

**UNIVERSIDADE FEDERAL DO ABC**

**EDUARDO RENESTO ESTANQUIERE**

**UMA REVISÃO DE MÉTODOS DE MONTE CARLO PARA TRANSPORTE DE LUZ  
EM COMPUTAÇÃO GRÁFICA**

**SANTO ANDRÉ**

**2020**

**UNIVERSIDADE FEDERAL DO ABC**

**EDUARDO RENESTO ESTANQUIERE**

**UMA REVISÃO DE MÉTODOS DE MONTE CARLO PARA TRANSPORTE DE LUZ  
EM COMPUTAÇÃO GRÁFICA**

Projeto apresentado para a Universidade  
Federal do ABC, para obtenção de  
aprovação na disciplina Projeto Dirigido.  
Profa. Dra. Marisa Soares - Orientadora

**SANTO ANDRÉ**

**2020**

## Resumo

Apresentamos um histórico e uma introdução ao problema do transporte de luz em computação gráfica. Fazemos uma revisão de cinco métodos de Monte Carlo para a solução da equação de renderização, sendo eles Path Tracing Ingênuo, Amostragem de Importância Múltipla, Path Tracing Bidirecional, Metropolis Light Transport e Gradient Domain Path Tracing. Resumimos as ideias principais e casos ótimos para cada método.

**Palavras-chave:** computação gráfica, transporte de luz, renderização

## **Abstract**

We present a historical review and an introduction about the light transport problem in computer graphics. We analyze five Monte Carlo methods for solving the Rendering Equation, them being Naive Path Tracing, Multiple Importance Sampling, Bidirectional Path Tracing, Metropolis Light Transport and Gradient Domain Path Tracing. We summarize the main ideas and optimal cases for each method.

**Keywords:** computer graphics, light transport, rendering

# Sumário

<b>Resumo</b>	<b>3</b>
<b>Abstract</b>	<b>4</b>
<b>Introdução</b>	<b>6</b>
A Equação de Renderização . . . . .	8
<b>Objetivos</b>	<b>9</b>
Gerais . . . . .	9
Específicos . . . . .	9
<b>Metodologia</b>	<b>10</b>
<b>Análise</b>	<b>10</b>
Path Tracing Ingênuo . . . . .	10
Amostragem Múltipla de Importância . . . . .	11
Path Tracing Bidirecional . . . . .	12
Metropolis Light Transport . . . . .	13
Gradient Domain Path Tracing . . . . .	14
<b>Conclusão</b>	<b>15</b>
<b>Referências Bibliográficas</b>	<b>17</b>

## Introdução

Em computação gráfica, especialmente no contexto de síntese de imagens, é central o problema da renderização fotorrealística. Em termos gerais, consiste em, dada uma cena tridimensional, construir algoritmicamente uma imagem (matriz de *pixels*) que seria idêntica a uma fotografia daquela cena.

Trabalhos iniciais na área durante a década de 60 se concentraram no problema da superfície escondida, que consiste em determinar quais faces de um objeto não são visíveis para um observador em determinada cena. Esforços para resolver esse problema resultaram em uma melhoria significativa de performance em renderização, o que tornou possível um maior foco na aparência das imagens.

Nesse contexto, Bui Tuong Phong propôs, em 1975, um modelo matemático para a iluminação de superfícies renderizadas (PHONG, 1975). O modelo considera que a cor em um ponto da superfície é dado pela soma de três efeitos: a iluminação *ambiente*, a iluminação *especular* e a iluminação *difusa*. A primeira é uma aproximação da contribuição de luz vinda de todas as direções da cena; a segunda é a luz refletida dependente da posição do observador, e a última é a luz refletida igualmente em todas as direções a partir do objeto em si. A Figura 1 ilustra as três componentes.

Algebricamente, o modelo pode ser escrito como uma equação simples:

$$I = I_a + k_d(\vec{N} \cdot \vec{L}) + k_s(\vec{V} \cdot \vec{L})^n$$

onde

- $I$  é a intensidade de luz no ponto de interesse
- $I_a$  é a contribuição advinda da iluminação ambiente,
- $k_d$  é o coeficiente de difusão difusa,
- $\vec{N}$  é o versor normal à superfície onde ocorreu a intersecção,
- $\vec{L}$  é um versor em direção à fonte de luz,
- $k_s$  é o coeficiente de reflexão especular,
- $\vec{V}$  é um versor dado pela reflexão da direção da superfície ao ponto de observação em relação a  $\vec{N}$ , e

- $n$  é o fator *shininess*, particular para cada material.

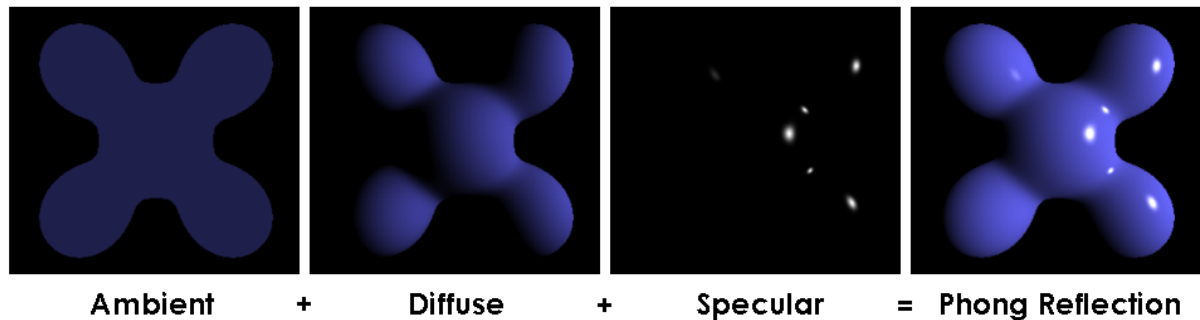


Figura 1: As componentes do modelo de Phong. Fonte: Wikipedia

Embora aproxime satisfatoriamente iluminação direta (isto é, causada diretamente pelas fontes de luz da cena), o modelo de Phong não leva em consideração iluminação indireta, onde existe contribuição luminosa advinda de outros objetos na cena, além de não empregar modelagens para os efeitos de reflexão e de refração. Uma das soluções mais importantes para esse problema foi introduzida por Turner Whitted em 1979, com o *traçador de raios recursivo* (em inglês, *recursive ray tracer*) (WHITTED, 1979).

Tal algoritmo busca inspiração na Óptica Clássica e trata a luz como raios, traçando o caminho de tais raios entre a fonte de luz da cena e o observador. São gerados raios (chamados de *raios primários*) a partir do observador (utilizando algum modelo de câmera, como o *pinhole camera*), e, para cada raio, é realizado um teste de intersecção com os objetos constituintes da cena. Se, para dado objeto e para dado raio existir uma intersecção, então tal objeto é visível a partir do ponto de origem de tal raio.

O cálculo da iluminação é executado para cada intersecção, e é dado pela equação

$$I = I_a + k_d(\vec{N} \cdot \vec{L}) + k_s S + k_t T$$

onde

- $I$  é a intensidade de luz no ponto da intersecção
- $I_a$  é a contribuição advinda da iluminação ambiente,
- $k_d$  é o coeficiente de difusão difusa,
- $\vec{N}$  é o versor normal à superfície onde ocorreu a intersecção,
- $\vec{L}$  é um versor em direção à fonte de luz,

- $k_s$  é o coeficiente de reflexão especular,
- $S$  é a intensidade de luz vinda da direção  $\vec{R}$ ,
- $k_t$  é o coeficiente de transmissão, e
- $T$  é a intensidade de luz vinda da direção  $\vec{P}$ .

De imediato, nota-se que os fatores ambiente e difuso do modelo de Phong são mantidos. O fator especular é modificado e o fator de *transmissão* é adicionado. Ambas alterações utilizam a intensidade de luz chegando no ponto da intersecção a partir de direções específicas - para o fator especular, é utilizado o versor  $\vec{R}$ , causado pela reflexão pela normal; para o fator de transmissão, o versor  $\vec{P}$  é dado pela Lei de Snell.

No entanto, o que dá o nome à técnica é a maneira pela qual tais intensidades são calculadas. São gerados novos raios, partindo do ponto da intersecção em direção aos versores de interesse. Tais raios são traçados, recursivamente, da mesma maneira que os raios primários descritos acima.

## A Equação de Renderização

Uma formalização do problema da renderização fotorrealística se deu pela introdução da chamada *Equação de Renderização*, feita independentemente por (KAJIYA, 1986) e (IMMEL; COHEN; GREENBERG, 1986). Uma solução da equação de renderização corresponde a uma imagem fisicamente correta da cena.

A equação pode ser escrita na seguinte maneira, retirada de (PHARR; JAKOB; HUMPHREYS, 2017):

$$L_0(p, \omega_0) = L_e(p, \omega_0) + \int_{\Omega} f(p, \omega_0, \omega_i) L_i(p, \omega_i) |\cos\theta_i| d\omega_i$$

onde

- $L_0$  é a *radiância* saindo do ponto  $p$  na direção  $\omega_0$ ,
- $L_e$  é a radiância emitida no ponto  $p$  na direção  $\omega_0$ ,
- $\Omega$  é uma esfera unitária ao redor de  $p$ ,
- $f(p, \omega_0, \omega_i)$  é a chamada função de distribuição de dispersão bidirecional (*BSDF*, das iniciais em inglês) avaliada no ponto  $p$  e nas direções de reflexão e incidência  $\omega_0$  e  $\omega_i$ , respectivamente,



- $L_i(p, \omega_i)$  é a radiância incidente no ponto  $p$  a partir da direção  $\omega_i$ , e
- $\theta_i$  é o ângulo formado entre a direção  $\omega_i$  e o vetor normal da superfície no ponto  $p$ .

Qualitativamente, a equação modela que a luz em certo ponto é a soma de um termo emissivo ( $L_e$ ) com um termo reflexivo (representado pela integral). É interessante notar que o fator  $L_i$  na integral representa toda a luz na cena que chega no ponto  $p$ .

Para fins de exemplo, suponha um ponto  $p'$  na cena, onde  $p' \neq p$  e  $p$  e  $p'$  são visíveis entre si. Desta forma,  $L_0(p, \omega_0)$  dependerá do valor de  $L_0(p', \omega_0)$ , que por sua vez dependerá novamente de  $L_0(p, \omega_0)$ . Isso mostra que a equação de renderização é uma equação integral recursiva, e portanto pode apenas ser resolvida analiticamente para casos extremamente simples. Na maioria das cenas, serão necessários métodos numéricos para resolvê-la – tais métodos, especialmente os baseados em integração de Monte Carlo, serão apresentados neste trabalho.

Segundo (PHARR; JAKOB; HUMPHREYS, 2017), o traçado de raios de Whitted resolve a equação de renderização utilizando a simplificação de tomar apenas as direções  $\omega_i$  tais que apontem para fontes de luz da cena. Desta maneira, a integral é convertida em uma soma finita, facilmente calculada iterativamente. Essa aproximação desconsidera a chamada *iluminação global*, que é caracterizada por contribuição luminosa vinda de objetos na cena que não são intrinsecamente fontes de luz.

Damos o nome de *Transporte de Luz* ao problema de resolver a equação de renderização.

Este trabalho foca em apresentar técnicas que solucionem a equação de renderização de maneira mais precisa utilizando, principalmente, métodos de Monte Carlo.

## Objetivos

### Gerais

Analisar diferentes maneiras de utilizar métodos de Monte Carlo para resolver o problema do transporte de luz, a fim de construir uma boa base teórica para prosseguir estudos e projetos futuros na área.

### Específicos

- Indexar diferentes métodos quanto a seus pontos fortes e fracos;

- Estudar a maneira com que métodos de Monte Carlo são desenvolvidos e modificados;
- Levantar fontes bibliográficas relevantes para estudos futuros;

## Metodologia

Selecionamos quatro publicações contendo cinco técnicas, sendo três artigos e uma tese de doutoramento. Tais publicações foram encontradas a partir do banco de referências DBLP, dos anais da conferência SIGGRAPH, e pelo livro *Physically Based Rendering* (PHARR; JAKOB; HUMPHREYS, 2017). Também buscamos influências no software *Mitsuba Renderer* (JAKOB, 2010), que apresenta implementações de várias técnicas sob uma licença livre.

Fizemos, então, uma pesquisa bibliográfica revisando essas cinco técnicas.

Embora todos os métodos apresentados convirjam para a mesma cena dado tempo infinito, cada um apresenta vantagens para cenas particulares.

Deste modo, analisamos as técnicas de acordo com as cenas onde elas têm maior eficiência, além de considerar a velocidade de convergência.

## Análise

### Path Tracing Ingênuo

A primeira solução, e a mais simples com cálculo da Iluminação Global, foi apresentada por Kajiya no mesmo artigo em que o autor desenvolve a Equação de Renderização (KAJIYA, 1986).

O autor desenvolve formalmente o método, tomando inspiração em trabalhos anteriores na área de simulação de transporte de calor. Ao final, o algoritmo é uma mudança facilmente integrada em traçadores de raios no estilo de Whitted. Em tradução livre, “*Ao invés de traçar uma árvore de ramificações [i.e. os três novos raios no modelo de Whitted], apenas trace um caminho com os raios escolhidos probabilisticamente*” (KAJIYA, 1986). O tipo de cada raio também é definido probabilisticamente, de acordo com uma distribuição pré-definida.

Por ser um método de Monte Carlo, é afetado pela *Lei dos Grandes Números*, isto é, se faz necessário um grande número de amostras (em inglês, *samples*) para se obter uma variância pequena. Em transporte de luz, uma variância alta se manifesta na presença de ruído na imagem

solução.

O resultado é uma imagem realista, com iluminação global precisa e sombreamento correto. No entanto, não apresenta otimização especial nenhuma, portanto o tempo de convergência pode ser alto (i.e. serão necessários muitos *samples por pixel*) dado uma cena complexa o suficiente.

## **Amostragem Múltipla de Importância**

Introduzido por Eric Veach em sua tese de doutorado (VEACH, 1998), resume-se em melhorar as escolhas aleatórias a fim de diminuir a variância da solução, o que diminui o tempo de convergência da avaliação da integral.

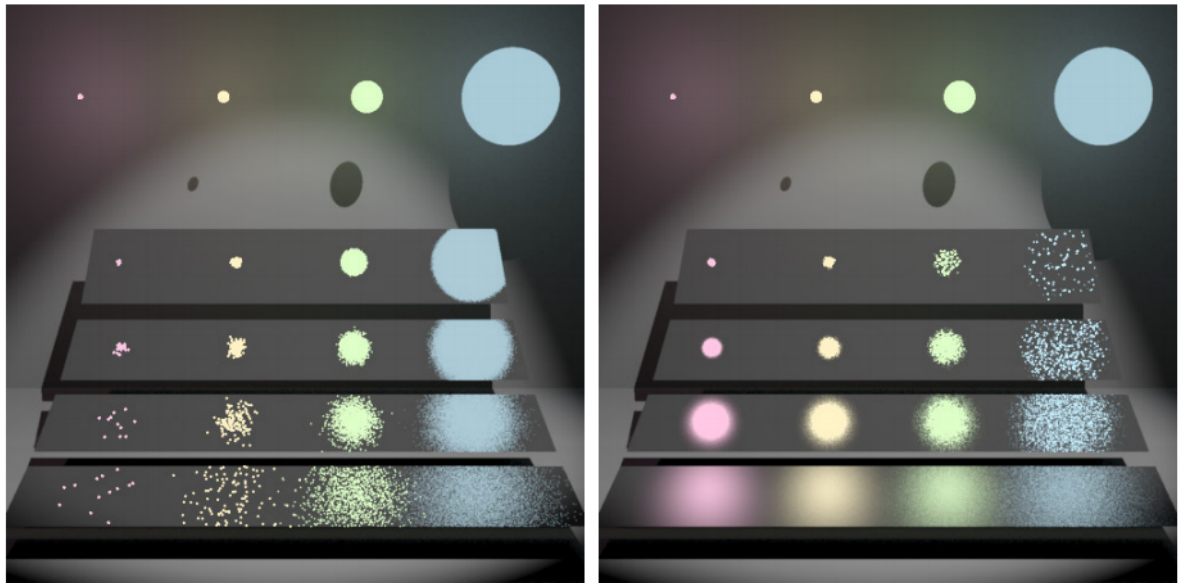
Veach afirma que existem duas estratégias comuns de realizar a amostragem no processo de renderização Monte Carlo:

- (a) Amostrar a BSDF: raios são traçados escolhendo direções aleatórias com probabilidade proporcional à BSDF da superfície atual.
- (b) Amostrar a fonte de luz: raios são traçados escolhendo pontos aleatórios na superfície da fonte de luz;

A figura 2, retirada da tese, mostra a diferença entre as duas estratégias para uma cena específica. São mostradas quatro superfícies lustrosas, com rugosidades diferentes e crescentes, começando de cima. As duas imagens foram geradas com o mesmo número de amostras por pixel.

Note que (a) apresenta resultados satisfatórios rapidamente para superfícies espelhadas, enquanto (b) resulta em pontos dispersos e ruidosos. Do mesmo modo, (a) apresenta ruído em superfícies difusas, onde (b) tem vantagem.

A Amostragem Múltipla de Importância (*Multiple Importance Sampling, MIS*, em inglês) nos dá uma maneira de combinar as duas estratégias sob demanda durante a execução da integração. O autor desenvolve diversas heurísticas e métodos para efetuar tal combinação. O resultado é uma solução que converge rapidamente para cenas tanto predominantemente difusas quanto predominantemente reflexivas.



(a) Sampling the BSDF

(b) Sampling the light sources

Figura 2: Amostrar a fonte de luz vs BSDF. Fonte: (VEACH, 1998)

## Path Tracing Bidirecional

Também introduzido por Veach em sua tese de doutorado, o Path Tracing Bidirecional faz uma modificação na maneira com que os caminhos entre a fonte de luz e o observador são criados e amostrados.

Nas técnicas mostradas até agora, os caminhos são gerados a partir da câmera e traçados em direções escolhidas por alguma estratégia. O Path Tracing Bidirecional gera um número  $s$  de caminhos partindo do observador e  $t$  partindo da fonte de luz, e tenta conectá-los. Caso for possível, cada segmento de caminho é amostrado independentemente utilizando importâncias múltiplas, como descrito acima.

Essa modificação possibilita a adaptação do processo dependendo da complexidade da cena a partir do ajuste dos parâmetros  $s$  e  $t$ . Para cenas onde superfícies de interesse têm pouca chance de serem amostradas, como aquelas que possuem fontes de luz pequenas e de difícil acesso, uma combinação específica de  $s$  e  $t$  gerará caminhos que contribuirão para uma convergência mais rápida e com menor variância.

A figura 3 mostra uma chamada *Árvore de Veach* para uma cena considerada difícil – a única fonte de luz é a luminária e a maior parte da iluminação é indireta. Note que valores diferentes de  $s$  e  $t$  contribuem para diferentes efeitos da cena serem desenvolvidos.

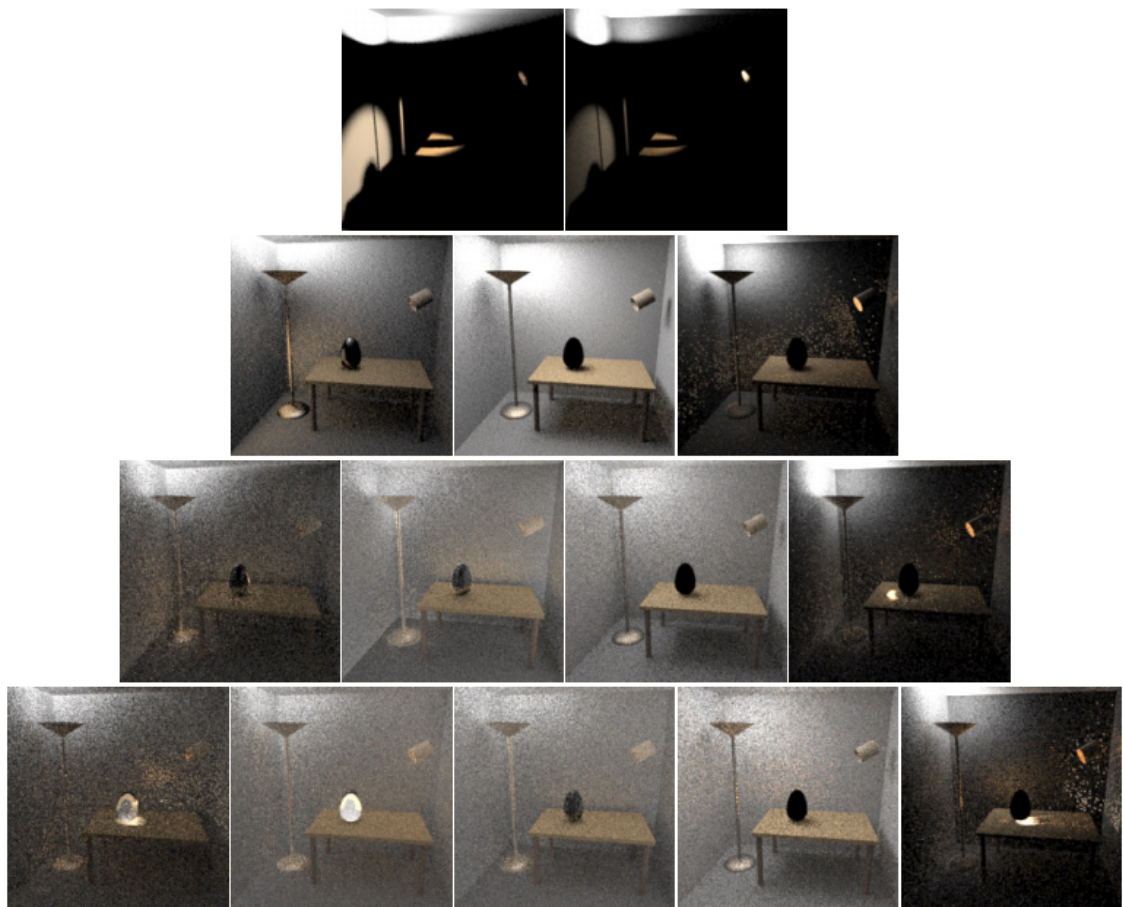


Figura 3: Pirâmide de Veach para uma cena. A  $i$ -ésima linha corresponde a  $s + t = i + 1$ , a  $j$ -ésima imagem em cada linha corresponde a  $s = j$ . Fonte: (VEACH, 1998)

Combinando várias configurações para  $s$  e  $t$  utilizando Amostragem Múltipla de Importância, é possível gerar resultados formalmente ótimos. Deste modo, o algoritmo se sobressai em cenas onde existem poucas e pequenas fontes de luz – especialmente cenas dentro de cômodos.

### Metropolis Light Transport

Outro algoritmo desenvolvido por Veach (VEACH; GUIBAS, 1997), também é uma modificação do path tracing ingênuo. É uma adaptação do algoritmo de Metropolis-Hastings ao problema da solução da Equação de Renderização. Portanto, é considerado um método de Monte Carlo baseado em Cadeias de Markov.

O ponto principal da modificação está na maneira com a qual os caminhos são gerados. Dado um caminho inicial, o algoritmo faz uma *mutação* específica no mesmo caminho. Tal mutação pode ser *aceita* ou *rejeitada* dependendo de uma distribuição de probabilidade calculada para refletir a contribuição da mesma na imagem final.

As mutações são feitas localmente no espaço do caminho. Deste modo, caminhos que contribuem mais para a redução da variância serão favorecidos, aumentando em alta escala a velocidade de convergência. Por utilizar Cadeias de Markov, os caminhos importantes serão “lembrados” e mutados com mais frequência, resultando em uma maior eficiência.

O resultado é um algoritmo que consegue renderizar facilmente cenas consideradas difíceis, como aquelas que possuem grande contribuição de iluminação indireta. Outros efeitos também são acelerados, como a produção de cáusticas.

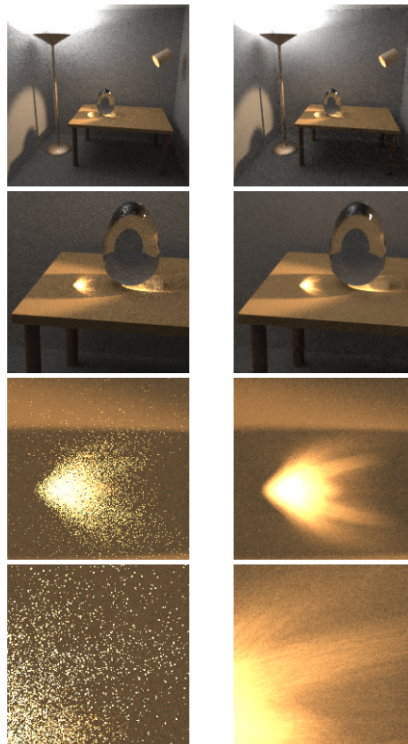


Figura 4: Comparação entre Path Tracing Bidirecional (esquerda) e MLT (direita). Fonte: (VEACH; GUIBAS, 1997)

Na figura 4, o autor faz uma comparação entre o Metropolis Light Transport e o Bidirectional Path Tracing. Em todas as imagens, ambos algoritmos foram executados por uma mesma quantidade de tempo, com o fim de mostrar a diferença de velocidade de convergência entre os dois. Note que o MLT consegue produzir muito mais rapidamente os efeitos de iluminação indireta e as cáusticas criadas pelo ovo de vidro.

## Gradient Domain Path Tracing

Introduzido por Kettunen et al. (KETTUNEN et al., 2015), é uma modificação do path tracing ingênuo que faz parte de uma família de métodos de transporte de luz chamada de *renderização*

*gradiente*. O algoritmo gera um *mapa de gradientes*, que consiste na diferença entre um pixel e seus vizinhos. Esse gradiente é usado, então, para resolver a Equação de Poisson, cuja solução pode ser utilizada para acelerar o processo de Monte Carlo (HUA et al., 2019).

O mapa de gradientes é gerado amostrando normalmente a cena como no Path Tracing ingênuo, com a adição de gerar um raio extra correspondendo a um pixel vizinho. Tal adição torna possível a medição de diferença finita entre dois pixels.

O artigo desenvolve uma longa análise do comportamento do gradiente para diversas cenas, concluindo que a solução da equação de Poisson a partir do mesmo resulta em uma boa reconstrução da imagem apenas, geralmente, em situações de alta frequência. Então, a informação reconstruída é combinada com a amostragem Monte Carlo da cena, gerando assim uma solução total da equação de renderização.

Quando comparado com outros métodos, o Gradient Domain Path Tracing consegue gerar soluções com variância muito menor durante o mesmo tempo de processamento. Na figura 5 é possível ver o comportamento do algoritmo em comparação com o Path Tracing Ingênuo. Note que a quantidade de ruído é muito menor mesmo com poucos samples por pixel.

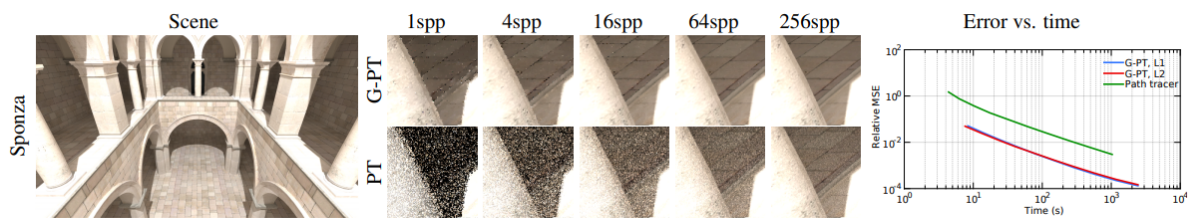


Figura 5: Gradient Domain Path Tracing (topo) vs Path Tracing (base). Fonte: (KETTUNEN et al., 2015)

## Conclusão

Fizemos um análise bibliográfica de cinco diferentes técnicas baseadas em Monte Carlo para a resolução da Equação de Renderização, a qual introduzimos em uma revisão histórica e conceitual.

O quadro abaixo apresenta uma síntese geral dos algoritmos discutidos.

Técnica	Fonte	Conceito central	Casos mais eficientes	Casos menos eficientes
Path Tracing Ingênuo	(KAJIYA, 1986)	Traçar caminhos aleatórios entre o observador e as fontes de luz	Cenas simples	Cenas com fontes de luz escassas e pequenas
Amostragem Múltipla de Importância	(VEACH, 1998)	Combinar eficientemente amostragens das BRDFs e das fontes de luz	Cenas com múltiplos tipos de superfícies	Cenas com fontes de luz escassas e pequenas
Path Tracing Bidirecional	(VEACH, 1998)	Criar caminhos partindo tanto da fonte de luz quanto do observador e conectá-los	Cenas com fontes de luz pequenas; cenas em cômodos	Cenas com obstáculos entre as fontes de luz e as superfícies
Metropolis Light Transport	(VEACH; GUIBAS, 1997)	Utilizar cadeias de Markov para gerar caminhos com maior importância	Cenas com geometrias “difíceis”; cenas onde boa parte da luz atravessa poucos caminhos	-



Técnica	Fonte	Conceito central	Casos mais eficientes	Casos menos eficientes
Gradient Domain Path Tracing	(KETTUNEN et al., 2015)	Reconstruir a imagem partindo de um gradiente amostrado e da equação de Poisson	Cenas que gerariam ruído de alta frequência	Cenas com muitas superfícies especulares

## Referências Bibliográficas

- HUA, B. et al. A Survey on Gradient-Domain Rendering. **Comput. Graph. Forum**, v. 38, n. 2, p. 455–472, 2019.
- IMMEL, D. S.; COHEN, M. F.; GREENBERG, D. P. **A radiosity method for non-diffuse environments**. (D. C. Evans, R. J. Athay, Eds.)Proceedings of the 13th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1986, Dallas, Texas, USA, August 18-22, 1986. **Anais...ACM**, 1986Disponível em: <<https://doi.org/10.1145/15922.15901>>
- JAKOB, W. **Mitsuba renderer**, 2010.
- KAJIYA, J. T. **The rendering equation**. (D. C. Evans, R. J. Athay, Eds.)Proceedings of the 13th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1986, Dallas, Texas, USA, August 18-22, 1986. **Anais...ACM**, 1986Disponível em: <<https://doi.org/10.1145/15922.15902>>
- KETTUNEN, M. et al. Gradient-Domain Path Tracing. **ACM Trans. Graph.**, v. 34, n. 4, 2015.
- PHARR, M.; JAKOB, W.; HUMPHREYS, G. **Physically based rendering: from theory to implementation**. [s.l.] Morgan Kaufmann, 2017.
- PHONG, B. T. Illumination for Computer Generated Pictures. **Commun. ACM**, v. 18, n. 6, p. 311–317, 1975.
- VEACH, E. **Robust Monte Carlo Methods for Light Transport Simulation**. tese de doutorado—Stanford, CA, USA: Stanford University, 1998.
- VEACH, E.; GUIBAS, L. J. **Metropolis light transport**. (G. S. Owen, T. Whitted, B. Mones-Hattal, Eds.)Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1997, Los Angeles, CA, USA, August 3-8, 1997. **Anais...ACM**, 1997Disponível em: <<https://doi.org/10.1145/258734.258775>>
- WHITTED, T. **An improved illumination model for shaded display**. (T. A. DeFanti et al., Eds.)Proceedings of the 6th Annual Conference on Computer Graphics and Interactive Techni-

ques, SIGGRAPH 1979, Chicago, Illinois, USA, August 8-10, 1979. **Anais...**ACM, 1979Disponível em: <<https://doi.org/10.1145/800249.807419>>