CHAPTER 31

# Variance Reduction via Footprint Estimation in the Presence of Path Reuse

*Johannes Jendersie*
*Clausthal University of Technology*

## ABSTRACT

Multiple importance sampling is a tool to weight the results of different samplers with the goal of a minimal variance for the sampled function. If applied to light transport paths, this tool enables techniques such as bidirectional path tracing and vertex connection and merging. The latter generalizes the path probability measure to merges—also known as *photon mapping*. Unfortunately, the resulting heuristic can fail, resulting in a noticeable increase of noise. This chapter provides an insight into why things go wrong and proposes a simple-to-implement heuristic that is closer to an optimal solution and more reliable over different scenes. The trick is to use footprint estimates of sub-paths to predict the true variance reduction that is introduced by reusing all the photons.

## 31.1 INTRODUCTION

In light transport simulation there is a multitude of sampling strategies, visualized in Figure 31-1. By tracing paths, beginning at a sensor, we can randomly find light sources in the scene and compute their contributions through the sampled paths. Additionally, next event estimation, the connection toward a known light source, helps to locate smaller light sources. Both sampling strategies together form the *path tracing* (PT) algorithm, introduced by Kajiya [8]. In *bidirectional path tracing* (BPT) [9, 13] all possible connections between a random view sub-path and a random light sub-path provide additional samplers. BPT is able to find caustic paths using the camera connection, something not possible in PT.


E. Haines, T. Akenine-Möller (eds.), *Ray Tracing Gems*, https://doi.org/10.1007/978-1-4842-4427-2_31

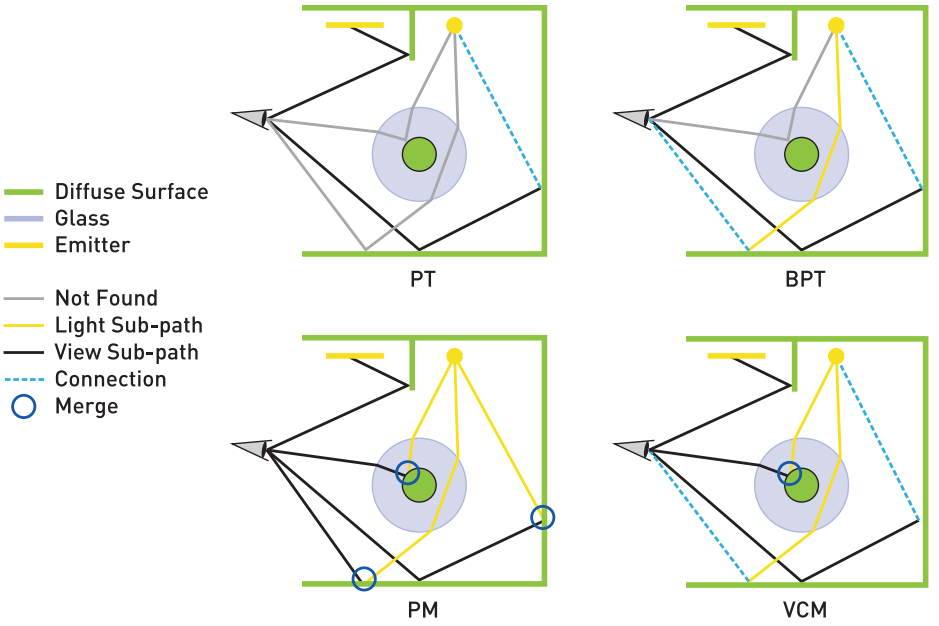


**Figure 31-1.** *Visualization of paths that are found and not found by different methods. The area and the point lights at the ceiling cause different challenging light effects.*

Veach [14] introduced several weighting strategies to combine all these different samplers in an almost-optimal way. The most-used strategy is known as the *balance heuristic*, which reads

$$w_a = \frac{n_a p_a}{\sum_{b \in S} n_b p_b}, \tag{1}$$

where $a$ and $b$ are indices of samplers, from the set $S$ of possible samplers, with probability densities $p_{\{a,b\}}$. If a sampler is applied multiple times, this is accounted for by the factors $n_{\{a,b\}}$. In PT and BPT, $n$ is always one. The result of the balance heuristic is a weight $w$ for each sampler such that all weights on a path sum up to one. For example, if there are different possibilities to find the same path, the weighted average of all options is used.

Another successful method for light transport simulation is *photon mapping* (PM), introduced by Jensen [7]. In a first pass, light sub-paths are traced and their vertices are stored as photons. In a second pass, view sub-paths are generated and nearby photons are merged with the current path. This introduces a small bias but is capable of finding even more light effects (reflected and refracted caustics) than BPT. Georgiev et al. [3] as well as Hachisuka et al. [4] derived a path probability that makes merges (a.k.a. photon mapping) compatible with Veach's MIS weights. Since all $n_\Phi$ photons can be found at each view sub-path vertex, there is a high amount

of path reuse. This decreases the variance of merges, which is modeled by setting $n = n_\Phi$ in the balance heuristic (Equation 1) for the merge samplers. The combined algorithm consisting of BPT and PM is called *vertex connection and merging* (VCM) [3].

Unfortunately, the choice of $n_\Phi$ in the balance heuristic can cause severe miscalculations with respect to variance, as first observed by Jendersie et al. [6]. Figure 31-2 demonstrates a scene with noticeable problems. While, compared to BPT, VCM only adds new samplers and therefore should only decrease the variance, it shows more noise. In most scenes this effect is not as noticeable as in the chosen example. However, using our new heuristic reduces the variance in any scene compared to the previous heuristic, leading to faster convergence. We call our method *optimized VCM* (OVCM).

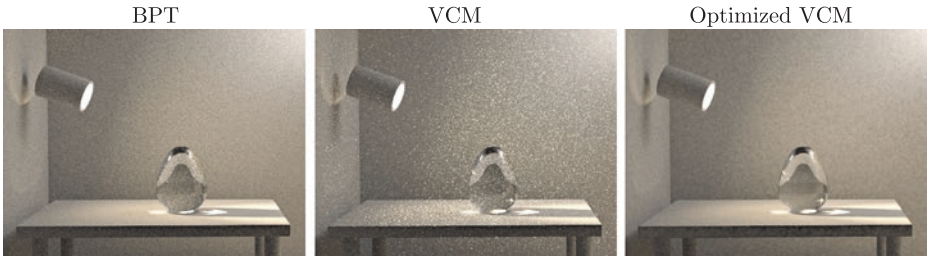


**Figure 31-2.** *Closeups of Veach's BPT test scene with 50 samples each. BPT outperforms VCM due to a nonoptimal MIS weight. OVCM improves the merge weights and achieves a lower variance.*

## 31.2 WHY ASSUMING FULL REUSE CAUSES A BROKEN MIS WEIGHT

The variance of a complete transport path consists of the variances of both the view and light sub-paths. Mathematically, the variance of the entire path is a product of two random variables $X$ (without loss of generality, the view sub-path) and $Y$ (the light sub-path):

$$V[XY] = V[X]E[Y]^2 + V[Y]E[X]^2 + V[X]V[Y]. \tag{2}$$

In addition to the variances of the two sub-paths, the total variance depends on the expected values $E[X]$ and $E[Y]$, which at the vertex are the incoming radiance ($E[Y]$) and its adjoint, the incoming importance ($E[X]$).

By using $n_\Phi$ photons, the variance of the light sub-paths in a merge is reduced by a factor of $n_\Phi$. However, if most of the total variance stems from the view sub-path, the true gain for the path's variance is much smaller than $n_\Phi$. Using $n_\Phi$ photons will reduce $V[Y]$, and so we divide only the second and third terms in Equation 2 by this factor. Therefore, if the total variance is dominated by the first term, $V[X]E[Y]^2$, then

the variance with respect to path reuse will not change much and using $n_\Phi$ directly will cause a severe overestimation of the event's likelihood.

## 31.3 THE EFFECTIVE REUSE FACTOR

To integrate the preceding observation into the balance heuristic, we want to change the factor $n$ for each of the merge samplers. The optimal factor tells us how much the variance of the full path is reduced when merging multiple sub-paths. Starting at the variance property from Equation 2, the true effective use of a merge sampler is

$$n = \frac{V[X]E[Y]^2 + V[Y]E[X]^2 + V[X]V[Y]}{V[X]E[Y]^2 + \frac{1}{n_\Phi}\left(V[Y]E[X]^2 + V[X]V[Y]\right)}, \tag{3}$$

which is the quotient of the variance with and without using the light sub-path sampler $n_\Phi$ times. Naturally, Equation 3 falls back to $n = 1$ if $n_\Phi = 1$. Also, $n = n_\Phi$ is reached only if we have zero view-path variance. This theoretical solution still lacks applicability because we need stable estimates of $V$ and $E$ based on the single sub-path sample we have at hand.

Jendersie et al. [6] used an additional data structure to query the expected number of photons (an estimate of $E[Y]$). Further, their method VCM* applied the estimate in a different way than shown here. However, there are several problems with an approach that uses a dedicated data structure: noise within the estimator itself, discretization artifacts, falsely counting photons from different paths into $E[Y]$, and, finally, scalability problems for large scenes. We present a new, more direct heuristic in the following.

### 31.3.1 AN APPROXIMATE SOLUTION

The most difficult question is, "What are the quantities $V$ and $E$ of a path?" A path consists of multiple Monte Carlo sampling events and one connection or merge event. In the case where the sampling densities $p$ and the target function $f$ are proportional, the ratio $f/p$, which is calculated in a Monte Carlo integrator, is constant for any sample, and thus the variance of the estimator is zero [14, Section 2.2]. If $p$ and $f$ are almost proportional, the variance of a Monte Carlo sampler is close to zero: $V \approx 0$.

In importance sampling, we choose sampling probability density functions (PDFs) $p$ proportional to the BSDF, but our target function is the BSDF multiplied by the incoming radiance. Mathematically, we can divide this product into two subproblems

by using Equation 2. Consider the example of direct lighting: We have the incoming radiance as the expected value of one random variable with zero variance ($E[Y] = L_i$, $V[Y] = 0$) from the deterministic direct light computation and the sampled view sub-path with $E[X] = 1$ (by the design of the pinhole camera model) and $V[X] = \varepsilon$ (because of jittering in a pixel). The final variance is then $E[Y]^2V[X] = L_i\varepsilon$, which can be large, even if the sampler is close to the BSDF (or camera model, in this case).

The same applies to longer paths. Each of the two sub-paths contains multiple Monte Carlo sampling events, whose variance can be approximated with a small $\varepsilon$. To approximate the incoming radiance and importance, we need a density per square meter and steradian for each sub-path. Alternatively, we can also compute the inverse of a density—the footprints $A[X]$ and $A[Y]$. See Figure 31-3. Conceptually, the footprint of a path is the projection of a pixel or a photon onto some surface after some number of bounces.

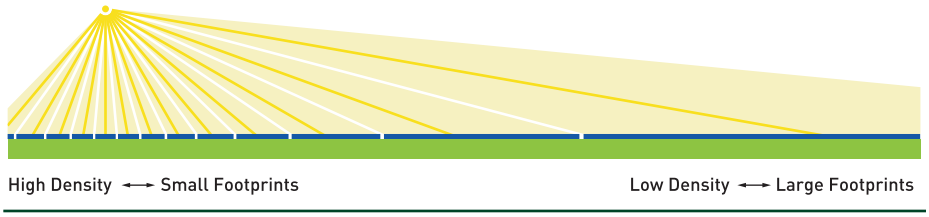


**Figure 31-3.** *Analogy between footprint size (blue) and density. The higher the density p, the smaller the footprint A of each particle: $A \propto 1/p$.*

Assuming we can compute this footprint $A$ and that the samplers have small variances $\varepsilon$, Equation 3 becomes

$$n \approx \frac{\varepsilon \frac{1}{A[Y]^2} + \varepsilon \frac{1}{A[X]^2} + \varepsilon^2}{\varepsilon \frac{1}{A[Y]^2} + \frac{1}{n_\Phi}\left(\varepsilon \frac{1}{A[X]^2} + \varepsilon^2\right)}$$

$$\frac{1}{A} \gg \varepsilon \Rightarrow n \approx \frac{\frac{1}{A[Y]^2} + \frac{1}{A[X]^2}}{\frac{1}{A[Y]^2} + \frac{1}{n_\Phi}\frac{1}{A[X]^2}}$$

$$= \frac{A[X]^2 + A[Y]^2}{A[X]^2 + \frac{1}{n_\Phi}A[Y]^2}. \tag{4}$$

In the second line we replaced $\varepsilon^2$ with zero, under the assumption that this third term is dominated by the other two. The $\varepsilon$ in front of the other terms cancels out naturally. Finally, we obtain an approximation of the optimal effective reuse factor $n$ if we can provide the footprint of each sub-path.

### 31.3.2 ESTIMATING THE FOOTPRINT

For our application, the most important thing about the footprint is to capture events that change the density noticeably. Rough surfaces cause one such event, which means that it is essential to include the BSDF in the computation.

Estimates were researched previously and used for antialiasing [5] (details in Chapter 20) as well as adaptive reconstruction kernels [2]. To estimate the size of a projected pixel, Igehy introduced *ray differentials* [5]. These are capable of estimating the anisotropic footprint after multiple specular interactions, but they lack any handling of rough BSDFs. Suykens et al. [12] introduced a heuristic treatment of BSDFs to Igehy's ray differentials, calling them *path differentials*. However, their approach requires an arbitrary scale parameter that is hard to determine. Schjøth et al. [11] explored what they call *photon differentials*, which are essentially ray differentials used for photons. Again, a treatment of BSDFs is missing and only specular bounces are handled. The most convenient solution so far is *5D covariance tracing* from Belcour et al. [2], which contains the first proper handling of BSDFs but is expensive to compute and store.

Inspired by covariance matrices, we developed a simplified heuristic. Since we are interested in only the area of the footprint and not in its anisotropic shape, it suffices to store and update two scalar values: the searched area $A$ and a solid angle $\Omega$, which is used to derive the change of the area. Like in covariance tracing, we use convolutions to model the change of the footprint by interaction events.

A heuristic similar to ours was used by Bekaert et al. [1], in their Equation 7, to estimate the kernel size for a photon mapping event. The difference to our heuristic is that Bekeart et al. used only the previous PDF to update $A$. We additionally introduce the cumulative solid angle $\Omega$ that is based on all previous PDFs.

Beginning on any point in a source area, the next path segment will have a footprint of its own. The segment footprint can be computed, assuming the traced segment is a cone with solid angle $\Omega$. A convolution is required to combine any source position with the target area. It is performed by adding the square root of the segment footprint and the source area and squaring their sum again. This is depicted in Figure 31-4 at the bottom, assuming a circular footprint (note that $\pi$ cancels out in the final result). The same result is obtained if we convolve two squares or any other regular polygon. So, the next question is, how to obtain the two areas?

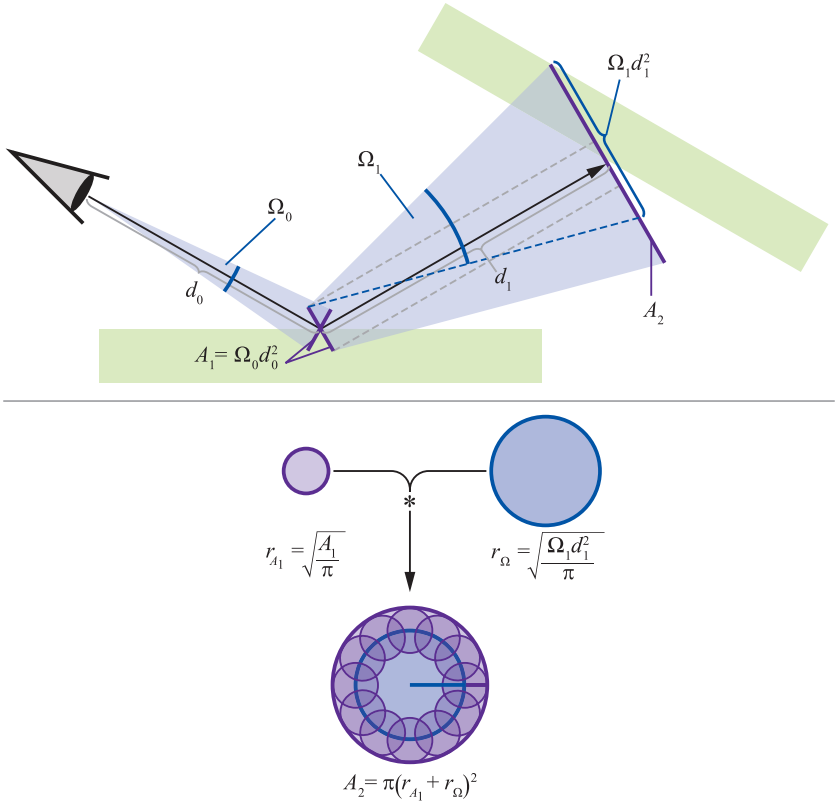


**Figure 31-4.** *Our footprint heuristic. Due to BSDF scattering the solid angle grows with each interaction ($\Omega_1 > \Omega_0$). The footprint size depends on the previous area as well as $\Omega d^2$ for the new scattering. The combination of the two areas is achieved by a convolution (bottom).*

The source area is one of the following:

- The area of the light source for the first vertex on a light path.
- The area of the sensor in realistic camera models.
- A projection of the incoming area toward the outgoing direction.

The third option applies to all intermediate vertices. Tests have shown that simply using $A_{out} = A_{in}$ resulted in the best MIS weight. The alternative, to divide the area by the incoming cosine and to multiply by the outgoing cosine (i.e., using real projections), turned out to be slower and to have lower quality. The simple copy is reasonable because for our application the footprint should always be growing in diffuse scattering events. Using the previous area guarantees a monotonic function.

Next, we need an estimate of the footprint of a single path segment. This is given by the directional sampler's density $p$ used in any of the events (light sources, cameras, and intermediate vertices). The sampling density of a direction has the unit $\mathrm{sr}^{-1}$. Inverting it gives us a solid angle $\Omega = 1/p$ of the sample. Similar to ray differentials, the previous angular variance must be considered. Therefore, we apply a convolution of the solid angles, which gives

$$\Omega_k = \left( \sqrt{\Omega_{k-1}} + \frac{1}{\sqrt{p}} \right)^2. \tag{5}$$

Finally, the incoming area at the new vertex can be computed with

$$A_k = \left( \sqrt{A_{k-1}} + \sqrt{\Omega_{k-1} d_{k-1}^2} \right)^2, \tag{6}$$

where $\Omega d^2$ is the area of a spherical cap with solid angle $\Omega$ at the distance $d$, i.e., the footprint from the scattering on the last path segment. This is also shown in the top of Figure 31-4.

So far, the footprint assumes that all paths start at the same source. If there are multiple light sources (or cameras), a photon (or importon) is emitted with a probability of $p_0 < 1$. This leads to an increase of the area, and the final footprint of sub-path $X$ becomes

$$A[X] = \frac{A_{k,X}}{p_{0,X}}. \tag{7}$$

The newly derived footprint heuristic in this section has to be used with care. It is based on geometrical observations and lacks any kind of curvature-based changes or anisotropy, which are worth exploring in the future. So far, this heuristic focuses mainly on the parts that are required by our variance estimation. Therefore, it complements the previous ray differentials approach, but it does not replace it. Our heuristic would fail if used for antialiasing or adaptive photon mapping.

Another difficulty is the generalization to volume transport. To begin, it seems straightforward to use Equation 5 for scattering events, too. However, it would be necessary to track the spread along the ray direction, depending on the density of the medium. That is, it is necessary to track a volume instead of the area $A$.

## 31.4 IMPLEMENTATION IMPACTS

Having an estimate for the footprint area of each sub-path in a merge, we can use Equation 4 to estimate the proper multiplier $n$ for the MIS weight. To accomplish this, we store two float values $\sqrt{\Omega}$ and $\sqrt{A}$ per vertex. The direct use of these square roots avoids the repeated square root in Equations 5 and 6. Further, we found that dividing by $\sqrt{p}$ may lead to severe numerical problems by quickly exceeding the range of float or double precision for $\Omega_k$. Therefore, we introduced an $\epsilon = 1 \times 10^{-2}$ into the quotient. Additionally, Equation 4 can be reordered to obtain a more robust solution. The final computed values in the implementation are

$$\sqrt{\Omega_k} = \sqrt{\Omega_{k-1}} + \frac{1}{\epsilon + \sqrt{p}}, \tag{8}$$

$$\sqrt{A_k} = \sqrt{A_{k-1}} + d_{k-1}\sqrt{\Omega_{k-1}}, \tag{9}$$

$$n_k = \frac{\left(\frac{p_{0,Y}}{p_{0,X}}\left(\frac{\sqrt{A_{k,X}}}{\sqrt{A_{k,Y}}}\right)^2\right)^{2+c} + 1}{\left(\frac{p_{0,Y}}{p_{0,X}}\left(\frac{\sqrt{A_{k,X}}}{\sqrt{A_{k,Y}}}\right)^2\right)^{2+c} + \frac{1}{n_\Phi}}. \tag{10}$$

In Equation 10 we added an artificial constant $c$ to the exponent. Similar to the exponents in the power heuristic which amplify a decision, this increases the differentiation between the sub-path estimates. We found that using $c = 0.5$ improved the results for rough surfaces and remained almost the same for other events. Therefore, we use this artificial modification in all the following experiments.

### 31.4.1 PERFORMANCE CONSEQUENCES

Unfortunately, the sub-paths must be iterated completely for each MIS-weight computation to obtain the sum of all other events. In the usual weight computation this can be optimized by storing partial sums for the sub-paths. This optimization is possible only because the path probabilities are pure products that share a lot of terms [10, p. 1015ff]. With the sums in $n_k$, this optimization is not possible anymore, leading to a loss of performance. Compared to a fast standard implementation with constant cost, the cost with our heuristic becomes linear in path length. In practical tests this resulted in a loss of 0.3%–3%, depending on the scene. The impact is higher if the scene is simple (low tracing and low material evaluation

cost) or the paths are long. For more realistic complex scenes, the impact is often below 1%.

Since the fast implementation stores a different set of values per vertex, VCM and our OVCM require the same amount of memory.

## 31.5 RESULTS

As shown in the previous section, our new heuristic is simple to implement and has better performance and lower memory consumption than the previous solution, VCM* [6]. Nonetheless, it is equally capable of avoiding the overestimated merge importance. Figures 31-5 and 31-6 show a convergence comparison of five scenes rendered with different methods. In Figure 31-5 the full renderings are given for reference. Figure 31-6 shows the square root of the sample variance over a high number of iterations for different algorithms. In this visualization darker is better. A black image would reveal a perfect estimator with zero variance. Since the variance often scales with the brightness of the image, the standard deviation images look like contrast enhanced versions of the rendering itself.

**Figure 31-5.** *The full renderings associated with the zoomed images in Figure 31-6.*

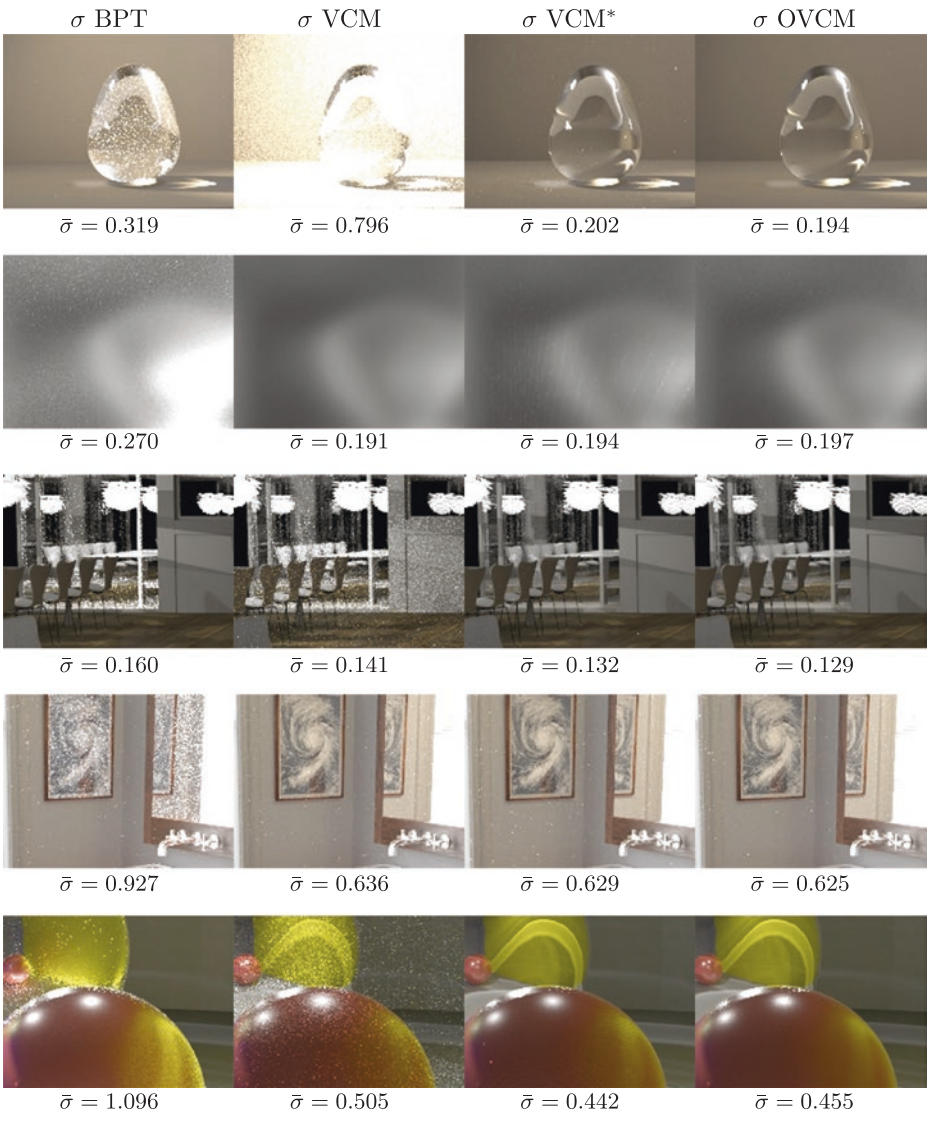


**Figure 31-6.** *Here we show closeups of the standard deviation σ (darker is better) of the full renderings in Figure 31-5 over a high number of iterations (10k–65k). The average sample standard deviation for the entire image is given by* $\overline{\sigma}$ .

In all cases VCM* and OVCM perform similarly. Both of them are superior to BPT or VCM. However, they are still not optimal, which becomes noticeable in a direct comparison: Different surfaces are darker in either of the two approaches. An optimal solution would have the lowest variance everywhere. Since these differences are barely visible, the average standard deviations give a better comparison. According to these numbers, VCM* and OVCM are equally good on average.

There is also room for improvement. Equation 3 formulates the optimal solution. Our proposed heuristic completely removes all variances $V$ and approximates the expected values $E$ with another heuristic, which, for example, neglects the curvature or texture changes in the footprint area. By improving the estimates of $V$ and $E$, we expect that the artificial $c = 0.5$ from Equation 10 will become redundant and that the heuristic will be closer to the optimum.

## ACKNOWLEDGMENTS

Most scenes shown here are taken from the repository of *Physically Based Rendering* (third edition) [10]. The colorful balls scene is courtesy of T. Hachisuka.

## REFERENCES

[1] Bekaert, P., Slusallek, P., Cools, R., Havran, V., and Seidel, H.-P. A Custom Designed Density Estimator for Light Transport. Research Report MPI-I-2003-4-004, Max-Planck Institut für Informatik, 2003.

[2] Belcour, L., Soler, C., Subr, K., Holzschuch, N., and Durand, F. 5D Covariance Tracing for Efficient Defocus and Motion Blur. *ACM Transaction on Graphics 32*, 3 (July 2013), 31:1–31:18.

[3] Georgiev, I., Křivánek, J., Davidovič, T., and Slusallek, P. Light Transport Simulation with Vertex Connection and Merging. *ACM Transactions on Graphics (SIGGRAPH Asia) 31*, 6 (2012), 192:1–192:10.

[4] Hachisuka, T., Pantaleoni, J., and Jensen, H. W. A Path Space Extension for Robust Light Transport Simulation. *ACM Transactions on Graphics (SIGGRAPH Asia) 31*, 6 (Nov. 2012), 191:1–191:10.

[5] Igehy, H. Tracing Ray Differentials. In *Proceedings of SIGGRAPH* (1999), pp. 179–186.

[6] Jendersie, J., and Grosch, T. An Improved Multiple Importance Sampling Heuristic for Density Estimates in Light Transport Simulations. In *Eurographics Symposium on Rendering EI&I Track* (July 2018), pp. 65–72.

[7] Jensen, H. W. Global Illumination Using Photon Maps. In *Eurographics Workshop on Rendering* (1996), pp. 21–30.

[8] Kajiya, J. T. The Rendering Equation. *Computer Graphics (SIGGRAPH)* (1986), 143–150.

[9] Lafortune, E. P., and Willems, Y. D. Bi-Directional Path Tracing. In *Conference on Computational Graphics and Visualization Techniques* (1993), pp. 145–153.

[10] Pharr, M., Jakob, W., and Humphreys, G. *Physically Based Rendering: From Theory to Implementation*, third ed. Morgan Kaufmann, 2016.

[11] Schjøth, L., Frisvad, J. R., Erleben, K., and Sporring, J. Photon Differentials. In *Computer Graphics and Interactive Techniques* (Dec. 2007), pp. 179–186.

[12] Suykens, F., and Willems, Y. D. Path Differentials and Applications. In *Eurographics Workshop on Rendering* (June 2001), pp. 257–268.

[13] Veach, E., and Guibas, L. J. Bidirectional Estimators for Light Transport. In *Photorealistic Rendering Techniques*. Springer, 1995, pp. 145–167.

[14] Veach, E., and Guibas, L. J. Optimally Combining Sampling Techniques for Monte Carlo Rendering. In *Proceedings of SIGGRAPH* (1995), pp. 419–428.