ISSN 0361-7688, Programming and Computer Software, 2015, Vol. 41, No. 5, pp. 253–257. © Pleiades Publishing, Ltd., 2015.
Original Russian Text © A.M. Gruzdev, V.A. Frolov, A.V. Ignatenko, 2015, published in Programmirovanie, 2015, Vol. 41, No. 5.

Practical Approach to the fast Monte-Carlo Ray-Tracing[1]

A. M. Gruzdev[a], V. A. Frolov[b], and A. V. Ignatenko[a]

[a] *Moscow State University, Moscow, 119992 Russia*

[b] *Keldysh Institute for Applied Mathematics, Russian Academy of Sciences, Miusskaya pl. 4, Moscow, 125047 Russia*

e-mail: aleks-gruzdev92@yandex.ru, vfrolov@graphics.cs.msu.ru, ignatenko@graphics.cs.msu.ru

Received January 20, 2015

Abstract—The paper proposes a new high-quality approach to fast Monte-Carlo path-tracing. The key feature of the approach is screen-space filtering with the help of additional information (depth, normal direction, etc.) of the illumination separated from material color. It allows to reach high-quality and edge-aware filtering. The proposed method yields 1–2 times speed-up without putting significant restrictions on the ray-tracing algorithm.

Keywords: path-tracing, multidimensional filtering, indirect illumination, global illumination

DOI: 10.1134/S0361768815050035

1. INTRODUCTION

Monte-Carlo path tracing [1] is a physically correct and high-quality rendering algorithm, though its major disadvantage is a slow convergence. Rendering can be sped up by computing small number of paths per pixel and removing the remaining noise with the help of filtering algorithms. The paper proposes an algorithm surpassing similar methods in filtering quality and significantly decreasing image rendering time.

2. RELATED WORKS

2.1. Irradiance Cache

Irradiance cache [2] is an approach to speed up rendering equation in a set of points. The method is supposed to work with the diffuse component of illumination. Indirect diffuse illumination changes smoothly, so it can be computed with the help of a precise algorithm in a limited set of points and interpolated to all other points.

The paper [3] discusses a speed-up approach to Monte-Carlo path tracing with the help of combination of the irradiance cache [2] and a filtering algorithm. That approach significantly reduces the amount of samples needed for creating irradiance cache using filtering, while selecting values from the cache. In the case of photon mapping, the filtering is used during the final gathering stage. The filter's weights are selected individually for each point of the cache according to geometry complexity at that point. The algorithm yields fivefold speed-up compared to the irradiance cache. It avoids artifacts peculiar to irradiance cache; however, the method has artifacts due to filtering and complicates the operation of selecting from the cache.

2.2. Adaptive Filtering

The algorithm [4] speeds up the progressive Monte-Carlo ray tracing with the help of filtering output of each iteration. The authors propose filtering with the Epanechnikov kernel [5] with varying size according to a rule of thumb depending on the density of Monte-Carlo samples. Considering only the samples density, rather than scene features, the approach preserves edges poorly.

2.3. Multidimensional Filtering

Authors of the paper [6] propose a multidimensional filter for image denoising, which is applicable to arbitrary signals processing as well. Via additional dimensions, the filter considers a large amount of additional information besides the color. The methods yields high-quality results comparable to bilateral filtering [7]; it has linear complexity and a high performance.

2.4. Wavelet Transform Filtering

The wavelet transform filtering is proposed in [8] for real-time irradiance filtering in screen space. The paper discusses an improved method of bilateral filtering with the use of the wavelet transform A-Trous [9] adding an additional weight function for edges recog-

[1] The article was translated by the authors.

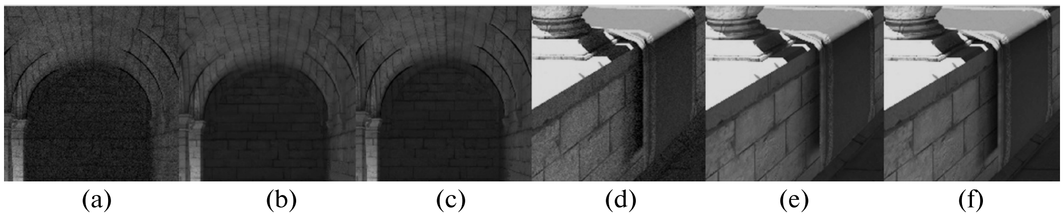

Fig. 1. Example of the proposed approach results: (a) no filtering, (b) the proposed approach (1 min), (c) groundtruth (50 min)

nition to the transform. The function is computed based on additional information about pixels: normal direction and world space position. The additional weight value in the wavelet transform depends on pixels distance by normal and position. The method preserves edges well, though it has colorful spots artifacts on homogeneous areas. GPU implementation of the proposed approach processes four images in 512 × 512 resolutions per second.

2.5. Adaptive Distribution of Monte-Carlo Samples

The paper [10] discusses an adaptive distribution method of Monte-Carlo samples in the screen space. A multidimensional distribution function is defined in the screen space by considering every sample as a multidimensional point (e.g., it includes lens coordinates and time). After anisotropic interpolation, the function is used for choosing the next samples. More samples correspond to areas of strong variation: edges, soft shadows, areas out of camera focus.

MSE minimization. An iterative method of computing Monte-Carlo samples and rendering is described in [11]. It chooses the most optimal distribution of a fixed number of Monte-Carlo samples in the screen space minimizing MSE [12] on every step. Since MSE value cannot be computed without a groundtruth image, an approximation [5] is used. After that, an image reconstruction based on the computed samples is performed: the resulted image is blurred by a set of predefined filters (Gaussian filtering with various kernel sizes) and, then, the best MSE minimizing value is selected for each pixel. Estimated error of the current iteration is used in the next one for setting up the distribution.

Non-local means denoising. An iterative approach to Monte-Carlo sampling from [13] yields a significant speed up of Monte-Carlo ray tracing surpassing the method from [11]. Every iteration consists of three steps. On the first step, a fixed number of rays are computed and the result is stored in a buffer. Then, a non-local means filtering [14] of the buffer is performed on the second step. On the third step, a discrepancy per pixel is computed. On the first step of the next iteration, the rays distribution is chosen based on the given discrepancy. The first iteration uses a uniform distribution. The computation costly non-local means filtering limits performance and rendering one image in the resolution 1024 × 1024 takes 8-16 seconds with GPU acceleration.

2.6. Guided Filtering

A guided filtering based on additional information per pixel is proposed in [15]. The guided filtering may surpass bilateral filtering in the case of lucky choice of the additional information. The paper [16] discusses using the guided filtering in Monte-Carlo ray tracing. The proposed method separates direct and indirect illuminations. The indirect illumination is processed by the guided filtering with normal map, depth map, and an additional color buffer as the additional information. The additional color buffer is computed by tracing additional guide rays through pixel centers. The proposed method yields better quality and performance than the cross-bilateral filtering [7] and A-Trous filtering [8].

2.7. Probabilistic Filtering

The authors of [17] propose an effective samples filtering considering the Monte-Carlo ray tracing as a Poisson process. After clustering all samples of each pixel by ray depth, the Poisson process assumption allows comparing samples distributions in clusters of neighbor pixels, so that the filter can reuse illumination from the neighboring pixels. Also, the filter uses normal and depth information in order to preserve edges. In the postprocessing of images with low amount of rays per pixel, the proposed method surpasses the cross-bilateral filtering [7] in quality.

A multidimensional filtering considering stochastic relation between random variables in Monte-Carlo ray tracing and a computed sample value is proposed in [18]. Each sample is considered as a multidimensional vector consisting of random variables and some additional information about the scene (e.g., distribution of samples over the lens). If a noise on the image is caused by the random distribution of sampling over the lens, then there exists a functional relation between the distribution of lens points and the image noise. Knowledge of that relation allows reducing weights of affected by the noise pixels in the filter kernel. The approach uses the bilateral filtering [7] with parameters based on the mutual information between

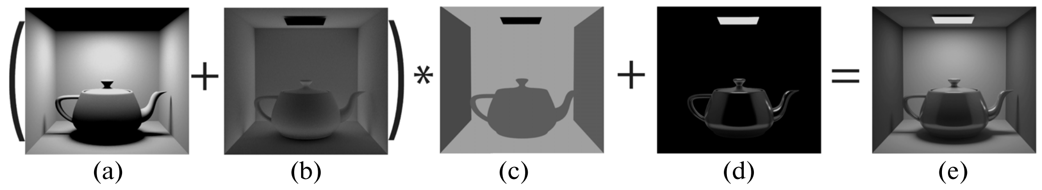

Fig. 2. The final image computing: (a) the direct illumination intensity; (b) the indirect illumination intensity; (c) the texture color; (d) the reflected illumination; (e) the final image.

the additional information distribution and random variables at the point.

2.8. Summary

The irradiance cache [2, 3] solves the task of speeding up images rendering, but it has artifacts and requires significant changes of the rendering algorithm.

The papers [10, 11, 13] discuss a more efficient distribution of samples in the screen space rather than the uniform one, though they require changing ray tracer as well.

The filtering approaches allow one to avoid significant changes of the ray tracing algorithm. The methods [4, 7] process images without any information about the scene geometry, so they do not preserve edges. Similar approaches [6, 8, 15, 17, 18] use the additional information about the scene in order to improve filtering quality on edges. The method [6] combines the idea of the guided filtering [15] and the non-local means filtering [14] improving quality on homogenous areas.

3. SUGGESTED APPROACH

This paper proposes an application of ideas from [6] to the task of Monte-Carlo path tracing. The method is based on the assumption that the scene image F can be represented in the form of equation (3.1), see also Fig. 2.

$$F = (D + I) * T + R, \quad (3.1)$$

where D is direct illumination intensity, I is indirect illumination intensity, T is texture color, and R is illumination reflected from mirror surfaces.

In such a way, the ray tracer should be able to compute those components separately. The key idea of the approach is separating the illumination intensity from pixel color. It allows us to separate filtering the noisy illumination intensity from texture pattern, i.e., reuse neighbor pixels with the same illumination but different texture color. Hence, the proposed filter does not blur textures in contrast to [6] and does not interpret texture pattern as an edge.

The proposed approach can be separated into the following three steps:

1. Illumination preprocessing: transferring pixels to the non-local means space [16, 6].

2. Filtering illumination. A separable filter from [6] is applied using the color in non-local means space, normal map, and depth map as the additional information.

3. Combining illumination and generating the final image according to formula (3.1).

The first two steps are applied to the direct illumination (D) and indirect illumination (I). After that, the third step is made using the outputs of the previous steps. During rendering image according to formula

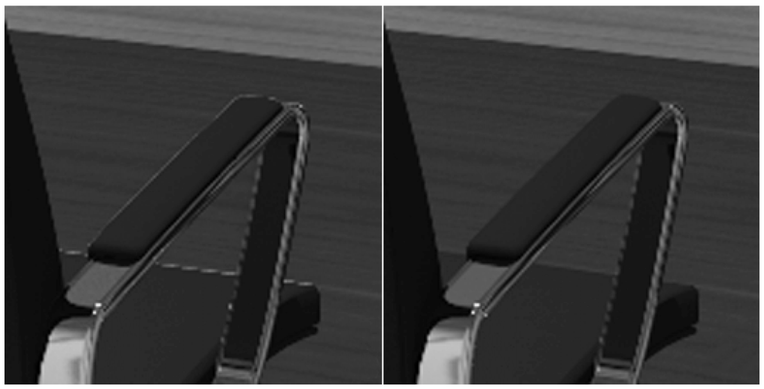

Fig. 3. Removing the artifacts on edges.

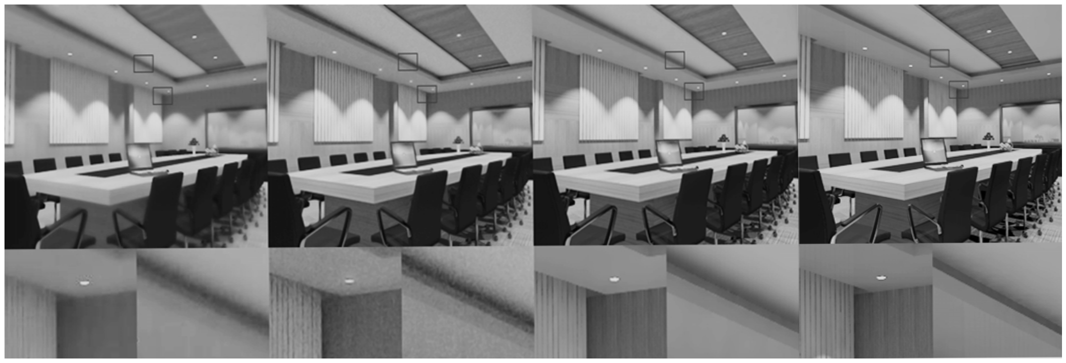

Fig. 4. Comparison with the alternative methods: (a) Adobe Photoshop (bilateral filtering); (b) Adaptive manifolds [6]; (c) The proposed approach; (d) The irradiance cache [2].

Table 1. Comparison of the proposed approach with the alternative methods by the MSE metric

MSE	Adaptive Manifolds filtering	Adobe Photoshop denoising	Proposed algorithm	Irradiance cache
"Meeting room"	7.73	7.15	1.50	2.05
"Cornel box"	8.12	7.05	1.64	1.58
"Bedroom"	6.73	6.11	3.12	2.65

Table 2. Performance comparison with the alternative methods

Header	Synthesis of input information	Filtering on central processor	Toatal time on central processor	Filtering on graphical processor	Toatal time on graphical processor	Irradiance cache	Groundtruth synthesis
"Cornel box"	53.40c	6.42c	59.82c	0.16c	53.56c	24.02c	611.13c
"Meeting room"	56.83c	6.40c	63.23c	0.16c	56.99c	98.23c	629.04c
"Bedroom"	70.21c	6.53c	76.74c	0.17c	70.38c	111.41c	716.22c
"Sponza"	69.04c	6.59c	75.63c	0.16c	69.20c	26.83c	643.65c

(3.1), artifacts on edges may appear because assumption (3.1) is not true on the edges. The reason is the fact that the pixels belonging to the edges include illumination from several different surfaces.

3.1. Antialiasing

A technique similar to the MSAA method [19] is used to eliminate the artifacts (Fig. 3). In order to remove aliasing artifacts, the filter selects sets of edge pixels G based on the additional information. They differ from neighboring pixels in normal direction and depth value. For each pixel from G, the ray tracer generates 16 additional sub-pixels (4×4) containing the normal and depth. If a pixel belongs to the set G, then it is filtered 16 times with different additional information values from the sub-pixels, and the result is averaged.

3.2. Reflected Illumination

The reflected illumination is not processed by the proposed filter because it requires another set of additional information. However, the algorithm can be applied recursively by reflected ray depth; i.e., the algorithm is applied in the same way to the reflected illumination: saving the illumination intensity, computing the additional information for the reflected rays. The processed illumination of the previous reflection level is used in formula (3.1) for the current level.

4. RESULTS

The test data set was prepared in the resolution 1024×768. The noisy images were rendered with 256 paths per pixel, and the groundtruth images, with 4096 paths per pixels.

Testing results are shown in Tables 1 and 2. The proposed approach has a significant advantage over Adaptive Manifold filtering [6] and Adobe Photoshop denoising, though its results are equal to the irradiance cache results [2] (Fig. 4). Nevertheless, the proposed method has no typical artifacts of the irradiance cache in the corners and is much easier to implement.

REFERENCES

1. Ritschel, T., Dashsbacher, C., Grosch, T., and Kautz, J., The state of the art in interactive global illumination, *Comput. Graphics Forum,* 2012, vol. 31, pp. 160–188.
2. Pharr, M. and Humphreys, G., *Physically Based Rendering: From Theory to Implementation*, San Francisco: Morgan Kaufmann, 2004.
3. Kontkanen, J., Rasanen, J., and Keller, A., Irradiance filtering for Monte Carlo ray tracing, *Monte Carlo and Quasi-Monte Carlo Methods*, Berlin: Springer, 2006, pp. 259–272.
4. Suykens, F. and Willems, Y., Adaptive filtering for progressive Monte-Carlo image rendering, *Proc. WSCG 2000*, 2000.
5. Silverman, B., *Density Estimation for Statistics and Data Analysis*, London: Chapmann and Hall, 1986.
6. Gastal, E. and Oliveira, M., Adaptive manifolds for real-time high-dimensional filtering, *ACM Trans. Graphics*, 2012, vol. 31, no. 4, article 33.
7. Kornprobst, P. and Tumblin, J., *Bilateral Filtering: Theory and Applications*, Now Publishers, 2009.
8. Dammertz, H., Sewtz, D., Hanika, J., and Lensch, H., Edge-avoiding A-Trous wavelet transform for fast global illumination filtering, *Proc. of the Conf. on High Performance Graphics*, 2010, pp. 67–75.
9. Holschneider, M., Kronald-Martinet, R., Morlet, J., and Tchamitchian, P., *A Real-Time Algorithm for Signal Analysis with the Help of the Wavelet Transform*, New York: Springer, 1989.
10. Hachicuka, T., Wojciech, J., Weistroffer, R., Dale, K., Humphreys, G., Zwicker, M., and Jensen, H., Multidimensional adaptive sampling and reconstruction for ray tracing, *ACM Trans. Graphics,* 2008, vol. 27, no. 3, article 33.
11. Rousselle, F., Knaus, C., and Zwicker, M., Adaptive sampling and reconstruction using greedy error minimization, *Proc. of 2011 SIGGRAPH Asia Conf.* (New York, 2011), New York: ACM, 2011, pp. 159:1–159:12.
12. Lehmann, E. and Casella, G., *Theory of Point Estimation*, New York: Springer, 1998.
13. Rousselle, F., Knaus, G., and Zwicker, M., Adaptive rendering with non-local means filtering, *ACM Trans. Graphics*, 2012, vol. 31, no. 6, p. 195.
14. Buades, A., Coll, B., and Morel, J.M., A non-local algorithm for image denoising, *Proc. of IEEE Int. Conf. on Computer Vision and Pattern Recognition*, 2005, vol. 2, pp. 60–65.
15. He, K., Sun, J., and Tang, X., Guided image filtering, *Computer Vision-ECCV 2010*, Berlin: Springer, 2010, pp. 1–14.
16. Bauszat, P., Eisemann, M., and Magnor, M., Guided image filtering for interactive highquality global illumination, *Comput. Graphics Forum,* 2011, vol. 30, no. 4, pp. 1361–1368.
17. Doidge, I. and Jones, M., Probabilistic illumination-aware filtering for Monte Carlo rendering, *The Visual Comput.*, 2013, vol. 29, nos. 6–8, pp. 707–716.
18. Sen, P. and Darabi, S., On filtering the noise from the random parameters in Monte-Carlo rendering, *ACM Trans. Graphics*, 2012, vol. 31, no. 3.
19. Leler, W., Human vision, anti-aliasing, and the cheap 4000 line display, *ACM SIGGRAPH Comput. Graphics*, 1980, vol. 14, no. 3, pp. 308–313.