

# Noise Dependence on the Number of Rays in Bidirectional Stochastic Ray Tracing with Photon Maps

S. V. Ershov<sup>a,\*</sup>, E. D. Birukov<sup>a</sup>, A. G. Voloboy<sup>a</sup>, and V. A. Galaktionov<sup>a,\*</sup>

<sup>a</sup> Keldysh Institute of Applied Mathematics, Russian Academy of Sciences,  
Moscow, 125047 Russia

\*e-mail: voloboy@gin.keldysh.ru

Received December 19, 2020; revised December 25, 2020; accepted January 14, 2021

**Abstract**—The classical Monte Carlo ray tracing is a powerful technique for modeling almost all effects in geometric optics; however, it can be prohibitively slow in many cases, such as generation of images seen by an objective or camera with a point aperture. For this reason, numerous modifications of this technique are used, among which is the bidirectional stochastic ray tracing with photon maps. A drawback of all stochastic methods is the undesirable noise. The noise level, i.e., the variance of the pixel luminance calculated for one iteration step, depends on various parameters, such as the number of rays traced from the light source and from the camera, the method of merging their trajectories, the integration sphere radius, etc. The choice of the optimal parameters makes it possible to minimize the noise level for the given computation time. This is the topic of the current paper. It is shown that the variance of the pixel luminance is the sum of three functions scaled by the reciprocal of the number of rays traced from the light source and from the camera, where the functions themselves are independent on the number of rays. Therefore, given these functions, one can predict the noise for any number of rays and thus find the optimal set of parameters. The calculation of these functions based on the data obtained by ray tracing is a nontrivial problem. The paper proposes a practical method for their calculation, and demonstrates that a single such calculation is able to predict the variance for an arbitrary number of rays. Therefore, the noise can be minimized due to the optimal choice of the number of rays.

DOI: 10.1134/S036176882103004X

## 1. INTRODUCTION

Presently, modeling of light propagation is widely used in realistic computer graphics and for designing new materials and optical systems [1]. It is also used in architectural, automobile and aircraft design. If wave effects may be neglected in a problem, then stochastic ray tracing methods are a good choice. This group of methods mainly includes the simulation of radiation transport using the Metropolis method [2] and stochastic ray tracing [3]. The classical forward ray tracing beginning from the light source is inefficient for image generation, and for this reason it is replaced by bidirectional modifications of this method [4–6]. Among them, we consider the so-called bidirectional stochastic ray tracing with photon maps (BDPM – Bidirectional Photon Mapping) [5, 7]. A drawback of all stochastic methods is that they produce noisy results. Therefore, the noise reduction problem is always important, and it is considered in many works, e.g., see [8–10].

The level of noise in BDPM mainly depends on the random scattering of the forward and backward rays, on the choice of the vertex for their merging (or, in other words, on the vertex of the camera ray trajectory

at which photon maps are used for estimating the luminance), and, finally, on the number of forward and backward rays traced in one iteration step. The majority of studies is devoted to the first two issues (e.g., [9–12]), and the number of rays got less attention. However, this is an important factor, and it often happens that the number of forward rays is already redundant and its further increase only increases the computation time but does not reduce the noise. In other cases, it may happen that the number of forward rays is indeed critical, while the number of traced backward rays is redundant.

This situation is illustrated in Table 1, where the dependence of variance (RMS) after a fixed computation time on the number of forward rays (from the light source)  $N_F$  and backward rays (from the camera through the pixel)  $N_B$  is shown. The scene description can be found in the section *Results*. In this example, there is almost no dependence on the number of backward rays  $N_B$  (indicated in columns), while there is a minimum of noise level at  $N_F \approx 3000$  depending on the number of rays emitted from the source. Here we see that the general claim that *the more rays the lower the noise* does not hold.

**Table 1.** Mean RMS over the region of the image marked by a red box in Fig. 2 after 1000 seconds of computation for various numbers of rays  $N_F$  and  $N_B$

$N_F \backslash N_B$	5	10	25	100
100	0.135	0.135	0.135	0.135
1000	0.101	0.101	0.101	0.101
3000	0.099	0.099	0.099	0.099
10000	0.101	0.101	0.101	0.101
30000	0.110	0.110	0.109	0.109
100000	0.136	0.136	0.135	0.135

It is usually difficult to predict which fraction of the forward and backward rays is optimal: however, a good choice can speed up the computations by several fold.

In this paper, we consider the optimization problem. In [8], a general rule determining noise in the BDPM was derived. It states that the variance of the pixel luminance is the sum of three components scaled by the reciprocal of the number of rays. These three components are independent of the number of rays and, therefore, they can be calculated once and then used for predicting the dependence of the noise level on the number of rays traced in one iteration step. In this way, we are going to predict the optimal number of rays. In other words, if we know these three components, then we are able to predict the level of noise for any number of rays.

Even though the mathematical expression is trivial, these values are difficult to calculate in the process of ray tracing. In this paper, we describe a method for their effective calculation, and it is shown how they can be used for choosing the optimal number of rays.

## 2. NOISE IN THE BIDIRECTIONAL RAY TRACING WITH PHOTON MAPS

In the BDPM, the computations are typically iterative. At each iteration step,  $N_F$  light paths and  $N_B(p)$  paths from the camera are traced for each pixel  $p$ . Next, we check each pair, and if the light path goes sufficiently close to a vertex on the path from the camera, we *join* them to obtain the complete path from the camera to the light source. Next, we calculate the contribution of this *joined* path to the pixel luminance and increase the pixel luminance by this magnitude. The accumulated sum divided by  $N_F N_B(p) N_I$  (where  $N_I$  is the number of iterations) converges to the luminance expectation. This mean value is independent of  $N_F$  and  $N_B(p)$ , but its variance (noise) can vary significantly. For example, if there are too many light ray trajectories and the ray trajectories issued from the camera are few, then tracing the redundant light rays only wastes time without improving the image quality; therefore, it is reasonable to reduce the number of the redundant rays.

The optimal number of rays (which is common for all light paths in each scene and individual for each pixel for the paths from the camera) is the number that minimizes the noise after a fixed computation time. To solve this minimization problem, we need to know how the variance of the value obtained in one iteration step depends on the number of rays. This dependence is described by a simple algebraic formula [8]; however, its terms are not easy to find numerically.

Here and below, we always deal with values for one specific pixel  $p$ ; for this reason, we will omit the pixel indication and will write  $N_B$  instead of  $N_B(p)$ .

It was shown in [8] that the luminance variance at a given pixel *calculated during one iteration step* is

$$V = \frac{1}{N_F N_B} (\langle\langle C^2 \rangle\rangle - \langle\langle C \rangle\rangle^2) + \frac{1 - N_F^{-1}}{N_B} (\langle\langle C \rangle_F^2 \rangle_B - \langle\langle C \rangle\rangle^2) + \frac{1 - N_B^{-1}}{N_F} (\langle\langle C \rangle_B^2 \rangle_F - \langle\langle C \rangle\rangle^2)$$

where  $C(X^{(F)}, X^{(B)})$  is the contribution of the merged path from the source  $X^{(F)}$  and the path from the camera  $X^{(B)}$  to the pixel luminance, its mean  $\langle\langle C \rangle\rangle$  clearly coincides with the limiting (exact) pixel luminance  $L$ , and  $\langle\cdot\rangle_F$  and  $\langle\cdot\rangle_B$  denote averaging over the ensemble of the paths issued from the light source and the camera, respectively.

Note that this functional form is independent of whether or not the multiple importance sampling [13] is used and is even independent of whether vertex merging or vertex connection by an additional trajectory segment [9] is used; however, the quantities  $\langle\langle C^2 \rangle\rangle$ ,  $\langle\langle C \rangle_F^2 \rangle_B$ , and  $\langle\langle C \rangle_B^2 \rangle_F$  depend on the computation strategy. In any case, they are independent of the number of rays, but they can depend on the radius of the integration sphere (i.e., on the radius of vertex merging).

## 3. CALCULATION OF THE COEFFICIENTS IN NOISE FORMULA

The calculation of  $\langle\langle C^2 \rangle\rangle$  is trivial (as in the calculation of the sample variance): by adding  $C$  to the accumulated pixel luminance, we also add  $C^2$  to  $\langle\langle C^2 \rangle\rangle$  accumulated in this pixel. That is, at each iteration step we calculate

$$C \equiv \frac{1}{N_B N_F} \sum_{j=1}^{N_B} \sum_{i=1}^{N_F} C^2(X_i^{(F)}, X_j^{(B)}),$$

and the mean of this value  $C_{\text{avg}}$  over the iterations converges to the desired value:  $C_{\text{avg}} \rightarrow \langle\langle C^2 \rangle\rangle$ . However, this method is certainly inapplicable for  $\langle\langle C \rangle_F^2 \rangle_B$  and  $\langle\langle C \rangle_B^2 \rangle_F$ .

Indeed, assume that  $N_F$  is so large that, for each ray issued from the camera, the sum  $\frac{1}{N_F} \sum_{i=1}^{N_F} C(X_i^{(F)}, X_j^{(B)})$  gives a good approximation to  $\langle C \rangle_F(X_j^{(B)})$ . At the same time, the number of rays passing through the same pixel  $N_B$  is small; therefore the averaging  $\langle C \rangle_F^2(X_j^{(B)})$  over this small number of rays is insufficient. However, here we can add averaging over iterations because at each iteration step we have a random independent set of rays issued from the camera, and this yields a good estimate of  $\langle \langle C \rangle_F^2 \rangle_B$ .

Certainly, this is just a thought experiment, and this method cannot be practically used because the number of rays issued from the light source is not large enough for providing a sufficiently accurate estimate of  $\langle C \rangle_F(X_j^{(B)})$  within one iteration step. Averaging over iterations in this case is extremely difficult because this quantity should be calculated for each trace  $X_j^{(B)}$  issued from the camera, and it occurs only once. It is clear that we actually will hit a neighborhood of this path regularly, so that if we will calculate and store the grid function  $X_j^{(B)}$  (on a sufficiently fine grid), the aim will be achieved. However, this is very inefficient since the space dimension, which equals the maximum length of the trace issued from the camera, is fairly large.

However strange it may seem, the solution is simple. Assume for convenience that the number of rays issued from the camera and from the light source is independent of the iteration step. To calculate  $\langle \langle C \rangle_F^2 \rangle_B$ , we divide the whole set  $N_F$  of rays issued from the source into two parts (not necessarily equal) consisting of  $N_{F_1}$  and  $N_{F_2}$  rays, respectively, and calculate at each iteration step

$$B \equiv \frac{1}{N_B} \sum_{j=1}^{N_B} \left( \left( \frac{1}{N_{F_1}} \sum_{i \in \text{half}_1} C(X_i^{(F)}, X_j^{(B)}) \right) \times \left( \frac{1}{N_{F_2}} \sum_{i \in \text{half}_2} C(X_i^{(F)}, X_j^{(B)}) \right) \right).$$

The mean  $B_{\text{avg}}$  of this quantity over the iterations is nothing more nor less than the mean over the ensemble of  $N_F$  rays issued from the source and  $N_B$  rays issued from the camera. These two means are independent and, therefore, they commute. Let us first average  $B$  over the ensemble of rays issued from the source:  $B_{\text{avg}} = \langle \langle B \rangle_F \rangle_B$ . Since the rays in the first and second parts of this ensemble are obviously independent, the mean of the product of the two internal sums is the product of the means over these halves:

$$\langle B \rangle_F \equiv \frac{1}{N_B} \sum_{j=1}^{N_B} (\langle C \rangle_{F_1}(X_j^{(B)}) \langle C \rangle_{F_2}(X_j^{(B)})).$$

Furthermore, since the statistical characteristics of the rays in the first and second parts are certainly iden-

tical, the means over the halves of the ensemble and over the entire ensemble are also identical:

$$\langle B \rangle_F \equiv \frac{1}{N_B} \sum_{j=1}^{N_B} \langle C \rangle_F^2(X_j^{(B)})$$

The remaining averaging over the ensemble of the rays issued from the camera yields

$$B_{\text{avg}} = \langle \langle B \rangle_F \rangle_B = \langle \langle C \rangle_F^2 \rangle_B$$

Thus, the second coefficient in the noise formula can be calculated to an arbitrary accuracy by the sequence of iterations even if each iteration contains a small number of rays.

The computation of  $\langle \langle C \rangle_B^2 \rangle_F$  is organized similarly by dividing into two parts the set of rays issued from the camera and passing through the given pixel and by calculating at each iteration step the quantity

$$F \equiv \frac{1}{N_F} \sum_{i=1}^{N_F} \left( \left( \frac{1}{N_{B_1}} \sum_{j \in \text{half}_1} C(X_i^{(F)}, X_j^{(B)}) \right) \times \left( \frac{1}{N_{B_2}} \sum_{j \in \text{half}_2} C(X_i^{(F)}, X_j^{(B)}) \right) \right)$$

Its mean  $F_{\text{avg}}$  over the iterations converges to  $\langle \langle C \rangle_B^2 \rangle_F$ .

Note that the set of rays issued from the camera may be partitioned into unequal parts. However, this is senseless because the sums over these two parts appear in the formulas symmetrically, and there is no reason to improve one of the parts at the expense of the other. In practice, it is simpler to use the partition into even and odd rays.

Now, knowing the RMS means and the limiting pixel luminance  $L$ , we can calculate the variance of the contribution of one iteration to this luminance as

$$V = \frac{1}{N_F N_B} (C_{\text{avg}} - L^2) + \frac{1 - N_F^{-1}}{N_B} (B_{\text{avg}} - L^2) + \frac{1 - N_B^{-1}}{N_F} (F_{\text{avg}} - L^2); \quad (1)$$

after  $N_I$  iterations, the variance will be  $\frac{V}{N_I}$ .

Note that  $C_{\text{avg}}$ ,  $B_{\text{avg}}$ , and  $F_{\text{avg}}$ , as well as  $L$ , depend on the pixel but are independent of the number of rays. However, as has been mentioned above,  $C_{\text{avg}}$ ,  $B_{\text{avg}}$ , and  $F_{\text{avg}}$  can depend on the radius of the integration sphere. This is a very important property because it makes it possible to predict the noise for an arbitrary number of rays. Therefore, we may make one preliminary calculation for a certain, maybe even a very poor choice of the number of rays, find  $C_{\text{avg}}$ ,  $B_{\text{avg}}$ ,  $F_{\text{avg}}$ , and  $L$ , and then calculate the optimal number of rays (in the general case,  $N_B$  will be specific for each pixel), which ensures the minimal noise.

#### 4. FEATURES OF DIRECT LIGHTING

In the BDPM method, the direct illuminance can be taken into account in two different ways. In the first method, all the rays issued from the source—both direct (not scattered) and indirect (scattered) are processed uniformly. That is, the direct illuminance is also taken from the photon maps. In the second method, the direct illuminance is calculated straightforwardly: each vertex of the path from the camera is connected to the light source, its luminance is calculated along this segment, and sum with the luminance estimate is taken from the photon maps [14] (formula (2) for  $w_0 = 0$  and  $w_1 = 1$ ). As a rule, the second method is more effective because in it the direct luminance component does not contain the noise of the photon maps.

It is clear that in both cases the noise is described by formulas of Section 3. Indeed, if all the luminance components are taken from the photon maps as in the first case, then the number of nonzero terms  $C(X_i^{(F)}, X_j^{(B)})$  is much less than the maximal number  $N_F N_B N_p$  (where  $N_p$  is the number of pixels in the image) because the probability of trace “intersection” is low. For this reason, the computation of the sums in  $C$ ,  $B$ , and  $F$  will be efficient.

In the second method, the situation is different; indeed, in this case *each* path from the camera makes certain contribution due to direct illuminance. Now, *all*  $C(X_i^{(F)}, X_j^{(B)})$  are nonzero, even if  $X_i^{(F)}$  and  $X_j^{(B)}$  do not intersect. The computation of the sums in  $C$ ,  $B$ , and  $F$  becomes two expensive. Therefore, it is reasonable to slightly transform the second method to simplify computations.

More precisely, to avoid this effect, we should separate the contribution of the direct illuminance in the sums because it is independent of the ray issued from the light source and may, therefore, be taken out of the summation sign:

$$C(X_i^{(F)}, X_j^{(B)}) = C^{(I)}(X_i^{(F)}, X_j^{(B)}) + C^{(0)}(X_j^{(B)})$$

where  $C^{(I)}(X_i^{(F)}, X_j^{(B)})$  is the contribution of the intersection of the camera and indirect rays issued from the source and  $C^{(0)}(X_j^{(B)})$  is the contribution of the direct illuminance obtained by connecting the node on the camera ray with the light source.

Now, we write  $C$  as

$$C = \frac{1}{N_B N_F} \sum_{j=1}^{N_B} \sum_{i=1}^{N_F} (C^{(I)}(X_i^{(F)}, X_j^{(B)}) + C^{(0)}(X_j^{(B)}))^2$$

$$\begin{aligned} &= \frac{1}{N_B N_F} \sum_{j=1}^{N_B} \sum_{i=1}^{N_F} ((C^{(I)}(X_i^{(F)}, X_j^{(B)}))^2 \\ &\quad + 2C^{(I)}(X_i^{(F)}, X_j^{(B)})C^{(0)}(X_j^{(B)}) \\ &\quad + \frac{1}{N_B} \sum_{j=1}^{N_B} (C^{(0)}(X_j^{(B)}))^2 \end{aligned}$$

Here, the sums contain the same number of non-zero terms as in the absence of direct illuminance (i.e., for  $C^{(0)} = 0$ ).

Next, we transform the expression for  $B$  as

$$\begin{aligned} B &= \frac{1}{N_B} \sum_{j=1}^{N_B} \left( \left( \frac{\sum_{i \in \text{half}_1} C^{(I)}(X_i^{(F)}, X_j^{(B)})}{N_{F_1}} + C^{(0)}(X_j^{(B)}) \right) \right. \\ &\quad \left. \times \left( \frac{\sum_{i \in \text{half}_2} C^{(I)}(X_i^{(F)}, X_j^{(B)})}{N_{F_2}} + C^{(0)}(X_j^{(B)}) \right) \right) \\ &= \frac{1}{N_B} \sum_{j=1}^{N_B} B_1(X_j^{(B)}) B_2(X_j^{(B)}) \end{aligned}$$

where

$$B_k(X_j^{(B)}) \equiv C^{(0)}(X_j^{(B)}) + \frac{1}{N_{F_k}} \sum_{i \in \text{half}_k} C^{(I)}(X_i^{(F)}, X_j^{(B)})$$

Similarly,

$$\begin{aligned} F &= \frac{1}{N_F} \sum_{i=1}^{N_F} ((F_1^{(I)}(X_i^{(F)}) + F_1^{(0)})(F_2^{(I)}(X_i^{(F)}) + F_2^{(0)}) \\ &= \frac{1}{N_F} \sum_{i=1}^{N_F} ((F_1^{(I)}(X_i^{(F)})F_2^{(I)}(X_i^{(F)}) \\ &\quad + F_1^{(I)}(X_i^{(F)})F_2^{(0)} + F_2^{(I)}(X_i^{(F)})F_1^{(0)}) + F_1^{(0)}F_2^{(0)} \end{aligned}$$

where

$$\begin{aligned} F_k^{(I)}(X_i^{(F)}) &\equiv \frac{1}{N_{B_k}} \sum_{i \in \text{half}_k} C^{(I)}(X_i^{(F)}, X_j^{(B)}) F_k^{(0)} \\ &\equiv \frac{1}{N_{B_k}} \sum_{i \in \text{half}_k} C^{(0)}(X_j^{(B)}) \end{aligned}$$

Now, all the sums contain the same number of non-zero terms as in the absence of direct illuminance (i.e., for  $C^{(0)} = 0$ ); therefore, the computations are fairly efficient. The computation time is approximately the same as in the calculation of the image luminance itself.

#### 5. RESULTS

As an example, we used the well-known benchmark scene Cornell Box. An isotropic point light source is located slightly below the ceiling center. All the surfaces are Lambertian with the albedo 0.5. The method BDPM was used without multiple importance sampling, and the ray issued from the camera was forcefully interrupted after the second diffuse intersection. The integration sphere radius was

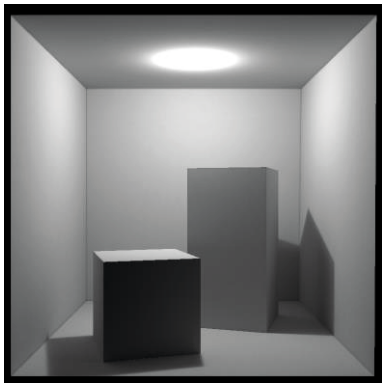


Fig. 1. Virtual photo of the model Cornell Box scene.

1/120 of the scene size. The direct luminance was calculated by connecting the surface point to the light source rather than taken from photon maps.

The image obtained by the virtual camera is shown in Fig. 1.

The three terms of the noise were calculated as described in Sections 3 and 4, and the total variance of the contribution to the luminance in one iteration was found by formula (1). For comparison, it was also calculated as sample variance over a series of iterations, which is possible because the contributions of different iterations are independent.

The noise amplitude, i.e.,  $\sqrt{V}$ , calculated using these two methods is shown in Fig. 2. It is seen that the results are almost undistinguishable.

To illustrate the contribution of all three terms in formula (1) to the total variance, they are shown in Fig. 3 as the (R, G, B) components of the image. More precisely,

the quantities  $\sqrt{\frac{1}{N_F N_B} (C_{\text{avg}} - L^2)}$ ,  $\sqrt{\frac{1 - N_F^{-1}}{N_B} (B_{\text{avg}} - L^2)}$ , and  $\sqrt{\frac{1 - N_B^{-1}}{N_F} (F_{\text{avg}} - L^2)}$  are written to the R, G, and B channels of the image, respectively.

It is seen that the image in Fig. 3 is mainly green, i.e., the second component is dominant. Therefore, to reduce the noise in this case, we should mainly reduce this dominant second term, i.e., increase  $N_B$  because  $C_{\text{avg}}$ ,  $B_{\text{avg}}$ ,  $F_{\text{avg}}$ , and  $L$  are independent of the number of rays. The increase of the number of rays  $N_F$  issued from the light source has almost no effect and only slows down the computation. The result for  $N_B$  increased by a factor of three is shown in Fig. 3b.

In practice, it is not so much the variance of the contribution of one iteration  $V$  that matters as the variance of the luminance after a given computation time  $T$ . During this time,  $N_I = \frac{T}{\tau}$  iteration steps will be performed, and the variance of the resulting image will be

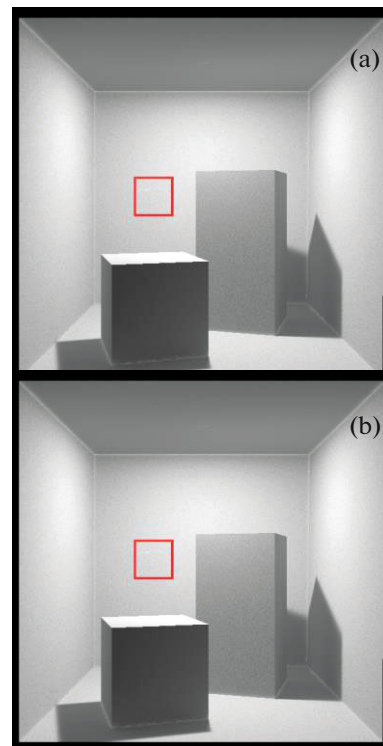
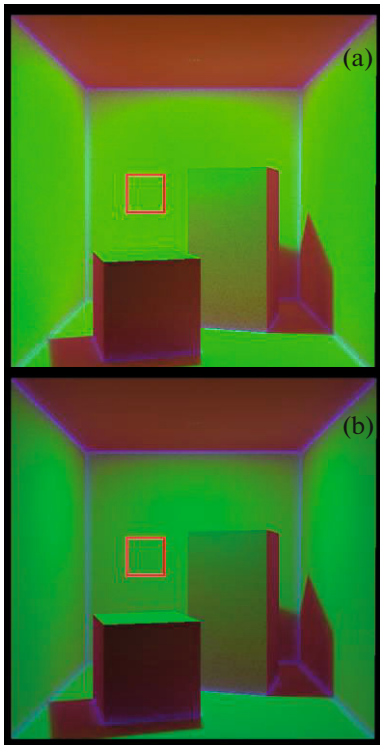


Fig. 2. Noise  $\sqrt{V}$  for  $N_B = 25$  and  $N_F = 1000$ —(a, above) benchmark value found as the sample variance over the sequence of iterations; (b, below)  $V$  is found by formula (1). The mean RMS over the region within the red box in the upper image is 1.2767 and 1.2769 in the lower image.

$V_T = \frac{\tau V}{T}$ . While  $V$  monotonically decreases with increasing number of rays, the time taken by one iteration step monotonically increases. These two “opposing factors” provide the optimum in the number of rays, as shown in Fig. 4 and Table 2. They illustrate the behavior of the mean RMS (i.e.  $\sqrt{V_T}$ ) over the red box for the benchmark scene under consideration, depending on the number of rays per one iteration step for  $T = 1000$  s. Note that the number of rays issued from the camera was the same for all pixels (not only within the red box).

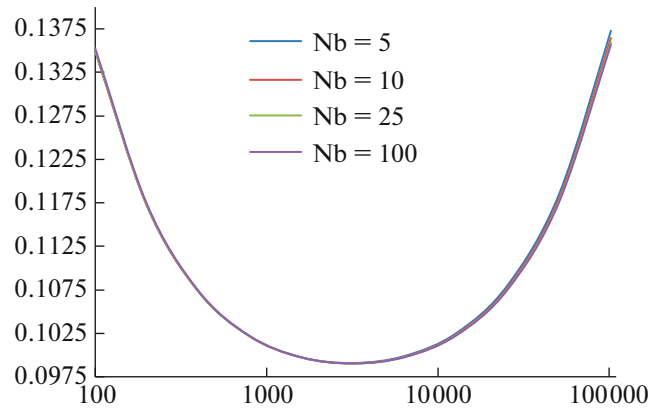
It is clearly seen in Fig. 4 that the dependence on the number of rays issued from the camera (shown by plots of different colors) is very weak in this example. On the contrary, the dependence on the number of rays issued from the light source (plotted on the horizontal axis) has a minimum in a neighborhood of  $N_F = 3100$ . The exact value of this minimum can slightly vary depending on the number of rays issued from the camera. However, this minimum is very flat; i.e., RMS is almost invariable in a wide range of  $N_F$  variation, and its growth becomes noticeable only when the deviation from the optimum is by several orders of magnitude.



**Fig. 3.** Contributions of  $\sqrt{\frac{1}{N_F N_B} (C_{\text{avg}} - L^2)}$ ,  $\sqrt{\frac{1 - N_F^{-1}}{N_B} (B_{\text{avg}} - L^2)}$ , and  $\sqrt{\frac{1 - N_B^{-1}}{N_F} (F_{\text{avg}} - L^2)}$  to the total noise written in the color channels R, G, and B of the image—(a, above)  $N_B = 25$ , (b, below)  $N_B = 100$ ; in both cases  $N_F = 1000$ . The mean values over the region within the red box in the upper image are (0.3812065, 1.21883, 0.01907) and (0.19602, 0.610129, 0.0204105) in the lower image; thus, the first two components decreased by a factor of  $\sqrt{4}$ . The total RMS, i.e.  $\sqrt{V}$ , averaged over the red box is 1.27689 in the upper image and 0.639361 in the lower image.

Note that the smooth curve in Fig. 4 is constructed on the basis of a lot of points obtained by calculating the RMS over one iteration by the *exact* formula (1), and the time taken by one iteration step is found by the approximate formula  $\tau = \alpha N_F + \beta N_B + \gamma N_B N_F$  the coefficients in which are obtained by fitting to the data in Table 1. This approximation works very well.

More detailed information is shown in Table 2 for the case  $N_B = 25$ . The time  $\tau$  of one iteration step is composed of the time spent on tracing the rays issued from the camera  $\tau_{\text{BMCRT}}$ , tracing the rays issued from the light source  $\tau_{\text{FMCRT}}$ , and the time spent on merging these traces  $\tau_{\text{merge}}$ . In addition to the total noise  $\sqrt{V_T}$ , Table 2 shows the variance over one iteration  $\sqrt{V}$ , the number of iterations performed in the time  $T$ , the time spent on one iteration  $\tau$ , and its components.



**Fig. 4.** The mean RMS over the region of the image marked by the red box in Fig. 2 after 1000 seconds of computation for various  $N_B$  and  $N_F$  (on the horizontal axis).

It is clearly seen from Table 2 that the variance in one iteration decreases with increasing number of rays issued from the light source (which could be expected), while the final noise increases. The analysis of the time components allows us to estimate their contribution to the total tracing time and understand why an increase in the number of rays ultimately leads to degradation in the quality of the final image.

## 6. CONCLUSIONS

A practical method for computing the coefficients  $\langle\langle C^2 \rangle\rangle$ ,  $\langle\langle C \rangle^2 \rangle_B$ , and  $\langle\langle C \rangle^2 \rangle_F$  appearing in the formula for the BDPM noise on the basis of quantities obtained by ray tracing is proposed. A practical method for computing the coefficients for the case of the straightforward calculation of the direct luminance is also considered.

The final noise value can be found using the sample variance of a series of values accumulated in a sequence of iteration steps. This yields its value only for a specific number of rays; however, it does not allow us to predict how the variance will change for another number of rays. Meanwhile, it is easy to calcu-

**Table 2.** Statistical data for tracing 25 rays issued from the camera per pixel depending on the number of rays issued from the source during 1000 s for the benchmark scene

$N_F$	1000	3000	10000	30000	100000	300000
$\sqrt{V_T}$	0.1	0.098	0.1	0.11	0.13	0.18
$\sqrt{V}$	1.277	1.234	1.224	1.222	1.21	1.202
$N_i$	160	158	149	130	89	47
$\tau$	6.25	6.33	6.7	7.7	11.25	21.47
$\tau_{\text{BMCRT}}$	6	6	6	6	6	6
$\tau_{\text{FMCRT}}$	$\approx 0$	$\approx 0$	$\approx 0$	$\approx 0$	0.05	0.17
$\tau_{\text{merge}}$	0.25	0.33	0.7	1.7	5.2	15.3

late, knowing the coefficients  $\langle\langle C^2 \rangle\rangle$ ,  $\langle\langle C \rangle_F \rangle_B$ ,  $\langle\langle C \rangle_B \rangle_F$ . Therefore, given the results of only one computation, we are able to predict the variance (the contribution of one iteration step) for an arbitrary number of rays.

If the time spent on tracing the rays issued from the camera and the light source is known together with the time spent on their merging (as shown in Table 2), then the optimal number of rays can be calculated, which is the number of rays that ensures the minimum noise during the given computation time.

## REFERENCES

- Zhdanov, D.D., Galaktionov, V. A., Voloboy, A.G., Zhdanov, A.D. Garbul', A.A., Potemin, I.S., and Sokolov, V. G., Photorealistic rendering of images formed by augmented reality optical systems, *Program. Comput. Software*, 2018, vol. **44**, no. 4, pp. 213–224. <https://doi.org/10.1134/S0361768818040126>
- Sik, M. and Krivanek, J., Survey of Markov chain Monte Carlo methods in light transport simulation, *IEEE Trans. Visual. Comput. Graph.*, 2018, vol. **26**, no. 4, pp. 1821–1840.
- Pharr, M. and Humphreys, G., *Physically Based Rendering: From Theory to Implementation*, San Francisco: Morgan Kaufmann, 2010, 2nd ed.
- Dodik, N., Implementing probabilistic connections for bidirectional path tracing in the Mitsuba Renderer, Sep. 2017. <https://www.cg.tuwien.ac.at/research/publications/2017/dodik-2017-pcbpt/>.
- Jensen, H.W. and Christensen, P., High quality rendering using ray tracing and photon mapping, *ACM SIGGRAPH 2007 Courses, Ser. SIGGRAPH'07*. 2007, New York: ACM, 2007. <http://doi.acm.org/10.1145/1281500.1281593/>
- Veach, E., A dissertation: Robust Monte-Carlo methods for light transport simulation, 1997. [http://graphics.stanford.edu/papers/veach\\_thesis/thesis.pdf](http://graphics.stanford.edu/papers/veach_thesis/thesis.pdf).
- Vorba, J., Bidirectional photon mapping, *Proc. of CESC 2011: The 15th Central European Seminar on Computer Graphics*, Prague: Charles University, 2011, pp. 25–32. <https://cgg.mff.cuni.cz/~jaroslav/papers/2011-bdpm/vorba2011-bdpm.pdf>.
- Ershov, S.V., Zhdanov, D.D., and Voloboy, A.G., Estimation of noise in calculation of scattering medium luminance by MCRT, *Math. Montisnigri*, 2019, vol. **XLV**, pp. 60–73.
- Georgiev, I., Krivánek, J., Davidovic, T., and Slusallek, P., Light transport simulation with vertex connection and merging, *ACM Trans. Graph.* 2012, vol. 31, no 6, pp. 192:1–192:10. <http://doi.acm.org/10.1145/2366145.2366211>.
- Hachisuka, T., Pantaleoni, J., and Jensen, H.W., A path space extension for robust light transport simulation, *ACM Trans. Graph.* 2012, vol. 31, pp. 191:1–191:10.
- Ershov, S.V. and Voloboy, A.G., Calculation of MIS weights for bidirectional path tracing with photon maps in presence of direct illumination, *Math. Montisnigri*, 2020, vol. **XLVIII**, pp. 86–102.
- Sbert, M., Havran, V., and Szirmay-Kalos, L., Multiple importance sampling revisited: Breaking the bounds, *EURASIP J. on Advances Signal Proc.*, 2018, vol. **15**, pp. 1–15.
- Hachisuka, T. and Jensen, H.W., Stochastic progressive photon mapping, *ACM SIGGRAPH Asia 2009 Papers*, 2009, pp. 141:1–141:8. <http://doi.acm.org/10.1145/1661412.1618487>.
- Ershov, S.V., Birukov, E.D., and Voloboy, A.G., Efficient calculation of optimal weights of multiple importance sampling in the bidirectional ray tracing with photon maps, *Preprint of the Keldysh Institute of Applied Mathematics, Russ. Acad. Sci.*, Moscow, 2020, no. 107, p. 22. <https://doi.org/10.20948/prepr-2020-107>

*Translated by A. Klimontovich*