Pacific Graphics (2018)
H. Fu, A. Ghosh, and J. Kopf (Editors)

Short Paper

Efficient Metropolis Path Sampling for Material Editing and Re-rendering

Tomoya Yamaguchi Tatsuya Yatagawa Shigeo Morishima

Graduate School of Advanced Science and Engineering, Waseda University, Japan
tomoya.tomoya@akane.waseda.jp, tatsy@acm.org, shigeo@waseda.jp

Abstract

This paper proposes efficient path sampling for re-rendering scenes after material editing. The proposed sampling method is based on Metropolis light transport (MLT) and distributes more path samples to pixels whose values have been changed significantly by editing. First, we calculate the difference between images before and after editing to estimate the changes in pixel values. In this step, we render the difference image directly rather than calculating the difference in the images by separately rendering the images before and after editing. Then, we sample more paths for pixels with larger difference values and render the scene after editing by reducing variances of Monte Carlo estimators using the control variates. Thus, we can obtain rendering results with a small amount of noise using only a small number of path samples. We examine the proposed sampling method with a range of scenes and demonstrate that it achieves lower estimation errors and variances over the state-of-the-art methods.

Keywords: Ray tracing; Metropolis light transport; Material editing

CCS Concepts

•***Computing methodologies*** → ***Ray tracing;*** *Graphics processors;*

1. Introduction

In computer graphics, physics-based rendering is one of the most important research topics. Most of the rendering algorithms primarily aim to render given scenes from scratch. Moreover, the scenes are generally re-rendered when the materials or illuminations are changed at actual graphics productions. An excellent survey for such material editing was provided by Schmidt et al. [SPN*14]. Recently, several studies have introduced another type of re-rendering method, which leverages the high-quality rendering image for a scene before editing [GG15, RJN16]. In these studies, the scene after editing was only roughly rendered using a small number of path samples and was combined with high-quality rendering for the scene before editing. Among these studies, Rousselle et al. [RJN16] applied the control variates that reduce the variance of Monte Carlo integration, to re-render the edited scenes. They referred to this method as image-space control variates. Their method renders the scene after editing by two different strategies, which they refer to as "reuse" and "re-render." Then, pixel values of these images are blended using ratios determined from variances of Monte Carlo estimators for the two strategies. Although the method achieved high-quality re-rendering using a small number of path samples, its efficiency can be improved further because it samples the paths uniformly over all pixels.

2. Efficient Path Sampling for Scene Editing

The proposed method extends the image-space control variates [RJN16] by distributing path samples more effectively for scene re-rendering. The key observation of the proposed method is that only a limited number of pixels are changed by material editing; the rest of the pixels remain unchanged. According to this observation, we distribute path samples to significantly edited pixels based on a new importance function that represents the contribution of the sampled path to the final edited image. In this section, we first explain techniques of Monte Carlo integration and image-space control variates. Then, we introduce how the previous method can be extended by non-uniform path sampling.

2.1. Image-space control variates

Monte Carlo integration estimates the integral of function $f(x)$ using N random samples $\{x_i : i = 1, 2, \ldots, N\}$ following a probability density function $p(x)$. The estimate for integral F is written as follows:

$$\langle F \rangle_N = \frac{1}{N} \sum_{i=1}^{N} \frac{f(x_i)}{p(x_i)}. \tag{1}$$

The control variates reduce the variance of $\langle F \rangle_N$ using the estimate $\langle H \rangle_M$ for another function $h(x)$ with M samples (typically $M \gg N$).

DOI: 10.2312/pg.20181271

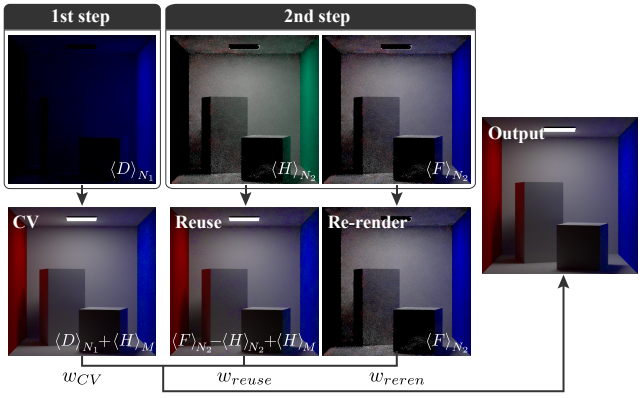

Figure 1: *Overview of the proposed method. The difference image is rendered in the first step, and the importance function is defined with it. The difference is also used to evaluate the Monte Carlo estimates "CV." In the second step, the Monte Carlo estimates, "reuse" and "re-render," are evaluated using the importance function. Finally, the estimates are blended by the weights to reduce the estimation variance.*

If the values of $\{f(x_i)\}$ and $\{h(x_i)\}$ have strong correlation, the variance of $\langle F\rangle_N$ can be reduced by the following formula:

$$\langle F\rangle_N = \langle H\rangle_M + \frac{1}{N}\sum_{i=1}^{N}\frac{h(x_i)-f(x_i)}{p(x_i)}. \tag{2}$$

In the technique of image-space control variates [RJN16], the estimate F corresponds to a pixel value of the image after editing. This estimate is referred to as "re-render" in the formula represented by Eq. 1 and "resuse" in the formula represented by Eq. 2. For clarity, we denote these two values as $\langle F\rangle_N^{reren}$ and $\langle F\rangle_N^{reuse}$. In the method of image-space control variates, the two pixel values for re-render and reuse images are blended with blending ratios defined by a covariance matrix of the Monte Carlo samples:

$$\mathbf{w}_j = \begin{bmatrix} w_j^{reuse} \\ w_j^{reren} \end{bmatrix} = \frac{\Sigma_j^{-1}\mathbf{e}}{\mathbf{e}^\top\Sigma_j^{-1}\mathbf{e}}, \tag{3}$$

where $\mathbf{e}$ is a two-dimensional vector $\mathbf{e} = [1,1]^\top$, and Σ_j is the covariance matrix of j-th pixel.

2.2. Metropolis path sampling

To improve the path sampling efficiency using image-space control variates, we vary the number of path samples for each pixel and distribute more path samples to the pixels whose values have been changed significantly by material editing. Figure 1 outlines the proposed method comprising two steps, i.e., rendering of the difference image and the final edited image. In what follows, we elaborate on the processes of the two steps. We adopted multiplexed Metropolis light transport (MMLT) [HKD14] for both the steps. For simplicity of explanation, we refer to the scenes before and after editing as scene S (source) and scene T (target) henceforth.

In the first step, we roughly render the difference image between images before and after editing. The difference image is rendered by accumulating differences of contributions by the same set of path samples rather than obtaining the difference between scenes S and T rendered separately with different path samples. A vector of random numbers $\bar{u}_i$ is sampled in the primary sample space and mapped to one of the path spaces of S and T. Let $x_s(\bar{u}_i)$ and $x_t(\bar{u}_i)$ be the mappings for the two path spaces; the difference of pixel j is estimated by multiple importance sampling, represented as follows:

$$\langle D_j\rangle \approx \sum_{r\in\{s,t\}}\frac{1}{N_r}\sum_{i=1}^{N_r} w_r(\bar{u}_i)\frac{h(x_r(\bar{u}_i))-f(x_r(\bar{u}_i))}{p(x_r(\bar{u}_i))}, \tag{4}$$

where w_r denotes the weight for multiple importance sampling. For scene editing, we define weight w_r with probability densities in the path spaces:

$$w_r(\bar{u}_i) = \frac{p(x_r(\bar{u}_i))}{p(x_s(\bar{u}_i))+p(x_t(\bar{u}_i))}.$$

By evaluating Eq. 4 using the MMLT method, we can more frequently sample paths providing larger gaps between contributions $f(x)$ and $h(x)$. Considering that material editing affects a limited number of pixels, the above sampling strategy yields the difference image more effectively than sampling the same number of paths for every pixel. The similar idea of rendering a gradient image is leveraged in gradient-domain MLT [LKL*13]. Different from this method, the first step of our method is followed by the second step to improve the rendering quality, particularly for the regions with significant changes in their appearance.

In the second step, we render the edited image using image-space control variates. Different from the original study [RJN16], we allocate more path samples to the pixels with larger difference values. For this purpose, we define an importance function using the difference image, as specified in the first step. The importance function is defined by considering all the luminance of scene S, luminance of scene T, and the difference:

$$Q(x) = \frac{h^*(x)+f^*(x)}{2}|D(x)|^*,$$

where the superscript $*$ denotes an operation to obtain the luminance from per-spectrum pixel values. The acceptance probability of the MMLT method for the transition $(\bar{u},r)\to(\bar{v},r')$ is defined using the importance function:

$$a((\bar{u},r)\to(\bar{v},r')) = \frac{w_{r'}(\bar{v})Q(x_{r'}(\bar{v}))}{w_r(\bar{u})Q(x_r(\bar{u}))}.$$

Thus, a series of path samples can be obtained by the MMLT method using the acceptance probability; further, both the estimates $\langle F\rangle_{N_2}^{reuse}$ and $\langle F\rangle_{N_2}^{reren}$ using Eqs. 1 and 2 can be computed. Here, N_2 is the number of path samples used in the second step. In addition, we can define an estimate $\langle F\rangle_{N_1}^{CV}$ using the difference image, as given in the first step:

$$\langle F\rangle^{CV} = \langle H\rangle_M + \langle D\rangle_{N_1},$$

where $\langle D\rangle$ denotes the estimate derived by Eq. 4 and N_1 is the number of samples used in the first step.

We blend the three estimates $\langle F\rangle^{CV}$, $\langle F\rangle^{reuse}$, and $\langle F\rangle^{reren}$ using

Figure 2: *Rendering result of the proposed method and its comparison to the previous approach [RJN16]. The image is separated diagonally into several regions. Each region corresponds to either of reference image, the result of [RJN16], and that of ours. The results are rendered with 256 samples per pixel on average.*

a similar weighting scheme as Eq. 3. As the path samples used in the first and second steps are uncorrelated, the covariance matrix $\Sigma' \in \mathbb{R}^{3\times 3}$ for the proposed method is defined as follows:

$$\Sigma' = \begin{bmatrix} \mathrm{Var}\left[\langle F \rangle_{N_1}^{CV}\right] & \mathbf{0} \\ \mathbf{0} & \Sigma \end{bmatrix},$$

where Σ is the original covariance matrix in Eq. 3 that [RJN16] used. The blending weights are defined similarly to Eq. 3 using a three-dimensional vector $\mathbf{e}' = [1,1,1]^\top$. The variance of each pixel is calculated in the same manner as that for standard Monte Carlo integration. Suppose that N_j paths $x_{j,1},\ldots x_{j,N_j}$ are sampled in Markov chains for pixel j, we calculate the variance of a Monte Carlo estimate I_j of the pixel value of j as follows:

$$\mathrm{Var}\left[\langle I_j \rangle_{N_j}\right] = \frac{1}{N_j}\sum_{i=1}^{N_j}\left(\frac{f(x_{j,i})}{p(x_{j,i})} - \langle I_j \rangle_{N_j}\right)^2.$$

Because the path samples obtained by Markov chains are weakly correlated to each other, the above definition for mutually independent samples cannot be strictly available. However, we did not find any drawbacks by ignoring the correlation between samples in practice. We defined a covariance matrix for each pixel and applied the NL-means filter for the matrix entries as performed by [RJN16]. The filter size was 7×7 in our implementation.

3. Results

The proposed method was implemented as a plug-in on PBRT v3 [PJH16] which is an open-source physics-based renderer. The following experiments were performed on a computer with two 1.70 GHz Intel Xeon E5-2603 v4 CPUs and 16 GB RAM. The computation times for rendering with 12 parallel processes were noted. The rendering results are computed with $N_1 = 128$ and $N_2 = 128$ unless otherwise specified. To fairly evaluate the performance of the proposed method, we rendered reference images with the primary sample space MLT (PSSMLT) [KSKAC02] using 20,000 path samples per pixel. We also implemented the original image-space

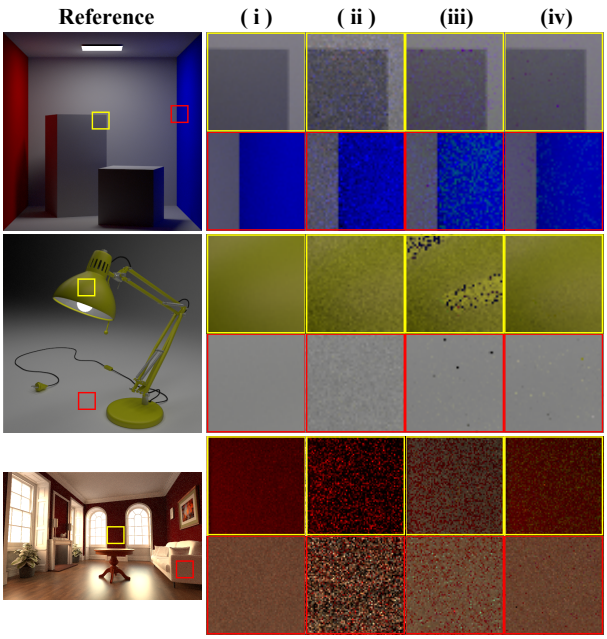

Figure 3: *Comparison of rendering results. The square regions marked in the left images (reference) are enlarged and compared among the (i) reference image, (ii) [KSKAC02], (iii) [RJN16], and (iv) proposed method.*

Table 1: *Comparison of computation times and RMS errors. These error values are obtained by averaging the values of 10 independent trials. The error values are magnified by a factor of* 10^2

Scene		[KSKAC02]	[RJN16]	Ours
Cornell box	RMSE	2.642	1.667	0.587
	Time (s)	72	76	82
Lamp	RMSE	1.080	1.595	0.847
	Time (s)	90	95	109
Living room	RMSE	16.30	1.934	1.386
	Time (s)	150	155	168

Table 2: *RMSE for the proposed method with different N_1 and N_2. The error values are computed as those in Table 1.*

(N_1, N_2)	(32, 224)	(64, 192)	(128, 128)	(256, 0)
Cornell box	0.764	0.647	0.587	**0.374**
Lamp	1.585	**0.831**	0.847	1.072
Living room	1.426	1.992	1.386	**0.489**

control variates [RJN16] using the PSSMLT method. Note that the proposed method uses the MMLT method not for considering the length of camera and light paths, as in the original paper [HKD14], but for considering the multiple importance sampling among the path spaces of two scenes. Therefore, the reference images are rendered by the PSSMLT method instead of the MMLT method.

Figure 2 depicts the results of the proposed method and its com-

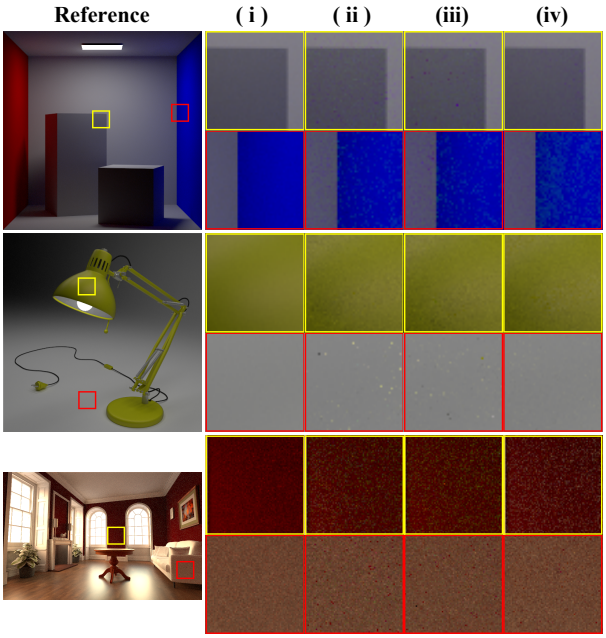

Figure 4: *Comparison of rendering results for different N_1 and N_2. The right images show the enlarged areas for (i) reference, (ii) $N_1 = 64$, (iii) $N_2 = 128$, and (iv) $N_1 = 256$. For our results, $N_1 + N_2 = 256$, and the same square regions in Fig. 3 are enlarged.*

parison to those of the image-space control variates [RJN16]. As can be seen in this figure, the proposed method and the previous method render the scenes with significantly small amounts of noise compared to the standard PSSMLT for rendering the scenes from scratch. Figure 3 shows several enlarged parts in the results. Although the results of this study and those of [RJN16] are visually similar, our results demonstrate a smaller amount of noise, particularly in the parts where the materials are edited. Simultaneously, the parts where the appearances do not change show approximately the same amount of noise with [RJN16]. The quantitative comparisons of root-mean square errors (RMSEs) and computational times of the above methods are illustrated in Table 1. As shown in the table, the proposed method achieves smaller RMSEs than the previous methods without any additional computation times over [RJN16]. Accordingly, the proposed method can effectively reduce the amount of noise with the same number of path samples. The three examples, i.e., "Cornell box," "Lamp" and "Living room," are rendered in 800×512, 512×512 and 512×512, respectively. Rendering results and comparisons of additional examples are included in the supplementary materials.

Figure 4 depicts the influence of the ratio of N_1 and N_2, i.e., the number of path samples per pixel used for the first and second steps, on the results of the proposed method. As the error values indicate in the left charts, RMSEs gradually decrease as the value of N_1 increases. In other words, the second step in which the paths are sampled based on a difference image does not contribute to reducing the errors. In contrast, as shown in the enlarged images in the left, the rendering results by $N_1 = N_2 = 128$ are most visually similar to the reference images. The reason for this contradiction is explained as follows. We can obtain the results that have more accurate pixels values in non-edited regions using large paths for the first step. The pixel area of such non-edited regions is typically smaller than that of edited regions; therefore, the overall RMSEs decrease. In contrast, when more path samples are used in the second step, the results have more accurate pixel values for edited regions. When artists check the material-edited results, it is more important to improve the appearances of such edited regions.

4. Conclusion

This paper proposed efficient path sampling for re-rendering scenes after material editing. The results demonstrated that the proposed method significantly reduces the errors and variances in the rendering results with a small number of path samples per pixel. For future work, we are interested in extending the proposed method with a multi-stage MLT [HH10] to gradually improves the quality of the difference image. In addition, we intend to implement the proposed method on GPU to enable an interactive preview of rendering results.

Acknowledgment

This study was granted in part by the Strategic Basic Research Program ACCEL of the Japan Science and Technology Agency (JPMJAC1602). Tatsuya Yatagawa was supported by a Research Fellowship for Postdoctoral Researchers of Japan's Society for the Promotion of Science (16J02280). Shigeo Morishima was supported by a Grant-in-Aid of Waseda Research Institute of Advanced Science and Engineering.

References

[GG15] GÜNTHER T., GROSCH T.: Consistent scene editing by progressive difference images. *Computer Graphics Forum 34*, 4 (2015), 41–51. doi:10.1111/cgf.12677. 1

[HH10] HOBEROCK J., HART J. C.: Arbitrary importance functions for metropolis light transport. *Computer Graphics Forum 29*, 6 (2010), 1993–2003. doi:10.1111/j.1467-8659.2010.01713.x. 4

[HKD14] HACHISUKA T., KAPLANYAN A. S., DACHSBACHER C.: Multiplexed metropolis light transport. *ACM Trans. Graph. 33*, 4 (2014), 100. doi:10.1145/2601097.2601138. 2, 3

[KSKAC02] KELEMEN C., SZIRMAY-KALOS L., ANTAL G., CSONKA F.: A simple and robust mutation strategy for the metropolis light transport algorithm. *Computer Graphics Forum 21*, 3 (2002), 531–540. doi:10.1111/1467-8659.t01-1-00703. 3

[LKL*13] LEHTINEN J., KARRAS T., LAINE S., AITTALA M., DURAND F., AILA T.: Gradient-domain metropolis light transport. *ACM Trans. Graph. 32*, 4 (2013), 95. doi:10.1145/2461912.2461943. 2

[PJH16] PHARR M., JAKOB W., HUMPHREYS G.: *Physically based rendering: From theory to implementation*, 3 ed. Morgan Kaufmann, 2016. doi:10.1016/C2013-0-15557-2. 3

[RJN16] ROUSSELLE F., JAROSZ W., NOVÁK J.: Image-space control variates for rendering. *ACM Trans. Graph. 35*, 6 (2016), 169. doi:10.1145/2980179.2982443. 1, 2, 3, 4

[SPN*14] SCHMIDT T., PELLACINI F., NOWROUZEZAHRAI D., JAROSZ W., DACHSBACHER C.: State of the art in artistic editing of appearance, lighting and material. *Computer Graphics Forum 35*, 1 (2014), 216–233. doi:10.1111/cgf.12721. 1