

Adjoint-Driven Russian Roulette and Splitting in Light Transport Simulation

Jiří Vorba*
Charles University in Prague
Weta Digital, Wellington

Jaroslav Krivánek†
Charles University in Prague

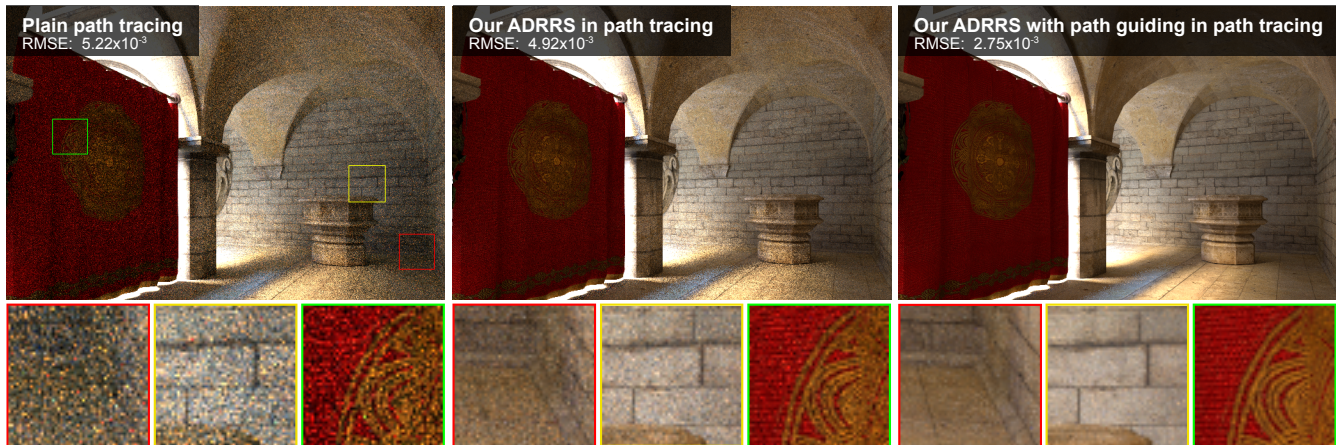


Figure 1: Russian roulette based on material reflectance, as traditionally applied in computer graphics, leads to suboptimal results in scenes with non-uniform light distribution or complex visibility (left). Our adjoint-driven Russian roulette and splitting (ADRRS) significantly increases the efficiency in such cases (middle). ADRRS complements the advantages of path guiding (PG) methods [Vorba et al. 2014] and in conjunction they provide superior results than either method alone (right). All images have been rendered by path tracing in 1 hour.

Abstract

While Russian roulette (RR) and splitting are considered fundamental importance sampling techniques in neutron transport simulations, they have so far received relatively little attention in light transport. In computer graphics, RR and splitting are most often based solely on local reflectance properties. However, this strategy can be far from optimal in common scenes with non-uniform light distribution as it does not accurately predict the actual path contribution. In our approach, like in neutron transport, we estimate the expected contribution of a path as the product of the path weight and a pre-computed estimate of the adjoint transport solution. We use this estimate to generate so-called weight window which keeps the path contribution roughly constant through RR and splitting. As a result, paths in unimportant regions tend to be terminated early while in the more important regions they are spawned by splitting. This results in substantial variance reduction in both path tracing and photon tracing-based simulations. Furthermore, unlike the standard computer graphics RR, our approach does not interfere with importance-driven sampling of scattering directions, which results in superior convergence when such a technique is combined with our approach. We provide a justification of this behavior by relating our approach to the zero-variance random walk theory.

Keywords: Russian roulette, splitting, light transport, importance sampling, zero-variance schemes.

Concepts: •Computing methodologies → Rendering;

*e-mail:jirka@cgg.mff.cuni.cz

†e-mail:jaroslav.krivanek@mff.cuni.cz

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. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request

1 Introduction

Recently, Monte Carlo light transport simulation has been adopted by the movie industry as a standard tool for producing photo-realistic imagery [Seymour 2014]. Since the computation of a noise-free movie frame can easily take tens of hours, various approaches to improve computation efficiency have received much attention [Křivánek et al. 2014b].

Two classic techniques to improve efficiency, *Russian roulette* (RR) and *splitting*, have been used in computer graphics for over 25 years [Arvo and Kirk 1990]. RR aims to save computation time by terminating transport paths with small contribution, while splitting (a.k.a. distributed ray tracing [Cook et al. 1984]), branches paths into several independent trajectories. In computer graphics, RR decisions have usually been based on local surface reflectivity [Jensen 2001; Dutré et al. 2006; Jakob 2010] or on the accumulated path weight (a.k.a. throughput) [Arvo and Kirk 1990; Jensen 1996; Veach 1997]. Splitting often relies on heuristics based on local BRDF roughness [Szirmay-Kalos and Antal 2005]. While simple to implement, these *local* approaches do not work well in scenes with non-uniform light distribution, as illustrated in Fig. 1.

The sub-optimal performance of RR and splitting – as currently employed in light transport simulation – is due to the respective decisions being oblivious to the actual distribution of light (when tracing paths from the camera) or visual importance (when tracing paths from the light sources). For example, when the surface reflectivity is used as the termination probability in RR, effort is often spent on sampling long paths on bright surfaces, while dark surfaces suffer from high variance due to early path termination. The latter is particularly acute in scenes with difficult visibility or dense participating media, where most contributions are due to long paths that

may absorb lots of energy but originate at bright sources. Unfortunately, the problem cannot be solved simply by making RR less aggressive across the entire scene, because doing so would waste resources on sampling longer paths even where not necessary.

We present *adjoint-driven Russian roulette and splitting* (ADRRS), a new approach for RR and splitting decisions. To address the shortcomings of the current approaches, we terminate or split paths according to an estimate of their *total expected contribution* to the image, relative to a *reference solution*. Paths with an expected contribution much higher than the reference are split, while paths with a low expected contribution have higher chance of being terminated. The expected contribution is calculated as a product of the current path weight and an estimate of the adjoint transport solution (i.e. equilibrium radiance for camera sub-paths or equilibrium visual importance for light sub-paths), which we pre-compute and cache in the scene. We show that this approach leads to good importance sampling of the path space, and, in turn, it can significantly increase the overall efficiency of the simulation.

Our method works in synergy with path guiding methods [Vorba et al. 2014; Bashford-Rogers et al. 2012; Hey and Purgathofer 2002; Jensen 1995], and in conjunction they provide superior results. This is an important advantage over the traditional local RR decisions, which counteract the guiding methods by terminating potentially important paths before they can even make a contribution. We provide a theoretical justification of this beneficial behavior based on the theory of zero-variance random walks [Křivánek and d’Eon 2014; Xu et al. 2001; Kalos 1963], which, as we show, is the basis of both the proposed method as well as the guiding schemes. In summary, we make the following contributions:

- We propose *adjoint-driven Russian roulette and splitting* (ADRRS) where paths are terminated or split according to their expected contribution to the image (Sec. 4).
- We provide a theoretical analysis of the close relation of ADRRS to the zero-variance random walk schemes, which explains its variance reduction properties (Sec. 7).
- We develop a solution for obtaining the paths’ expected contribution necessary to use ADRRS in practice (Sec. 5).
- We show that ADRRS can improve the efficiency of path guiding methods (Sec. 6).

2 Related Work

Particle transport. The origins of simulating the transport of neutral particles (e.g. neutrons or photons) by Monte Carlo (MC) processes goes back to the 1940s. The idea of using MC stems from the probabilistic nature of the particles’ behavior [Spanier and Gelbard 1969]. In the real world, a particle has a certain probability of being emitted in a given time interval and its further fate is also governed by probabilistic events: collisions, absorption and scattering. If a simulated particle follows the events according to the precise physical probabilities, the MC process is said to be *analog*. In such a simulation, all particles have equal, unit statistical *weight*. In order to improve computation efficiency, *non-analog simulations* can be designed by altering the probabilities of various events in the simulation. The particle weights are then modified upon each event so that the simulation remains unbiased [Lux and Koblinger 1991].

The close relation between MC light transport simulation in computer graphics and MC processes simulating the transport of neutral particles has been pointed out by several authors [Křivánek and d’Eon 2014; Christensen 2003; Veach 1997; Arvo and Kirk 1990]. While light transport is described by the rendering equation [Kajiya 1986], particle transport in physics is governed by the linear Boltzmann equation. The similarities between the two were first pointed out by Arvo and Kirk [1990], which allowed them to adopt useful

techniques, such as Russian roulette (RR) and splitting [Kahn 1956; Kahn and Harris 1951] in computer graphics.

Russian roulette and splitting. Arvo and Kirk [1990] suggest to use RR in non-analog simulations for unbiased termination of particles with low *weights*. If the weight drops below a threshold, the particle path is terminated with a certain probability. They also identify distributed ray tracing [Cook et al. 1984] to be an instance of *splitting* and discuss the circumstances under which splitting can reduce variance. However, they fail to propose a practical method without an exponential branching of the ray-tree. Jensen [1996] proposes to use material absorption as the RR termination probability in photon tracing to minimize photon weight fluctuations.

Some works in computer graphics derive the termination and splitting rates by direct optimization of *efficiency* (i.e. reciprocal of the product of variance and computation time). For example, Szirmay-Kalos and Antal [2005], using a series of simplifying assumptions, arrive at a RR/splitting heuristic based on local BRDF reflectivity and roughness, and user-specified global constants. Bolin and Meyer [1997] derive optimal termination and splitting rates through a variance analysis of nested estimators, but they do not describe a working method based on these results. In contrast, we provide a practical algorithm that relies on a well-founded theory, albeit not on the optimization of efficiency. While Veach [1997] proposes efficiency-optimized RR for light path connections in bidirectional path tracing, our work describes RR in a more general context.

Importance sampling. Unlike in graphics, in the neutron transport literature, RR and splitting are understood as *importance sampling* techniques [Veach 1997; Hammersley and Handscomb 1964; Spanier and Gelbard 1969]. For example, to reliably estimate the radiation escaping through a nuclear reactor shield, it is impractical to use an analog simulation since the probability of penetrating the thick shield by a particle is extremely low ($\leq 10^{-9}$). To solve similar problems in the nuclear engineering practice, users of the MC simulators define, usually semi-automatically, an *importance function* over the domain of interest [Wagner and Haghghat 1998]. The simulation then terminates particles in the parts of the domain designated as unimportant, while splitting them in high-importance regions. This strategy effectively adapts the number of surviving particles to the user-specified importance.

As mentioned above, a common practice in computer graphics is to drive RR decisions by the particle weight [Veach 1997; Jensen 1996; Arvo and Kirk 1990]. However, doing so results in poor importance sampling, because no information on the expected future behavior of the particle is taken into account. In our work, we show that – rather than relying solely on the particle weight – it is beneficial to drive RR and splitting also by the *adjoint* quantity (*radiance* when tracing a particle from the camera, and *visual importance* when starting from the light sources). This adjoint quantity value gives us an estimate of the path’s expected future behavior, which – when multiplied by the path weight – provides the expected total contribution of the path to the solution.

There is a vast body of work that uses adjoints in rendering surveyed by Christensen [2003]. Two closely related works are that of Keller and Wald [2000] and Georgiev and Slusallek [2010]. They both use importance-driven RR to randomly decide about depositing a photon and a virtual point light, respectively, while they use classical RR based on local reflectance properties during path construction. Consequently, they may need to sample a tremendous number of paths to achieve low variance in visually important regions with low illumination. Our work, in contrast, address this issue by employing adjoint-driven RR during path construction itself to directly influence length of sampled paths. The works of Szécsi et al. [2003] and Szirmay-Kalos and Antal [2005] reduce variance

due to RR by returning an irradiance estimate upon path termination (instead of the usual zero). In contrast, we exploit a similar estimate to compute optimal termination and splitting rate itself.

Path guiding methods. A direct approach to distributing paths according to visual importance is importance sampling of emission from light sources [Vorba et al. 2014; Bashford-Rogers et al. 2013; Dutré and Willems 1994], and of scattering directions during incremental path construction [Vorba et al. 2014; Bashford-Rogers et al. 2012; Hey and Purgathofer 2002; Jensen 1995]. In this way, paths are directly guided towards regions with high contribution to the computed image. Such guided path sampling is a non-analog simulation that typically leads to high local variation of particle weights [Vorba et al. 2014; Keller and Wald 2000; Suykens and Willems 2000]. In contrast, our adaptive RR and splitting achieves more balanced weights locally than the guiding methods.

Weight window. In neutron transport simulations, RR and splitting are combined in one variance reduction tool called the *weight window* [Hoogenboom and Légrády 2005; X-5 Monte Carlo team 2003; Booth and Hendricks 1984]. This technique is designed to keep the particle weight within a certain interval that may vary over the simulation domain. This interval can be user-specified or based on an automatic importance computation [Wagner and Haghghat 1998; Wagner 1997; Booth and Hendricks 1984]. If a particle weight is below or above the interval bound, RR or splitting is applied, respectively, so that the particle weight stays within the interval. Vorba et al. [2014] employs the weight window facility but the bounds are set manually and are not adapted to the scene. We achieve a significant performance gain by setting the weight window bounds according to a spatially and directionally-varying radiance or visual importance solution.

Booth and Hendricks [1984] set the interval bounds so that the weight window center is inversely proportional to an importance function. This keeps more particles in important regions and less in unimportant ones. The normalization constant for computing the weight window bounds from the importance is set heuristically so that particles are within the weight window immediately upon their emission. Based on an analysis of zero-variance random walk schemes, Wagner and Haghghat [1998; 1997] suggest a theoretically founded approach where an estimate of the final solution serves as the said normalization constant. Adopting this approach allows us to keep particle/path weights around the optimal levels.

While Booth and Hendricks [1984] compute the importance function solely by *forward* particle tracing, we follow Vorba et al. [2014] and use *interleaved* particle tracing from both the camera and light sources. This yields more reliable radiance or visual importance estimates even in scenes with difficult visibility.

Go-with-the-winners. Szirmay-Kalos and Antal [2005] base RR and splitting rate on efficiency analysis and use a heuristic variance approximation based on local material properties and scene-dependent parameters. They introduced the term “go-with-the-winners” to computer graphics that is often used in a general MC context to refer to RR and splitting. Note that Grassberger [2002] points out there is no fundamental difference between RR and

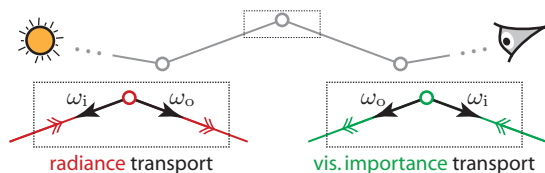


Figure 2: By convention, the direction ω_o is always aligned with the direction of the transported quantity.

splitting as described by Kahn [1951; 1956] and the “go-with-the-winners” strategy. The term was coined by Aldous and Vazirani [1994] as a mean for population control in randomized optimization algorithms.

3 Background

Light transport in a scene without participating media is described by the *rendering* [Kajiya 1986] or, alternatively, by the *visual importance* [Spanier and Gelbard 1969; Veach 1997, Sec. 3.7] transport equations (see also the supplemental). In the following text, we describe how the transport equations can be solved by particle tracing.

Particle tracing. By tracing M *visual importance* particles from the camera, we can estimate the pixel value I using the following MC estimator [Veach 1997]:

$$\langle I \rangle = \frac{1}{M} \sum_k \nu_i(\mathbf{y}_k, \omega_k) L_o^e(\mathbf{y}_k, \omega_k). \quad (1)$$

The above sum is updated when a particle k with its *weight* $\nu_i(\mathbf{y}_k, \omega_k)$, coming from a direction ω_k , collides at a location \mathbf{y}_k . In fact, the particle contributes to the sum only when the self emitted outgoing radiance $L_o^e(\mathbf{y}_k, \omega_k)$ is non-zero. Note that the above estimator corresponds to unidirectional *path tracing* [Kajiya 1986] without explicit connections to light sources (next event estimation); extension to next event estimation is discussed in Sec. 5.4.

The particle weight is given by the product of the emitted visual importance [Veach 1997, p. 91], and the bidirectional scattering distribution functions (BSDFs) and geometric factors along the particle path, divided by the probability density (pdf) of generating the path. To describe the way in which the weight is updated during a collision at some surface point \mathbf{y} , we distinguish between the *incident* weight $\nu_i(\mathbf{y}, \omega_i)$ just before the collision, the weight after application of RR/splitting $\hat{\nu}(\mathbf{y}, \omega_i)$, and the *outgoing* weight $\nu_o(\mathbf{y}, \omega_o)$, just after the scattering. The last two are related by the weight update formula

$$\nu_o(\mathbf{y}, \omega_o) = \hat{\nu}(\mathbf{y}, \omega_i) \frac{f_s(\mathbf{y}, \omega_o \rightarrow \omega_i) |\cos \theta_o|}{p(\omega_o | \mathbf{y})}, \quad (2)$$

where $p(\omega_o | \mathbf{y})$ is a directional pdf for sampling the scattering direction ω_o , f_s denotes BSDF and the arrow notation in f_s marks the direction of light flow. The outgoing weight ν_o after one collision then enters the next collision as its incoming weight ν_i . Extension for participating media would include attenuation between \mathbf{y} and the next collision, which we leave for future work.

Let us emphasize that, throughout the paper, we use a convention, depicted in Fig. 2, that ω_o always points in the direction of the transported quantity. Thus Eq. (2) also holds for *light tracing* [Dutré and Willems 1994] with the minor modification that the arguments of the BSDF need to be swapped.

Russian roulette and splitting. To avoid sampling infinite particle paths, tracing of a particle can be terminated using Russian Roulette: At any collision, the particle survives with a given probability $P_{\text{surv}} > 0$ and its weight is divided by P_{surv} to keep the estimator unbiased. Similarly, the particle path can be split into n independent paths, and dividing the particle weight by n again keeps the resulting estimator unbiased.

Various approaches have been proposed in computer graphics to determine the survival probability or the splitting factor as discussed in Sec. 1 and 2. An important contribution of this paper is an approach to Russian roulette and splitting where the survival probability and the splitting factor are determined in a manner that yields significant variance reduction.

4 Adjoint-Driven RR and Splitting

In this section, we describe the theory behind our *adjoint-driven Russian roulette and splitting* (ADRRS) approach to termination and splitting along incrementally sampled paths. Practical rendering algorithms based on this approach are described in Sec. 5 and 6. In Sec. 7, we show its variance reduction properties and also that our approach is motivated by zero-variance sampling schemes.

4.1 Unified Russian Roulette and Splitting

In our approach to Russian roulette (RR) and splitting, we follow previous work [Szirmay-Kalos and Antal 2005; Wagner and Haghghat 1998; Booth and Hendricks 1984] and we base the respective decisions on a single real value $q > 0$. We consider a particle that has just left a collision (or emitting) event at \mathbf{x} , as shown in Fig. 3. After sampling its outgoing direction ω_o^x and determining the position \mathbf{y} of the next collision, we contribute the source radiance/importance from \mathbf{y} to the solution (Eq. (1) for path tracing). Then, we determine $q(\mathbf{y}, \omega_i)$, as described in the next section, and if $q < 1$ we play RR to randomly terminate the path with probability $1 - q$. Conversely, if $q > 1$, we split the path into q new paths. (Details on dealing with non-integer q are given in Sec. 5.1.) To compensate for termination or splitting at \mathbf{y} , the incoming particle weight ν_i is divided by q to obtain the weight $\hat{\nu}$ of each survived or split particle:

$$\hat{\nu}(\mathbf{y}, \omega_i) = \frac{\nu_i(\mathbf{y}, \omega_i)}{q(\mathbf{y}, \omega_i)}. \quad (3)$$

Each particle resulting from the collision at \mathbf{y} is then traced using the same procedure independently.

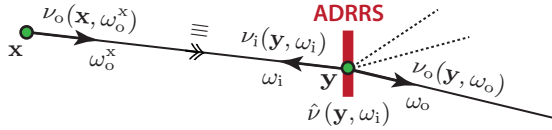


Figure 3: After we have accounted for a particle’s contribution from a collision at \mathbf{y} , we apply our ADRRS to decide about the particle’s termination/splitting. All potentially spawned particles at \mathbf{y} have weight $\hat{\nu}(\mathbf{y}, \omega_i)$ and are scattered and traced independently.

4.2 Determining the RR/Splitting Factor q

In our ADRRS, the RR/splitting factor q at \mathbf{y} is directly proportional to the *total expected contribution* $E[c(\mathbf{y}, \omega_i)]$ of the particle that collided at \mathbf{y} to the computed measurement I (e.g. a pixel value) [Wagner and Haghghat 1998; Booth and Hendricks 1984]:

$$q(\mathbf{y}, \omega_i) = \frac{E[c(\mathbf{y}, \omega_i)]}{I} = \frac{\nu_i(\mathbf{y}, \omega_i) \Psi_o^r(\mathbf{y}, \omega_i)}{I}. \quad (4)$$

For a path traced from a light source, the adjoint Ψ stands for the visual importance W , while for a path traced from the camera, it stands for radiance L . Here, we use only its reflected part $\Psi^r = \Psi - \Psi^e$ without the source term Ψ^e , because we are interested in the expected contribution of the particle that is scattered at \mathbf{y} .

The idea behind Eq. (4) is to compare $E[c]$ to the true value of the measurement I . We are likely to terminate particles that are not expected to make a contribution larger than I . This, in turn, saves resources for sampling particles with a more significant expected contribution. In contrast, when $E[c]$ exceeds the true measurement value I , we split the particle path. This results in a better exploration of the relevant regions of the path space, albeit at the expense of some additional computational resources.

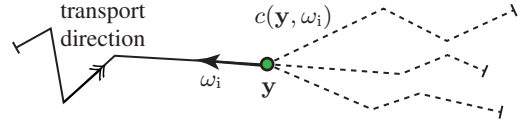


Figure 4: Realizations of the path contribution variable $c(\mathbf{y}, \omega_i)$ correspond to the different possible particle paths beyond \mathbf{y} .

Particle contribution and its expected value. The particle contribution $c(\mathbf{y}, \omega_i)$ is a random variable associated with a particle that has reached the point \mathbf{y} from the direction ω_i and has the weight $\nu_i(\mathbf{y}, \omega_i)$. The variable is distributed over all possible realizations of the particle path beyond \mathbf{y} , as shown in Fig. 4. For example in path tracing, the outcome of c for one such specific realization is given by the particle’s contribution to the sum in the measurement estimator in Eq. (1). Note that each particle sampled beyond \mathbf{y} is an unbiased estimate of $\Psi_o^r(\mathbf{y}, \omega_i)$. Thus the expected contribution $E[c(\mathbf{y}, \omega_i)]$ is given by the product of the path weight ν_i and the outgoing reflected adjoint Ψ_o^r .

4.3 Weight Invariant in ADRRS

We design our ADRRS so that it maintains the following invariant:

$$\hat{\nu}(\mathbf{y}, \omega_i) = \frac{I}{\Psi_o^r(\mathbf{y}, \omega_i)}, \quad (5)$$

which holds for particle weight in a zero-variance (ZV) scheme [Wagner and Haghghat 1998]. While this invariant arises naturally under the ZV scheme (see the supplemental), our ADRRS keeps it through termination and splitting. Our termination/splitting rate q (Eq. (4)) follows directly from Eq. (5) and the weight update formula after termination/splitting (Eq. (3)). Note that ADRRS keeps this invariant with arbitrary emission and scattering probabilities. This principle allows us to justify the importance sampling properties of ADRRS in Sec. 7 through inspection of q .

5 Algorithm

In this section we develop a practical solution for incremental path sampling (either from the camera or from the light sources) based on our *adjoint-driven Russian roulette and splitting* (ADRRS), described earlier. Evaluating the RR/splitting factor q according to Eq. (4) requires knowing the final measurement I as well as the value of the adjoint transport quantity Ψ everywhere in the scene, none of which are readily available up front. Our solution builds on an approximate estimate of those quantities obtained in a preprocessing step, as described in Sec. 5.2 and 5.3. To make the resulting algorithm robust to the inaccuracies of these estimates, we apply the *weight window* facility as described next.

We conduct the following exposition in a general tenor that applies to tracing paths in either direction. We recall that the adjoint Ψ stands for the radiance L for a camera path (as in path tracing), and for the visual importance W in the case of a path traced from the light sources (as in light or photon tracing). Differences between the two cases are pointed out when necessary.

5.1 Weight Window

The weight window is a classic technique from neutron transport used to control particle weights through RR and splitting [Hoogenboom and Légrády 2005; Booth and Hendricks 1984]. In our algorithm, it amends the ADRRS step shown in Fig. 3.

The weight window defines an interval of acceptable particle weights (δ^-, δ^+) (see Fig. 5). A particle with a weight ν_i that enters

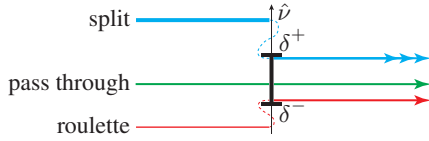


Figure 5: The weight window ensures – through selective RR or splitting – that the weight of all particles that pass the window is within the interval of acceptable weights $\langle \delta^-, \delta^+ \rangle$.

a weight window may be terminated, pass unchanged, or be split. In any case, the weight window ensures that the weight $\hat{\nu}$ of each leaving particle stays within the window bounds, i.e. $\hat{\nu} \in \langle \delta^-, \delta^+ \rangle$. If the weight of a particle entering the window is below the lower bound, $\nu_i < \delta^-$, we play Russian roulette with the survival probability $q = \nu_i / \delta^-$. If the particle survives, its new weight $\hat{\nu}$ is set to $\nu_i / q = \delta^-$. Particles entering the window with a weight inside the bounds, $\nu_i \in \langle \delta^-, \delta^+ \rangle$, pass the window intact (i.e. $\hat{\nu} = \nu_i$). Finally, if $\nu_i > \delta^+$, the particle is split into $q = \nu_i / \delta^+$ new particles, each with a weight of $\hat{\nu} = \nu_i / q = \delta^+$.

If the particle is to be split into a non-integer number of new particles, we use an *expected-value split* approach [Booth 1985]. We split the particle into $n = \lfloor q \rfloor$ new particles with the probability $n + 1 - q$, and into $n + 1$ particles otherwise. Irrespective of that decision, each new particle is assigned a weight of $\hat{\nu} = \nu_i / q$. Although this splitting strategy does not preserve the original weight exactly, the total weight is still preserved in an expected-value sense and thus the estimator stays unbiased.

Weight window bounds. To calculate the weight window bounds for a particle incident from the direction ω_i at \mathbf{y} , we start by setting the window center $C_{\text{ww}} = (\delta^- + \delta^+) / 2$ to the desired particle weight given by Eq. (5),

$$C_{\text{ww}} = \frac{\tilde{I}}{\tilde{\Psi}_o^r(\mathbf{y}, \omega_i)}. \quad (6)$$

The measurement estimate \tilde{I} is invariant along the entire particle path while the adjoint quantity estimate $\tilde{\Psi}_o^r(\mathbf{y}, \omega_i)$ depends on the scattering location \mathbf{y} and the incoming direction ω_i . Computation of these two estimates is explained in Sec. 5.2 and 5.3.

Booth and Hendricks [1984] as well as Wagner and Haghghat [1997] all suggest to set the window width as $\delta^+ = s\delta^-$, with the ratio parameter $s = 5$. The formula for computing the weight window lower bound then reads

$$\delta^- = \frac{2C_{\text{ww}}}{1 + s}. \quad (7)$$

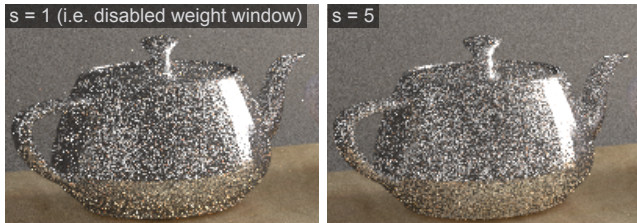
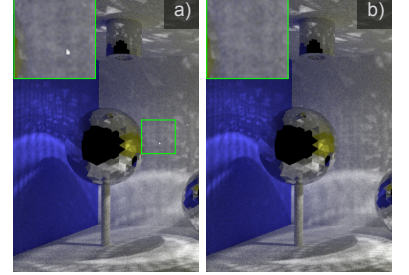


Figure 6: Weight window increases robustness of our ADRRS. It relaxes termination and splitting when the path weight is close to the weight window center not to introduce additional noise. It also compensates for imprecisions in the adjoint estimate (glossy teapot). Left: Without weight window. Right: Our weight window size. Images are rendered by light tracing in 30 minutes.

We have experimented with the parameter s and verified that performance is not particularly sensitive to its value [Booth 2006].

A practical consequence of using weight window are more relaxed RR and splitting decisions than those given by Eq. (4). As a result, the algorithm is more robust to the inaccuracies of our measurement and adjoint estimates. The weight window achieves this by allowing some leeway on the particle weight before any RR/splitting action is taken (Fig. 6).

Let us emphasize that weight window is different from clamping of the RR/splitting factor q to a finite interval, which we do apply on top of the weight window output. When RR is to be played, we additionally force the survival probability q to be above 0.1 (Fig. (b) on the right). Very small survival probabilities could otherwise result in high variance in some cases due to the inaccuracies in the measurement and adjoint estimates (Fig. (a)).



5.2 Adjoint Solution Estimate $\tilde{\Psi}_o^r(\mathbf{y}, \omega_i)$

To setup the weight window bounds at \mathbf{y} , we need an estimate of the outgoing reflected adjoint quantity $\tilde{\Psi}_o^r(\mathbf{y}, \omega_i)$. One could use a photon/importance density estimate but that would be neither accurate nor fast enough. Alternatively, a solution similar to radiance caching [Křivánek et al. 2005; Gassenbauer et al. 2009] could be used that stores the spatial-directional distribution of the adjoint. However, we have found that a simpler approach outlined below and illustrated in Fig. 7 (a) provides fairly robust estimates without having to store any directional information.

We obtain the outgoing adjoint at \mathbf{y} from a *pre-computed cache*. Instead of the full spatial-directional distribution of $\Psi_i(\mathbf{y}, \omega)$, we pre-compute and cache an estimate $\tilde{G}(\mathbf{y})$ of the quantity

$$G(\mathbf{y}) = \int \Psi_i(\mathbf{y}, \omega) \cos \theta_{\mathbf{y}} d\omega, \quad (8)$$

that corresponds to *irradiance* or *diffuse visual importance*. Here $\theta_{\mathbf{y}}$ is the angle between the direction ω and the surface normal at \mathbf{y} . An estimate of the reflected outgoing adjoint is then calculated as

$$\tilde{\Psi}_o^r(\mathbf{y}) = \frac{\kappa(\mathbf{y})}{\pi} \tilde{G}(\mathbf{y}), \quad (9)$$

where $\kappa(\mathbf{y})$ is the total material reflectivity at \mathbf{y} . We obtain $\tilde{G}(\mathbf{y})$ by querying a spatial cache at \mathbf{y} .

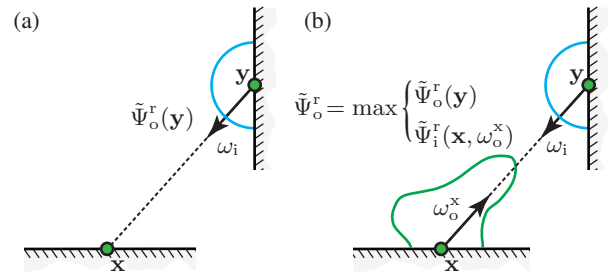


Figure 7: Adjoint solution estimate in our basic ADRRS implementation (a) and when ADRRS is combined with path guiding (b).

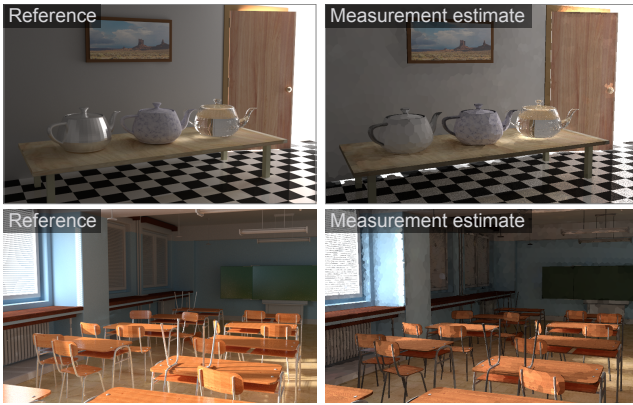


Figure 8: Measurement estimates \tilde{I} for two example scenes.

Pre-processing and caching. We base our pre-processing stage on the iterative scheme proposed by Vorba et al. [2014], where they interleave shooting particles from the camera and from light sources for faster convergence. Our basic implementation differs from Vorba et al. in that the traced particles are used to compute an approximation to *irradiance* and *diffuse visual importance*, respectively (i.e. \tilde{G}), rather than for fitting directional distributions. (A more advanced implementation that uses Vorba et al.’s directional distributions will be described in Sec. 6.)

We also adopt their spatial caching approach. When an estimate of \tilde{G} is not available we use *kernel density estimation* to compute a new estimate from nearby particles (e.g. from photons when we currently trace from the camera). Each cached estimate has a validity radius where the record can be reused. The radius is never allowed to be larger than the furthest particle used for the estimation and the caching scheme makes the cache denser in places of strong light field changes. We refer the reader to the work of Vorba et al. [2014] for a more detailed description of their caching scheme.

We refine \tilde{G} in every iteration using progressive kernel density estimation and at the same time, we estimate its relative error (i.e. $\text{stddev}(\tilde{G})/\tilde{G}$). Our ADRRS is applied at a collision only if the relative error of the associated cache record is below a threshold value of 30%. Otherwise we use a large weight window with the globally-fixed size as described by Vorba et al. [2014] and in Sec. 6. Note that unlike bidirectional path tracing [Veach 1997] or vertex connection and merging [Georgiev et al. 2012; Hachisuka et al. 2012] we do not construct a combined estimator to compute \tilde{G} and rather keep our implementation simple. To achieve smoother estimates of \tilde{G} we average it over nearby cached estimates.

5.3 Measurement Estimate \tilde{I}

When calculating the weight window center using Eq. (6), Booth [1984] recommends, instead of \tilde{I} , using a normalization constant so that the particle weight is exactly in the weight window center after the first collision. We adopt a more principled approach due to Wagner and Hachisuka [1997], where they use an approximation \tilde{I} of the measurement I that we eventually strive to calculate. In this way, the particle weights $\hat{\nu}(\mathbf{y}, \omega_i)$ oscillate around the ideal value $I/\Psi_o^r(\mathbf{y}, \omega_i)$, which is motivated by the zero-variance theory, as discussed in Sec. 7.

In our implementation, the actual meaning of \tilde{I} depends on the direction of path sampling. For paths from the camera (path tracing), we set \tilde{I} to be an estimate of the pixel value that the respective path passes through. We compute the pixel value from the pre-computed

irradiance cache (Sec. 5.2) using four jittered primary rays. We query the cache immediately on diffuse and glossy surfaces while continuing the path on purely specular surfaces. If no non-specular surface is hit within ten bounces, we treat the surface of the 10th bounce as diffuse and query the irradiance cache. Fig. 8 shows the resulting estimates for two example scenes.

For paths from light sources (light or photon tracing), we set \tilde{I} to the average pixel value estimated as above, over the entire image. We use this approach because we do not know up front which pixel the path would contribute to. While this heuristic has worked well in our tests, a thorough analysis of ADRRS when calculating several measurements (i.e. pixel values) simultaneously would be an interesting avenue for future work.

5.4 Path Sampling Algorithm

Algorithm 1 shows simplified pseudocode for processing a collision in our path sampling algorithm. The procedure receives the previous and the current collision locations \mathbf{x} and \mathbf{y} respectively, the incident direction ω_i at \mathbf{y} , and the incident particle weight ν_i . Its steps should be self-explanatory. Below we discuss some additional details of the full path sampling algorithm.

First collision. When tracing a particle from the camera, we initialize the RR/splitting factor q to 1 at the first collision so no RR or splitting is carried out there. In Fig. 3, \mathbf{x} is the camera vertex and \mathbf{y} corresponds to the first collision. Applying our ADRRS in this case would only serve to compensate the variations of the light contribution within one pixel, which are usually small. On the other hand, when tracing particles from light sources, we initialize q according to our algorithm right at the first collision, so RR/splitting can take place there. (In this case, in Fig. 3, \mathbf{x} is a light source vertex.)

Tree pruning. To avoid overly bushy ray trees due to splitting, we impose a maximum splitting factor at each collision to 100. In addition, we limit the size of the entire ray tree by the following heuristic. We associate a number s_{count} with every event along a path which conservatively estimates the total number of rays in the tree. We initialize s_{count} to 1 and we multiply it at every collision by the splitting factor determined at that event. We disable splitting once $s_{\text{count}} > 1000$.

Next-event estimation. So far, we have only discussed unidirectional path or light tracing algorithms, but our ADRRS naturally extends to using next-event estimation (i.e. explicit connections to light sources or the camera). A theoretical justification is based on the idea of replacing self-emission, L_o^e or W_o^e , by the sources of first-scattering events (e.g. direct illumination on surfaces serves as the new emission term). All the derivations can then be carried out with those re-defined source terms as before [Hoogenboom 2008]. In practice, we do the following. Suppose we have reached a scattering event at \mathbf{y} and determined the integer splitting factor of n . At this point, we in fact draw n pairs of sample rays, one ray in each pair by sampling a scattering direction ω_o and another by explicit light source or sensor sampling. The direct illumination contributions of these rays are then combined using multiple importance sampling [Veach 1997].

6 Combining ADRRS with Path Guiding

In this section we combine our ADRRS with *path guiding* methods based on adjoint-driven importance sampling of scattering directions [Vorba et al. 2014; Bashford-Rogers et al. 2012; Hey and Purgathofer 2002; Jensen 1995]. We show on the method of Vorba et al. that ADRRS works in synergy with path guiding and the combination leads to superior results than either method alone.

Algorithm 1 Pseudocode showing the ADRRS-related steps in processing a collision event along an incrementally constructed path.

```

1: procedure HANDLECOLLISION( $\mathbf{x}, \mathbf{y}, \omega_i, \nu_i$ )
2:   //  $\mathbf{x}$  ... previous collision location,  $\mathbf{y}$  ... current collision location
3:   //  $\omega_i$  ... incident direction at  $\mathbf{y}$ ,  $\nu_i$  ... particle weight
4:   CONTRIBUTE( $\mathbf{y}, \omega_i, \nu_i$ ) // Eq. (1)
5:    $\tilde{G} := \text{LOOKUPCACHE}(\mathbf{y})$  // Sec. 5.2
6:    $\tilde{\Psi}_o^r := \text{CALCADJOINT}(\tilde{G}, \mathbf{x}, \mathbf{y}, \omega_i)$  // Eq. (9) or (11)
7:    $\langle \delta^-, \delta^+ \rangle := \text{CALCWBOUNDS}(\tilde{\Psi}_o^r, \tilde{I})$  // Eqns. (6) and (7)
8:    $[n, \hat{\nu}] := \text{APPLYWW}(\nu_i, \langle \delta^-, \delta^+ \rangle)$  // Sec. 5.1
9:   for  $j = 1 \dots n$  do
10:     $\omega_o := \text{SAMPLEDIR}(\mathbf{y})$ 
11:     $\nu_o := \text{UPDATEWEIGHT}(\hat{\nu}, \omega_i, \omega_o)$  // Eq. (2)
12:     $\mathbf{z} := \text{INTERSECT}(\mathbf{y}, \omega_o)$ 
13:    HANDLECOLLISION( $\mathbf{y}, \mathbf{z}, -\omega_o, \nu_o$ ) // Recurse to next event
14:   end for
15: end procedure

```

Motivation. Vorba et al. [2014] observed that the classic RR, based only on particle weights, is adverse to path guiding in scenes with difficult visibility. The reason is that a particle guided towards a high-contribution region may be terminated before being able to reach it. In fact, aggressive RR may offset the advantages of path guiding entirely, and just add overhead.

To address this problem, Vorba et al. employ RR with very low threshold $\delta^- = 10^{-6}$ on relative change of weight, leading to vastly increased average path length. Such an approach is not satisfactory because effort is wasted on sampling long paths in unimportant regions (see Fig. 10), while other paths may still be terminated prematurely. In addition to RR, they apply splitting if the particle weight exceeds two times its original value. Effectively, they use a globally fixed weight window with an extensive size of $s = 2 \times 10^6$.

Our approach. We address the above problem by using our ADRRS, which allows us to sample close-to-optimal path lengths without any adverse effects on path guiding. Moreover, ADRRS effectively improves the quality of path space importance sampling over baseline path guiding, as discussed in Sec. 7.1 and shown in Figs. 10 and 13. We base our implementation on the on-line learning algorithm of Vorba et al. [2014]. We pre-compute and cache the diffuse quantity \tilde{G} (Sec. 5.2) together with Vorba et al.’s directional sampling distributions. ADRRS is applied both in the training and the rendering stages. As opposed to the basic approach from Sec. 5, the scattering directions ω_o are sampled from the pre-computed directional distributions as in Vorba et al.’s work. Furthermore, we exploit the cached directional distributions to obtain more accurate estimates of the adjoint, which substantially improves the robustness of our method, especially in scenes with *glossy materials* (Fig. 9).

Adjoint from directional distribution. We use the cached directional distribution $\tilde{p}(\omega | \mathbf{x})$ at \mathbf{x} to estimate the incoming adjoint at \mathbf{x} reflected from \mathbf{y} , that does not involve the source illumination from \mathbf{y} , as

$$\tilde{\Psi}_i^r(\mathbf{x}, \omega_o^x) = \frac{\tilde{p}(\omega_o^x | \mathbf{x}) \tilde{G}(\mathbf{x})}{\cos \theta_x} - \Psi_o^e(\mathbf{y}, \omega_i). \quad (10)$$

This equation follows from the fact that the directional distribution is designed such that $\tilde{p}(\omega_o^x | \mathbf{x}) \propto \Psi_i(\mathbf{x}, \omega_o^x) \cos \theta_x$ with $\tilde{G}(\mathbf{x})$ (Eq. (8)) being the normalization factor [Vorba et al. 2014].

In practice, we obtain the final adjoint estimate as shown in Fig. 7 (b):

$$\tilde{\Psi}_o^r(\mathbf{y}, \omega_i) = \max \left\{ \underbrace{\tilde{\Psi}_o^r(\mathbf{y})}_{\text{Eq. (9)}}, \underbrace{\tilde{\Psi}_i^r(\mathbf{x}, \omega_o^x)}_{\text{Eq. (10)}} \right\}. \quad (11)$$

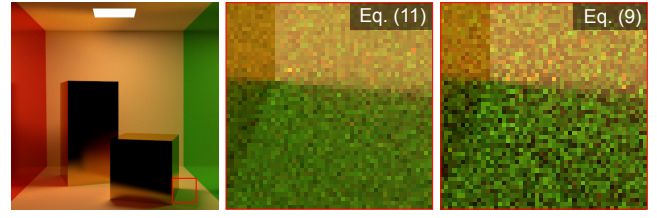


Figure 9: A glossy surface with (middle) and without (right) the conservative adjoint estimate given by Eq. (11). Both insets were rendered at 64 samples per pixel.

This conservative estimate produces a lower weight window center and thus lower termination rates, which eliminates some high-frequency noise otherwise appearing in certain scenarios especially in the presence of glossy materials (Fig. 9).

7 ADRRS and Zero-Variance Schemes

It has been known for a long time in neutron transport [Kalos 1963], and recently pointed out in computer graphics [Křivánek and d’Eon 2014; Xu et al. 2001], that particle paths can be constructed such that the estimator in Eq. (1) has zero variance (ZV). In other words, the solution can be found with only one particle path. While this cannot be achieved without knowing the computed solution in advance, zero-variance schemes are an invaluable tool for studying and designing variance reduction techniques.

Recall from Sec. 4.3 that the termination/splitting rate q in ADRRS is designed to keep the same principle (Eq. (5)) that governs particle weight in the ZV scheme. In this section, we study the importance sampling properties of ADRRS through the inspection of q .

7.1 Zero Variance, Importance Sampling, and ADRRS

To study importance sampling properties of ADRRS and its relation to ZV schemes, we derive, in the supplemental document, the following equation for the RR/splitting rate q at a collision \mathbf{y} :

$$q(\mathbf{y}, \omega_i) = \frac{p_{zv}(\omega_o^x | \mathbf{x})}{p(\omega_o^x | \mathbf{x})}. \quad (12)$$

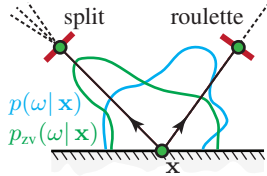
That is to say, the RR/splitting rate

at \mathbf{y} is given by the ratio of the zero-variance pdf to the actually used pdf for sampling the scattering direction ω_o^x at the *preceding* vertex \mathbf{x} . The pdf $p_{zv}(\omega | \mathbf{x})$ that ensures ZV estimation (see the supplemental document) is proportional to the product of the cosine-weighted BSDF lobe and the directional distribution of the adjoint quantity incident at \mathbf{x} : $p_{zv}(\omega | \mathbf{x}) \propto \Psi_i(\mathbf{x}, \omega) f_s^+(\mathbf{x}, \omega_i^x \rightarrow \omega) |\cos \theta_x|$. Here, we use ω_i^x to denote the incident direction at \mathbf{x} . Note that $f_s^+(\cdot, \omega_i^x \rightarrow \omega_o^x) = f_s(\cdot, \omega_i^x \rightarrow \omega_o^x)$ for light tracing while $f_s^+(\cdot, \omega_i^x \rightarrow \omega_o^x) = f_s(\cdot, \omega_o^x \rightarrow \omega_i^x)$ for path tracing. To simplify the following discussion, we assume, with no bearing on our results, that \mathbf{y} is not on a source or sensor; general form of Eq. (12) is presented in the supplemental document.

Interestingly, Eq. (12), which is derived for infinitesimal weight window (i.e. $s = 1$ in Eq. (6)), shows that q takes the same form at any collision irrespective of any previous collisions before \mathbf{x} (i.e. there is no weight term ν_i of a particle incident at \mathbf{x}). We can thus limit ourselves to discussing the effect of ADRRS on the variance of a local estimator of the (hemi)spherical integral at \mathbf{x} .

RR and splitting as importance sampling.

Suppose we have drawn the scattering direction ω_o^x from a general pdf $p(\omega|\mathbf{x})$. Eq. (12) states that the factor q is determined by comparing the pdf value for the sampled direction to the pdf value $p_{zv}(\omega_o^x|\mathbf{x})$ dictated by the ZV scheme. We keep samples untouched in those parts of the (hemi)sphere where $p(\omega_o^x|\mathbf{x}) = p_{zv}(\omega_o^x|\mathbf{x})$, we split where our sampling rate is too low, i.e. $p(\omega_o^x|\mathbf{x}) < p_{zv}(\omega_o^x|\mathbf{x})$, while we randomly terminate where we place too many samples, i.e. $p(\omega_o^x|\mathbf{x}) > p_{zv}(\omega_o^x|\mathbf{x})$.



Residual variance. In the case of RR, the above procedure is equivalent to *rejection sampling*, where $p(\omega|\mathbf{x})$ serves as the proposal density and $p_{zv}(\omega|\mathbf{x})$ is the target density. However, splitting cannot reduce any variance introduced due to sampling from $p(\omega|\mathbf{x})$ (as opposed to the ideal pdf $p_{zv}(\omega|\mathbf{x})$) because splitting only occurs at the next collision event \mathbf{y} . Only variance from subsequent bounces can be reduced. It is therefore not effective when most of the variance is gained through the use of an inappropriate scattering pdf at \mathbf{x} . An example is shown and discussed in Sec. 8. Booth [2012] provides further discussion of this limitation.

Zero-variance sampling. By using ADRRS on top of a ZV scheme (i.e. when $p(\omega|\mathbf{x}) = p_{zv}(\omega|\mathbf{x})$ for all ω), we still obtain a ZV estimator because q becomes q_{zv} , the ZV termination rate (see the supplemental document). In other words, it follows from Eq. (12) that $q = 1$ when \mathbf{y} is not on a light source or sensor and thus no termination or splitting takes place which is inline with the ZV scheme. However, there is a clear difference between direct sampling from $p_{zv}(\omega|\mathbf{x})$ in the ZV scheme and ADRRS, where we can sample from an arbitrary $p(\omega|\mathbf{x})$. While the former has zero variance, i.e. it solves the integral with a single sample, ADRRS only strives for variance reduction through RR-implied rejection and splitting at \mathbf{y} .

Path guiding methods. Path guiding methods strive to sample scattering directions from a pdf $p(\omega)$ that closely approximates the ZV pdf $p_{zv}(\omega)$ [Vorba et al. 2014; Hey and Purgathofer 2002; Jensen 1995]. Path guiding and ADRRS work in synergy to approximate the optimal ZV scheme even more closely. To see why, recall the rejection sampling interpretation of ADRRS. On one hand, the better the path guiding distribution (i.e. the closer the proposal pdf $p(\omega)$ to the target ZV pdf $p_{zv}(\omega)$), the less work for RR and splitting. On the other hand, should the path guiding distribution $p(\omega)$ fail to approximate the ZV pdf $p_{zv}(\omega)$ closely enough, our ADRRS still steps in to improve the effective particle distribution.

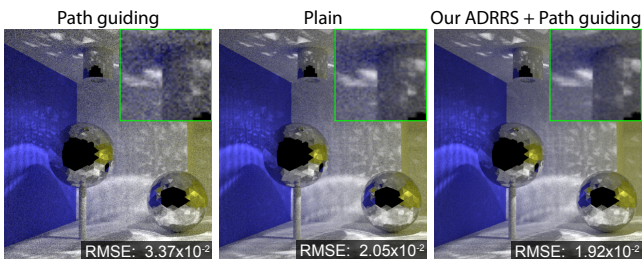


Figure 10: Path guiding (PG) of Vorba et al. [2014] (left) applied in Progressive Photon Mapping (PPM) does not match the efficiency of plain PPM (middle) due to its overhead coming mainly from sampling long paths. PPM with our ADRRS and PG (right) achieves superior results than plain PPM even in this simple scene. All images took 5 minutes to render (including the training time).

8 Results

Here, we experimentally validate the theoretical outcomes of Sec. 7. Results in Fig. 13 confirm the strong variance reduction capabilities of our adjoint-driven Russian roulette and splitting (ADRRS) in path tracing (PT) and light tracing (LT). Additionally, we show the results of the combination of ADRRS with path guiding (PG) [Vorba et al. 2014], which yields a practical and efficient algorithm capable of rendering complex scenes with difficult visibility. Interestingly, this combination is beneficial even in simple scenes where the overhead of path guiding, coming from excessive path lengths, would offset its advantages if used without ADRRS (Fig. 10). We encourage the reader to view the supplemental material for all our results rendered with PT, LT and also progressive photon mapping (PPM) [Hachisuka et al. 2008].

Setup. All images in Fig. 13 were rendered for 1 hour on an Intel Core i7-2600K CPU using 8 logical cores. Our implementation is based on the Mitsuba renderer [Jakob 2010] and the path guiding code of Vorba et al. [2014]. We set the maximum path length to 40 bounces. The ‘plain’ light and path tracing algorithms use classic albedo-based RR from the fifth bounce on, and no splitting.

We include the pre-computation time in the reported total time of all our results. The pre-computation times of PG and our ADRRS with path guiding (ADRRS + PG) are listed in Table 1. Results rendered with our ADRRS-only use the same pre-computation as ADRRS + PG to achieve the same quality of cached irradiance/visual importance. This allows us to compare the effect of adding PG on top of ADRRS. Unlike ADRRS + PG, the PG-only results do not use our ADRRS in the training stage.

	ADRRS + PG	PG
Crytek sponza	150s	114s
Veach door	222s	192s
Living room	462s	426s
Classroom	588s	462s

Table 1: Training times of ADRRS + PG and PG-only.

Scenes. We adopt three scenes from the work of Vorba et al. [2014], namely *Living room*, *Veach door* and *Classroom*, without any change. Our fourth scene is the *Crytek Sponza* [2010]. Common to all the scenes is complex visibility with many regions lit only by high-order indirect illumination.

In the *Crytek Sponza* scene with mainly diffuse materials, the sunlight enters the atrium and indirectly illuminates most of the shot. Our ADRRS alone achieves substantial variance reduction in comparison to standard path and light tracing. Using path guiding on top of our ADRRS (ADRRS+PG) yields superior result without any of the spike noise present in path guiding alone.

All the illumination in the *Veach door* scene enters through the door ajar from the back room. We use Vorba et al.’s version of the scene, which differs from that of Lehtinen et al. [2013] in that the light source size is roughly 250× smaller. This makes the Vorba et al.’s scene more realistic – and substantially more challenging (see Fig. 11). In PT, our ADRRS + PG combination significantly reduces the spike noise produced by path guiding alone. However, unlike in the *Crytek Sponza* scene, some of this noise still remains. This is due to the combination of the small light source and specular reflections on the floor, which effectively disqualifies any next event estimation. When a path guiding distribution on a wall fails to accurately target the caustic-like illumination due to specular light source reflection on the floor, ADRRS cannot remedy the situation by splitting that is decided on the wall (vertex \mathbf{x} in Fig. 3). This

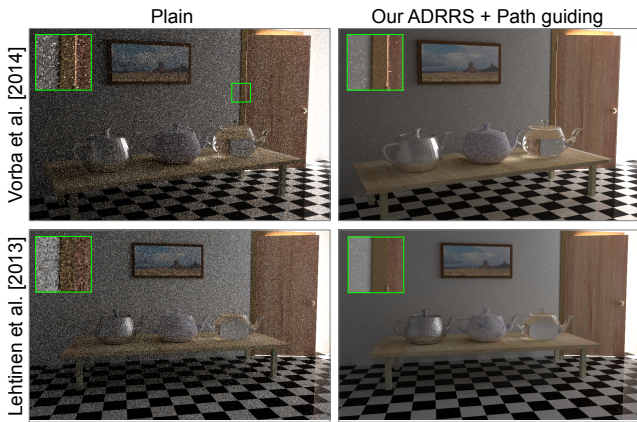


Figure 11: Two versions of the Veach Door scene. We rendered 40 light bounces by plain path tracing (left) and path tracing with our ADRRS and path guiding (right). The Vorba et al. version (top) is more challenging than the version of Lehtinen et al. (bottom): its $250\times$ smaller light source is responsible for the spiky noise.

is because physical splitting on the specular reflection on the floor (vertex y) is ineffective (see “Residual Variance” in Sec. 7.1)

Illumination in both the **Living room** and **Classroom** scenes is similar. The Sun is shining through window glass covered by curtains and jalousies respectively, and large parts of the scenes are lit by light after many bounces. While in LT, our ADRRS alone provides excellent results, the results in PT expose the ADRRS limitations discussed in Sec. 7.1. Referring back to the rejection sampling interpretation of ADRRS, the proposal distribution could be the floor BRDF, while the target zero-variance distribution is close to a delta-distribution as it encompasses the sunlight passing through the window-panes. ADRRS cannot reduce variance in this case because the physical split can only occur on the glass (vertex y in Fig. 3), where it would be ineffective. Nevertheless, the combination of our ADRRS with path guiding addresses the problem and yields superior results.

Effect of RR and splitting. We show separately the effect of our adjoint-driven RR (ADRR) and adjoint-driven splitting in Fig. 12. We rendered the Crytek Sponza scene for 20 minutes with a guided path tracer (a), guided path tracer with our ADRR (no splitting) (b), and guided path tracer with full ADRRS (c). The variance reduction in (b) stems from sampling nearly optimal path lengths by our ADRR. The splitting in (c) significantly improves sampling of regions where light is transported through several bounces.

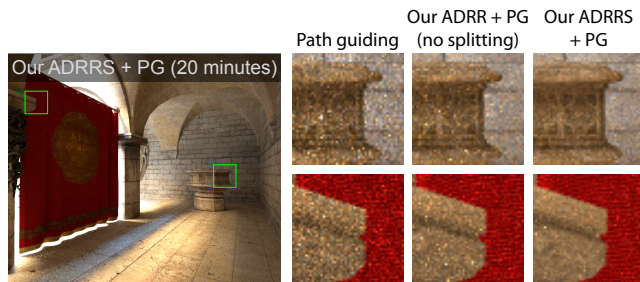


Figure 12: Our adjoint-driven RR without splitting (middle) in guided path tracing reduces variance of guided path tracing (left). Application of splitting (right) provides a substantial additional variance reduction in regions reached by light after several events.

9 Limitation, Discussion, and Future Work

Inaccurate adjoint and measurement estimates. Grossly inaccurate adjoint or measurement estimates can produce increased variance, as in the top right image in Fig. 11 under the table. This can happen for example due to light leaks well-known from photon mapping, such as those visible in Fig. 8. This limitation is common to all variance reduction techniques based on estimated quantities, and in practice it can be alleviated by adaptive image sampling.

Efficiency-driven RR and splitting. While the close relation of our ADRRS to zero-variance path sampling schemes provides a solid justification of its variance reduction properties, nothing in the theory suggests that ADRRS would be optimal with respect to efficiency. As such, an efficiency-driven RR and splitting is an important avenue for future research.

Splitting and combined estimators. We have shown that splitting is an effective variance reduction tool in unidirectional path sampling algorithms. It would be interesting to extend its use to bidirectional algorithms based on combining various estimators [Křivánek et al. 2014a; Georgiev et al. 2012; Hachisuka et al. 2012]. A challenge associated with this idea would be the development of proper combination weights that would respect the correlation of split paths due to their common shared prefix [Popov et al. 2015].

Participating media. When simulating transport in media, particles typically undergo many more events than on surfaces before making an image contribution. At the same time, whenever a light source or the camera is inside a medium, there can be tremendous variation of the respective adjoint quantity. For those reasons, it is likely that extending our method to participating media would result in a substantially greater increase in efficiency than on surfaces.

10 Conclusion

We have introduced an approach for selective path termination and splitting that we call adjoint-driven Russian roulette and splitting (ADRRS). The termination and splitting decisions are driven by a pre-computed estimate of the adjoint transport quantity so that a significant variance reduction is achieved. We have provided a theoretical justification of the variance reduction properties – and its limits – by juxtaposing ADRRS to a zero-variance path sampling scheme. To make the method practical and robust, we have introduced the idea of adaptive weight window from the neutron transport field. We have shown that our ADRRS complements the directional importance sampling techniques (path guiding) and together they result in robust and efficient simulations even in fairly simple *unidirectional* methods such as path and light tracing. These are easier to implement than combined path integral estimators such as bidirectional path tracing [Veach 1997] or vertex connection and merging [Georgiev et al. 2012; Hachisuka et al. 2012] and are favored in practice for their easy combination with a broad scale of production features.

Acknowledgements

The scenes in Fig. 10 and Fig. 11 are courtesy of Toshiya Hachisuka and Jaakko Lehtinen, respectively. The work was supported by the Charles University Grant Agency, project GA UK 340915, by the grant SVV–2016–260332, and by the Czech Science Foundation grants 13–26189S and 16–18964S. Many thanks to Iliyan Georgiev and Oskar Elek for their detailed feedback on the manuscript and to Peter Pearson for language corrections.

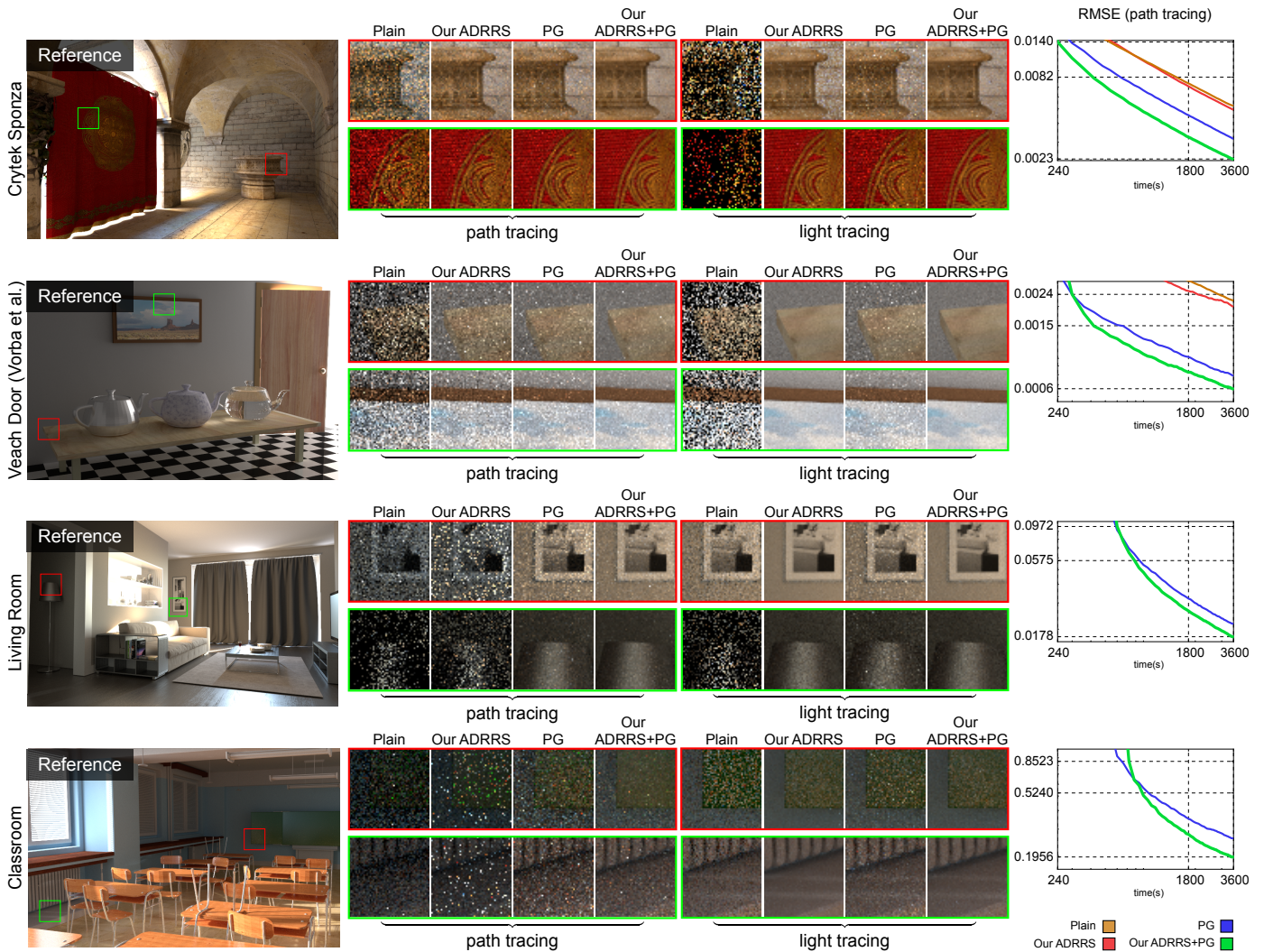


Figure 13: We render four scenes dominated by indirect lighting for 1 hour. Results in the figure come from path and light tracing, respectively, with albedo-based RR applied from the fifth bounce on (Plain), our ADRRS, path guiding alone (PG), and our ADRRS with PG. Our ADRRS achieves substantial variance reduction over the albedo-based RR (Plain). Superior results are achieved by complementing our ADRRS with path guiding. The training time is included in the reported 1 hour. Complete images are shown in the supplemental material.

References

- ALDOUS, D., AND VAZIRANI, U. 1994. “Go with the winners” algorithms. In *Proc. 35th IEEE Symp. on Found. of Comp. Sci.*
- ARVO, J., AND KIRK, D. 1990. Particle transport and image synthesis. In *Proc. SIGGRAPH '90*, ACM, 63–66.
- BASHFORD-ROGERS, T., DEBATTISTA, K., AND CHALMERS, A. 2012. A significance cache for accelerating global illumination. *Computer Graphics Forum* 31, 6, 1837–51.
- BASHFORD-ROGERS, T., DEBATTISTA, K., AND CHALMERS, A. 2013. Importance driven environment map sampling. *IEEE Trans. Vis. Comput. Graphics* 19.
- BOLIN, M. R., AND MEYER, G. W. 1997. An error metric for Monte Carlo ray tracing. In *In Rendering Techniques '97*.
- BOOTH, T. E., AND HENDRICKS, J. S. 1984. Importance estimation in forward MC calculations. *Nuc. Tech./Fusion* 5, 1.
- BOOTH, T. E. 1985. Monte Carlo variance comparison for expected-value versus sampled splitting. *Nucl. Sci. Eng.* 89, 4.
- BOOTH, T., E. 2006. Genesis of the weight window and the weight window generator in MCNP - a personal history. Tech. Rep. LA-UR-06-5807, July.
- BOOTH, T., E. 2012. Common misconceptions in Monte Carlo particle transport. *Applied Radiation and Isotopes* 70.
- CHRISTENSEN, P. H. 2003. Adjoints and importance in rendering: An overview. *IEEE Trans. Vis. Comput. Graphics* 9, 3, 329–340.
- COOK, R. L., PORTER, T., AND CARPENTER, L. 1984. Distributed ray tracing. *SIGGRAPH Comput. Graph.* 18, 3 (Jan.).
- DUTRÉ, P., AND WILLEMS, Y. 1994. Importance-driven Monte Carlo light tracing. In *Eurographics Workshop on Rendering*.
- DUTRÉ, P., BALA, K., AND BEKAERT, P. 2006. *Advanced Global Illumination*, 2nd ed. A. K. Peters.

- GASSENBAUER, V., KŘIVÁNEK, J., AND BOUATOUCH, K. 2009. Spatial directional radiance caching. *Computer Graphics Forum (EGSR 2009)* 28, 4, 1189–1198.
- GEORGIEV, I., AND SLUSALLEK, P. 2010. Simple and Robust Iterative Importance Sampling of Virtual Point Lights. *Proceedings of Eurographics (short papers)*.
- GEORGIEV, I., KŘIVÁNEK, J., DAVIDOVIČ, T., AND SLUSALLEK, P. 2012. Light transport simulation with vertex connection and merging. *ACM Trans. Graph. (SIGGRAPH Asia '12)* 31, 6.
- GRASSBERGER, P. 2002. Go with the winners: a general Monte Carlo strategy. In *Comp. Phys. Commun.*, vol. 147, 64–70.
- HACHISUKA, T., OGAKI, S., AND JENSEN, H. W. 2008. Progressive photon mapping. *ACM Trans. Graph. (SIGGRAPH Asia '08)* 27, 5 (Dec.).
- HACHISUKA, T., PANTALEONI, J., AND JENSEN, H. W. 2012. A path space extension for robust light transport simulation. *ACM Trans. Graph. (SIGGRAPH Asia '12)* 31, 6.
- HAMMERSLEY, J., AND HANDSCOMB, D. 1964. *Monte Carlo Methods*. Chapman and Hall, New York.
- HEY, H., AND PURGATHOFER, W. 2002. Importance sampling with hemispherical particle footprints. In *SCCG*.
- HOOGENBOOM, EDUARD, J., AND LÉGRÁDY, D. 2005. A critical review of the weight window generator in MCNP. *Monte Carlo 2005 Topical Meeting* (Apr.).
- HOOGENBOOM, EDUARD, J. 2008. Zero-variance Monte Carlo schemes revisited. *Nucl. Sci. Eng.* 160, 1–22.
- JAKOB, W., 2010. Mitsuba renderer. <http://mitsuba-renderer.org>.
- JENSEN, H. W. 1995. Importance driven path tracing using the photon map. In *Eurographics Workshop Rendering*, 326–335.
- JENSEN, H. W. 1996. Global illumination using photon maps. In *Proceedings of the Eurographics Workshop on Rendering Techniques '96*, Springer-Verlag, London, UK, UK, 21–30.
- JENSEN, H. W. 2001. *Realistic Image Synthesis Using Photon Mapping*. A. K. Peters, Ltd., Natick, MA, USA.
- KAHN, H., AND HARRIS, T., E. 1951. Estimation of particle transmission by random sampling. In *Nat. Bur. of Stand. Appl. Math. Ser.*, vol. 12, 27–30.
- KAHN, H. 1956. Use of different Monte Carlo sampling techniques. In *Symp. on Monte Carlo Methods*, New York: Wiley.
- KAJIYA, J. T. 1986. The rendering equation. *SIGGRAPH Comput. Graph.* 20, 4 (Aug.), 143–150.
- KALOS, M. H. 1963. Importance sampling in Monte Carlo shielding calculations: I. neutron penetration through thick hydrogen slabs. In *Nuclear Science and Engineering*, vol. 16, 227–234.
- KELLER, A., AND WALD, I. 2000. Efficient importance sampling techniques for the photon map. In *Proc. Fifth Fall Workshop Vision, Modeling, and Visualisation*, 271–279.
- KŘIVÁNEK, J., GAUTRON, P., PATTANAIK, S., AND BOUATOUCH, K. 2005. Radiance caching for efficient global illumination computation. *IEEE Trans. Vis. Comp. Graph.* 11, 5.
- KŘIVÁNEK, J., GEORGIEV, I., HACHISUKA, T., VÉVODA, P., ŠIK, M., NOWROUZEZAHRAI, D., AND JAROSZ, W. 2014. Unifying points, beams, and paths in volumetric light transport simulation. *ACM Trans. Graph.* 33, 4 (Aug.), 1–13.
- KŘIVÁNEK, J., KELLER, A., GEORGIEV, I., KAPLANYAN, A., FAJARDO, M., MEYER, M., NAHMIA, J.-D., KARLÍK, O., AND CANADA, J. 2014. Recent advances in light transport simulation: Some theory and a lot of practice. In *ACM SIGGRAPH 2014 Courses*, ACM, New York, NY, USA, SIGGRAPH '14.
- KŘIVÁNEK, J., AND D'EON, E. 2014. A zero-variance-based sampling scheme for Monte Carlo subsurface scattering. In *ACM SIGGRAPH 2014 Talks*, ACM, New York, NY, USA.
- LEHTINEN, J., KARRAS, T., LAINE, S., AITTALA, M., DURAND, F., AND AILA, T. 2013. Gradient-domain metropolis light transport. *ACM Trans. Graph.* 32, 4.
- LUX, I., AND KOBLINGER, L. 1991. *Monte Carlo particle transport methods: neutron and photon calculations*. CRC Press.
- MEINL, F., 2010. Crytek sponza. <http://www.crytek.com/cryengine/cryengine3/downloads>.
- POPOV, S., RAMAMOORTHY, R., DURAND, F., AND DRETTAKIS, G. 2015. Probabilistic connections for bidirectional path tracing. *Computer Graphics Forum (Proc. of EGSR)* 34, 4.
- SEYMOUR, M., 2014. Manuka: Weta digital's new renderer. <http://www.fxguide.com/featured/manuka-weta-digitals-new-renderer/>.
- SPANIER, J., AND GELBARD, ELY, M. 1969. *Monte Carlo principles and neutron transport problems*. Addison-Wesley.
- SUYKENS, F., AND WILLEMS, Y. D. 2000. Density control for photon maps. In *Proceedings of the Eurographics Workshop on Rendering Techniques 2000*, Springer-Verlag, London, UK.
- SZÉCSI, L., SZIRMAY-KALOS, L., AND KELEMEN, C. 2003. Variance reduction for russian roulette. *Journal of WSCG*.
- SZIRMAY-KALOS, L., AND ANTAL, G. 2005. Go with the winners strategy in path tracing. In *Journal of WSCG*, vol. 13.
- VEACH, E. 1997. *Robust Monte Carlo methods for light transport simulation*. PhD thesis, Stanford University.
- VORBA, J., KARLÍK, O., ŠIK, M., RITSCHER, T., AND KŘIVÁNEK, J. 2014. On-line learning of parametric mixture models for light transport simulation. *ACM Trans. Graph. (SIGGRAPH '14)* 33, 4 (July).
- WAGNER, J., C., AND HAGHIGHAT, A. 1998. Automated variance reduction of Monte Carlo shielding calculations using the discrete ordinates adjoint function. In *Nucl. Sci. Eng.*, vol. 128.
- WAGNER, J., C. 1997. *Acceleration of Monte Carlo shielding calculations with an automated variance reduction technique and parallel processing*. PhD thesis, The Pennsylvania State Univ.
- X-5 MONTE CARLO TEAM. 2003. MCNP – A general Monte Carlo N-particle transport code, version 5. Tech. Rep. LA-UR-03-1987, Los Alamos National Laboratory, Apr.
- XU, Q., SUN, J., WEI, Z., SHU, Y., MESSELODI, S., AND CAI, J. 2001. Zero variance importance sampling driven potential tracing algorithm for global illumination. In *Journal of WSCG*. 2001, vol. 9.