A Spatial Target Function for Metropolis Photon Tracing

ADRIEN GRUSON
Université de Rennes 1, IRISA
MICKAËL RIBARDIÈRE
Université de Poitiers, XLIM-SIC
MARTIN ŠIK and JIŘÍ VORBA
Charles University in Prague, Faculty of Mathematics and Physics
RÉMI COZOT and KADI BOUATOUCH
Université de Rennes 1, IRISA
and
JAROSLAV KŘIVÁNEK
Charles University in Prague, Faculty of Mathematics and Physics

The human visual system is sensitive to relative differences in luminance, but light transport simulation algorithms based on Metropolis sampling often result in a highly nonuniform relative error distribution over the rendered image. Although this issue has previously been addressed in the context of the Metropolis light transport algorithm, our work focuses on Metropolis photon tracing. We present a new target function (TF) for Metropolis photon tracing that ensures good stratification of photons leading to pixel estimates with equalized relative error. We develop a hierarchical scheme for progressive construction of the TF from paths sampled during rendering. In addition to the approach taken in previous work, where the TF is defined in the image plane, ours can be associated with compact spatial regions. This allows us to take advantage of illumination coherence to more robustly estimate the TF while adapting to geometry discontinuities. To sample from this TF, we design a new replica exchange Metropolis scheme. We apply our algorithm in progressive photon mapping and show that it often outperforms alternative approaches in terms of image quality by a large margin.

Categories and Subject Descriptors: I.3.7 [**Computer Graphics**]: Three-Dimensional Graphics and Realism—*Display algorithms*

General Terms: Algorithms, Performance

Additional Key Words and Phrases: Global illumination, light transport simulation, Markov chain Monte Carlo, Metropolis-Hastings algorithm, progressive photon mapping

ACM Reference Format:

Adrien Gruson, Mickaël Ribardière, Martin Šik, Jiří Vorba, Rémi Cozot, Kadi Bouatouch, and Jaroslav Křivánek. 2016. A spatial target function for Metropolis photon tracing. ACM Trans. Graph. 36, 1, Article 4 (November 2016), 13 pages.
DOI: http://dx.doi.org/10.1145/2963097

1. INTRODUCTION

The recent widespread adoption of Monte Carlo (MC) light transport simulation in the practice of realistic rendering has revealed several limitations of the existing algorithms. One of the most pressing issues is their low efficiency in scenes with complex visibility, where just a small fraction of emitted light contributes to the image, often only after multiple interactions with surfaces. Such scenes are difficult, especially for algorithms employing photon density estimation [Hachisuka and Jensen 2009]. Although such methods often excel at dealing with complex light transport, a good sampling strategy for subpaths originating from the light sources is critical to their success. Indeed, when their sampling is blind to the camera location, most computation effort may be wasted on processing paths that make no image contribution [Hachisuka and Jensen 2011; Vorba et al. 2014].

Our work focuses on improving light subpath sampling for photon density estimation. Specifically, we use Metropolis sampling to guide light subpaths toward the camera. Although some previous works have taken this route [Fan et al. 2005; Chen et al. 2011; Hachisuka and Jensen 2011], the existing algorithms usually result in a highly nonuniform distribution of the generated path vertices, or "photons", producing a nonuniform image error. This is undesirable, as a few high-error regions force the rendering calculation to continue for a long time.

To address this problem, we propose a target function (TF) (i.e., nonnormalized target distribution) for a Metropolis photon sampler that aims to equalize relative error over several radiance measurements. Whereas a similar strategy has previously been used in Metropolis light transport (MLT) [Veach 1997; Hoberock and Hart 2010], our work focuses specifically on photon tracing in progressive photon mapping [Hachisuka and Jensen 2009]. By using

This work was supported by Charles University in Prague, projects GA UK 164815 and 340915, by grant SVV-2016-260332, and by Czech Science Foundation grants 13-26189S and 16-18964S.
Authors' addresses: A. Gruson; email: adrien.gruson@gmail.com; M. Ribardière; email: mickael.ribardiere@univ-poitiers.fr; M. Šik; email: martin_sik@centrum.cz; J. Vorba; email: jirka@cgg.mff.cuni.cz; R. Cozot; email: remi.cozot@irisa.fr; K. Bouatouch; email: kadi@irisa.fr; J. Křivánek; email: jaroslav.krivanek@mff.cuni.cz.

DOI: http://dx.doi.org/10.1145/2963097

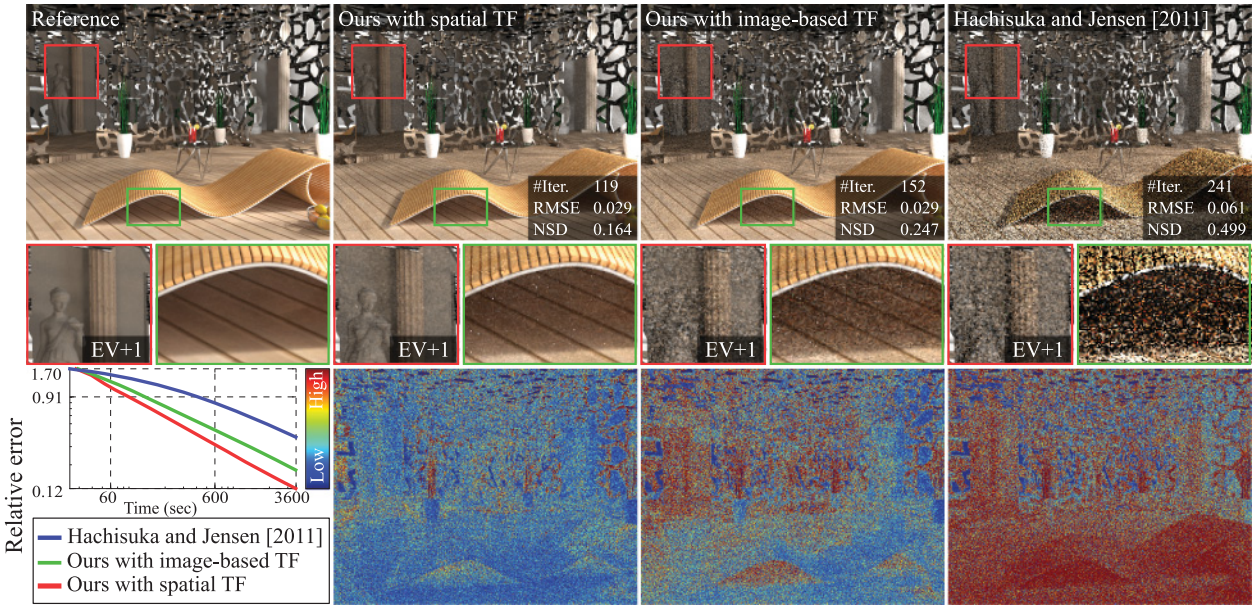

Fig. 1. We propose a new TF for Metropolis photon tracing that distributes relative error (bottom row) roughly uniformly over the image plane. It produces images of superior quality than state-of-the-art methods such as Hachisuka and Jensen [2011] in scenes with complex visibility and high dynamic range of indirect illumination. As shown in this equal-time comparison (30 minutes), our new spatial formulation of the TF (second row) produces superior results to the more common image-based TF (third row).

our TF for Metropolis sampling of light subpaths, the generated photons cover the regions of interest more evenly and in turn yield a roughly uniform distribution of relative error (i.e., perceived noise). Equalizing relative—as opposed to absolute—error is desirable because the human visual system is sensitive to relative differences in luminance. Furthermore, the uniformity of noise is beneficial for image filtering (denoising) algorithms [Zwicker et al. 2015]. Let us emphasize that our method is designed to equalize error on diffuse surfaces, although the error will usually still vary on glossy ones.

Our TF is given by the inverse of the expected number of photons contributing to each measurement (e.g., a pixel value). Such a TF is unknown at the outset, and computing it is as hard as rendering the image itself. To address this issue, we develop a hierarchical scheme for progressively constructing the TF from paths generated during rendering.

Although the TF can be calculated in image space as in previous work [Veach 1997; Hoberock and Hart 2010], we show that defining it over spatial regions organized in a spatial hierarchy yields improved results, as illustrated in Figure 1. The spatial TF benefits from the fact that illumination is often more coherent over compact spatial regions than over image regions; this in turn yields more accurate TF estimates.

Excessive sample correlation could result from using the basic Metropolis algorithm to sample from our TF because of its multimodal shape and relatively high dynamic range. To ensure good exploration without "getting trapped" in the TF peaks, we design a new replica exchange [Neal 1996] Metropolis scheme following multiple parallel Markov chains.

We apply our tracing algorithm in progressive photon mapping [Hachisuka and Jensen 2009] and show that it often outperforms the existing photon tracing algorithms, as well as the multistage MLT [Hoberock and Hart 2010] algorithms, by a large margin. Our main contributions are the following:

—A theoretical derivation of a TF that ensures an approximately equal distribution of relative error among a number of MC estimators that share a common set of samples (i.e., paths)
—A heuristic TF for light subpath tracing in photon density estimation, motivated by the preceding derivation
—A scheme for estimating the TF progressively during rendering based on either a spatial or image-based hierarchy
—A robust replica exchange design for Metropolis-based sampling of paths according to the estimated TF.

2. RELATED WORK

The specific problem treated in this paper – equalizing the distribution of error in the image – has not received much attention in the rendering research dealing with photon mapping or Metropolis sampling. We therefore review works from a wider area, that is, algorithms designed to handle scenes with difficult visibility, while focusing on approaches based on Metropolis sampling.

Metropolis Light Transport. MLT [Veach and Guibas 1997] was the first use of the Metropolis-Hastings (MH) algorithm [Hastings 1970] in light transport simulation. A major advantage of the MH algorithm is its ability to generate samples (light paths) proportionally to an arbitrary scalar TF (i.e., nonnormalized target distribution), which may include the otherwise difficult-to-sample path visibility. This unique feature theoretically makes MLT and related methods well suited to rendering large and/or highly occluded scenes.

The MH algorithm generates a Markov chain of samples (in our case, a sequence of paths), where a new sample is generated by mutating the current one. Intricate path mutation strategies have been devised to help the MLT algorithm deal with complex light transport, such as specular-diffuse-specular interactions. But even the most advanced mutation strategies known to date [Jakob and

Marschner 2012; Hanika et al. 2015] often fail to converge to the desired result in acceptable time [Vorba et al. 2014].

Equalizing Image Error in MLT. Veach and Guibas [1997] use a TF given by the luminance value of the pixel contribution of a full transport path. A well-known problem of this TF is a low number of paths contributing to dim image regions, resulting in their large relative error. Veach [1997] addressed this issue with his two-stage MLT, where a low-resolution image is rendered first and the inverse of the pixel luminance values is then used to rescale the original TF for the actual rendering. The idea is that if all pixels receive the same expected number of samples, relative error will be equalized. The results of our derivation in the context of photon tracing are in line with the preceding Veach observation.

Hoberock and Hart [2010] pointed out some important deficiencies of two-stage MLT that may deteriorate the image quality compared to plain MLT. They propose a multistage MLT algorithm, where an estimate of the rendered image is continually refined during the rendering and used to progressively update the TF. Our work is based on a similar idea, but we address the case of photon density estimation, which provides improved robustness to complex transport thanks to its inherent subpath reuse and regularization properties [Kaplanyan and Dachsbacher 2013b]. Furthermore, we show that by defining our TF over scene surfaces, as opposed to the image plane, we can better benefit from illumination coherence for more accurate progressive TF calculation. Consequently, our approach outperforms the multistage MLT in the majority of our test scenes (see Section 5).

Various other TFs for MLT have been developed. In gradient-domain MLT [Lehtinen et al. 2013], gradient of path contributions is used as the TF so that image discontinuities are explored better than flat areas. Multiplexed MLT [Hachisuka et al. 2014] uses a TF given by the path contribution modulated by the multiple importance sampling (MIS) weight [Veach 1997] of the sampling technique used to generate the path. That way, appropriate sampling techniques are selected more often.

Metropolis Sampling in Photon Density Estimation. We have chosen photon density estimation as the basis of our approach to be able to benefit from its robustness to complex light transport. As discussed in Section 1, a good photon tracing strategy is critical for good performance of photon density estimation, and ours is not the first work to apply MH sampling for this purpose. Fan et al. [2005] ran the original MLT algorithm and used selected vertices of the sampled paths as photons. More closely related are the works of Hachisuka and Jensen [2011], Chen et al. [2011], and Zheng and Zheng [2015]. All of these algorithms are designed for use in stochastic progressive photon mapping (SPPM) [Hachisuka and Jensen 2009], and their purpose is to guide light subpaths toward a set of measurement points distributed on scene surfaces. Our work follows this general scheme. These methods differ mostly by the TF for the MH sampler. Hachisuka and Jensen [2011] use the path visibility—that is, a binary variable indicating whether or not the path contributes to any of the measurement points. Chen et al. [2011] modulate the visibility by a somewhat arbitrary function constructed by filtering and nonlinear remapping of local photon density estimated in a pilot photon tracing pass. Zheng and Zheng [2015] additionally modulate the visibility by a quantity representing visual importance [Christensen 2003] so that salient areas receive more photons. A major shortcoming of these methods is their tendency to generate paths that lead to a highly varying error of radiance estimates at the measurement points. Our TF also incorporates path visibility, but we modulate it to ensure a more uniform relative error distribution.

Hachisuka and Jensen [2011] introduce important optimizations of the MH sampler itself, namely adaptive mutation size [Rosenthal 2011] and replica exchange [Neal 1996; Kitaoka et al. 2009], which we adopt and extend in our work. Gruson et al. [2014] improve on the photon tracing algorithm of Hachisuka and Jensen [2011] by employing different Markov chains—each with its own adaptive mutation size—for user-specified sections of the scene, such as interiors and exteriors.

Metropolis Sampling for Virtual Point Lights. A related problem to photon tracing is the distribution of virtual point lights (VPLs) for many-lights rendering algorithms [Dachsbacher et al. 2014]. In this context, Segovia et al. [2007] propose the use of multiple-try Metropolis to guide VPLs toward the camera.

Local Path Sampling. Metropolis sampling is not the only approach to guide light subpaths toward the camera. Another option is to devise suitable local sampling probability density functions (pdfs) for constructing light subpaths vertex by vertex. These pdfs can be constructed adaptively based on the observed contributions of previously generated paths [Dutré and Willems 1995] or by using directional density estimation from importance particles distributed in the scene in a preprocessing phase [Jensen 1995; Peter and Pietrek 1998; Bashford-Rogers et al. 2014]. We compare our results to a state-of-the-art work from the latter category [Vorba et al. 2014] in Section 5. Christensen [2003] provides a comprehensive survey of the use of importance in rendering.

3. ACHIEVING UNIFORM RELATIVE ERROR

The objective of our work is to develop a photon tracing algorithm that produces uniform relative error of radiance estimates on a set of measurement points distributed on scene surfaces. We build on the works of Hachisuka and Jensen [2011] and Chen et al. [2011], where the MH algorithm generates samples in the primary-sample space $\mathcal{U}$ [Kelemen et al. 2002], which are then transformed into the path space Ω to produce the actual photon subpaths. We achieve the desired uniform error distribution by importance sampling in the primary-sample space using a suitably defined scalar TF $\hat{T} : \mathcal{U} \rightarrow \mathbb{R}$ for the MH sampler.

In the rest of this section, we give a derivation of a canonical TF that ensures an approximately equalized distribution of relative error among several radiance estimates on diffuse surfaces. The result of this derivation is then used, in Section 4, as a basis for a heuristically defined TF employed in our rendering algorithm.

3.1 Problem Statement

Photon Mapping in the Path Space. We consider photon tracing in the context of SPPM [Hachisuka and Jensen 2009], where our goal is to evaluate radiance measurements I_k at a number of measurement points G_k. The latter are created by tracing subpaths from the camera until a sufficiently diffuse surface is encountered, where a measurement point is then deposited. The measurements are given by integrals of the form

$$I_k = \int_{\Omega} h_k(\bar{\mathbf{x}}) f(\bar{\mathbf{x}}) \mathrm{d}\mu(\bar{\mathbf{x}}), \tag{1}$$

where $\bar{\mathbf{x}}$ is a subpath starting from a light source and $h_k(\bar{\mathbf{x}}) f(\bar{\mathbf{x}})$ is the measurement contribution of that subpath. The function $f(\bar{\mathbf{x}})$, common to all of the measurements, is given by the product of the emitted radiance at the first path vertex, scattering terms at the interior vertices, and geometry and visibility terms for the path edges [Veach 1997]. The functions $h_k(\bar{\mathbf{x}})$, which are specific to each

measurement, are the density estimation kernels centered at the measurement points G_k.[1]

Photon Mapping in the Primary-Sample Space. We employ a formulation in the primary-sample space $\mathcal{U} = \cup_{i=1}^{\infty}[0,1)^i$ [Kelemen et al. 2002] (the union over i considers paths of different lengths), where the measurements I_k are given by

$$I_k = \int_{\mathcal{U}} \hat{h}_k(\bar{\mathbf{u}})\hat{f}(\bar{\mathbf{u}}) \left| \frac{\mathrm{d}\mu(\bar{\mathbf{x}})}{\mathrm{d}\bar{\mathbf{u}}} \right| \mathrm{d}\bar{\mathbf{u}} = \int_{\mathcal{U}} \frac{\hat{h}_k(\bar{\mathbf{u}})\hat{f}(\bar{\mathbf{u}})}{\hat{p}(\bar{\mathbf{u}})} \mathrm{d}\bar{\mathbf{u}}. \tag{2}$$

The mapping $\bar{\mathbf{x}} = P^{-1}(\bar{\mathbf{u}})$, $\bar{\mathbf{u}} \in \mathcal{U}$ from the primary-sample space to the path space is given by the inverse cumulative distribution function (cdf) of the probability distribution used for path sampling on the path space. The corresponding path pdf p is the product of local pdfs for light emission sampling, BRDF sampling, and Russian roulette, used to generate the path vertices from a vector of "random numbers" $\bar{\mathbf{u}}$. Here we have introduced the notation $\hat{h}_k(\bar{\mathbf{u}}) = h_k(P^{-1}(\bar{\mathbf{u}}))$ that we employ also for functions $\hat{f}$ and $\hat{p}$.

Objective. Our goal is to construct MC estimators $\langle I_k \rangle$ of the radiance measurements so that their relative error, or normalized standard deviation (NSD) $\mathrm{nsd}(\langle I_k \rangle) = \sqrt{\mathrm{var}(\langle I_k \rangle)/I_k^2}$, is the same for all k.

3.2 The Canonical TF

To ensure a uniform relative error, we use the MH algorithm to generate samples $\bar{\mathbf{u}}$ from a distribution proportional to the unnormalized TF $\hat{T}$ that we seek to derive.

General Case. As shown in Appendix A, under the assumption that the density estimation kernels at the measurement points do not overlap (i.e., have disjoint supports), we arrive at a piecewise constant TF $\hat{T}$ given by

$$\hat{T}_k \propto \frac{E[\langle I_k \rangle_{\mathrm{uni}}^2]}{I_k^2}. \tag{3}$$

Here, $\hat{T}_k$ is the (constant) TF value over the support of the kernel $\hat{h}_k$, and $E[\langle I_k \rangle_{\mathrm{uni}}^2]$ is the second moment of the estimator $\langle I_k \rangle$ that uses uniform sampling on the primary-sample space:

$$E[\langle I_k \rangle_{\mathrm{uni}}^2] = \int_{\mathrm{supp}(\hat{h}_k)} \left[\hat{h}_k(\bar{\mathbf{u}})\hat{f}(\bar{\mathbf{u}})/\hat{p}(\bar{\mathbf{u}})\right]^2 \mathrm{d}\bar{\mathbf{u}}. \tag{4}$$

Simplified Formulation. For our practical implementation, we further simplify the general result in Equation (3) by making two assumptions:

(1) We assume that photons are emitted proportionally to the emitted radiance and scattered or absorbed proportionally to the BRDF (i.e., the common strategy used for photon tracing). The path space pdf p of this strategy is roughly proportional to the contribution function (i.e., $\hat{p}(\bar{\mathbf{u}}) \approx \hat{f}(\bar{\mathbf{u}})/c$), where c is the "flux" of the resulting photons.
(2) Furthermore, we assume piecewise constant density estimation kernels (i.e., $\hat{h}_k(\bar{\mathbf{u}}) = 1/s_k$, with $s_k = \pi r_k^2$) for $\bar{\mathbf{u}}$ such that the last vertex of the corresponding path $\bar{\mathbf{x}} = P^{-1}(\bar{\mathbf{u}})$ is within the radius r_k from the measurement point G_k and $\hat{h}_k(\bar{\mathbf{u}}) = 0$ otherwise.

[1]Note that for the TF derivation, we make a simplifying assumption that the BRDF at the measurement points is not a part of the measurement itself. As a result, our TF does not adapt to the BRDF, and measurements on glossy surfaces will have higher errors than on diffuse surfaces, due to the additional variance from BRDF evaluation that is not importance sampled. Generalizing our derivation to directional BRDFs is left for future work.

With these assumptions, the second moment given by Equation (4) becomes equal to $c^2/s_k^2 \int_{\mathrm{supp}(\hat{h}_k)} \mathrm{d}\bar{\mathbf{u}}$, and from Equation (1) we have $I_k^2 = (c/s_k \int_{\mathrm{supp}(\hat{h}_k)} \mathrm{d}\bar{\mathbf{u}})^2$. Equation (3) then simplifies to

$$\boxed{\hat{T}_k \propto \frac{1}{P_k^{\mathrm{uni}}} \qquad \text{with} \qquad P_k^{\mathrm{uni}} = \int_{\mathrm{supp}(\hat{h}_k)} \mathrm{d}\bar{\mathbf{u}}.} \tag{5}$$

Here, P_k^{uni} is actually the probability that a path generated by uniform sampling in the primary-sample space will make a nonzero contribution to the estimate at the point G_k. This becomes apparent if we write the preceding integral for P_k^{uni} directly in the path space: $P_k^{\mathrm{uni}} = \int_{\mathrm{supp}(h_k)} p(\bar{\mathbf{x}})\, \mathrm{d}\mu(\bar{\mathbf{x}})$. The net result of sampling in the primary-sample space proportionally to TF given by the inverse of P_k^{uni} (Equation (5)) is that all measurement points get an equal probability of receiving a nonzero contribution. In other words, equalizing relative error corresponds to equalizing the number of contributions.

3.3 Discussion

One could argue that none of the two preceding assumptions hold in practice (i.e., $\hat{p}$ is rarely exactly proportional to $\hat{f}$, and kernels may have some overlap). Nonetheless, we show empirically in Section 5 that the heuristic TF built on the result derived earlier still achieves approximately uniform error distribution.

Some algorithms utilize a path space importance sampling strategy for which the pdf $\hat{p}$ is not even approximately proportional to the contribution function $\hat{f}$. An example is found in the work of Vorba et al. [2014], where light subpaths are guided toward the camera by modifying the path space pdf $\hat{p}$. The use of light portals for photon emission would be another example. In such a case, no direct relation between the relative error and the number of contributions exists, as the measurement variance is not only due to the varying number of contributions but also due to the varying photon fluxes, given by the ratio $\hat{f}/\hat{p}$. The utility of the more general result in Equation (3) consists of its applicability even when such a more general path sampling strategy is used. Nonetheless, our algorithm, described in the next section, relies on a heuristic TF based on the result in Equation (5), whereas we leave the combination of our approach with alternative path sampling strategies for future work.

The simplified result in Equation (5) is in line with the argument that Veach [1997] made in the context of MLT: uniform relative error on pixels is achieved by equalizing the number of contributing paths. Although our derivation might appear as a lengthy way of arriving at a similar result, its value lies in clearly identifying the involved assumptions while also deriving the more general result in Equation (3).

Finally, although our derivation relies on the primary-sample space formulation, equivalent results can be achieved directly in the path space. As such, the derived formulas are not limited to the use with primary-sample space–style mutations.

4. ALGORITHM

In this section, we develop a heuristic TF based on the result derived in the previous section, along with an algorithm for estimating the TF in the course of rendering (Sections 4.1 through 4.3). Furthermore, we develop a replica exchange method for robustly sampling light subpaths according to this heuristic TF (Section 4.4).

The basis for our heuristic TF is the canonical TF given by the inverse of the expected number of photons contributing to each measurement point (Equation (5)). This quantity is unknown at the outset, and we estimate it from statistics of the generated path

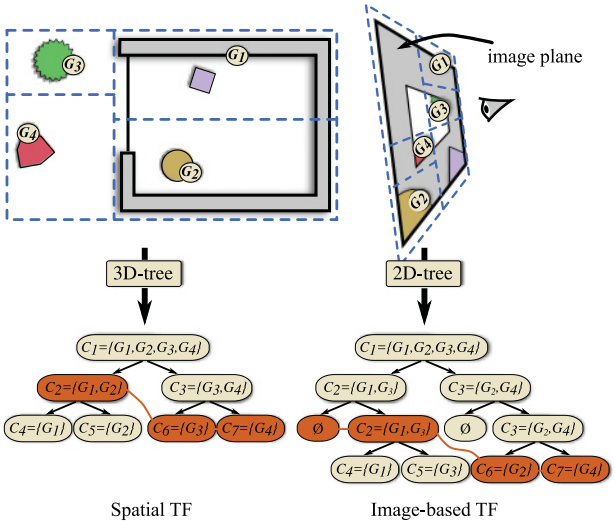

Fig. 2. Regions used to maintain the statistics for the TF estimation are defined by the nodes of a spatial hierarchy constructed prior to rendering. The regions can be defined either in world space (spatial TF, left) or on the image plane (image-based TF, right). For any given algorithm iteration, the current set of regions is given by a cut through the hierarchy (highlighted in orange).

vertices (photons) maintained over a hierarchy of spatial or image regions. The statistics are subject to variance, especially in early stages of calculation. To obtain robust TF estimates despite the variance, we initially average the statistics over large regions, which are then adaptively refined.

Algorithm Overview. The algorithm works in iterations and refines the TF estimates as it progresses. Iteration i uses the TF $\hat{T}_i$ for sampling light subpaths, whereas updating MC estimators will be used to calculate an updated TF $\hat{T}_{i+1}$ for the next iteration. In this way, the TF approaches the desired TF given by Equation (5). Note that the TF is only used to sample light subpaths (i.e., photons), whereas the camera subpaths used to distribute the measurements points always use an independent MC sampler.

To calculate the TF according to Equation (5), we need to estimate, for each measurement point G_k, the probability P_k^{uni} of receiving a photon under uniform sampling in the primary-sample space. The estimates of P_k^{uni} need to be carried over from one iteration to another so that their variance can eventually vanish. This is complicated by the fact that each iteration uses a new, independently generated set of measurement points. SPPM [Hachisuka and Jensen 2009] solves this by maintaining statistics associated with image pixels.

Spatial and Image Regions. We adopt the idea of statistics maintained over the iterations, but we depart from SPPM in that we associate these statistics to hierarchically organized regions C. These are defined either in the 3D space (spatial regions) or on the image plane (image regions) as illustrated in Figure 2. A *spatial region* is defined by an axis-aligned 3D box, and a measurement point belongs to a given region if it is contained by the associated box. An *image region* is defined by an axis-aligned rectangle on the image plane: a measurement point belongs to a region if it was generated by tracing a camera subpath through the associated rectangle. Further details on the region construction and their use will be given in Section 4.2.

We refer to a TF calculated using statistics from the spatial regions as a spatial TF, whereas a TF calculated using statistics from image regions is referred to as an image-based TF. As shown in Figure 1, the spatial TF is more robust to illumination changes caused by geometry discontinuities because it averages statistics over compact spatial regions, where illumination can more likely be expected to be coherent. On the other hand, the image-based TF averages statistics over image regions that may encompass spatially distant locations with completely different illumination (see Figure 2). This in turn can negatively impact the accuracy of the estimated TF.

4.1 TF Calculation

To calculate the TF, we associate with each spatial or image region C a statistic $\kappa(C)$, which is a running estimate of the probability P_k^{uni} averaged over the measurement points G_k that have so far been generated in this region. The statistic $\kappa(C)$ persists across the iterations and is continually updated.

Let us start by estimating the probability $P_k^{\text{uni}} = \int_{\text{supp}(\hat{h}_k)} \mathrm{d}\bar{\mathbf{u}}$ (Equation (5)) for a single measurement point. We use the photons generated in iteration i to evaluate $\psi_i(G_k)$, an MC estimator of the probability P_k^{uni} for each measurement point G_k. If we used uniform sampling in the primary-sample space, this could be calculated simply as a fraction of paths that make a nonzero contribution to G_k. However, this would be a bad estimator because such paths are extremely scarce in the highly occluded scenes that we consider. We can obtain a more accurate estimate by using paths sampled by the Metropolis sampler using path visibility as the TF $\hat{T}_i^V$ [Hachisuka and Jensen 2011]. Using these paths, $\psi_i(G_k)$ is a standard importance sampling estimator of the probability P_k^{uni},

$$\psi_i(G_k) = \frac{1}{N} \sum_{\bar{\mathbf{u}}_j \in \mathcal{S}_k} \frac{1}{\hat{T}_i^V(\bar{\mathbf{u}}_j)/b_i}, \tag{6}$$

because the paths are sampled from the pdf given by $\hat{T}_i^V(\bar{\mathbf{u}}_j)/b_i^V$ (i.e., the normalized TF). The normalization by $b_i^V = \int \hat{T}_i^V(\bar{\mathbf{u}})\mathrm{d}\bar{\mathbf{u}}$ is carried out at the end of the iteration, as b_i^V is not known earlier. The calculation of the normalization factor is detailed in Section 4.4 and in the supplemental document. In the preceding formula, N is the number of light subpaths sampled in each iteration, and the sum runs over the paths that fall into the support of the kernel at G_k (i.e., $\mathcal{S}_k = \{\bar{\mathbf{u}}_j \mid P^{-1}(\bar{\mathbf{u}}_j) \in \text{supp}(h_k)\}$).

To estimate the overall probability P^{uni} for a region C, we sum $\psi_i(G_k)$ over all measurement points in that region:

$$\psi_i(C) = \sum_{G_k \in C} \psi_i(G_k). \tag{7}$$

Finally, the estimator κ for the next iteration is calculated as

$$\kappa_{i+1}(C) = [\kappa_i(C) + \psi_i(C)] \frac{s_{i+1}(C)}{s_i(C)}. \tag{8}$$

Here, $s_i(C)$ is the sum of the kernel sizes $s_k = \pi r_k^2$ at the measurement points contained by the region C in iteration i:

$$s_i(C) = \sum_{G_k \in C} s_k. \tag{9}$$

Equation (8) is a recursive form of a weighted sum of the estimators $\psi_i(C)$ over the iterations, similar to SPPM [Hachisuka and Jensen 2009]. The fraction $s_{i+1}(C)/s_i(C)$ accounts for the fact that a spatial region may contain a different number of measurement points in each iteration and that the measurement point radii r_k may be reduced as the calculation progresses.

A normalized estimate of the overall probability $P_i^{\text{uni}}(C)$ for a region C at iteration i can be obtained by dividing $\kappa_i(C)$ by $N_i(C)$, the number of iterations in which the region C contained at least one measurement point: $P_i^{\text{uni}}(C) = \kappa_i(C)/N_i(C)$. This normalization disregards iterations in which the region C did not contain any measurement points, because such iterations cannot make any contribution to $\kappa_i(C)$ despite the region C possibly being reachable by light subpaths. Considering such iterations in the normalization would make the estimate $P_i^{\text{uni}}(C)$ biased.

Two Components of the TF. We could now use Equation (5) to define the TF as an inverse of the per-region probability $P_i^{\text{uni}}(C)$ defined earlier. But because $P_i^{\text{uni}}(C)$ is constant in each region, the resulting TF would be piecewise constant as well, which would be a poor approximation of the desired TF, especially for large regions. We now show how to take advantage of the region statistics without forcing the TF to be constant in each of them. This provides a more accurate TF approximation.

We use the fact that the probability P_k^{uni} at any measurement point can be approximated as a product of the local area density of photons $D^{\text{uni}}(G_k)$ (under uniform sampling in the primary-sample space) and the kernel size s_k (i.e., $P_k^{\text{uni}} = D^{\text{uni}}(G_k)s_k$). Since the kernel sizes s_k are known, they can be factored out in the TF calculation and provide the desired modulation within each region, which yields nonconstant per-region TF.

The local density $D^{\text{uni}}(G_k)$ at any measurement point is approximated by the average density $D^{\text{uni}}(C)$ over the spatial region containing G_k. At iteration i, the average density in a region can be calculated as

$$D_i^{\text{uni}}(C) = \frac{P_i^{\text{uni}}(C)}{s_i(C)} = \frac{1}{N_i(C)} \frac{\kappa_i(C)}{s_i(C)}. \tag{10}$$

As before, $N_i(C)$ is the number of iterations in which the region C contained at least one measurement point.

TF for a Measurement Point. We could now calculate the TF $T_i(G_k)$ at any measurement point as the inverse of $D^{\text{uni}}(G_k)s_k$. However, both the local density $D^{\text{uni}}(G_k)$ and the kernel size s_k can have a very high dynamic range. Using them directly could result in a TF that would be difficult to explore for the Metropolis algorithm. For this reason, we compress the dynamic range of both the inverse density and the inverse kernel size. The final formula for the TF at a measurement point G_k then reads as follows:

$$T_i(G_k) = \underbrace{\left[\epsilon_1 \frac{1}{\frac{D_i^{\text{uni}}(C(G_k))}{\max_l\{D_i^{\text{uni}}(C_l)\}} + \epsilon_1}\right]}_{\text{inverse average density in region}} \cdot \underbrace{\left[\epsilon_2 \frac{1}{\frac{s_k}{\max_l\{s_l\}} + \epsilon_2}\right]}_{\text{inverse kernel size}}. \tag{11}$$

Here we have used $C(G_k)$ to denote the region that contains the measurement point G_k. Dividing by the maxima is used to normalize the range of the respective quantities: $\max_l\{D_i^{\text{uni}}(C_l)\}$ is the maximum of all region densities, and similarly, $\max_l\{s_l\}$ is the maximum kernel size over all measurement points. Both of these maxima consider only regions and measurement points in the current iteration. The parameters ϵ_1 and ϵ_2 ensure that the range of the respective component of the TF remains within the interval $[\epsilon/(1+\epsilon), 1]$, where $\epsilon \in \{\epsilon_1, \epsilon_2\}$. The use of ϵ_1 serves as a protection against paying unnecessary attention to insignificant absolute errors in dim regions (which would have extremely large TF if ϵ_1 were not used). We use $\epsilon_1 = \epsilon_2 = 10^{-4}$ in all of our results. Figure 3 illustrates the two components of the TF.

TF for a Full Path. To compute the TF $T_i(\bar{\mathbf{x}})$ of an entire light subpath $\bar{\mathbf{x}}$, we take the maximum of the TF values for the measurement points $\mathcal{K}$ impacted by the path vertices. Taking the maximum is motivated by the preference for exploring difficult regions:

$$T_i(\bar{\mathbf{x}}) = \max_{k \in \mathcal{K}}\{T_i(G_k)\}. \tag{12}$$

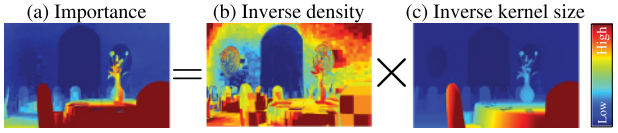

Fig. 3. The TF $\hat{T}(G_k)$ for a measurement point (a) is given by the product of two components: the inverse of the average photon density in the spatial region that contains the measurement point (b) and the inverse of the density estimation kernel size at that point (c). The images show the TF for measurement points associated with the image pixels for the scene from Figure 7 (top). Before display, each of the three images is normalized independently by its maximum.

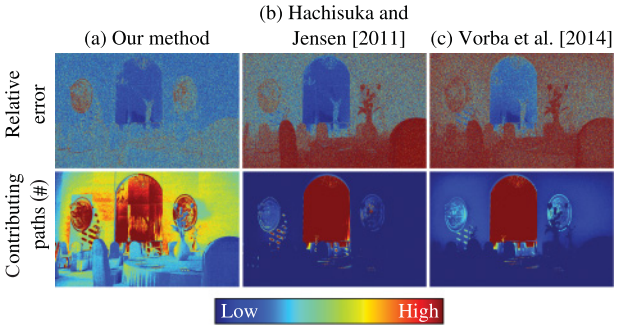

Fig. 4. Top row: Our method produces mostly uniform distribution of relative error across the image plane (a), whereas the error distribution is highly nonuniform in the images generated with the methods of Hachisuka and Jensen [2011] (b) and Vorba et al. [2014] (c). The relative error is calculated as an absolute difference from a reference image divided by the same reference image. Bottom row: The average number of contributions received by the measurement points. We see that the relative error is indeed roughly inversely proportional to this quantity. The images are based on 10 minutes of rendering.

Discussion. Several factors contribute to the fact that our final estimated TF does not exactly correspond to the desired TF given by Equation (5). These are namely the variance of the TF estimation, averaging over regions, range compression in Equation (11), and taking the maximum over path vertices in Equation (12). However, we show in Figure 4 that the resulting TF still equalizes the relative error well.

4.2 Hierarchy of Spatial and Image Regions

Statistics in small regions could be subject to high variance, especially in the early stages of calculation. As a result, the estimated TF could contain some artificial spikes that would negatively affect its ability to equalize error. We adopt a natural solution where variance in the early iterations is reduced by averaging the statistics over larger regions. The regions are then refined as the calculation progresses.

To define the regions C, we build a hierarchy prior to rendering, as shown in Figure 2. A 3D-tree over the scene defines the spatial regions, whereas image regions are defined by a 2D-tree over the image. Each node of this structure corresponds to one region with its associated statistic $\kappa(C)$. The set of regions used to define the

TF $\hat{T}_i$ in iteration i is given by a cut through the tree. The idea of maintaining photon statistics in a spatial hierarchy bears some similarity to the work of Jakob et al. [2011], whereas the image-based TF is closely related to the multistage MLT algorithm of Hoberock and Hart [2010].

Hierarchy Construction. To construct the hierarchy of spatial regions, we distribute a pilot batch of measurement points in the scene by tracing four paths per pixel and construct a kD-tree over these points in a top-down manner using the median-split rule. The use of the median split ensures that regions on the same hierarchy level contain a similar number of measurement points and will therefore receive a similar number of photons. The 2D-tree of image regions is simply a regular octree in the 2D image space. The time spent on constructing the hierarchy is insignificant compared to the remaining running time.

Region Refinement Strategy. The initial cut of the tree contains only the root node that groups all of the measurement points. We refine the regions C as the rendering progresses so as to improve the accuracy with which the desired TF is represented. The goal is to avoid using regions with a high variance of their κ statistic; otherwise, our TF would be inaccurate and possibly counterproductive. We achieve this by estimating a measure of relative error of the κ statistic for each region using the formula $\text{err} = |\kappa_{\text{odd}} - \kappa_{\text{even}}|/\kappa$, where κ_{odd} and κ_{even} are estimates of κ for the region constructed using only odd or even samples (paths), respectively [Rousselle et al. 2012]. After each iteration, we refine the spatial subdivision such that the error for all regions is kept at less than 0.05 and the number of paths contributing to the region is at least 10,000. The resulting TF values using this subdivision is illustrated later in Figure 8 and the supplemental material. This adaptive refinement strategy produces significantly improved results over a simple refinement based on the number of iterations.

4.3 TF Calculation Summary

Algorithm 1 gives a summary of our algorithm, where only the steps related to TF calculation are shown; steps necessary for image rendering are similar to regular SPPM [Hachisuka and Jensen 2009]. Before the rendering, we build the hierarchy of regions (line 1). The algorithm then proceeds in iterations, each of which starts by distributing a new set of measurement points by tracing camera subpaths through image pixels (line 4). After that, we calculate the average density $D_i^{\text{uni}}(C)$ for all regions (line 5), which allows evaluation of the TF for all measurement points (line 6). We then run our replica exchange Metropolis algorithm, described in Section 4.4, which samples photon paths using the TF $\hat{T}_i$ while updating the statistic $\psi_i(G_k)$ for the measurement points (line 9). After the path sampling has finished, we gather the measurement point statistics and calculate the statistic $\kappa_{i+1}(C)$ for all regions (line 12). Note that the multiplication by $s_{i+1}(C)$ in Equation (8) is carried out at the beginning of the next iteration, when this value is known. In the final step, we evaluate our refinement criterion and possibly subdivide the hierarchy as described in Section 4.2 (line 13).

ALGORITHM 1: Overview of Our Algorithm with References to Equations Used to Calculate the TF

```
1:  ConstructRegionHierarchy()                    // Section 4.2
2:  for i = 0 ... N_iter − 1 do
3:      // Step 1: Initialization
4:      DistributeMeasurementPts()
5:      UpdateRegionDensity()                     // Equation (10)
6:      AssignTargetFuncToMP()                    // Equation (11)
7:
8:      // Step 2: Photon tracing
9:      RunMetropolisPathSampling()               // Equation (6)
10:
11:     // Step 3: Statistics update
12:     UpdateRegionStatistics()                  // Equation (8)
13:     RefineRegions()                           // Section 4.2
14: end for
```

4.4 Sampling from the TF

We conclude the algorithm description by presenting the variant of the Metropolis algorithm that we use for sampling from our TF. As shown in Figure 3, the TF has a relatively high dynamic range and could have several modes. Such functions are hard to explore via basic Metropolis sampling because the generated Markov chain tends to get trapped in the modes and produces highly correlated samples. To ensure good exploration, we design a new replica exchange Metropolis scheme [Neal 1996; Kitaoka et al. 2009; Hachisuka and Jensen 2011]. A general idea of replica exchange Metropolis is to follow several Markov chains in parallel with a probabilistic exchange of states between any two chains. Using "tempered" TFs for some of the chains ensures good exploration with low correlation even if one of the chains follows a function with a complex shape. We build on the work of Hachisuka and Jensen [2011], to which we refer for more details.

Our Four-Chain Replica Exchange Design. Our design features four chains. The first chain uses uniform, independent proposals and a constant TF. The TF of the second chain is the path visibility (a binary variable indicating whether or not the path contributes to any of the measurement points). These two chains behave in the same manner as in the work of Hachisuka and Jensen [2011]. The third chain uses as its TF the inverse of the kernel size (the second factor in Equation (11); also see Figure 3). Samples generated by this chain are distributed more evenly in the screen projection of the visible regions than the samples from the second (visibility) chain. Finally, the fourth chain follows our complete TF given by Equation (11). The chains with simpler TFs (uniform, visibility) help to efficiently explore the entire state space, whereas the remaining two chains explore local features of their more complex TFs. Technical details are given in the supplemental document.

There are two major advantages to using this four-chain design over a simpler one that would combine uniform sampling with sampling from our TF. The first benefit, as already discussed, is lower sample correlation. A no less important advantage is a more robust calculation of the normalization factor for the chain corresponding to our TF, $b_i = \int \hat{T}_i(\bar{\mathbf{u}})\bar{\mathbf{u}}$. Although this integral can be easily estimated by using paths from the first chain, the estimate often suffers from a high variance because of the complex shape of our TF. We instead calculate the normalization factor by using samples from one chain to estimate the normalization of the next chain as detailed in the supplemental document. This provides substantially more stable estimates as shown in Figure 5.

Finally, our four-chain scheme samples paths from several different distributions given by the respective TFs. To use all of these samples while minimizing the variance of the result, we combine them using MIS [Veach 1997] as detailed in the supplemental document. Figure 6 illustrates the benefits of this approach.

Markov Chain Initialization. No density estimates are available in the first iteration of our algorithm, so we initialize our TF to the inverse kernel size (second term in Equation (11)). In this case, we employ the unbiased initialization technique of Veach and Guibas [1997] and choose the initial state by importance resampling

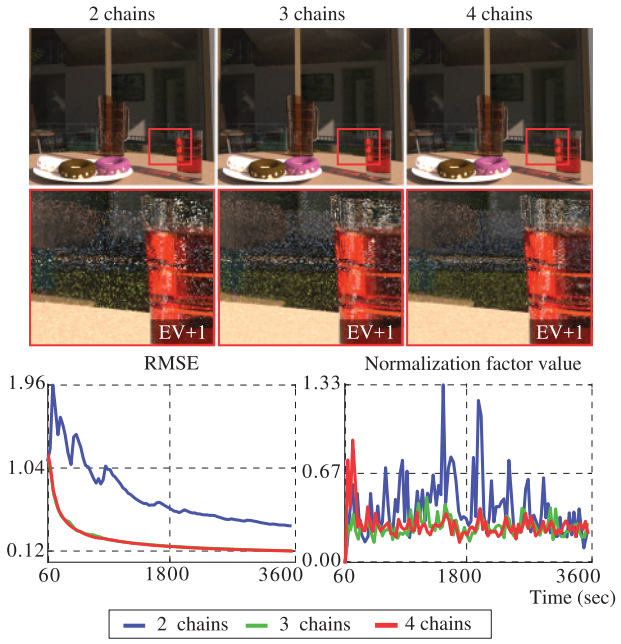

Fig. 5. Comparison of our method using two Markov chains (uniform and our TF), three chains (uniform, inverse kernel size, and our TF), and all four chains (uniform, visibility, inverse kernel size, and our TF). Adding more chains limits the possibility of "getting stuck" in one state, which reduces image noise (top) and yields more uniform convergence of the RMSE to noise-free images (bottom left). Furthermore, it also yields lower variance in the estimated normalization factor value (bottom right).

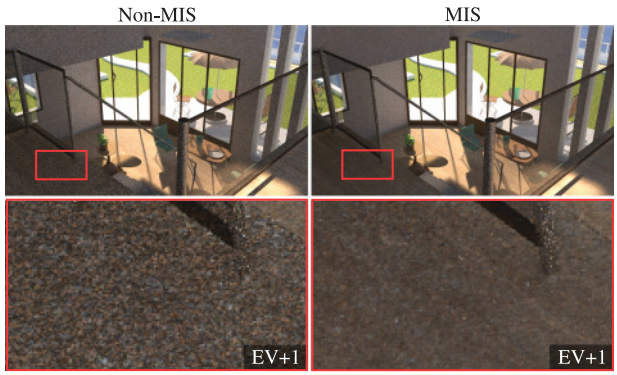

Fig. 6. Comparison of our method without (left) and with (right) the use of MIS to combine the contributions of paths generated by the different chains. MIS helps to reduce the overall noise and suppresses bright spots in the generated images. The non-MIS image uses a simple average of contributions from all four chains weighted by the number of samples taken by each chain.

from among 100,000 uniform samples. In the subsequent iterations, we simply initialize the chains by their last state from the previous iteration. Since the TF may change from one iteration to another, we apply a burn-in period by discarding the first 10,000 samples before we start accumulating the results. Although this design prevents us from formally proving unbiasedness or consistency of our method, the plots shown later in Figure 8 provide empirical evidence of the method's convergence.

5. RESULTS

Figure 7 compares the results of SPPM to different methods for sampling photon paths: the state-of-the-art methods of Hachisuka and Jensen [2011] and Vorba et al. [2014], and ours. We implemented the methods in the Mitsuba renderer [Jakob 2010]. The three algorithms share the same code for distributing and looking up the measurement points, the photon search radius is initialized using ray differentials to approximate the projected pixel size, and then it is kept constant (i.e., no radius reduction is used). The pixel-size radius is small enough to avoid any objectionable blurring, and the fact that the radius is not reduced preserves the asymptotic convergence rate of MC algorithms. We emit 10 million photon paths per iteration for all methods and generate one measurement point per image pixel. All scenes were rendered for 1 hour at a resolution of 960×540 on a 2×Intel Xeon CPU E5640 with 2.67GHz using eight logical threads. Since we use photon mapping to render all images, we left out paths with only highly glossy vertices, which cannot be efficiently handled by photon mapping. These paths can be added by brute-force MC path tracing or by specialized algorithms [Velázquez-Armendáriz et al. 2015]. In the supplemental material available at the http://cgg.mff.cuni.cz/~jaroslav/papers/2016-stf/, we provide our code, scripts to regenerate some of the results, and additional results.

In the Dinner hall scene, only the entrance hall far from the camera is strongly illuminated. This poses a problem to the previous algorithms, as they expend most of the sample budget on these distant bright areas—at the expense of the visually important dim parts of the scene. The Villa interior scene showcases a common setup in architectural visualization where the camera sees a brightly lit exterior but most of the image is occupied by the much dimmer interior. Again, prior algorithms leave the dim interior very noisy, whereas our algorithm ensures that image error is reduced equally in the whole image. The Underwater and Canyon scenes feature a high range of depth from the camera and—in Canyon—an extremely high dynamic range of illumination. In such cases, our method delivers an almost noise-free image, which is not the case with prior algorithms.

Figure 8 shows the TF and the RMSE and NSD convergence plots for the four scenes in Figure 7. Our method has a significantly lower RMSE and NSD across all scenes except the diffuse Canyon, where the method of Vorba et al. [2014] achieves a lower RMSE. Nonetheless, the NSD of our method is lower even in this case.

Spatial Versus Image-Based TF. Figure 1 showcases the advantage of our spatial TF over the image-based one. Due to specular reflections, many regions in the image-based formulation mix together statistics over measurement points from extremely differently illuminated parts of the scene. As a result, the image-based TF is unable to guide paths to appropriate scene areas because it only captures average behavior over the image regions. Our proposed spatial TF formulation, on the other hand, deals with this situation gracefully, as the spatial regions are well localized and exhibit more illumination coherence.

The Villa interior and Underwater scenes serve as worst-case scenarios for the spatial TF. Due to their large physical size, a deep spatial hierarchy is necessary to focus on the region observed by the camera. The supplemental material shows that even in such cases, the spatial TF yields only a negligibly higher NSD than the image-based TF.

Comparison to MLT and Multistage MLT. In Figure 9, we compare our algorithm to the primary-sample space MLT (PSSMLT) [Kelemen et al. 2002] and MLT with Manifold exploration (MEMLT) [Jakob and Marschner 2012] in their original and

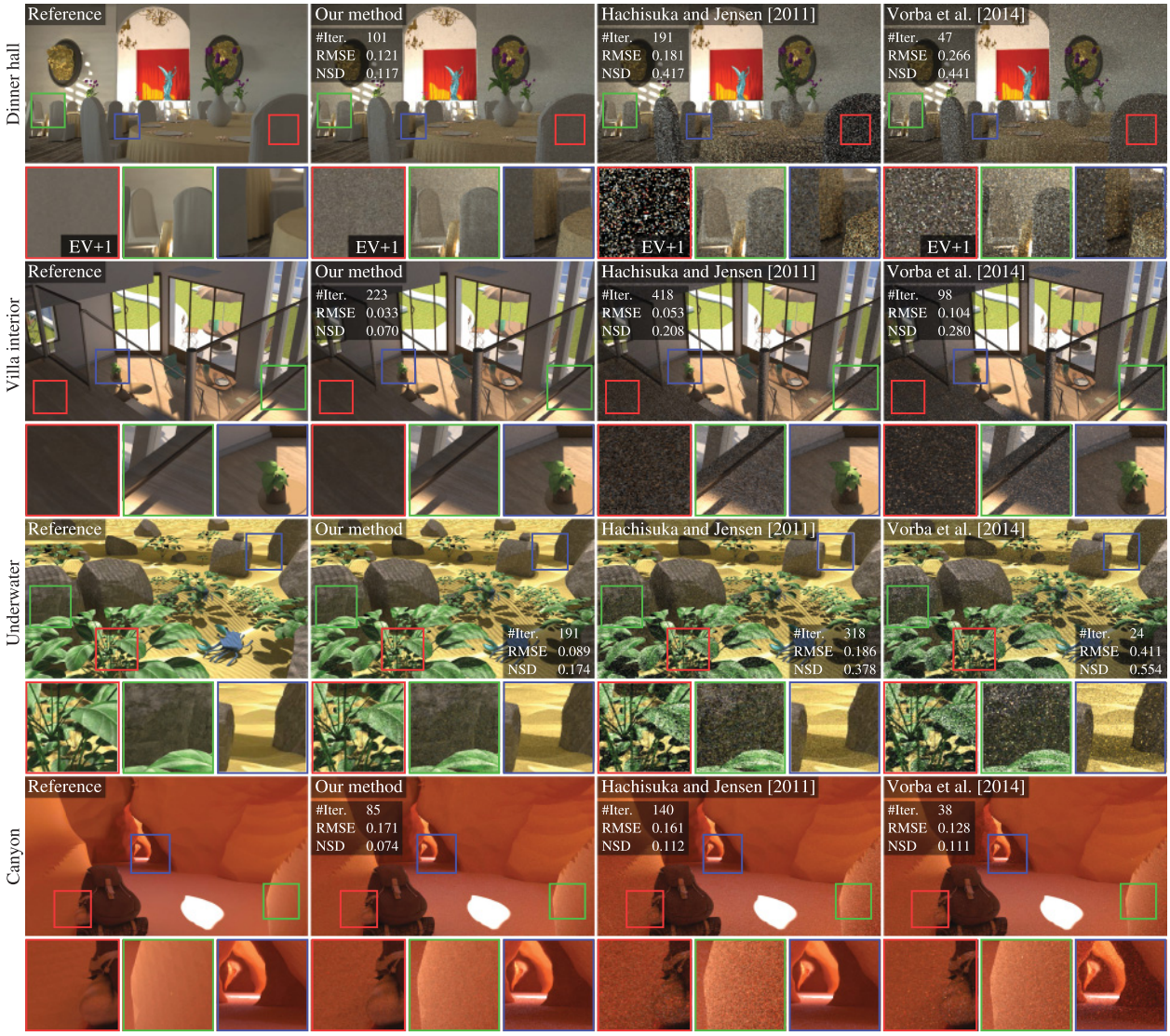

Fig. 7. Equal-time comparison (1 hour) of SPPM with three different photon tracing algorithms: our method with the spatial TF, Hachisuka and Jensen [2011], and Vorba et al. [2014].

multistage variants [Hoberock and Hart 2010]. From the positive-negative difference images in the insets, we can see that the MLT images are far from being converged, irrespective of whether or not the multistage variant is employed. This is true even for the MEMLT images that appear to be noise free. Of all of our test scenes, only in the diffuse Canyon scene (shown in the supplemental material) does the MLT algorithm outperform our method. In the other scenes, MLT presumably suffers from insufficient path space exploration due to complex specular transport and/or complex visibility. Our algorithm does not suffer from this problem thanks to the four-chain replica exchange design together with the path-space regularization afforded by the use of photon density estimation.

Performance in Simple Scenes. Advanced photon tracing methods such as ours are not necessary in simple scenes where most of the paths generated by uniform sampling in primary-sample space already contribute to the image. Although our method has some computational overhead and increased variance due to sample correlation over plain SPPM [Hachisuka and Jensen 2009], we show in Figure 10 that it does not significantly impair performance even in an extremely simple case.

Handling of Glossy Surfaces. Measurements points are deposited on the first surface encountered by a camera subpath with BRDF roughness above a user-specified threshold (*RD*). Figure 11 shows the results of our method and Hachisuka and Jensen [2011] with different threshold values in a highly glossy scene. Although our method does not achieve uniform relative error on glossy surfaces, it still produces better results than those of Hachisuka and Jensen [2011]. A low roughness threshold (first row, $RD = 0.05$) results in noisy images because measurement points are directly deposited on the glossy surfaces, which yields high variance of the photon map radiance estimates. The threshold value actually used in our results (second row, $RD = 0.39$) produces less noise. However,

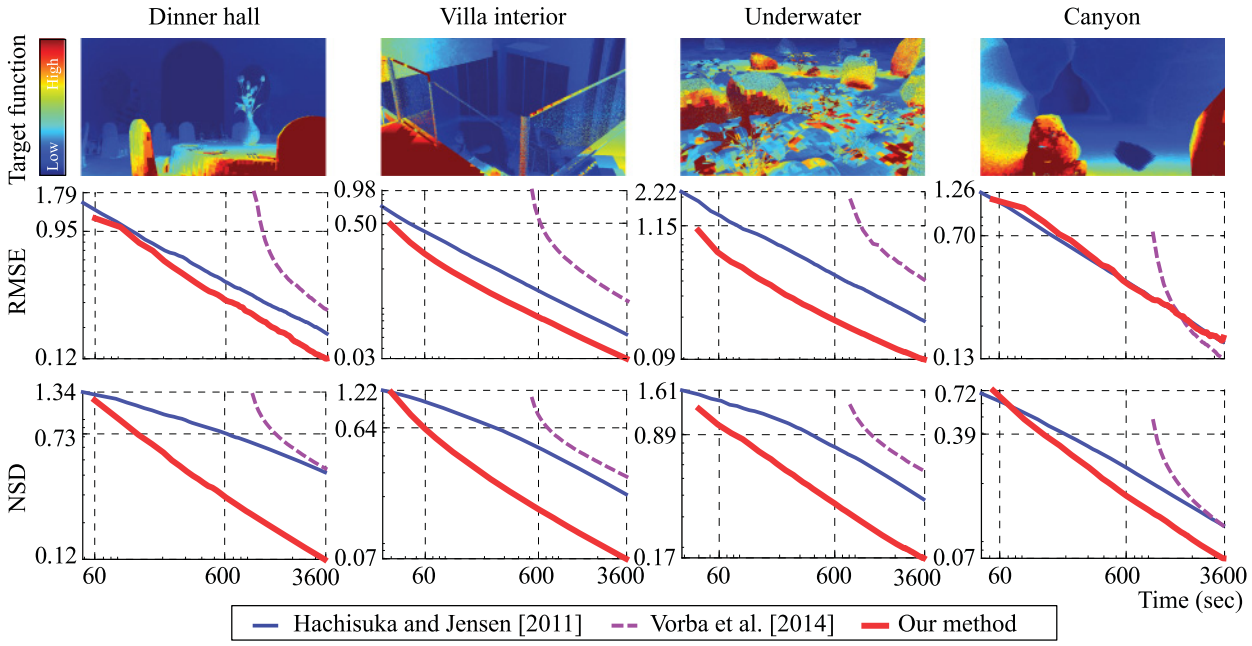

Fig. 8. The top row shows the spatial TF associated with our method after 1 hour of rendering for the scenes in Figure 7. In the bottom two rows, we give log-log plots of the evolution of the root mean square error (RMSE) and NSD over 1 hour of rendering. The plots for the method of Vorba et al. [2014] are offset from the time axis origin because of the training phase.

Fig. 9. Equal-time (1-hour) comparison of our algorithm to PSSMLT [Kelemen et al. 2002] and MEMLT [Jakob and Marschner 2012] in their original and multistage variants [Hoberock and Hart 2010]. To discount the effect of incorrectly estimated normalization factors, the PSSMLT and MEMLT images have been rescaled to match the average luminance of the reference image. The positive (green)/negative (red) difference images in the insets have been created using HDRITools (https://bitbucket.org/edgarv/hdritools). In this comparison, we have added to the results of our algorithm purely specular paths calculated with brute-force path tracing in an additional 10 minutes (included in the 60-minute total rendering time of our method).

transport along paths with all vertices below the roughness threshold is completely missing from the images. The missing energy can be recovered using brute-force path tracing (bottom left).

Additional Results. Due to space limitations, we only show a small selection of our results in the article. A more complete dataset is shown in the supplemental material, available through the TOG editorial office. For all scenes, we show the results of our method with the spatial and image-based TF, as well as the TF based only on the inverse kernel size (second term in Equation (11)). Furthermore, we provide results of the following algorithms: photon tracing with the TF of Hachisuka and Jensen [2011] and Chen et al. [2011], guided photon tracing [Vorba et al. 2014], path tracing,

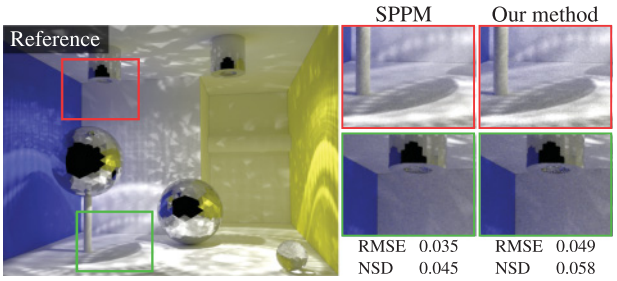

Fig. 10. Our method does not significantly impair the performance of plain SPPM in extremely simple scenes, where most of the paths generated by uniform sampling in the primary-sample space contribute to the image. To generate these images, we ran SPPM and our method for 5 minutes.

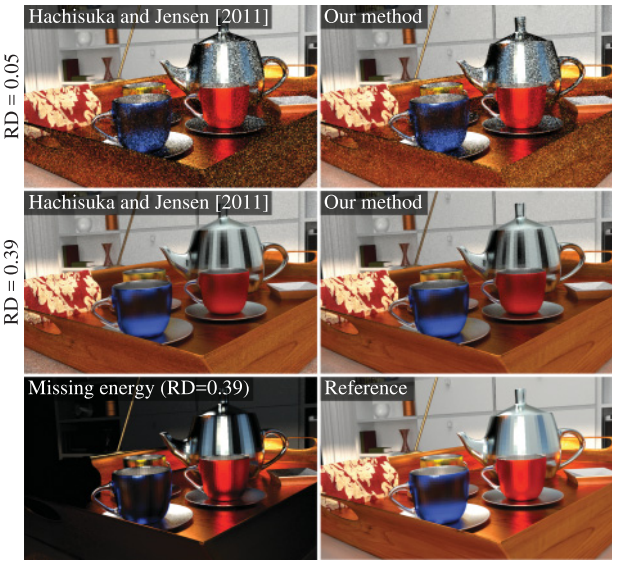

Fig. 11. Comparison of our method and Hachisuka and Jensen [2011] in a highly glossy scene. The first and second rows show rendering results for different roughness threshold (*RD*) after 1 hour of rendering. The bottom left image shows the energy missing from the second row results, calculated by path tracing in 10 minutes.

bi-directional path tracing [Lafortune and Willems 1993; Veach 1997], PSSMLT [Kelemen et al. 2002] and MEMLT [Jakob and Marschner 2012] in their original and multistage [Hoberock and Hart 2010] variants, and energy redistribution path tracing [Cline et al. 2005]. We conclude that our algorithm outperforms all of these methods except in the mostly diffuse Canyon scene, where MLT performs better because there is no complex glossy or specular transport.

6. LIMITATIONS AND FUTURE WORK

In our current implementation, statistics from differently illuminated parts of the scene—such as interior and exterior—can be averaged in one spatial region. This could negatively affect the ability of the TF to equalize error. Considering surface orientation or using a spatial subdivision that better aligns with the scene geometry to define the regions would alleviate this problem.

We currently calculate an independent estimate of the normalization factor in each rendering iteration, which can only be done by sampling a high number of photon paths per iteration. Devising a progressive normalization factor estimate would enable running a higher number of "smaller" iterations, thus improving sampling of effects that rely on camera subpaths such as glossy reflections.

As mentioned earlier, our TF derivation disregards view-dependent BRDFs on the measurement points, which results in increased error on glossy surfaces (see Figure 11). Extending the TF to take the BRDF into account would resolve this issue.

A combination of our method and adaptive progressive photon mapping (APPM) [Kaplanyan and Dachsbacher 2013a] could further improve efficiency. An interesting issue associated with this would be overcoming the assumption of independent and identically distributed photons on which the adaptive bandwidth selection in APPM relies, and which is not fulfilled in our method.

Finally, the objective of equalizing relative error is well suited when the individual measurements are directly related to the resulting pixel values, as in progressive photon mapping. However, this criterion is certainly not optimal when the light subpaths are used in a more complex way, such as in combined bi-directional algorithms [Georgiev et al. 2012; Hachisuka et al. 2012]. It would be interesting to investigate suitable sampling distributions that would equalize relative pixel error in such algorithms.

7. CONCLUSIONS

We have presented a new TF for Metropolis photon tracing along with a robust scheme for calculating the function during rendering. Its objective is to equalize the relative error of several radiance measurements estimated using a shared set of photon paths. We have identified the assumptions under which this goal can be achieved by making the number of contributions to each measurement equal. But our derivation also exposes a more general form of the TF with potential application in advanced importance sampling schemes.

Additionally, we have proposed an approach for progressive refinement of the TF estimates in the course of rendering. We have shown that a spatial, as opposed to image-based, definition of the TF provides an important advantage in certain cases. Finally, we have developed a new replica exchange Metropolis algorithm that can sample from the TF without excessive sample correlation. We have shown that this approach outperforms most existing MC-rendering algorithms in scenes with difficult visibility and high luminance range.

APPENDIX

A. TARGET FUNCTION DERIVATION

We derive the general form of the target function (TF) given by Equation (3). To achieve uniform error, we importance sample in the primary-sample space by using random variables $\bar{U}$ from the probability density function (pdf) $\hat{T}(\bar{\mathbf{u}})/b$ with $b = \int \hat{T}(\bar{\mathbf{u}})\mathrm{d}\bar{\mathbf{u}}$. Here, $\hat{T}$ is the (unnormalized) TF that we seek to derive, and the division by b ensures that $\hat{T}(\bar{\mathbf{u}})/b$ is a properly normalized pdf. This yields a classic one-sample Monte Carlo (MC) estimator of the measurements I_k:

$$\langle I_k \rangle = \frac{\hat{h}_k(\bar{U})\hat{f}(\bar{U})/\hat{p}(\bar{U})}{\hat{T}(\bar{U})/b}. \tag{13}$$

We now derive $\hat{T}$ such that the relative error given by $\mathrm{nsd}(\langle I_k \rangle) = \sqrt{\mathrm{var}(\langle I_k \rangle)/I_k^2}$ is the same for the estimators of all the measurements I_k. The variance of $\langle I_k \rangle$ is given by $\mathrm{var}(\langle I_k \rangle) = E[\langle I_k \rangle^2] - I_k^2$. Because we assume that the functions $\hat{h}_k$ have disjoint supports

and that the TF $\hat{T}$ is constant in the support of the individual $\hat{h}_k$'s (i.e., $\hat{T}(\bar{\mathbf{u}}) = \hat{T}_k$ for $\bar{\mathbf{u}} \in \text{supp}(\hat{h}_k)$), the second moment simplifies to

$$E[\langle I_k \rangle^2] = \frac{1}{\hat{T}_k/b} \underbrace{\int_{\text{supp}(\hat{h}_k)} \left[\hat{h}_k(\bar{\mathbf{u}})\hat{f}(\bar{\mathbf{u}})/\hat{p}(\bar{\mathbf{u}})\right]^2 \mathrm{d}\bar{\mathbf{u}}}_{E[\langle I_k \rangle^2_{\text{uni}}]}. \tag{14}$$

In other words, $E[\langle I_k \rangle^2]$ is equal to a rescaled version of $E[\langle I_k \rangle^2_{\text{uni}}]$, the second moment of the MC estimator that uses uniform sampling on the primary-sample space. The relative error now becomes

$$\text{nsd}(\langle I_k \rangle) = \sqrt{\frac{1}{\hat{T}_k/b} \frac{E[\langle I_k \rangle^2_{\text{uni}}]}{I_k^2} - 1}. \tag{15}$$

Setting the TF according to Equation (3) ensures that the relative error is indeed equal for all measurements.

ACKNOWLEDGMENTS

The Mirror Balls scene in Fig. 10 is courtesy of Toshiya Hachisuka. The Dinner hall scene (Fig. 7 and 9) is courtesy of Gregzaal. The Tea time scene (Fig. 11) is courtesy of Xev. The work was supported by Charles University in Prague, projects GA UK 164815 and 340915, by the grant SVV-2016-260332, and by the Czech Science Foundation grant 16-18964S. Thanks to Oskar Elek for proofreading the paper.

REFERENCES

T. Bashford-Rogers, K. Debattista, and A. Chalmers. 2014. Importance driven environment map sampling. *IEEE Transactions on Visualization and Computer Graphics* 20, 6, 907–918. DOI:http://dx.doi.org/10.1109/TVCG.2013.258

Jiating Chen, Bin Wang, and Jun-Hai Yong. 2011. Improved stochastic progressive photon mapping with Metropolis sampling. *Computer Graphics Forum* 30, 4, 1205–1213. DOI:http://dx.doi.org/10.1111/j.1467-8659.2011.01979.x

P. H. Christensen. 2003. Adjoints and importance in rendering: An overview. *IEEE Transactions on Visualization and Computer Graphics* 9, 3, 329–340. DOI:http://dx.doi.org/10.1109/TVCG.2003.1207441

David Cline, Justin Talbot, and Parris Egbert. 2005. Energy redistribution path tracing. *ACM Transactions on Graphics* 24, 3, 1186–1195. DOI:http://dx.doi.org/10.1145/1073204.1073330

Carsten Dachsbacher, Jaroslav Křivánek, Miloš Hašan, Adam Arbree, Bruce Walter, and Jan Novák. 2014. Scalable realistic rendering with many-light methods. *Computer Graphics Forum* 33, 1, 88–104. DOI:http://dx.doi.org/10.1111/cgf.12256

Philip Dutré and Yves Willems. 1995. Potential-driven Monte Carlo particle tracing for diffuse environments with adaptive probability density functions. In *Proceedings of the Eurographics Workshop on Rendering*.

Shaohua Fan, Stephen Chenney, and Yu-Chi Lai. 2005. Metropolis photon sampling with optional user guidance. In *Proceedings of the Eurographics Symposium on Rendering*.

Iliyan Georgiev, Jaroslav Křivánek, Tomáš Davidovič, and Philipp Slusallek. 2012. Light transport simulation with vertex connection and merging. *ACM Transactions on Graphics* 31, 6, Article No. 192. DOI:http://dx.doi.org/10.1145/2366145.2366211

Adrien Gruson, Mickael Ribardière, Remy Cozot, and Kadi Bouatouch. 2014. Rendu progressif base Metropolis-Hasting dans des scenes a contextes topologiques multiples. *Revue Electronique Francophone d'Informatique Graphique* 8, 1.

Toshiya Hachisuka and Henrik Wann Jensen. 2009. Stochastic progressive photon mapping. *ACM Transactions on Graphics* 28, 5, Article No. 141. DOI:http://dx.doi.org/10.1145/1618452.1618487

Toshiya Hachisuka and Henrik Wann Jensen. 2011. Robust adaptive photon tracing using photon path visibility. *ACM Transactions on Graphics* 30, 5, Article No. 114. DOI:http://dx.doi.org/10.1145/2019627.2019633

Toshiya Hachisuka, Anton S. Kaplanyan, and Carsten Dachsbacher. 2014. Multiplexed Metropolis light transport. *ACM Transactions on Graphics* 33, 4, 100:1–100:10.

Toshiya Hachisuka, Jacopo Pantaleoni, and Henrik Wann Jensen. 2012. A path space extension for robust light transport simulation. *ACM Transactions on Graphics* 31, 6, Article No. 191. DOI:http://dx.doi.org/10.1145/2366145.2366210

Johannes Hanika, Anton Kaplanyan, and Carsten Dachsbacher. 2015. Improved half vector space light transport. *Computer Graphics Forum* 34, 4, 65–74.

W. K. Hastings. 1970. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika* 57, 1, 97–109.

Jared Hoberock and John C. Hart. 2010. Arbitrary importance functions for Metropolis light transport. *Computer Graphics Forum* 29, 6, 1993–2003. DOI:http://dx.doi.org/10.1111/j.1467-8659.2010.01713.x

Wenzel Jakob. 2010. Mitsuba Renderer. Retrieved September 23, 2016, from http://www.mitsuba-renderer.org.

Wenzel Jakob and Steve Marschner. 2012. Manifold exploration: A Markov chain Monte Carlo technique for rendering scenes with difficult specular transport. *ACM Transactions on Graphics* 31, 4, Article No. 58. DOI:http://dx.doi.org/10.1145/2185520.2185554

Wenzel Jakob, Christian Regg, and Wojciech Jarosz. 2011. Progressive expectation-maximization for hierarchical volumetric photon mapping. *Computer Graphics Forum* 30, 4, 1287–1297. DOI:http://dx.doi.org/10.1111/j.1467-8659.2011.01988.x

Henrik Wann Jensen. 1995. Importance driven path tracing using the photon map. In *Rendering Techniques'95*. Eurographics. Springer, 326–335.

Anton S. Kaplanyan and Carsten Dachsbacher. 2013a. Adaptive progressive photon mapping. *ACM Transactions on Graphics* 32, 2, Article No. 16. DOI:http://dx.doi.org/10.1145/2451236.2451242

Anton S. Kaplanyan and Carsten Dachsbacher. 2013b. Path space regularization for holistic and robust light transport. *Computer Graphics Forum* 32, 2, 63–72.

Csaba Kelemen, László Szirmay-Kalos, Gyorgy Antal, and Ferenc Csonka. 2002. A simple and robust mutation strategy for the Metropolis light transport. *Computer Graphics Forum* 21, 3531–540.

Shinya Kitaoka, Yoshifumi Kitamura, and Fumio Kishino. 2009. Replica exchange light transport. *Computer Graphics Forum* 28, 8, 2330–2342. DOI:http://dx.doi.org/10.1111/j.1467-8659.2009.01540.x

Eric P. Lafortune and Yves D. Willems. 1993. Bi-directional path tracing. In *Proceedings of the Compugraphics Conference (Compugraphics'93)*. 145–153.

Jaakko Lehtinen, Tero Karras, Samuli Laine, Miika Aittala, Frédo Durand, and Timo Aila. 2013. Gradient-domain Metropolis light transport. *ACM Transactions on Graphics* 32, 4, 95:1–95:12.

Radford M. Neal. 1996. Sampling from multimodal distributions using tempered transitions. *Statistics and Computing* 6, 4, 353–366. DOI:http://dx.doi.org/10.1007/BF00143556

Ingmar Peter and Georg Pietrek. 1998. Importance driven construction of photon maps. In *Rendering Techniques'98*. Eurographics. Springer, 269–80.

Jeffrey S. Rosenthal. 2011. Optimal proposal distributions and adaptive MCMC. In *Handbook of Markov Chain Monte Carlo*, S. Brooks, A. Gelman, G. L. Jones, and X.-Li Meng (Eds.). Chapman & Hall/CRC.

Fabrice Rousselle, Claude Knaus, and Matthias Zwicker. 2012. Adaptive rendering with non-local means filtering. *ACM Transactions on Graphics* 31, 6, Article No. 195. DOI:http://dx.doi.org/10.1145/2366145.2366214

Benjamin Segovia, Jean Claude Iehl, and Bernard Péroche. 2007. Metropolis instant radiosity. *Computer Graphics Forum* 26, 3, 425–434.

Eric Veach. 1997. *Robust Monte Carlo Methods for Light Transport Simulation*. Ph.D. Dissertation. Stanford University, Stanford, CA.

Eric Veach and Leonidas J. Guibas. 1997. Metropolis light transport. In *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH'97)*. 65–76.

Edgar Velázquez-Armendáriz, Zhao Dong, Bruce Walter, and Donald P. Greenberg. 2015. Complex luminaires: Illumination and appearance rendering. *ACM Transactions on Graphics* 34, 3, Article No. 26. DOI:http://dx.doi.org/10.1145/2714571

Jiří Vorba, Ondřej Karlík, Martin Šik, Tobias Ritschel, and Jaroslav Křivánek. 2014. On-line learning of parametric mixture models for light transport simulation. *ACM Transactions on Graphics* 33, 4, Article No. 101.

Quan Zheng and Chang-Wen Zheng. 2015. Visual importance-based adaptive photon tracing. *Visual Computer* 31, 6–8, 1001–1010. DOI:http://dx.doi.org/10.1007/s00371-015-1104-0

Matthias Zwicker, Wojciech Jarosz, Jaakko Lehtinen, Bochang Moon, Ravi Ramamoorthi, Fabrice Rousselle, Pradeep Sen, Cyril Soler, and Sung-Eui Yoon. 2015. Recent advances in adaptive sampling and reconstruction for Monte Carlo rendering. *Computer Graphics Forum* 34, 2, 667–681. DOI:http://dx.doi.org/10.1111/cgf.12592

Received October 2015; revised March 2016; accepted June 2016