Go with the Winners Strategy in Path Tracing [1]

László Szirmay-Kalos, György Antal

Dept. of Control Eng. and Information Tech.
TU Budapest, Budapest
Magyar Tudósok krt. 2., H-1117, Hungary
szirmay@iit.bme.hu, gy_antal@yahoo.com

Mateu Sbert

Inst. of Applied Math. and Informatics
University of Girona, Girona
Campus Montilivi, E-17071, Spain
mateu@ima.udg.es

ABSTRACT

This paper proposes a new random walk strategy that minimizes the variance of the estimate using statistical estimations of local and global features of the scene. Based on the local and global properties, the algorithm decides at each point whether a Russian-roulette like random termination is worth performing, or on the contrary, we should split the path into several child paths. In this sense the algorithm is similar to the go-with-the-winners strategy invented in general Monte Carlo context. However, instead of establishing thresholds to make decisions, we compute the number of child paths on a continuous level and show that Russian roulette can be interpreted as a kind of splitting using fractional number of children. The new method is built into a path tracing algorithm, and a minimum cost heuristic is proposed for choosing the number of reflected rays. Comparing it with the classical path tracing approach we concluded that the new method reduced the variance significantly.

Keywords: Global illumination, random walk, Monte Carlo methods.

1 Introduction

Random walk global illumination algorithms evaluate an infinite sequence of integrals of the following form:

$$L^r = \int_{\Omega_1} w_1 \cdot \left(L^e + \int_{\Omega_2} w_2 \cdot (L^e + \ldots)\ d\omega_2 \right) d\omega_1 \qquad (1)$$

where L^r is the reflected radiance, L^e is the emission, w is the scattering density, usually expressed as the product of the BRDF and the cosine of the orientation angle, and Ω_i is the set of directions of possible illumination.

When the first outer integral is estimated by Monte Carlo techniques n_1 random directions are obtained with a probability density p_1, and the following quadrature is computed

$$\hat{L}^r = \frac{1}{n_1} \cdot \sum_{i=1}^{n_1} \frac{w_1(\omega_1^i)}{p_1(\omega_1^i)} \cdot L_i^{\text{in}} = \sum_{i=1}^{n_1} W_1^i \cdot L_i^{\text{in}}.$$

where $L_i^{\text{in}} = L_i^e + L_i^r$ is the sum of the emission and the reflected radiance at the hit point of the traced ray. If $n_1 > 1$, then the random path is split into n_1 paths at this point. When $n_1 = 1$ *splitting* does not happen. Term $\frac{w_1(\omega_1^i)}{p_1(\omega_1^i)} \cdot L^e$ can be immediately added to the estimate, but the computation of reflected radiance L_i^r poses a similar integration problem, which can be solved by repeating the same procedure. Each step l we add n_l number of $W_l \cdot L^e$ terms to the quadrature, where *potential* W_l can be expressed in a product form

$$W_l = W_{l-1} \cdot \frac{1}{n_l} \cdot \frac{w_l(\omega_l^i)}{p_l(\omega_l^i)}. \qquad (2)$$

Sampling a random direction is not the only way to estimate L^r in a point if a random approximation of the radiance function is available in the scene. Taking this random approximation as the real radiance and computing its reflection by deterministic connections also lead to the estimate of all remaining terms of the infinite Neumann series. This corresponds to *joining* the path with

[1] Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. The Journal of WSCG, Vol.13, ISSN 1213-6964. WSCG'2005, January 31-February 4, 2005 Plzen, Czech Republic.

the results of other paths obtained earlier. Finally, we might decide not to continue the computation of the path. This is called *termination.*

As we walk along the random path, we make a decision at in each step. Should we spawn new random rays, or should we estimate the reflected radiance directly? If random rays are sampled, what is their optimal number? Each of these decisions results in a term in the integral quadrature and also an error in the estimate. The complete rendering algorithm will evaluate many recursive integrals with a lot of random paths, thus we make a lot of decisions that affect both the error of the integral quadratures and the total computation time. In this paper we propose an approach that minimizes this total computation error and keeps the computational time low.

Section 2 reviews the previous work, and particularly the go-with-the-winners strategy. In section 3 we present a theoretical analysis of the simultaneous application of Russian roulette, splitting and joining, and extend the concept of the go-with-the-winners strategy to use continuous scale. Then in section 4 the cost-variance optimization is discussed. Finally, in section 5 we present a minimum cost heuristic for choosing the number of reflected rays in stochastic ray tracing.

2 Previous work

Path splitting, joining and termination have been intuitively and partially applied in several random walk global illumination algorithms.

Russian-roulette[AK90] terminates the walk randomly. When the path is terminated, no deterministic estimation takes place, and the illumination of this point is supposed to be zero. The probability of the random termination is the albedo of the visited point, or the luminance of the albedo in case of spectral rendering. In order to compensate the not computed terms, when the integrand is really computed, it is divided by the continuation probability. There are several problems of classical Russian-roulette. It increases the variance inversely proportional to the continuation probability. On the other hand, spectral rendering poses another problem to Russian roulette, where the contributions are transferred on different wavelengths simultaneously, but the continuation probability should obviously remain a scalar value. If this scalar is the luminance of the albedo, then the estimation can be very poor if the spectrum of the reflection does not coincide with the transferred potential. For example, when a path visits first a red, then a green surface, then the contribution will be zero, but this is not recognized by Russian-roulette. These problems have been pointed out in [SSKK03].

The variance introduced by Russian roulette can also be reduced by setting the termination probability globally and not locally. It means that the continuation probability is the average albedo of the whole scene, and not the local albedo. Such approach was used by Keller [Kel97], when the continuation probability has been determined separately, and also in ray-based stochastic iteration algorithms, where the contraction ratio of the integral operator has been determined on the fly [SK99]. However, global termination probability may also cause infinite variance.

In random walk algorithms that reuse light paths, we also have a random estimate of the incoming illumination, which can be obtained without continuing the random walk. The acquisition of this estimate may require data-structure searches (*photon-map* [JC95, Chr00], *irradiance caching* [WRC88], *discontinuity buffer* [WKB$^+$02]) or tracing deterministic shadow rays (*bi-directional path tracing* [LW93, VG95], *virtual light sources algorithm* [Kel97, WKB$^+$02], and *path reuse* [BSH02]).

The benefits of path termination, splitting and joining can also be combined. In path reuse methods, paths are terminated by Russian-roulette, and its visited points are joined with other paths. Since such methods may generate a complete path in many different ways a clever weighting scheme should be applied, as proposed by *multiple importance sampling* [Vea97].

Considering these, we can conclude that termination, splitting and joining have already shown up in many different random walk algorithms, and even their intuitive optimal combination has been emerged. On the other hand, Bolin and Meyer [BM97] analyzed the variance of Russian-roulette and splitting.

In this paper we follow this direction of the previous work in order to find optimal termination/splitting/joining, which results in the smallest error. This work has been inspired by a general Monte Carlo strategy called *go with the winners* [AV94, Gra01] that can include many approaches dealing with termination and splitting [Kah56]. In this method, the decision is made according to the accumulated potential W, which is compared with two predefined constants W^- and W^+ ($W^- < W^+$). If $W < W^-$, then Russian roulette is executed with probability W/W^-. If $W^- \leq W < W^+$, then the path is extended by a single ray. If $W^+ \leq W$, then the random path splits to n subpaths (say $n = 10$), and the potential is divided by n.

3 Random walks with termination, splitting and joining

Suppose that $l-1$ steps of the random walk have already been computed and we are facing the decision of what to do having potential W_{l-1}. If the walk is continued, then W_l needs to be found, and n_l estimates of reflected radiance $\hat{L}_l^r$ are added to the quadrature. If the walk is not continued, then the estimate should cover all $l, l+1, \ldots$ steps, which can also be added to the quadrature. The random termination can also be imagined similarly to splitting, but now we use $n_l \leq 1$ number of random directions in average. The average value comes from the fact that sometimes the path is not continued at all.

At a given point of the random walk, parameter n_l must be determined to minimize the error. Each sample contributes to the square error of the integral quadrature proportionally to its own square error, which equals to the variance in the unbiased case, and to the sum of the variance and the square of the bias in the biased case. Thus the decision should be made to minimize the introduced error.

The variance computation is discussed for splitting and random termination separately.

3.1 Splitting: $n_l \geq 1$

When n_l random directions are used, the estimator of the contribution of paths of length l is

$$\hat{L}_l^r = W_{l-1} \cdot \frac{1}{n_l} \cdot \sum_{i=1}^{n_l} \frac{w_l(\omega_l^i)}{p_l(\omega_l^i)} \cdot L_i^{\text{in}},$$

where ω_l^i is the ith random sample of integrand variable ω_l, and $p_l(\omega_l^i)$ is the probability density of obtaining this sample. This means breaking the paths to n_l children, where child i has

$$W_l^i = W_{l-1} \cdot \frac{1}{n_l} \cdot \frac{w_l(\omega_l^i)}{p_l(\omega_l^i)}$$

potential, and $W_l^i \cdot L_i^{\text{in}}$ is the contribution of this path. When using this formula in practical algorithms, we can usually assume that p_l mimics w_l, i.e. where w_l is non-zero, their ratio $w_l/p_l = a_l$ is — at least approximately — constant. This constant is the probability that the light is not absorbed, and is called the *albedo*.

The variance of this contribution is:

$$\frac{W_{l-1}^2}{n_l^2} \cdot D^2 \left[\frac{w_l}{p_l} \cdot L^{\text{in}}\right] = \frac{W_{l-1}^2}{n_l^2} \cdot a_l^2 \cdot D^2 \left[L^{\text{in}}\right].$$

The total variance of the family of n_l samples obtained by splitting the path is n_l times this variance since the children can be assumed to be independently generated. Thus the total variance of the family of paths is:

$$\frac{W_{l-1}^2}{n_l} \cdot a_l^2 \cdot D^2 \left[L^{\text{in}}\right]. \tag{3}$$

3.2 Random termination: $n_l < 1$

In this case the average number of samples to continue a path is less than 1. It corresponds to the case when the probability of path continuation is n_l. When no random sample is taken, the result of all remaining steps — i.e. the contribution of paths of length $l, l+1, \ldots$ — is estimated by a known constant value, for example by 0 as suggested by Russian-roulette. To be general, let us assume that we have a random estimate $\hat{L}$, which is available without spawning random rays. The expected value of estimate $\hat{L}$ may or may not be equal to the exact integral value L^r, which can be expressed by bias ΔL in this estimation:

$$E[\hat{L}] = L^r + \Delta L, \quad L^r = E\left[\frac{w_l}{p_l} \cdot L^{\text{in}}\right].$$

When the walk is decided to be terminated, we use available estimate $\hat{L}$. If the walk is continued, then a linear combination of actually computed radiance $W_{l-1} \cdot w_l/p_l \cdot L^{\text{in}}$ and estimate $W_{l-1} \cdot \hat{L}$ is inserted in the estimator, that is, we use

$$W_{l-1} \cdot \left(\alpha \cdot \frac{w_l}{p_l} \cdot L^{\text{in}} + \beta \cdot \hat{L}\right).$$

The α and β values of this linear combination can be determined from the requirement that the expected value of this estimator should be correct:

$$n_l \cdot W_{l-1} \cdot \left(\alpha \cdot E\left[\frac{w_l}{p_l} \cdot L^{\text{in}}\right] + \beta \cdot E\left[\hat{L}\right]\right) +$$

$$(1 - n_l) \cdot W_{l-1} \cdot E\left[\hat{L}\right] =$$

$$W_{l-1} \cdot L^r \cdot (1 - (1 - \alpha - \beta) \cdot n_l) +$$

$$W_{l-1} \cdot \Delta L \cdot (1 - (1-\beta) \cdot n_l).$$

Note that the cases of continuation and termination have been weighted with n_l and $1 - n_l$, respectively, since these are their probabilities. To make this estimate unbiased, it should be equal to $W_{l-1} \cdot L^r$, thus $\alpha + \beta = 1$ should hold, and the following term should be zero

$$W_{l-1} \cdot \Delta L \cdot (1 - \alpha \cdot n_l).$$

Even if $\hat{L}$ is biased (i.e. ΔL is not zero), the bias of the random walk estimate can be made zero by setting $\alpha = 1/n_l$. Using this assumption, the variance of the estimate is

$$n_l \cdot W_{l-1}^2 \cdot E\left[\left(\frac{w_l/p_l \cdot L^{\text{in}}}{n_l} - \frac{(1-n_l)\cdot \hat{L}}{n_l}\right)^2\right] +$$

$$(1-n_l) \cdot W_{l-1}^2 \cdot E\left[\hat{L}\right]^2 - W_{l-1}^2 \cdot (L^r)^2 =$$

$$\left(\frac{1}{n_l} - 1\right) \cdot W_{l-1}^2 \cdot E\left[\left(\frac{w_l}{p_l} \cdot L^{\text{in}} - \hat{L}\right)^2\right] +$$

$$W_{l-1}^2 \cdot D^2\left[\frac{w_l}{p_l} \cdot L^{\text{in}}\right].$$

This formula can be used to obtain the variance for a given n_l. Note that if $\hat{L}$ is not far from an unbiased estimator, i.e. $\hat{L} \approx L^r = E\left[\frac{w_l}{p_l} \cdot L^{\text{in}}\right]$, then

$$E\left[\left(\frac{w_l}{p_l} \cdot L^{\text{in}} - \hat{L}\right)^2\right] \approx E\left[\left(\frac{w_l}{p_l} \cdot L^{\text{in}} - L^r\right)^2\right],$$

which equals to $D^2\left[\frac{w_l}{p_l} \cdot L^{\text{in}}\right]$, and thus the variance is approximately

$$\frac{W_{l-1}^2}{n_l} \cdot D^2\left[\frac{w_l}{p_l} \cdot L^{\text{in}}\right] = \frac{W_{l-1}^2}{n_l} \cdot a_l^2 \cdot D^2\left[L^{\text{in}}\right].$$

Note that the variance has the same formula as derived for the case of splitting (equation 3).

3.3 Estimation of $D^2\left[L^{\text{in}}\right]$

We face the problem that incoming radiance L^{in} is a random variable and is not known. The variance of L^{in} can come from two different sources. On the one hand, for fixed ω, the incoming radiance is estimated by continuing the random walk, which obtains the estimate by random simulation. On the other hand, even if we exactly knew the conditional expectation $\tilde{L}^{\text{in}}(\omega)$ of $L^{\text{in}}(\omega)$ for fixed incoming direction ω, then the variation of this expectation for different incoming directions would be another source of the error. Formally, we can write

$$D^2\left[L^{\text{in}}\right] = E\left[\left(L^{\text{in}} - E\left[L^{\text{in}}\right]\right)^2\right] =$$

$$\int_\Omega E\left[\left(L^{\text{in}} - E\left[L^{\text{in}}\right]\right)^2 \mid \omega\right] \cdot p_l(\omega) d\omega =$$

$$\int_\Omega E\left[\left(L^{\text{in}}(\omega) - \tilde{L}^{\text{in}}(\omega)\right)^2\right] \cdot p_l(\omega)\, d\omega +$$

$$\int_\Omega \left(\tilde{L}^{\text{in}}(\omega) - E\left[L^{\text{in}}\right]\right)^2 \cdot p_l(\omega)\, d\omega.$$

The first term in this sum describes how well the algorithm can estimate the radiance of a single point, and is approximated by a global constant V_R. The second term, on the other hand, represents how quickly the incoming radiance changes in the domain of the random directions, which is prescribed by the local BRDF. For instance, if the examined point is an ideal mirror, then BRDF sampling samples just a single direction, and the second term is zero. Generally, the second term gets bigger as the size of the set of possible directions grows. As can be shown the dependence is quadratic, that is, the second term is proportional to the square of the size of the directional domain. Let us consider a simple, Phong-like BRDF with shininess parameter s. Diffuse and mirror like materials can be imagined as special cases of $s = 0$ and $s = \infty$, respectively. The size of the domain of a Phong-like BRDF is $2\pi/(s+1)$ [LW94], thus the second term is approximated by $V_V/(s+1)^2$, where V_V is a general global constant.

Summarizing, the total variance of the children of a single parent is approximated as

$$\frac{W_{l-1}^2}{n_l} \cdot a_l^2 \cdot \left(V_R + \frac{V_V}{(s_l+1)^2}\right).$$

4 Variance-cost optimization

In the previous section we determined the variance associated with splitting and random termination with incoming radiance estimation. The variance is inversely proportional to value n, which stands for the average number of continued path at this point. On the other hand, if ray tracing is responsible for a major part of the computation time, then the cost is proportional to n. The goal is to obtain the most accurate result paying the lowest cost, that is, to minimize the total variance of the result with a constraint on the total number of rays. Formally, the optimization goal has the form

$$\sum_k \sum_l \sigma_{k,l}^2 / n_{k,l},$$

where k considers each light path and l each ray of a path, and

$$\sigma_{k,l}^2 = D^2 \left[W_{k,l-1} \cdot \frac{w_{k,l}}{p_{k,l}} \cdot L_k^{\mathrm{in}} \right] \approx$$

$$W_{k,l-1}^2 \cdot a_{k,l}^2 \cdot \left(V_R + \frac{V_V}{(s_{k,l}+1)^2} \right),$$

with constraint $\sum_k \sum_l n_{k,l} = N$, where $n_{k,l}$ is the average number of paths leaving the lth sample point of path k, and N is the total number of rays used to compute the whole image. Using the Lagrange multiplier method, we have to find the minimum of

$$\sum_k \sum_l \frac{\sigma_{k,l}^2}{n_{k,l}} + \lambda \cdot \left(\sum_k \sum_l n_{k,l} - N \right).$$

Making the partial derivatives equal to zero, we obtain

$$n_{k,l} = N \cdot \frac{\sigma_{k,l}}{\sum_{k'} \sum_{l'} \sigma_{k',l'}}.$$

It means that at each visited point number of child rays n_l should be proportional to

$$W_{k,l-1} \cdot a_{k,l} \cdot \sqrt{V_R + \frac{V_V}{(s_{k,l}+1)^2}}.$$

We could establish only a requirement of proportionality, and parameters V_R and V_V are left free. These parameters depend on the scene properties and may also be subjects for statistical estimation. On the other hand, we can follow a simple intuition. Assume that the accumulated potential and the albedo are maximum, that is $W_{l-1} \cdot a_l = 1$. If the surface is an ideal mirror, i.e. $s_l = \infty$, then a reasonable way to continue the path randomly with exactly one child. On the other hand, if the surface is purely diffuse, i.e. $s_l = 0$, and we may require the maximum number of children equal to $n_{\max}$. The optimal selection of $n_{\max}$ depends also on the properties of the scene. For example, if the illumination in the scene is homogeneous, i.e. a point receives similar illumination from all directions, then $n_{\max}$ is 1. As the illumination gets more and more heterogeneous, $n_{\max}$ is worth increasing. We used value 10 in the implementation, which seems to be a good choice for practical scenes. From these two requirements, V_R and V_V can be obtained, and the general formula for the number of children is

$$n_{k,l} = W_{k,l-1} \cdot a_{k,l} \cdot \sqrt{1 + \frac{n_{\max}^2 - 1}{(s_{k,l}+1)^2}}. \qquad (4)$$

If the material model consists of several different elementary materials (e.g. diffuse + specular), then the number of children should be computed separately using the albedo of the elementary BRDFs, and then the results should be added.

5 Variance based Go with the Winners Strategy

We propose a path tracing algorithm that is driven by the theoretical results of previous sections. Note that if path tracing used only BRDF sampling, then the probability of hitting small light sources would be very small. In order to avoid this problem, the illumination of small light sources is directly estimated at each point of the random walk. This technique, which is called *next event estimation* or *direct light source computation*, is also incorporated into both the reference and the new algorithm.

At each visited point number of child rays n_l is computed according to equation 4. If the computed n_l turns out to be less than 1, then the child ray is traced only with probability n_l. If we decide not to trace the child ray, then estimate $\hat{L}$ is used instead. If according to the random decision, we have to trace a child ray, then $(1 - n_l)/n_l \cdot \hat{L}$ is subtracted from the result. On the other hand, if the computed n_l is greater than 1, we find the nearest integer and spawn n_l child rays from this point. The potential passed with a child ray is divided by n_l.

The first problem that needs to be solved is to find an approximation of radiance $\hat{L}$. We could use, for example, a photon map, or a statistical estimation gained during the computation of previous paths. In the implementation we made a direct estimation in the following way [SSKK03].

Suppose that the scene is closed. In this case, we can approximate the average reflected radiance in the scene, which can be regarded as an estimate for $\hat{L}$. Note that we use the reflected radiance here, since the direct illumination is computed separately by next event simulation. The total emitted power of the light sources is

$$\Phi^e = \int_S \int_\Omega L^e(\vec{x}, \omega) \cdot \cos\theta \, d\vec{x} d\omega$$

where S is the set of all surface points, L^e is the emitted radiance and θ is the angle between the direction of the emission and the surface normal. This emitted power will be multiplied by the albedo at each reflection. Suppose that the average albedo in the scene is $\tilde{a}$. The reflected power in the scene is the sum of the single reflection, double reflection, etc., that is:

$$\Phi^r \approx \Phi^e \cdot (\tilde{a} + \tilde{a}^2 + \ldots) = \frac{\tilde{a}\Phi^e}{1 - \tilde{a}}.$$

From the average power, we can obtain the average radiance:

$$\hat{L}(\vec{x}, \omega) \approx \frac{1}{\pi S} \cdot \frac{\tilde{a}\Phi^e}{1 - \tilde{a}}.$$

Formula 4 contains the accumulated potential of the path, W_{l-1}. The computation of W_{l-1} poses no particular problem, as we increase the length of the path, the potential is updated according to equation 2. However, we have to take into account that in the global illumination problem the potential is not scalar, but a vector whose elements correspond to the wavelengths on which the computation is carried out. These vectors are multiplied as *diadic products*, that is, the result is also a vector of the same dimension, whose elements are the products of the respective elements in the two operands.

The albedo showing up in equation 4 is available as a local material property, as well as shininess parameter s_l. Note that the albedo also depends on the wavelength, thus diadic product is applied when it is multiplied with the potential.

The modified versions of equations 4 and 2 for the spectral case, denoting the diadic product by $\circ$ and the luminance of a spectrum by $\mathcal{L}$, is:

$$n_{k,l} = \mathcal{L}(W_{k,l-1} \circ a_{k,l}) \cdot \sqrt{1 + \frac{n_{\max}^2 - 1}{(s_{k,l}+1)^2}},$$

$$W_l = \frac{W_{l-1} \circ w_l(\omega_l^i)}{n_l \cdot p_l(\omega_l^i)}.$$

6 Simulation results

The proposed variance based go with the winner strategy has been implemented in a path tracing algorithm. The results are compared with the classical path tracing applying Russian roulette. The termination probability was set equal to the local albedo. In both algorithms we included direct light source computation (next event simulation) to handle small light sources.

To make the comparison fair, we allowed the two algorithms to use the same number of rays to compute the image. The new method distributed the available rays differently for pixels and for the different levels of recursion, aiming at the goal to place more rays at higher variance domains. We were surprised that when the two methods traced the same number of rays, the go-with-the-winner solution was about 20% faster. A possible explanation is that the new method applies much less recursive calls to generate child rays, and the rays resulted from splitting are much more coherent, thus the new method automatically provides better cache utilization. The rendering times were measured in the open source RenderX.NET [Ant04] global illumination framework, that is a software package written completely in C# targeting the .NET platform.

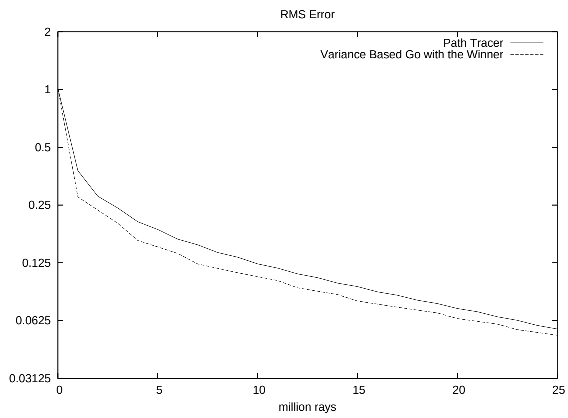
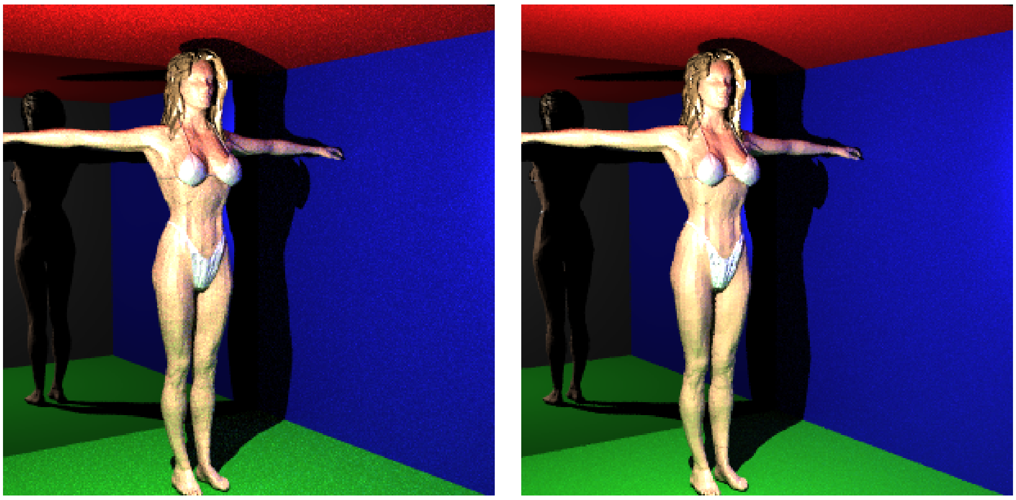


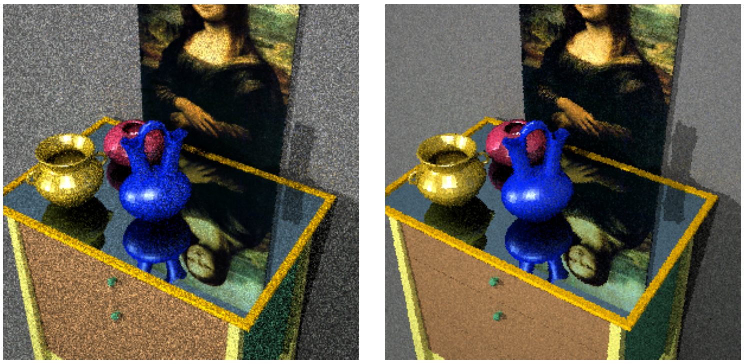
Figure 1: *Relative error curves obtained with the original path tracing algorithm and the proposed method for the Cornell Girl scene.*

The computed images are shown by figures 2 and 3, which demonstrate the superior performance of the new method. On the one hand, examining the error curves (figure 1) we can conclude that the new method can provide the same error level using about 30-50% less rays. The improved image quality is due to several features. The new method distributes the variance evenly in the pixels of the image, and does not devote unnecessary amount of computation to simpler parts. On the other hand, splitting allows to reuse path segments, which also saves time and make the saved time available to generate additional paths.



path tracing | go with the winner

Figure 2: Comparison of classical path tracing with Russian-roulette and path tracing using the go with the winner strategy for a Cornell Girl scene. Both images have been obtained by casting 9 million rays. The image resolution is 300×300.



path tracing
2 million rays, 19 sec

go with the winner
2 million rays, 15 sec

path tracing
10 million rays, 104 sec

go with the winner
10 million rays, 76 sec

go with the winner
111 million rays, reference

Figure 3: Comparison of classical path tracing with Russian-roulette and path tracing using the go with the winner strategy for the “Table with vases” scene. The image resolution is 300×300.

7 Conclusions

This paper proposed an extended go with the winner strategy to improve random walk global illumination algorithms. The basic idea is that at each visited point the variance caused by tracing the next random ray is estimated, and we split or randomly terminate the path to maintain a roughly constant variance in all steps. The variance estimation seems complicated at the first glance, but the implementation of the method is still straightforward. Having a random walk global illumination program, the required modifications are trivial to implement. The simple formula of equation 4 should be included, and based on the result several random rays should be generated, or if it is smaller than 1, this value will be the continuation probability of Russian roulette.

According to our measurements, this simple change can speed up the calculation by about 30-50% due to the better distribution of rays, and other 20% speed up is due to reducing the number of recursive calls and making the rays more coherent.

8 Acknowledgements

This work has been supported by the GameTools FP6-004363 EU project, OTKA ref. No.: T042735, by TIN 2004-07451-C03-01, and by the Spanish-Hungarian Action Fund. The scenes have been modeled by Maya that was generously donated by AliasWavefront.

REFERENCES

[AK90] J. Arvo and D. Kirk. Particle transport and image synthesis. In *SIGGRAPH '90 Proceedings*, pages 63–66, 1990.

[Ant04] Gy. Antal. RenderX.NET. 2004. http://www.sourceforge.com/projects/renderx-net.

[AV94] D. Aldous and U. Vazirani. Go with the winners algorithms. In *Proc. 35th IEEE Symp. on Foundations of Computer Science*, 1994.

[BM97] M. R. Bolin and G. W. Meyer. An error metric for Monte Carlo ray tracing. In *Rendering Techniques '97*, pages 57–68, 1997.

[BSH02] P. Bekaert, M. Sbert, and J. Halton. Accelerating path tracing by re-using paths. In *Proceedings of Workshop on Rendering*, pages 125–134, 2002.

[Chr00] P. Christensen. Faster photon map global illumination. *Journal of Graphics Tools*, 4(3):1–10, 2000.

[Gra01] P. Grassberger. Go with the winners: a general monte carlo strategy. In *Proceedings der CCP2001)*, 2001.

[JC95] H. W. Jensen and N. J. Christensen. Photon maps in bidirectional Monte Carlo ray tracing of complex objects. *Computers and Graphics*, 19(2):215–224, 1995.

[Kah56] H. Kahn. Use of different monte carlo sampling techniques. In *Symposium on Monte Carlo Method*, 1956.

[Kel97] A. Keller. Instant radiosity. *SIGGRAPH '97 Proceedings*, pages 49–55, 1997.

[LW93] E. Lafortune and Y. D. Willems. Bi-directional path-tracing. In *Compugraphics '93*, pages 145–153, Alvor, 1993.

[LW94] E. Lafortune and Y. D. Willems. Using the modified Phong reflectance model for physically based rendering. Technical Report RP-CW-197, Department of Computing Science, K.U. Leuven, 1994.

[SK99] L. Szirmay-Kalos. Stochastic iteration for non-diffuse global illumination. *Computer Graphics Forum (Eurographics'99)*, 18(3):233–244, 1999.

[SSKK03] L. Szécsi, L. Szirmay-Kalos, and C. Kelemen. Variance reduction for russian-roulette. *Journal of WSCG*, 11, 2003.

[Vea97] E. Veach. *Robust Monte Carlo Methods for Light Transport Simulation*. PhD thesis, Stanford University, 1997.

[VG95] E. Veach and L. Guibas. Optimally combining sampling techniques for Monte Carlo rendering. In *ACM SIGGRAPH '95 Proceedings*, pages 419–428, 1995.

[WKB+02] I. Wald, T. Kollig, C. Benthin, A. Keller, and P. Slussalek. Interactive global illumination using fast ray tracing. In *13th Eurographics Workshop on Rendering*, 2002.

[WRC88] G. J. Ward, F. M. Rubinstein, and R. D. Clear. A ray-tracing solution for diffuse interreflection. *Computer Graphics*, 22(4):85–92, 1988.