*. Artech House: Massachusetts, 1994.

- [44] METROPOLIS, N., ROSENBLUTH, A. E ROSENBLUTH, M. Equations of state calculations by fast computing machines. *Journal of Chemical Physics* 21, 6 (1953), 1087–1092.
- [45] METROPOLIS, N. E ULAM, S. The monte carlo method. *Journal of the American Statistical Association* (1949), 335–341.
- [46] MOTWANI, R. E RAGHAVAN, P. *Randomized Algorithms*. Cambridge University Press, 1995.
- [47] PEDRINI, H. E SCHWARTZ, W. *Análise de Imagens Digitais: Princípios, Algoritmos, e Aplicações*. Thomson Learning, 2008.
- [48] PHARR, M. E HUMPHREYS, G. *Physically Based Rendering: From Theory to Implementation*. Morgan Kaufmann Publishers, 2004.
- [49] PHONG, B. Illumination for Computer Generated Pictures. *Communications of the ACM* 18, 6 (Junho 1975), 311–317.
- [50] ROSS, S. *Introduction to probability models*. Academic Pr, 2006.
- [51] RUBINSTEIN, R. E KROESE, D. *Simulation and the Monte Carlo method*. Wiley-Interscience, 2007.
- [52] SCHEEL, A., STAMMINGER, M. E SEIDEL, H. Grid Based Final Gather for Radiosity on Complex Clustered Scenes. *Computer Graphics Forum* 21, 3 (2002), 547–556.
- [53] SCHEELF, A., STAMMINGERJ, M. E SEIDELF, H. Thrifty Final Gather for Radiosity. *Rendering Techniques 2001: Proceedings of the Eurographics Workshop in London, United Kingdom, June 25-27, 2001* (2001).
- [54] SHAO, M. E BADLER, N. A Gathering and Shooting Progressive Refinement Radiosity Method. Tech. Rep. MS-CIS-93-03, University of Pennsylvania, Philadelphia, PA 19104-6389, Janeiro 1993.
- [55] SHIRLEY, P. A Ray Tracing Method for Illumination Calculation in Diffuse-Specular Scenes. *Proceedings of Graphics Interface* (1990), 205–212.

- [56] SHIRLEY, P. E MORLEY, R. *Realistic Ray Tracing*. AK Peters, Ltd., 2003.
- [57] SMITS, B., ARVO, J. E SALESIN, D. An Importance-Driven Radiosity Algorithm. *Proceedings of the 19th Annual Conference on Computer Graphics and Interactive Techniques 26*, 2 (Julho 1992), 273–282.
- [58] STROUSTRUP, B. *The C++ Programming Language*. Addison-Wesley Professional, 1997.
- [59] TABELLION, E. E LAMORLETTE, A. An approximate global illumination system for computer generated films. In *International Conference on Computer Graphics and Interactive Techniques (2004)*, ACM New York, NY, USA, pp. 469–476.
- [60] VEACH, E. *Robust Monte Carlo methods for light transport simulation*. PhD thesis, Stanford University, 1997.
- [61] VEACH, E. E GUIBAS, L. Bidirectional estimators for light transport. In *Fifth Eurographics Workshop on Rendering (1994)*, pp. 147–162.
- [62] VEACH, E. E GUIBAS, L. Optimally combining sampling techniques for monte carlo rendering. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques (1995)*, ACM New York, NY, USA, pp. 419–428.
- [63] VEACH, E. E GUIBAS, L. Metropolis light transport. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques (1997)*, ACM Press/Addison-Wesley Publishing Co. New York, NY, USA, pp. 65–76.
- [64] WALLACE, J., COHEN, M. E GREENBERG, D. A Two-pass Solution to the Rendering Equation: A Synthesis of Ray Tracing and Radiosity Methods. *Proceedings of the 14th Annual Conference on Computer Graphics and interactive techniques 21*, 4 (Julho 1987), 311–320.
- [65] WARD, G. The radiance lighting simulation and rendering system. In *Proceedings of the 21st annual conference on Computer graphics and interactive techniques (1994)*, ACM New York, NY, USA, pp. 459–472.

- [66] WARD, G. E HECKBERT, P. Irradiance Gradients. *Conference: 3. Eurographics on Rendering Workshop, Bristol (United Kingdom)* (Maio 1992), 18–20.
- [67] WARD, G., RUBINSTEIN, F. E CLEAR, R. A Ray Tracing Solution for Diffuse Interreflection. *Proceedings of the 15th Annual Conference on Computer Graphics and Interactive Techniques 22, 4* (Agosto 1988), 85–92.
- [68] WHITTED, T. An Improved Illumination Model for Shaded Display. *Communications of the ACM 23, 6* (Junho 1980), 343–349.
- [69] YAFARAY. YafaRay Free Raytracer, 2008. <http://www.yafaray.org>, acesso em: 20 de abril de 2008.
- [70] YAFARAY. YafaRay XML File Format, 2008. <http://www.yafaray.org/development/documentation/XMLspecs>, acesso em: 20 de abril de 2008.