Direct calculation of the optimal weight for MIS

S.V. Ershov[1], A.G. Voloboy[1], D.D. Zhdanov[2], A.D. Zhdanov[2], V.A. Frolov[1]
voloboy@gin.keldysh.ru
[1]Keldysh Institute for Applied Mathematics, Moscow, Russia
[2]National Research University of Information Technologies, Mechanics, and Optics, St. Petersburg, Russia

A Monte-Carlo ray tracing is nowadays standard approach for lighting simulation and generation of realistic images. A widely used method for noise reduction in Monte-Carlo ray tracing is combing different means of sampling, known as Multiple Importance Sampling (MIS). For bi-directional Monte-Carlo ray tracing with photon maps (BDPM) the join paths are obtained by merging camera and light sub-paths. Since several light paths are checked against the same camera path and vice versa, the join paths obtained are not statistically independent. Thus the noise in this method does not obey the laws which are correct in simple classic Monte-Carlo with independent samples. And, correspondingly, the MIS weights that minimize that noise must also be calculated differently. In this paper we calculate these weights for a simple model scene directly minimizing the noise of calculation. This is a pure direct numerical minimization that does not involve any doubtful hypothesis or approximations. We show that the weights obtained are qualitatively different from those calculated from classic "balance heuristic" for Monte-Carlo with independent samples. They depend on the scene distance, but not only on scattering properties of the surfaces and the distribution of light source emission.

Keywords*: Monte-Carlo ray tracing, bi-directional ray tracing, photon maps, reduction of noise, multiple importance sampling, weights.*

1. Introduction

A powerful method of solution of the rendering equations is Monte Carlo ray tracing (MCRT). It is widely used in calculation of the global illumination [1, 2]. Its main problem is noise, and it strongly depends on the method of generation of random points. Therefore there were and are a lot of papers devoted to the optimal choice of the probability distribution of ray scattering [3–9]. One of the powerful approaches here is the so-called Multiple Importance Sampling (MIS). Its idea is that we generate several random samples (rays) according to different "strategies" i.e. probability distributions and then sum with weights their contributions to image luminance.

The mathematics behind that was produced in the famous thesis by E Veach [3] where the theorem was proved about several simple schemes of weight calculation. It was proved there that the resulting noise is close to its minimal value. This theorem applies to the classic MCRT method when successive random points are absolutely independent.

Lighting simulation meanwhile frequently uses not that simple MCRT but more advanced methods like bi-directional Monte-Carlo path tracing (BDPT), bi-directional Monte-Carlo ray tracing with photon maps (BDPM) [2], their combination termed sometimes BDCM [8, 9] etc. Here the successive trajectories are not quite independent, for example, in the BDPM *the same* forward path is "merged" with all the backward paths. Therefore the resulting joined full trajectories have common "tail" and thus are not independent.

As a result, the noise in these methods follows other rules than in the simple or classic MCRT [6]. Therefore the weights that minimize this noise are likely different from those which minimize the noise functional in the classic MCRT. We shall prove it for the example of a very simple model scene. In this scene the noise level is dictated by geometric factors, but not by object optical properties in form of bi-directional distribution function (BDF), while the Veach formulae [3, 4] relate weights to the BDFs along the ray path.

In this paper we calculate the optimal weights directly, i.e. find the minimum of the sample variance of the pixel value. This is performed for a simple model scene. We demonstrate that these weights depend on the *geometry* of the scene and on the number of light and camera rays per iteration, while the known MIS formulae from [3, 4] include only the BDFs and distribution of light source emission.

2. BDPM and weights in it

The basic idea of BDPM is that we trace several camera and several light rays. Then for each pair of light + camera paths, we try to merge them in a join trajectory that connects light and camera. If they do join we increment the accumulated luminance. Then the next pair is processed. After all light rays had been checked against all camera rays they all are discarded and new sets of rays are generated etc. Generation of the sets of rays and then cycling over all pairs constitute *one iteration* of the process. The luminance calculated in different iterations is statistically independent.

So, an iteration which uses N_B camera rays (through given pixel!!) and N_F light rays (for all pixels!) increases accumulated luminance of a pixel by

$$\Delta L = \frac{1}{N_F}\sum_{i=1}^{N_F}\frac{1}{N_B}\sum_{j=1}^{N_B} C_{i,j} \tag{1}$$

where $C_{i,j}$ is the contribution from the pair of i-th light and j-th camera rays. Similarly to it can be written as

$$C = \sum_m \sum_n w_{m+n,m} E^{(c)}(\vec{x}_m^{(c)}) E^{(l)}(\vec{x}_n^{(l)}) K(\vec{x}_m^{(c)} - \vec{x}_m^{(l)}) f(\cdots; \vec{x}_m^{(c)}) \tag{2}$$

where m cycles over all camera path vertices and n cycles over all light path vertices, K is the integration kernel and f is BDF in luminance units at the point $\vec{x}_m^{(c)}$. Like in [3, 4], $w_{k,m}$ is the weight for junction at the m-th camera vertex when the join path of k vertices (i.e. the light half of the join path has $k - m$ vertices). It must be a function of that full path such that

$$\sum_{m=0}^{k-1} w_{k,m} = 1$$

Notice that in principle we have different sets of weights for joint paths of different total length k (this is obvious because they are functions of k arguments).

3. Direct calculation of optimal weights

It follows from (1) and (2) that the increment of the pixel luminance from one algorithm iteration is also linear in weights: it is a sum of weights times some random functions:

$$\Delta L = \frac{1}{N_F N_B} \sum_{k=1}^{\infty} \sum_{m=0}^{k} \sum_{i,j} w_{k,m}(\vec{\zeta}_{i,j}) C_{k,m}(\vec{\zeta}_{i,j})$$

where i enumerates light rays in one iteration, j enumerates camera rays (through *this* pixel) in one iteration and $\vec{\zeta}$ is the join path from them. $C_{k,m}(\vec{\zeta}_{i,j})$ is the contribution from the this pair (i,j) constrained to conditions

1. The paths merge into join path
2. They merge at the m-th camera vertex
3. The join path has k vertices

If these conditions are not satisfied $C_{k,m}$ vanish. This resolves ambiguity how to define the join path if the camera and light halves did *not* merge.

The sets of join paths from different iterations are independent. So for this *linear* form the average over iterations (= the limiting luminance) is also linear in weights. The mean square of luminance value calculated in one iteration is a *quadratic* form in weights whose "coefficients" are averages that can be calculated in ray tracing.

Therefore we can find those weights that minimize the variance of the pixel luminance, i.e. the noise. These are the optimal weights. Its direct calculation that does not include an approximations and hypothesis is regrettably very expensive numerically. So we shall perform it for a simple model scene, but even this example will give us some important conclusions.

4. Simple model scene and calculations for it

Scene layout

To simplify our calculations we use a model scene where all join paths have the same (and small) length. It consists of 3 parallel planes with diffuse transparency; planes 1 and 3 have the Lambert BDF. BDF of the middle plane 2 is arbitrary and can be made very sharp (when direction of a transmitted ray is close to that of the incident ray). The planes are orthogonal to Oz and are positioned at $z = 0$, $z = L_2$ and $z = L_2 + L_3$ respectively. Camera looks at plane 1 at normal direction. We consider a single pixel such that the camera ray hits plane 1 at (0, 0, 0) point. The rightmost plane 3 is illuminated by light source from right side, see Fig. 1.

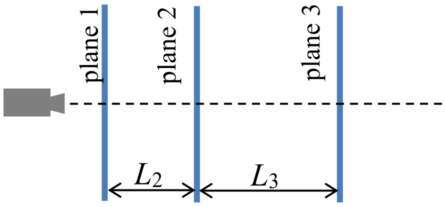

Fig. 1: The model scene

The spatial distribution of illumination of plane 3 is $I(x,y)$. Since transmittance is the Lambert the angular distribution of incident light is irrelevant.

Join paths and weights

This scene has no reflection and all paths connecting the camera and light source are qualitatively the same

$$\text{camera} \to \vec{x}_1 \to \vec{x}_2 \to \vec{x}_3 \to \text{light}$$

where $\vec{x}_1 = \vec{0}$ is fixed and the segment between lights source and plane 3 is ignored because does not affect the path contribution. Therefore the full path is completely described by its two variable vertices, $\vec{x}_2$ and $\vec{x}_3$.

The camera and light rays can meet at planes 1, 2 and 3 whose contributions are taken with weights $w_0(\vec{x}_2,\vec{x}_3)$, $w_1(\vec{x}_2,\vec{x}_3)$ and $w_2(\vec{x}_2,\vec{x}_3)$. Because of normalization $w_0 + w_1 + w_2$ it suffices to calculate w_0 and w_1.

If camera and light ray meet at plane m, it is ambiguous whether $\vec{x}_m$ is camera or light hit (they can differ by integration kernel radius). We choose camera hit then.

Calculation of contribution

Camera path is $(\vec{x}_1^{(c)}, \vec{x}_2^{(c)}, \vec{x}_3^{(c)})$ and light path is $(\vec{x}_3^{(l)}, \vec{x}_2^{(l)}, \vec{x}_1^{(l)})$ where $\vec{x}_m^{(c)}$ is the hit point of camera ray at the m-th plane, $\vec{x}_m^{(l)}$ is the hit point of light ray at the m-th plane (light ray goes from plane 3 to plane 2 then to plane 1), and $\vec{x}_1^{(c)} = \vec{0}$ is fixed. As said above we do not consider the light ray before it hits the plane 3; just we start the ray by choosing the point $\vec{x}_3^{(l)}$at random.

The contribution of these two sub-paths is

$$C = w_0(\vec{x}_2^{(l)}, \vec{x}_3^{(l)}) E^{(c)}(\vec{x}_1^{(c)}) E^{(l)}(\vec{x}_1^{(l)}) K(\vec{x}_1^{(c)} - \vec{x}_1^{(l)}) f_1(\cdots)$$
$$+ w_1(\vec{x}_2^{(c)}, \vec{x}_3^{(l)}) E^{(c)}(\vec{x}_2^{(c)}) E^{(l)}(\vec{x}_2^{(l)}) K(\vec{x}_2^{(c)} - \vec{x}_2^{(l)}) f_2(\cdots)$$
$$+ w_2(\vec{x}_2^{(c)}, \vec{x}_3^{(c)}) E^{(c)}(\vec{x}_3^{(c)}) E^{(l)}(\vec{x}_3^{(l)}) K(\vec{x}_3^{(c)} - \vec{x}_3^{(l)}) f_3(\cdots)$$

where $f_1 = f_2 = \pi^{-1}$ while

$$f_2 = \frac{1}{2\pi\beta^2} e^{-\frac{\vartheta^2}{2\beta^2}} \frac{1}{\cos\gamma}$$

where ϑ is the angle between the incident and scattered rays, γ is the angle between the scattered ray and the normal and β is the width. As to the integration kernel, we use the simplest one:

$$K(\vec{x}) = \frac{1}{\pi R^2} \begin{cases} 1, & |\vec{x}| \le R \\ 0, & |\vec{x}| > R \end{cases}$$

where R is integration radius. It is small.

Denoting the camera path as $\vec{\xi} \equiv (\vec{x}_1^{(c)}, \vec{x}_2^{(c)}, \vec{x}_3^{(c)})$ and light path as $\vec{\eta} \equiv (\vec{x}_3^{(l)}, \vec{x}_2^{(l)}, \vec{x}_1^{(l)})$ we can write

$$C(\vec{\xi},\vec{\eta}) = \sum_{m=0}^{2} w_m(\vec{x}_2(\vec{\xi},\vec{\eta}), \vec{x}_3(\vec{\xi},\vec{\eta}))C_m(\vec{\xi},\vec{\eta})$$

where

$$C_m \equiv E^{(c)}(\vec{x}_m^{(c)})E^{(l)}(\vec{x}_m^{(l)})K(\vec{x}_m^{(c)} - \vec{x}_m^{(l)})f_{m+1}(\cdots)$$

are independent from weights. Notice that usually in MCRT the ray energies can be calculated deterministically from the path. But even if they are random this does not affect our derivation, just $C_m$becomes “more random”.

Pixel luminance calculated during M iterations, each of which uses N_F light rays and N_B camera rays, is

$$L = \frac{1}{M}\sum_{s=1}^{M} \boldsymbol{\mathcal{C}}_s$$
$$\boldsymbol{\mathcal{C}}_s \equiv \frac{1}{N_B}\frac{1}{N_F}\sum_{i=1}^{N_F}\sum_{j=1}^{N_B} C(\vec{\xi}_{s,j}, \vec{\eta}_{s,i}) \tag{3}$$

The contribution from each iteration (3) is random variable and contributions from different iteration. Therefore the variance of calculated luminance is

$$Var(L) = \frac{1}{M} Var(\boldsymbol{\mathcal{C}})$$
$$Var(\boldsymbol{\mathcal{C}}) = \langle \boldsymbol{\mathcal{C}}^2 \rangle - \langle \boldsymbol{\mathcal{C}} \rangle^2$$

The averages are over the ray ensembles. They can be approximately estimated from the sum over iterations (the usual practice called sample mean and sample variance).

Tabulated weights

For numerical calculations let us subdivide the whole admissible area in $(\vec{x}_2, \vec{x}_3)$ space in cells. The weight is constant $w_{m,\alpha}$ within cell. Since formally $\vec{x}_2$ and $\vec{x}_3$ can be infinite we take some finite area and subdivide it in a usual way, unbounded space outside it constituting “the last cell”. Let $\chi_\alpha(\vec{x}_2, \vec{x}_3)$ be 1 inside the α-th cell and 0 outside it. Then the contribution from the s-th iteration (3) becomes

$$A_{\alpha,s}^{(m)} \equiv \boldsymbol{\mathcal{C}}_{\alpha,s}^{(m)} - \boldsymbol{\mathcal{C}}_{\alpha,s}^{(2)}$$
$$B_s \equiv \sum_{\alpha} \boldsymbol{\mathcal{C}}_{\alpha,s}^{(2)}$$
$$\boldsymbol{\mathcal{C}}_{\alpha,s}^{(m)} \equiv \frac{1}{N_B}\frac{1}{N_F}\sum_{i=1}^{N_F}\sum_{j=1}^{N_B} \chi_\alpha(\vec{\xi}_{s,j}, \vec{\eta}_{s,i}) C_m(\vec{\xi}_{s,j}, \vec{\eta}_{s,i})$$

Combining the tables that relate to the weights w_0 and w_1 into single array:

$$W \equiv \begin{pmatrix} w_0 \\ w_1 \end{pmatrix}$$
$$F_s \equiv \begin{pmatrix} F_s^{(0)} \\ F_s^{(1)} \end{pmatrix}$$
$$A_s \equiv \begin{pmatrix} A_s^{(0)} \\ A_s^{(1)} \end{pmatrix}$$

we can write

$$\boldsymbol{\mathcal{C}}_s = \langle A_s || W \rangle + B_s$$

Then, since the values for different s are independent the average value and the mean square are

$$\overline{\boldsymbol{\mathcal{C}}} = \langle \overline{A} || W \rangle + \overline{B}$$
$$\overline{\boldsymbol{\mathcal{C}}^2} = \langle W | D | W \rangle + 2\langle \overline{F} || W \rangle + \overline{B^2}$$

where the overbar denotes the average, and

$$D \equiv \overline{|A\rangle\langle A|}$$
$$\langle F| \equiv \overline{B\langle A|}$$

The mathematical expectation of pixel luminance is the same for all weights, thus *the limiting average* $\langle \overline{A}| = 0$. But for a finite number of iterations, when convergence is incomplete, the *sample average* can be slightly depending on weights. Thus the *sample average* $\langle \overline{A}| \neq 0$ and while calculating the *sample variance over a finite number of iterations* we must account for this dependence. This sample variance over M iterations is thus

$$Var(L) = \frac{1}{M}\Big(\langle W|D|W\rangle + 2\langle \overline{F} || W \rangle + \overline{B^2}$$
$$- \left(\langle \overline{A} || W \rangle + \overline{B}\right)^2\Big)$$

so the weights which minimize it satisfy

$$\left(D - |\overline{A}\rangle\langle\overline{A}|\right)|W\rangle = -|F\rangle + \langle B \rangle |\overline{A}\rangle$$

which is just a system of simultaneous linear equations.

However numerical experiments shown that the solution can be rather ragged. To improve the situation a regularization term can be added to the minimization equation which is a penalty for high gradients.

5. Results

We performed the calculations for the case when

- plane positions $L_2 = 1$, $L_3 = 3$,
- BDF of plane 2 has width $\beta = 3°$
- illumination density I is

$$I(r_3) = \begin{cases} \frac{1}{300}\left(\frac{r_{LS}}{a}\right)^2 = 3333 & r_3 \leq a \\ 1 & a < r_3 < r_{LS} \\ 0 & r_3 \geq r_{LS} \end{cases}$$

where $r_{LS} = 20$ is the radius of illuminated area and $a = 0.02$ is the “aperture” (radius) of its bright central part

- integration radius $R = 0.003$
- the number of rays $N_B = 100$, $N_F = 10^7$

The scene is axisymmetrical. Therefore all functions of $(\vec{x}_2, \vec{x}_3)$ actually depend on 3, not 4 variables: (r_2, r_3, φ) where φ is the angle between vectors $\vec{x}_2$ and $\vec{x}_3$ (notice that φ and $2\pi - \varphi$ give the same result!). BTW one can prove that the optimal weights are independent from φ.

As said above rays fill the area with r_2 and r_3 up to infinity. So we chose a finite area for each, now $0 \leq r_2 \leq \frac{1}{2}$, $0 \leq r_3 \leq 0.05$, subdivided it into equal cells and then added the last cell which completes to the whole infinite domain, e.g. $\frac{1}{2} < r_2 < \infty$.

Trial calculations were done for several numbers of cells. It happened that although the calculated weights differ, the noise level is nearly the same (as it is common for optimization). Since the weights are not needed per se, but only the noise reduction by them, we can use as small cells as enough to saturate the noise level.

It happened that it was enough 1 cell in φ (i.e. weights actually do not depend on it!), 2 cells in r_2 ($0 \le r_2 \le \frac{1}{2}$ and $\frac{1}{2} < r_2 < \infty$) and 26 cells in r_3 (the first 25 of size 0.002 and the last $0.05 < r_3 < \infty$).

The noise was calculated for the following cases. The calculation results are shown in Table 1:

1. rays meet at plane 1 only
2. rays meet at plane 2 only
3. rays meet at plane 3 only
4. rays meet at plane 3 only
5. optimal weights are used

Table 1. The calculation results

case	w_0	w_1	w_2	L×10⁵	RMS,%
1	1	0	0	26.484	208%
2	0	1	0	26.352	254%
3	0	0	1	25.997	146%
4	0	$\begin{cases}1, & r_3 \le a\\0, & r_3 > a\end{cases}$	$\begin{cases}0, & r_3 \le a\\1, & r_3 > a\end{cases}$	26.352	61%
5	0	optimal		25.961	54%

Optimal weights were calculated from statistic accumulated in 10000 iterations (Fig. 2). Optimization was constrained to $w_0 = 0$.

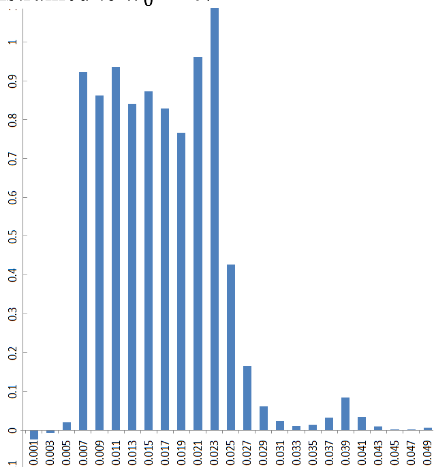

Fig. 2: The optimal weight w_1 as a function of r_2; $w_2 = 1 - w_1$; w_0 was constrained to 0

6. Conclusions

We see that even in case of a *direct* optimization (which gives the best result without false minima, approximations etc.) the gain is moderate; it is about 3fold as compared to the best "fixed BDD" strategy. This is not bad because 3fold in noise is equivalent to a 9fold increase of speed.

At qualitative level we see that the optimal weights *are not local* i.e. we cannot calculate the weight (which is as we remember a function of the vertices of join path) from that path only. Indeed, in the above calculation illumination of the rightmost plane was 3333 times lower for $r_3 > a$. Let us compare it with the case of uniform illumination. In this case the optimal weight is very close to $w_2 = 1$ (for all paths), so for a join path with $|\boldsymbol{x}_3| \le a$ the optimal weight is different for the uniform and not uniform illumination. In other words, the weight for this path depends on illumination outside it.

Surely the optimal weight is still a function of the join path but this function depends on the global scene characteristics.

Meanwhile in the "balance heuristic" or "power heuristic" [3, 4] this function is known in advance. Very roughly, it calculates the weight from the ratio of BDF at junction point to the sum of BDFs at all the vertices of the join path. We therefore conclude that the balance/power heuristic, derived for the usual MCRT, is not truly optimal for BDPM because there the "samples" (join paths) are correlated because use the same light and/or camera path several times.

Acknowledgments

The study was carried out within the framework of the RFBR grants 18-01-00569, 18-31-20032 and 20-01-00547.

References:

[1] Matt Pharr and Greg Humphreys. 2010. Physically Based Rendering, Second Edition: From Theory to Implementation (2nd ed.). Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

[2] H. W. Jensen, Global illumination using photon maps, in Proceedings of the Eurographics Workshop on Rendering Techniques '96, (London, UK, UK), pp. 21–30, Springer-Verlag, 1996.

[3] Eric Veach. A dissertation: Robust Monte-Carlo methods for light transport simulation, 1997.

[4] Jiri Vorba. Bidirectional photon mapping. In Proceedings of CESCG 2011: The 15th Central European Seminar on Computer Graphics, Prague, 2011.

[5] I. Georgiev, J. Křivánek, T. Davidovič, and Ph. Slusallek. 2012. Light transport simulation with vertex connection and merging. ACM Trans. Graph. 31, 6, Article 192 (November 2012)

[6] S. Ershov, D. Zhdanov, and A. Voloboy. Estimation of noise in calculation of scattering medium luminance by MCRT. Mathematica Montisnigri, XLV: 60–73, 2019.

[7] S. Ershov, D. Zhdanov, A. Voloboy, M. Sorokin. Treating diffuse elements as quasi-specular to reduce noise in bi-directional ray tracing // Keldysh Institute Preprints. 2018. No. 122. 30 p. doi:10.20948/prepr-2018-122-e

[8] S. Popov, R. Ramamoorthi, F. Durand, and G. Drettakis, Probabilistic Connections for Bidirectional Path Tracing, Computer Graphics Forum, 2015.

[9] N. Dodik, Implementing probabilistic connections for bidirectional path tracing in the Mitsuba Renderer, Sept. 2017.

About the authors

Sergey V. Ershov, PhD, senior researcher in the Keldysh Institute of Applied Math of RAS. E-mail: measure@spp.keldysh.ru

Alexey G. Voloboy, Doctor of Science in physics and mathematics, leading researcher in the Keldysh Institute of Applied Math of RAS. E-mail: voloboy@gin.keldysh.ru

Dmitry D. Zhdanov, PhD, associate professor in the Information Technologies, Mechanics and Optics (ITMO) University. E-mail: ddzhdanov@mail.ru

Andrey D. Zhdanov, PhD student in the Information Technologies, Mechanics and Optics (ITMO) University. E-mail: adzhdanov@itmo.ru

Vladimir A. Frolov, PhD, senior researcher in the Keldysh Institute of Applied Math of RAS. E-mail: vova@frolov.pp.ru