Path Differential-Informed Stratified MCMC and Adaptive Forward Path Sampling

TOBIAS ZIRR and CARSTEN DACHSBACHER, Karlsruhe Institute of Technology, Germany

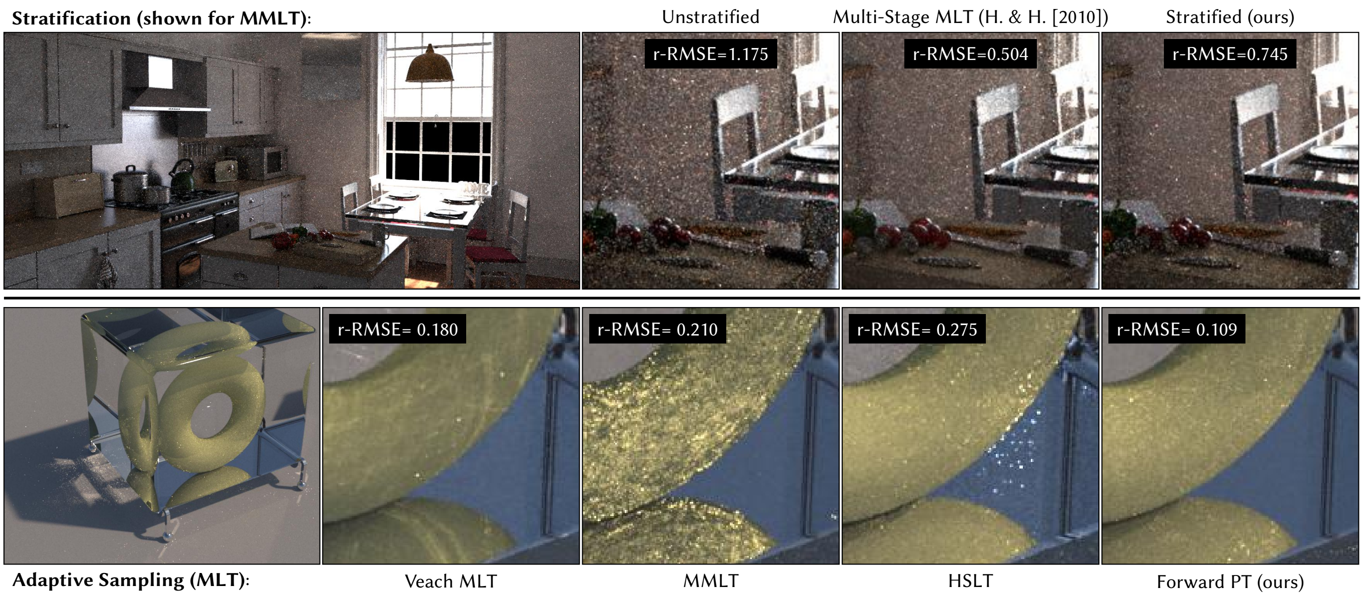


Fig. 1. Equal-time renderings (50s on 64 cores). *Top:* We propose a new analytical approach to Markov Chain stratification that improves sample distribution similarly to multi-stage MLT [Hoberock and Hart 2010], but at greatly reduced complexity, as it neither requires multiple rendering stages nor other pre-accumulated information. *Bottom:* We construct an adaptive rendering algorithm based on *purely forward path tracing* that is competitive with state-of-the-art bi-directional algorithms. Our analytic variance bounding scheme serves as a framework for the design and optimization of adaptive sampling distributions.

Markov Chain Monte Carlo (MCMC) rendering is extensively studied, yet it remains largely unused in practice. We propose solutions to several practicability issues, opening up path space MCMC to become an adaptive sampling framework around established Monte Carlo (MC) techniques. We address non-uniform image quality by deriving an analytic target function for image-space sample stratification. The function is based on a novel connection between variance and path differentials, allowing analytic variance estimates for MC samples, with potential uses in other adaptive algorithms outside MCMC. We simplify these estimates down to simple expressions using only quantities known in any MC renderer. We also address the issue that most existing MCMC renderers rely on bi-directional path tracing and reciprocal transport, which can be too costly and/or too complex in practice. Instead, we apply our theoretical framework to optimize an adaptive MCMC algorithm that only uses forward path construction. Notably, we construct our algorithm by adapting (with minimal changes) a full-featured path tracer into a single-path state space Markov Chain, bridging another gap between MCMC and existing MC techniques.

CCS Concepts: • **Computing methodologies → Ray tracing**.

Additional Key Words and Phrases: Variance Estimation, Controlled Correlation, Optimized Adaptive Sampling, Reversible Jump MC

ACM Reference Format:
Tobias Zirr and Carsten Dachsbacher. 2020. Path Differential-Informed Stratified MCMC and Adaptive Forward Path Sampling. *ACM Trans. Graph.* 39, 6, Article 246 (December 2020), 19 pages. https://doi.org/10.1145/3414685.3417856

1 INTRODUCTION

Markov Chain Monte Carlo (MCMC) light transport (MLT [Veach and Guibas 1997]) has been extensively studied as an adaptive variation of Monte Carlo (MC) rendering for over two decades. Although research of better mutation strategies and importance functions is ongoing, to this day, MCMC is largely ignored by the industry: The quality of results is oftentimes hard to control and non-uniform. Moreover, noise reduction and reconstruction techniques have become a successful and essential tool for sample-budgeted high-quality MC rendering in practice. However, the correlated nature of MCMC creates artifacts that prevent successful application of these techniques. Lastly, the heuristics of established mutation strategies often prove fragile in complex scenes.

Yet, recent advances in adaptive sampling and guiding show that there is unused potential in correlated sampling techniques. A major issue of these, however, is the reliance on information that can only

Authors' address: Tobias Zirr; Carsten Dachsbacher, Karlsruhe Institute of Technology, IVD, Am Fasanengarten 5, 76131, Karlsruhe, Germany.

be accumulated during the rendering process itself. Naïve MCMC does not require such information, but fails to differentiate between bright and difficult samples, which in turn can be fixed by stratification requiring similar information [Hoberock and Hart 2010]. To improve on this, we use path differentials, tracking the impact of stochastic scattering interactions on the influence of emitter surfaces, to predict information that previously had to be obtained from sample statistics in advance. We derive a novel analytic function that conservatively estimates variance caused by a given path sample.

As a proof of concept, we apply the gained insights to both stratification and optimal adaptive sampling in the context of MCMC:

- We stratify the target function of the Markov Chain (akin to Hoberock et al. [2010]) using our analytic variance bounds.
- We optimize the shape of proposal distributions in path space MCMC and by this push the capabilities of forward path tracing, to compete with other bi-directional methods.
- We complement the resulting small step based on forward path construction by embedding a full path tracer with minimal changes, importing established MC variance reduction and sample reuse, such as multiple importance sampling (MIS), next event estimation (NEE), and shared path prefixes.

Our MCMC renderer effectively becomes an adaptive path tracer, with the usual benefits of reduced algorithmic complexity, avoiding efforts to ensure reciprocity (e.g. for shading normals), and focussed sampling efforts due to tracing from the camera.

2 BACKGROUND

The goal of MC and MCMC rendering methods is to compute the light transported on all possible paths $\mathrm{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_k)$ (for all vertex counts $k \geq 2$) in a path space Ω to the individual pixels of a sensor. For a given path X the *measurement contribution* $f(\mathrm{X})$ [Veach 1998] (w.r.t. the area measure $\mathrm{d}A$ on all scene surfaces $\mathcal{M}$) is evaluated and the measurement for the j-th pixel with sensitivity $h_j(\mathrm{X})$ is computed by a path integral in the rendering equation:

$$I_j = \int_\Omega h_j(\mathrm{X}) f(\mathrm{X})\,\mathrm{dX}, \text{ with } \mathrm{dX} = \prod_{i=1}^{k} \mathrm{d}A(\mathbf{x}_i).$$

Integration by Stochastic Sampling. MC approaches to solving this rendering equation, i.e. by stochastic path sampling, leave a lot of flexibility. While in principle, every unbiased MC estimator converges to the correct solution, good importance sampling (IS) is crucial in order to sample high-throughput paths frequently and to reduce variance of the result. Multiple Importance Sampling (MIS) [Veach and Guibas 1995] robustly combines several estimators that efficiently reduce variance in different parts of path space, and has proven an essential technique for high-quality results.

MCMC rendering algorithms, introduced as Metropolis Light Transport (MLT) [Veach and Guibas 1997], make use of the Metropolis-Hastings algorithm resp. the generalizing Reversible Jump (RJ) framework [Geyer 2003; Green 1995, 2003] to enable correlated path sampling while maintaining a controlled distribution that allows MC integration. MCMC supports arbitrary mixed proposal distributions, but combination using MIS also proves to reduce variance, e.g. in bi-directional path mutations [Hachisuka et al. 2014] and incorporation of density estimation [Šik et al. 2016]. Primary Sample Space MLT (PSSMLT) [Kelemen et al. 2002] allows adding correlated sampling to existing MC estimators by moving from the state space Ω of paths X to the space $\mathcal{U}$ of random variables U used by the estimator.

Generalizations and Mixed Proposals in MCMC. Efficient correlated sampling across a mixture of sampling techniques has recently been derived from the observation that the choice of the state space does not matter [Bitterli et al. 2017; Otsu et al. 2017; Pantaleoni 2017]: In the RJ framework, proposal distributions arise from self-inverse maps that fuse arbitrary state variables with random variables, accounting for measure changes with their Jacobian determinants. We observe that despite many MCMC algorithms being fundamentally based on this framework, current techniques (in single-path state spaces) still artificially correlate the path lengths of proposal paths. We expand on the use of RJ by amortizing the cost of path construction across many path lengths in our large step mutation.

Multiple Proposals in MCMC. Other generalizations of MCMC include the Multiple Try Framework [Liu et al. 2000] (MT), allowing the generation of many proposal candidates at once in a single-path state space, by a weighted stochastic candidate selection [Pandolfi et al. 2010]. Bitterli [2015] applies this to the bi-directional mutation to reuse sampling effort of emitter and sensor subpaths across multiple candidates, with different connecting segments. Martino and Louzada [2017] note that good weights are crucial for robust candidate mixture, and note that MIS [Veach and Guibas 1995] is a good option. Craiu and Lemieux [2007] apply MT to bring Quasi-Monte Carlo methods into MCMC. MT in rendering has also been used for improved distribution of virtual point lights [Segovia et al. 2007b] and coherent proposal sampling using packet ray tracing [Segovia et al. 2007a]. We observe that MT can be formulated as a special case of RJ, enabling sampling many paths of differing length using multiple techniques at the same time. We show compatibility with MIS, and embed a full efficient path tracer in path space MCMC.

Stratification in MCMC. Stratification, the targeted distribution of samples across pixels and into regions (strata) with potential relevance to the image, is an important feature of MC estimators to accelerate the convergence of MC estimates. In MCMC, however, (pseudo-)deterministic sample placement is harder to achieve. While MT-based approaches have successfully applied it to individual mutations [Craiu and Lemieux 2007], in adaptive sampling the longer-term goal is *expected* stratification, i.e. to ensure an *expected* number of samples per stratum. For this, multi-stage MLT [Hoberock and Hart 2010] alters the *importance function* (*target function*) of the Metropolis-Hastings algorithm, which may be chosen almost arbitrarily without changing the MCMC result. The algorithm renders recursively refined images, using samples accumulated in previous images as guides. The stationary distribution of the Markov Chain follows the target function, which can be formed to enforce a uniform sample distribution in the limit. In the context of density estimation using photons, visibility-based [Hachisuka and Jensen 2011] and density-based [Gruson et al. 2016] target functions were proposed. Our approach also alters the target function to control sample distribution, but we remove the necessity of first aggregating information to guide the Markov Chain.

Stratification of Markov Chain Initiation. Besides controlling the Markov Chain, stratification can also be enforced more directly by breaking out of the pure MCMC framework: ERPT [Cline et al. 2005] only starts Markov Chains for high-energy samples in a stratifiable independent MC estimator. However, chains need to be short to not face the same stratification issues as standard MCMC, but they should be long to actually profit from the MCMC adaptive sampling. Moreover, adaptivity is currently limited in the ERPT framework as it requires equal chain lengths. Applying the ideas of previous work and this paper for expected stratification to ERPT might become interesting if this limitation is lifted. Recent work [Gruson et al. 2020] completely changes the flow of correlated sampling by individual Markov Chains per pixel, relying on replica exchange and a post-process iterative solver for information sharing between pixels. While this is interesting to enforce pixel stratification, it also currently impacts adaptivity to difficult light transport. The algorithm inherits a lot of complexity from gradient-domain (M)LT [Lehtinen et al. 2013], and further study is needed as to the practical benefits. We explore a different direction, towards simpler algorithms without multiple phases, to also allow progressive rendering.

Short-term Stratification and Step Sizes. Besides the long-term stationary distribution, the distribution of samples and thus the image quality in the short term (for low sample counts) heavily depends on the steps proposed by MCMC mutations. To not get stuck in subspaces, Veach and Guibas [1997] propose a mixture of *large step mutations* and *small step perturbations*. Since then, many heuristics have been studied [Šik and Křivánek 2018]. Zsolnai and Szirmay-Kalos [2013] propose scene-wide automatic parameter tuning, which by its nature has limited local adaptivity. Dedicated run-time effort can be put into adaptivity as in the geometric analysis of local visibility [Otsu et al. 2018]. In our small step mutation strategy, we optimize proposal distributions for minimal correlation (maximal exploration) while targeting a certain variance. The result is closely related to and provides a new perspective on the efficacy of previous decisions made in Manifold Exploration [Jakob and Marschner 2012] and Halfvector-Space Light Transport [Hanika et al. 2015; Kaplanyan et al. 2014], where image-projected derivatives can be used to control step sizes and correlation. In the context of PSSMLT, Szirmay-Kalos and Szécsi [2017] propose an adaptive isotropic scaling of steps according to the throughput of the current path. In our comparisons, we show cases where this is sufficiently adaptive and other cases where anisotropic distributions are crucial. Adaption to anisotropic distributions has been studied in the context of Hamiltonian MC [Li et al. 2015], deriving a multi-variate anisotropic Gaussian from quadratic surface approximations constructed using automatic differentiation, and optimized with Langevin MC [Luan et al. 2020], avoiding second-order differentiation. In our algorithm, we also use anisotropic Gaussian proposal steps, but allow a simpler implementation without requiring differentiability in the renderer.

Path Differentials in Light Transport. Path-space derivatives and path differentials (also known as ray differentials) are commonly used for level-of-detail, spatial filtering [Belcour et al. 2013, 2017; Christensen et al. 2018; Fascione et al. 2018; Suykens and Willems 2001], and real-time adaptive temporal filtering [Schied et al. 2018]. Similar derivatives can drive adaptive discrete partitionings of light transport, described as Pencil Tracing [Shinya et al. 1987], analytically computing indirect illuminance inside *pencils*, i.e. path space partitions, as radiance interpolants, constructed by adaptive line-trees [Bala et al. 1999], in adaptive radiosity [Suykens and Willems 2001], and to optimize irradiance cache placement [Schwarzhaupt et al. 2012]. Spectral differentials can be used for efficient approximation of dispersion [Elek et al. 2014]. The derivatives of direct visibility and higher-dimensional discontinuous transport have been used for inverse rendering [Li et al. 2018] and smooth reconstruction of gradient-domain Monte Carlo estimates [Lehtinen et al. 2013].

Path Guiding. Online-learning path guiding approaches [Müller et al. 2017; Simon et al. 2018; Vorba et al. 2014] identify complex subsets of path space from refined statistical sample representations over time. We derive an analytic function for variance bounding that computes similar information as density-based outlier classification (DBOR) [DeCoro et al. 2010; Zirr et al. 2018], which can be used to identify difficult paths [Bitterli and Jarosz 2019; Simon et al. 2018].

3 OVERVIEW OF OUR METHOD

In Sect. 4, we show ways to stratify MCMC samples by altered splat values, and what is required to control their variance. Sect. 5 derives a bound on this variance based on the *local* distribution of MC path samples. This local distribution is characterized by the differentials of an MC path construction strategy, i.e. the mapping from uniform random numbers to the vertex positions of light transport paths. We identify the spread of the emitter vertex (its marginal distribution) as a decisive source of uncontrolled variance for unidirectional path construction. This leads to a convenient characterization of variance via *path differentials* (also commonly known as (indirect) *ray differentials*), i.e. the spread of a path vertex due to the convolution of all random variables (previous interactions). Sect. 6 formalizes this connection with the necessary computations for variance bounding, and shows how to incorporate pre-existing variance reduction strategies like *next event estimation* (NEE), by virtually treating the penultimate vertex like a light source (leaving the variance-controlled path suffix untouched). Finally, Sect. 7 extends our variance characterization to adaptive sampling, optimizing the distribution of proposal paths according to local variance bounding criteria.

4 STRATIFICATION OF MCMC

In the following, we motivate the stratification improvements we strive for, and we give a high-level view on how to achieve them. We first take up the basic idea to get from unstratified to stratified sampling in MCMC light transport by altering the Metropolis-Hastings *importance function* (proportional to the targeted stationary distribution) with compensating sample weights [Hoberock and Hart 2010; Veach 1998]. From there, Section 4.2 makes a new connection to MC variance reduction techniques, which allows the construction of a new class of analytic importance functions. Note that Table 1 provides an overview of the notation used throughout the paper.

The Problem of MCMC Stratification. We are concerned with improved sampling in the image plane: Standard unstratified MCMC methods sample paths proportional to their contribution relative to the full image. In the worst case, this focuses all sampling on only the brightest pixels, leaving darker pixels severely undersampled:

Proposition 1. *A standard MLT importance function* $W(\mathrm{X}){=}f(\mathrm{X})$ *samples paths* X *contributing a fraction* $h_j(\mathrm{X})$ *to pixel* j *with a probability mass that directly corresponds to the expected pixel value* I_j*:*

$$\int_\Omega h_j(\mathrm{X}) \frac{W(\mathrm{X})}{\int_\Omega W(\mathrm{X})\,\mathrm{dX}}\,\mathrm{dX} \propto \int_\Omega h_j(\mathrm{X}) f(\mathrm{X})\,\mathrm{dX} = I_j. \tag{1}$$

Such sampling can result in extremely non-uniform sampling and thus perceived error across the image plane, impeding any benefits of MLT in practice. With our stratification, we still want adaptive sampling around paths whose contribution is actually difficult to estimate, but *elsewhere* aim at an approximately uniform distribution of samples in the image plane, independent of pixel brightnesses.

Proposition 2. *A pixel-stratifying importance function* $W'(\mathrm{X})$ *needs to ensure that there is a lower bound* $\tau > 0$ *for the expected mass of samples* X *that contribute fractions* $h_j(\mathrm{X})$ *for* each *pixel* j*:*

$$\int_\Omega h_j(\mathrm{X}) \frac{W'(\mathrm{X})}{\int_\Omega W'(\mathrm{X})\,\mathrm{dX}}\,\mathrm{dX} \geq \tau \quad \forall j. \tag{2}$$

Larger τ *improve stratification, up to equal sampling mass for every pixel* j*, but may reduce adaptive sampling (see Sect. 4.2).*

Note that this is in contrast to other uses of the term stratification for (pseudo-)deterministic sample placement in (quasi-)MC contexts, as we aim at a mere *expected* stratification. In the next section, we recap an approach proposed by previous works, before introducing our related general approach in Sect. 4.2.

4.1 Approximate Stratification by Integration

Stratifying importance functions $W'(\mathrm{X})$ can be constructed by using a-priori knowledge about the pixel values I_j, e.g. obtained in a pre-integration step [Hoberock and Hart 2010; Veach 1998]: If we set $W'(\mathrm{X}) = f(\mathrm{X}) \left(\sum_j h_j(\mathrm{X})/I_j\right)$, we find the expected sample mass in the supports $\mathrm{supp}(\cdot)$ of $h_j(\mathrm{X})$ to be:

$$\int_{\mathrm{supp}(h_j)} \frac{W'(\mathrm{X})}{\int_\Omega W'(\mathrm{X})\,\mathrm{dX}}\,\mathrm{dX} \propto \int_{\mathrm{supp}(h_j)} f(\mathrm{X}) \sum_k \frac{h_k(\mathrm{X})}{I_k}\,\mathrm{dX} \tag{3}$$

$$\geq \frac{\int_\Omega h_j(\mathrm{X}) f(\mathrm{X})\,\mathrm{dX}}{I_j} = 1 \;\; \forall j. \tag{4}$$

Therefore, stratification is ensured in the neighborhood around each pixel j, under the (unrealistic) assumption that we know all pixels I_j in advance. In practice, both the residual variance in the estimates for I_j used by $W'(\mathrm{X})$, and re-sampling artifacts due to stratified neighborhoods being too coarse, are clearly visible in many cases for such constructions of $W'(\mathrm{X})$ (see also our comparison to previous work in the results section).

4.2 Stratification by MC Strategies

We note that we can achieve perfect stratification in a much simpler way: A suitable importance function is any PDF $p(\mathrm{X})$ that distributes paths X such that the probability mass is the same for all pixels j:

$$\int_\Omega h_j(\mathrm{X}) p(\mathrm{X})\,\mathrm{dX} = \mathrm{const.}\ \forall j. \tag{5}$$

The stratification condition in Eq. (2) is therefore fulfilled. We may consider $p(\mathrm{X})$ the PDF of a MC estimator that generates sample values $S(\mathrm{X}) = f(\mathrm{X})/p(\mathrm{X})$. Effectively, we alter the importance function to $W''(\mathrm{X}) = f(\mathrm{X})/S(\mathrm{X})$ and compensate by splatting the MC estimates $S(\mathrm{X})$. However, we find that the variance of $S(\mathrm{X})$ also causes variance in the final MCMC result: For the stationary distribution $\pi''(\mathrm{X}) = W''(\mathrm{X}) = p(\mathrm{X})$, as for any other stationary distribution $\pi(\mathrm{X})$, the variance of the M-sample MCMC result is at least the variance of $S(\mathrm{X})$ (the series of adjacent autocovariance sum pairs is positive and decreasing [Geyer 1992, Theorem 3.1]):

$$\mathbf{V}\left[\frac{1}{M}\sum_{i=1}^{M}\frac{f_j(\mathrm{X}_i)}{\pi(\mathrm{X}_i)}\right] = \frac{1}{M^2}\sum_{i=1}^{M}\sum_{k=1}^{M}\mathrm{Cov}\left(S(\mathrm{X}_i), S(\mathrm{X}_k)\right) \tag{6}$$

$$\geq \frac{1}{M^2}\sum_{i=1}^{M}\mathrm{Cov}\left(S(\mathrm{X}_i), S(\mathrm{X}_i)\right) = \frac{\mathbf{V}\left[S(\mathrm{X})\right]}{M}. \tag{7}$$

Nevertheless, the acceptance rate of many large steps can be increased significantly, where the proposal distribution resembles $p(\mathrm{X})$. This can make the covariances vanish much faster.

4.3 Stratification by Variance-Bounded MC Estimates

Luckily, we have direct control over the splat values and resulting variances, without introducing bias, by altering the importance function accordingly. For that, we need a factor that downweights splats $S(\mathrm{X})$ inversely to variance, reliably detecting problematic paths X and quantifying their need for adaptive sampling: A lowered splat weight $\tilde{S}(\mathrm{X})$ increases the respective altered importance function $\tilde{W}(\mathrm{X})$, guiding the Markov Chain to allocate more time to exploring the corresponding space. Section 5 derives an analytic variance bounding scheme to obtain such a downweighting $\tilde{r}(\mathrm{X})$, such that the variance of the resulting splats $\tilde{S}(\mathrm{X})$ is controlled:

$$\boxed{\tilde{S}(\mathrm{X}) := \max\left\{\frac{S(\mathrm{X})}{\tilde{r}(\mathrm{X})},\ B\right\},\ \tilde{r}(\mathrm{X}) \geq 1. \quad \tilde{W}(X) := \frac{f(\mathrm{X})}{\tilde{S}(\mathrm{X})}}, \tag{8}$$

where B is the total image brightness. The lower bound B seems counterproductive at first, as in the worst case, the resulting target distribution[1] $\tilde{\pi}(\mathrm{X}) \propto \tilde{W}(\mathrm{X})$ is as low as the unstratified $\pi(\mathrm{X}) \propto f(\mathrm{X})$ for paths where $\tilde{S}(\mathrm{X}) = B$. However, this is intentional: For one, we never reduce sampling of such paths, while the typical splatting of expected acceptance values still directly splats low contributions (if the proposal distribution is sufficiently stratified). More importantly, the corresponding sampling probability mass remains free to be redistributed to the stratified sampling of all other paths: Note that our primary concern is to not get stuck in these paths with high contribution, using our altered target function to increase the acceptance rate via direct bright splats. For these, we verify the stratification condition in Eq. (2) and get $\tilde{\pi}(\mathrm{X}) \geq p(\mathrm{X})\tilde{r}(\mathrm{X})$. In experiments, we confirmed that the lower bound is crucial for good adaptive exploration of difficult paths.

---

[1] $\tilde{W}(\mathrm{X}) = \min\left\{p(\mathrm{X})\tilde{r}(\mathrm{X}), \frac{f(\mathrm{X})}{B}\right\}, \frac{\tilde{W}(\mathrm{X})}{\int \tilde{W}(\mathrm{X})\,\mathrm{dX}} \geq \min\left\{\frac{\cancel{B}}{\cancel{B}}\frac{f(\mathrm{X})}{\int f(\mathrm{X})\,\mathrm{dX}}, \frac{p(\mathrm{X})\tilde{r}(\mathrm{X})\cancel{B}}{\cancel{\int f(\mathrm{X})\,\mathrm{dX}}}\right\}.$

Table 1. Important notation used throughout the paper.

Notation	Description
$\Omega, \mathcal{U}$	Space of light paths X & of unit random vectors U
$\mathrm{X}, f(\mathrm{X}), f_j(\mathrm{X})$	Light path, its measurement contrib., $f(\mathrm{X})h_j(\mathrm{X})$
$\mathrm{U}, X(\mathrm{U})$	Random vector U, path X(U) constructed from it
$f(\mathrm{U})$	"PSS" contribution, $f(\mathrm{U})\,\mathrm{dU} \circ \mathrm{X}^{-1} = f(\mathrm{X})\,\mathrm{dX}$
$W(\mathrm{X}), W', \tilde{W}$	MCMC target functions, unaltered and altered
$S(\mathrm{X})$	Splat $\frac{f(\mathrm{X})}{p(\mathrm{X})}$ of an MC estimate with PDF $p(\mathrm{X})$
$\tilde{r}(\mathrm{X}) \geq 1$	Downweighting ratio quantifying variance of $S(\mathrm{X})$
$\tilde{S}(\mathrm{X}), \tilde{W}(\mathrm{X})$	Downweighted splat and altered target funct. $\frac{f(\mathrm{X})}{\tilde{S}(\mathrm{X})}$
$h_j(\mathrm{X}), g_i(\mathrm{X})$	Pixel filter for pixel j, and subfilter $g_i(\mathrm{X}) \leq h_j(\mathrm{X})$
$\lVert f \rVert_\alpha^{\propto m}$	α-norm of $f(\mathrm{U})$ w.r.t. $\hat{m}(\mathrm{U})\,\mathrm{dU}$, $\left(\frac{\int_\mathcal{U} f(\mathrm{U})^\alpha m(\mathrm{U})\,\mathrm{dU}}{\int_\mathcal{U} m(\mathrm{U})\,\mathrm{dU}}\right)^{\frac{1}{\alpha}}$
$g^d(\mathrm{X})$	Selects a deterministic transport subset in $g(\mathrm{X})$
$F(\mathbf{x}_k^\perp, \mathbf{o}_k)$	Marginal contribution of endpoint $(\mathbf{x}_k^\perp, \mathbf{o}_k)$ to $g(\mathrm{X})$
$[d], [\vert d]$	Equation holds with and without d and $\vert d$, resp.
$\int a\,\mathrm{d}\mu \stackrel{!}{=} \int \mathrm{d}\nu$	Defines the Radon-Nikodym derivative $a = \mathrm{d}\nu/\mathrm{d}\mu$

5 ANALYTIC VARIANCE BOUNDING SCHEME

For the described approach to stratification, we need a factor that quantifies the uncontrolled variance of MC weights (in Eq. 8). Our motivating example for causes of uncontrolled variance is a simple path tracer: There, a light source reflected in a mirror generates the same sample values as a light source hit via diffuse scattering. The difference causing variance is in the distribution of sample values: Via the mirror, paths hit the emitter deterministically; via diffuse scattering, only when rays randomly point towards the emitter. Thus, each case results in a different expected value, i.e. pixel value. For the deterministic *configuration* (via the mirror), path tracing works well, i.e. variance is controlled. For the non-deterministic configuration (diffuse scattering), variance is less acceptable (we would commonly add variance reduction, e.g. by NEE). Motivated by these observations, our variance bounding will depend on the impact of random variables, corresponding to scattering interactions along a path, on the caused variation of the endpoint at the emitter.

To control variance, we analyze relative pixel error (following previous adaptive sampling [Rousselle et al. 2012]), since our stratification aims to decouple sample density from pixel brightness. We split the relative variance into sample value range (L^p-norms constructed from higher-order moments) and distribution (we use the first moment, i.e. the pixel brightness I_j):

Proposition 3. *Let $f_j(\mathrm{U}) = f(\mathrm{U})\,h_j(\mathrm{U})$ be the sample values estimating a pixel value I_j by constructing paths $\mathrm{X}(\mathrm{U})$ from random vectors U. Let $J_j^{(\alpha)} = \int_\mathcal{U} f(\mathrm{U})^\alpha h_j(\mathrm{U})\,\mathrm{dU}$ be the α^{th} moment of these MC estimates for $\alpha \geq 2$. We can bound the relative variance (and the relative α^{th} moment $\frac{J_j^{(\alpha)}}{I_j{}^\alpha}$) by the $(\alpha-1)$-norm $\lVert f \rVert_{\alpha-1}^{\propto f_j}$ of sample values $f(\mathrm{U})$ w.r.t. $\hat{f}_j(\mathrm{U})\,\mathrm{dU}$ (where $\hat{f}_j(\mathrm{U})$ is normalized w.r.t. dU), divided by their first moment $\lVert f_j \rVert_1$:*

$$\frac{\mathbf{V}[f_j(\mathrm{U})]}{I_j{}^2} \leq \frac{J_j^{(2)}}{I_j{}^2} \leq \left(\frac{J_j^{(\alpha)}}{I_j{}^\alpha}\right)^{\frac{1}{\alpha-1}} = \frac{\lVert f \rVert_{\alpha-1}^{\propto f_j}}{\lVert f_j \rVert_1}. \tag{9}$$

Proof.

$$\frac{J_j^{(\alpha)}}{I_j{}^\alpha} = \frac{1}{I_j{}^{\alpha-1}} \frac{\int_\mathcal{U} f(\mathrm{U})^\alpha\, h_j(\mathrm{U})\,\mathrm{dU}}{\int_\mathcal{U} f(\mathrm{U})\, h_j(\mathrm{U})\,\mathrm{dU}} = \left(\frac{\lVert f \rVert_{\alpha-1}^{\propto f_j}}{\lVert f_j \rVert_1}\right)^{\alpha-1}. \tag{10}$$

$\lVert f \rVert_\alpha^{\propto f_j}$ grows with α [Axler 2020], the 2^{nd} moment bounds variance. □

Therefore, the relative variance is directly bounded by the ratio of the largest sample values to the integral we *want* to compute. We will now relate these quantities to what is *convenient* to compute.

5.1 Variance Bounding by Comparison of Configurations

Following our motivating example, a configuration where MC estimates become more reliable is light transport via deterministic scattering interactions. In this case, the only random variables in a path tracer are those of the camera. We exploit this observation for variance bounding by *virtually* constructing deterministic transport around *any* sampled path X (effectively disabling all random sampling at inner path vertices), and by then quantifying the relative change in variance compared to the *actual* transport (i.e, with all random sampling along the path enabled).

As it is impractical to reason about transport in the entire path space Ω, we need to enable a localized analysis. For this, we partition the space $\mathcal{U}$ of random variables U, from which a sampling strategy constructs paths X(U): The pixel filter is extended to $\mathcal{U}$ by setting $h_j(\mathrm{U}) := h_j(\mathrm{X}(\mathrm{U}))$, and partitioned by *subfilters* $g_i(\mathrm{U}) \leq h_j(\mathrm{U})$, such that $\sum_i g_i(\mathrm{U}) = h_j(\mathrm{U})$. Fig. 2 depicts (in green) a subfilter that selects a subset of values in each dimension of $\mathcal{U}$ and thus a subset of all paths X(U). To *locally* construct deterministic transport around a sampled path $\mathrm{X}_i = \mathrm{X}(\mathrm{U}_i)$, we define a convenient subfilter $g_i(\mathrm{U})$ (ultimately a Gaussian). We then swap $g_i(\mathrm{U})$ with a corresponding "deterministic" filter $g_i^d(\mathrm{U})$, which shrinks $g(\mathrm{U})$ until it is non-zero for exactly one sample U at each image-space position $\mathbf{p}_x(\mathrm{U})$, while keeping its sample value $f(\mathrm{U})$ (see also Fig. 2). We ensure that the filter contains the initial X_i (i.e. $g_i^d(\mathrm{X}_i) \neq 0$), and quantify changes in relative variance, comparing the deterministic to the actual transport:

Proposition 4. *Let $g_i(\mathrm{U})$ be a subfilter of $h_j(\mathrm{U})$ and $\mathrm{X}_i = X(\mathrm{U}_i)$ a contained path sample. We construct a 'deterministic' filter $g_i^d(\mathrm{U})$ that locally modifies light transport such that $g_i^d(\mathrm{U})f(\mathrm{U})$ transports energy on only one path X for each pixel position $\mathbf{p}_x(\mathrm{X})$, while conserving the probability volume $\int g_i(\mathbf{p}_x)\,\mathrm{d}\mathbf{p}_x \stackrel{!}{:=} \int g_i(\mathrm{U})\,\mathrm{dU} \circ \mathbf{p}_x{}^{-1}$ of the original subfilter (for any pixel positions $\mathbf{p}_x$).[2] We use a shrinking kernel $\hat{K}^d_{\mathbf{p}_x(\mathrm{U})}(\mathrm{U}-\mathrm{U}_i)$, narrowing towards U_i for $d \to 0$:*

$$g_i^d(\mathrm{U}) := g_i\big(\mathbf{p}_x(\mathrm{U})\big)\ \hat{K}^d_{\mathbf{p}_x(\mathrm{U})}\big(\mathrm{U}-\mathrm{U}_i\big), \tag{11}$$

where $\hat{K}^d_{\mathbf{p}_x}(\mathrm{U}-\mathrm{U}_i)$ is normalized for each $\mathbf{p}_x$ such that

$$\int g_i(\mathbf{p}_x)\,\hat{K}^d_{\mathbf{p}_x}(\mathrm{U}-\mathrm{U}_i)\,\mathrm{dU} \circ \mathbf{p}_x{}^{-1} \stackrel{!}{=} \int g_i(\mathbf{p}_x)\,\mathrm{d}\mathbf{p}_x.\ ^3 \tag{12}$$

Modifying light transport locally, only within a subfilter $g_i(\mathrm{X})$, corresponds to changing pixel filters $h_j(\mathrm{X})$ to:

$$h_j^{|d_i}(\mathrm{U}) := h_j(\mathrm{U}) - g_i(\mathrm{U}) + g_i^d(\mathrm{U}). \tag{13}$$

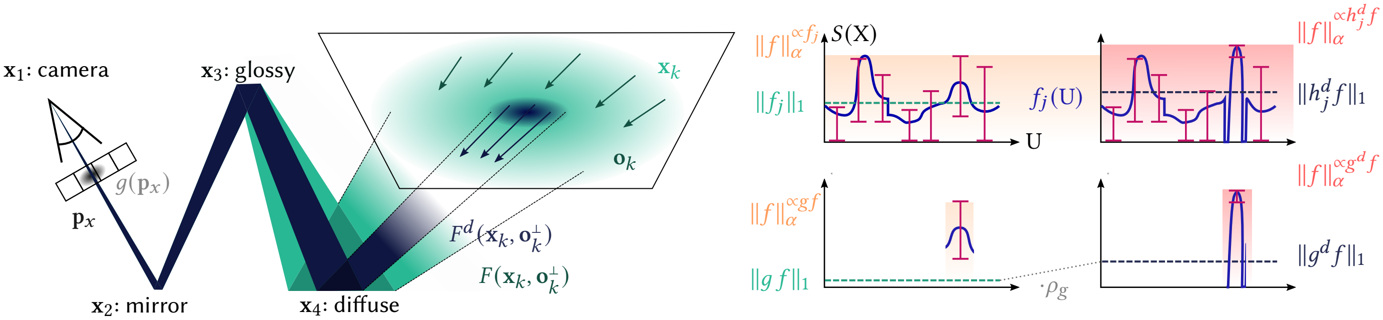


Fig. 2. Depiction of an exemplary subfilter $g(\mathrm{U})$ (in green), that selects subsets of values in each dimension of $\mathcal{U}$ and thus subsets of all paths $\mathrm{X}(\mathrm{U})$, and of an exemplary deterministic subfilter $g^d(\mathrm{U})$ (in blue), constructed for the "surrounding" subfilter $g(\mathrm{U})$ to effectively (virtually) replace all inner path vertices with specular interactions. The marginal endpoint contribution functions $F_{[d]}(\mathbf{x}_k^\perp, \mathbf{o}_k)$ are in practice computed by convolution within local environments defined by Gaussian subfilters, and they characterize the fraction of light transported within the subfilters for each emitter position $\mathbf{x}_k$ and direction $\mathbf{o}_k$. An exemplary distribution of sample values within one pixel is depicted on the right, with the moments (norms) used to compute variance bounds (for the actual and the locally altered transport, based on the full-pixel transport and based on only transport within the subfilters, as used in Eq. (15) for the conservative bounding).

We will show that the full $h_j^{|d_i}(\mathrm{U})$ does not need to be known for the following conservative variance bounding: We can compute a factor ρ_{g_i} that bounds the increase in relative variance, moving from locally altered, deterministic transport for paths weighted by a subfilter $g_i(\mathrm{U})$, to the unaltered transport. For every $d > 0$, we pick $\alpha \geq 2$ such that $\|f\|_{\alpha-1}^{\propto f\, h_j^{|d_i}} \geq \|f\|_{\alpha-1}^{\propto f_j}$ (as introduced in Prop. 3), and obtain:

$$\frac{J_j^{(2)}}{I_j{}^2} \underset{(9)}{\leq} \left(\frac{J_j^{(\alpha)}}{I_j{}^\alpha}\right)^{\frac{1}{\alpha-1}} \underset{(9)}{=} \frac{\|f\|_{\alpha-1}^{\propto f_j}}{\|f_j\|_1} \leq \underbrace{\frac{\|h_j^{|d_i} f\|_1}{\|h_j f\|_1}}_{=:\rho_{g_i}} \left(\frac{J_{j|d_i}^{(\alpha)}}{I_{j|d_i}{}^\alpha}\right)^{\frac{1}{\alpha-1}}. \tag{14}$$

The last term in Eq. (14) characterizes the controlled relative variance of the altered deterministic transport. Computing the variance bounding ratio ρ_{g_i} precisely is unfeasible, as it requires knowledge of the solution to the MC integration problem. However, we find a conservative bound based on only local information, regardless of the remaining light transport outside our subfilter $g(\mathrm{U})$:

Proposition 5. *Let $g(\mathrm{U})$ be a subfilter and let $g^d(\mathrm{U})$ define its locally altered deterministic transport. For any non-negative measurement contribution function $f(\mathrm{U})$, we get:*

$$\rho_g = \frac{\|h_j^{|d_i} f\|_1}{\|h_j f\|_1} \overset{(A.3)^4}{\leq} \frac{\int_{\mathcal{U}} g^d(\mathrm{U}) f(\mathrm{U})\, \mathrm{dU}}{\int_{\mathcal{U}} g(\mathrm{U}) f(\mathrm{U})\, \mathrm{dU}} =: r_g, \quad \text{if } \rho_g > 1. \tag{15}$$

□

Within local subfilters, bounding the variance ratio is not only feasible, but can be simplified to a mere single-point evaluation of the respective transport path distributions for the altered deterministic and the non-deterministic transport. In fact their marginals, which we compute by path differentials in Sect. 5.2, evaluated at a single 4D endpoint $(\mathbf{x}_k, \mathbf{o}_k)$ (comprised of an emitter position and direction), are sufficient to obtain a conservative ratio bound:

Proposition 6. *We denote the (marginal) contribution of all paths with endpoints and emitter directions $(\mathbf{x}_k, \mathbf{o}_k)$ to a subfilter $g(\mathrm{X})$ as $F_d(\mathbf{x}_k^\perp, \mathbf{o}_k)$ for locally altered deterministic, and $F(\mathbf{x}_k^\perp, \mathbf{o}_k)$ for the actual transport. (Let $(\mathbf{x}_k, \mathbf{o}_k)(\mathrm{U})$ be the endpoint of $X(\mathrm{U})$, then we set $\int F_{[d]}(\mathbf{x}_k^\perp, \mathbf{o}_k) L_e(\mathbf{x}_k^\perp, \mathbf{o}_k)\, \mathrm{d}(\mathbf{x}_k^\perp, \mathbf{o}_k) \overset{!}{=} \int g^{[d]}(\mathrm{U}) f(\mathrm{U})\, \mathrm{dU} \circ (\mathbf{x}_k, \mathbf{o}_k)_\circ^{-1}$ where $(\mathbf{x}_k, \mathbf{o}_k)^{-1}$ refers to the preimage). The ratio ρ_g is bounded by the ratio of endpoint distributions for the different configurations:*

$$\rho_g \leq \max_{\mathbf{x}_k, \mathbf{o}_k} \frac{F_d(\mathbf{x}_k^\perp, \mathbf{o}_k)}{F(\mathbf{x}_k^\perp, \mathbf{o}_k)} =: \tilde{r}_g. \tag{16}$$

Proof. Equations (15), (32), Appendix A. □

Densities with respect to projected vertices $\mathbf{x}_k^\perp$ avoid numerical singularities ($\perp$ denotes projection orthogonal to surface normals).

5.2 Computing the Variance Ratio $\tilde{r}_g$

Propositions 5 and 6 allow us to quantify relative variance caused by non-deterministic light transport within certain subfilters $g(\mathrm{U})$. In order to compute the respective integrals, we need to know $f(\mathrm{U})$, defined by a corresponding path sampling strategy $\mathrm{X}(\mathrm{U})$ such that $f(\mathrm{U})\, \mathrm{dU} \circ \mathrm{X}^{-1} = f(\mathrm{X})\, \mathrm{dX}$. Since path tracing is simple and works well with deterministic transport, we choose it as our $\mathrm{X}(\mathrm{U})$ and obtain the sample values $f(\mathrm{U}) = f(\mathrm{X})/p(\mathrm{X})$, where the PDF $p(\mathrm{X})$ applies local importance sampling to cancel out BSDF variance.

After importance sampling, variance in the resulting samples $f(\mathrm{U})$ is primarily due to the emitted light $L_e(\mathbf{x}_k^\perp, \mathbf{o}_k)$ at the endpoint of X. We write $f(\mathrm{U}) = A(\mathrm{U}) L_e(\mathbf{x}_k^\perp, \mathbf{o}_k)$ with an albedo $A(\mathrm{U}) \leq 1$ so that solutions for the marginal functions $F_{[d]}(\mathbf{x}_k^\perp, \mathbf{o}_k)$ of a subfilter $g(\mathrm{U})$ and its locally altered, deterministic subfilter $g^d(\mathrm{U})$ in Prop. 6 are:

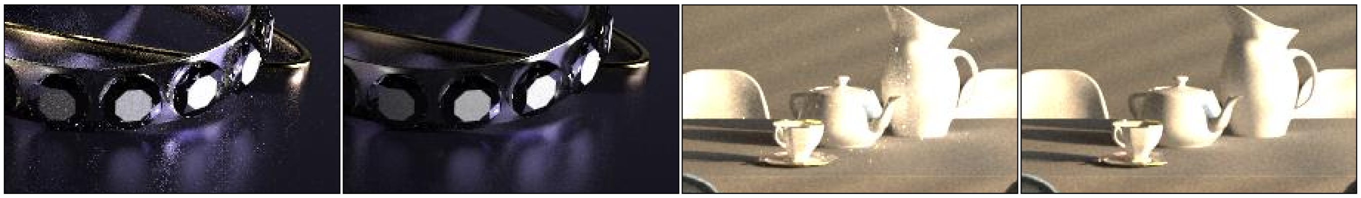
$$F_{[d]}(\mathbf{x}_k^\perp, \mathbf{o}_k) := \int_{\mathcal{U}} g^{[d]}(\mathrm{U}) A(\mathrm{U}) \delta\left((\mathbf{x}_k^\perp, \mathbf{o}_k)(\mathrm{U}) - (\mathbf{x}_k^\perp, \mathbf{o}_k)\right) \mathrm{dU}. \tag{17}$$

Given a sampled path X_i, we place an isotropic multivariate Gaussian around its random vector U_i to define a subfilter $g_i(\mathrm{U})$ in $\mathcal{U}$ that is easy to work with: This allows us to compute the marginal endpoint contribution functions $F(\mathbf{x}_k^\perp, \mathbf{o}_k)$ and $F_d(\mathbf{x}_k^\perp, \mathbf{o}_k)$ as convolutions of Gaussians at each interaction, if we assume separability (discussed in App. A.2; we assume albedo to be uncorrelated and

[2] $g(\mathbf{p}_x)$ effectively marginalizes $g(\mathrm{U})$ for all U in the preimage $\mathrm{X}^{-1} \circ \mathbf{p}_x{}^{-1}$ of $\mathbf{p}_x(\mathrm{X}(\mathrm{U}))$, i.e. for all paths X with certain pixel positions $\mathbf{p}_x$. It is the Radon-Nikodym derivative $\frac{\mathrm{d}\mu}{\mathrm{d}\nu}(\mathbf{p}_x)$ of $\mu(A) := \int_A g(\mathrm{U})\, \mathrm{dU} \circ \mathbf{p}_x{}^{-1}$ w.r.t. $\nu(\mathbf{A}) = \int_A \mathrm{d}\mathbf{p}_x$.

[3] The normalization constant is the Radon-Nikodym derivative $\frac{\mathrm{d}\mu}{\mathrm{d}\nu}(\mathbf{p}_x)$ of the unnormalized measure $\mu(A) := \int_A g(\mathbf{p}_x) K_{\mathbf{p}_x}^d(\mathrm{U} - \mathrm{U}_i)\, \mathrm{dU} \circ \mathbf{p}_x{}^{-1}$ w.r.t. $\nu(A) := \int_A g(\mathbf{p}_x)\, \mathrm{d}\mathbf{p}_x$.

[4] ρ_g is of the form $\frac{a+b}{a+c}$, where $a, b, c \geq 0$. For $b > c$, $\frac{a+b}{a+c} \leq \frac{b}{c}$, see also App. A.3.



Raw Splat Weights | Splats Bounded using $\tilde{r}$ | Raw Splat Weights | Splats Bounded using $\tilde{r}$

Fig. 3. Experiment validating that downweighting by the ratio $\tilde{r}$ reliably bounds variance: The samples of a path tracer are reweighted by $\tilde{r}^{-1}$. The results used the differential-free variance estimation derived in Sect. 6.5. Our supplementary materials show more scenes, also for the differential-based variant.

set $A(\mathrm{U}) = 1$ for practical computations). Similar local, convolution-based perspectives on light transport are well-established in previous work [Belcour et al. 2017]. In practice, this convolution corresponds to computing path differentials, i.e. the Jacobians of the mappings from random variable perturbations at each interaction to corresponding endpoint perturbations (Sect. 6). As the ratio $\tilde{r}_g$ is independent of normalization factors in $g(\mathrm{U})$ (they cancel out), for convenience we normalize $g(\mathrm{U})$ w.r.t. dU in our computations. Note that for Gaussian marginal endpoint distributions, the ratio is maximal in the center, since the covariance of the wider Gaussian contains that of the other, making the ratio Gaussian as well.

5.3 Applying the Variance Bound

The computed $\tilde{r}_g$ conservatively bounds the ratio by which large sample values in the bounded MC estimator deviate from the final result, compared to low-variance MC estimates of *deterministic* light transport. Prop. 3 showed that this ratio directly bounds the relative variance of a pixel. This information has several uses: We can drive optimized adaptive sample placement (see Sect. 7.1) to match a certain variance target. We can also use it to downweight sample values by the ratio, as in our stratification per Eq. (8), such that the variance of the resulting splats is bounded (see Sect. 4 and 6).

6 MCMC STRATIFICATION BY PATH DIFFERENTIALS

Practically, to implement our stratification (as per Sect. 4.3) in an MCMC algorithm (e.g. MLT, PSSMLT, MMLT), we compute the variance bounds derived in the previous section using *path differentials* [Belcour et al. 2017; Suykens and Willems 2001]. Path differentials (also *ray differentials*) are commonly computed for the purposes of level-of-detail and filtering [Christensen et al. 2018; Fascione et al. 2018]. We show how accurate path differentials can be computed efficiently and we introduce a conservative approximation thereof that works without derivative information, significantly simplifying computations and increasing numerical robustness.

6.1 Summary of Changes to MLT

To give an overview of the minimal required changes, the concrete steps to stratifying paths sampled by a Markov Chain are:

- For current path X, compute bounded-error pixel splat values $\tilde{S}(\mathrm{X})$ as per Sect. 4. For this, either accurate path differentials (Sect. 6.4) or a conservative approximation based on solely solid-angle PDF values (Sect. 6.5) can be used.
- Alter the unstratified importance function $f(\mathrm{X})$ to $f(\mathrm{X})/\tilde{S}(\mathrm{X})$.
- Accordingly, splats need to be multiplied by $\tilde{S}(\mathrm{X})$.

6.2 Marginalization by Path Differentials

To arrive at the variance bounding ratio $\tilde{r}_g$ in Prop. 6 by comparison of the marginal endpoint contributions $F(\mathbf{x}_k^\perp, \mathbf{o}_k)$ and $F_d(\mathbf{x}_k^\perp, \mathbf{o}_k)$, we marginalize their respective subfilters $g^{[d]}(\mathrm{U})$ by convolution of Gaussian kernels at each interaction $\mathbf{x}_i$ around a current path $X(\mathrm{U})$. In the process, we use path differentials to effectively project each respective covariance matrix forward onto the 4D space of endpoints $(\mathbf{x}_k^\perp, \mathbf{o}_k)$. The full covariance matrix Σ for $F(\mathbf{x}_k^\perp, \mathbf{o}_k)$ is:

$$\Sigma\left[F(\mathbf{x}_k^\perp, \mathbf{o}_k)\right] = \sum_{i=0}^{k-1} \begin{pmatrix} \frac{\mathrm{d}\mathbf{x}_k^\perp}{\mathrm{d}\mathbf{u}_i} \frac{\mathrm{d}\mathbf{x}_k^\perp}{\mathrm{d}\mathbf{u}_i}^T & \frac{\mathrm{d}\mathbf{x}_k^\perp}{\mathrm{d}\mathbf{u}_i} \frac{\mathrm{d}\mathbf{o}_k}{\mathrm{d}\mathbf{u}_i}^T \\ \frac{\mathrm{d}\mathbf{o}_k}{\mathrm{d}\mathbf{u}_i} \frac{\mathrm{d}\mathbf{x}_k^\perp}{\mathrm{d}\mathbf{u}_i}^T & \frac{\mathrm{d}\mathbf{o}_k}{\mathrm{d}\mathbf{u}_i} \frac{\mathrm{d}\mathbf{o}_k}{\mathrm{d}\mathbf{u}_i}^T \end{pmatrix}. \tag{18}$$

Both $\mathbf{x}_k^\perp$ and $\mathbf{o}_k$ are parameterized by two coordinates in the plane orthogonal to $\mathbf{o}_k$ of the current path X. Since isotropic scaling of subfilters cancels out in our variance bounding, the covariance matrices for each random variable are implicitly set to identity. To obtain differentials efficiently, we exploit the tridiagonal structure of a corresponding constraint matrix [Jakob and Marschner 2012], as explained in App. B.1. Doing this per vertex results in a run-time complexity of $O(k^2)$, but in App. B.2 we show that the convolution can be computed iteratively in $O(k)$ and constant storage.

6.3 Marginalization for Degenerate Distributions

In practice, we often face degenerate endpoint distributions: Path vertices with singularities in 2D (pinhole cameras, point lights, directional lights) limit light transport to a 2-manifold within our studied 4D space of endpoint position and emitter direction. To account for that, we always *project* to differentiable 2-manifolds $\mathcal{L}$, studying projected marginals $F_{[d]}^{\perp\mathcal{L}}$ for charted projected endpoints $\pi_{\mathcal{L}}(\mathbf{x}_k, \mathbf{o}_k)$ where: $\int F_{[d]}^{\perp\mathcal{L}}(\pi_{\mathcal{L}})\, \mathrm{d}\pi_{\mathcal{L}}(\mathbf{x}_k, \mathbf{o}_k) \overset{!}{=} \int F_{[d]}(\mathbf{x}_k^\perp, \mathbf{o}_k)\, \mathrm{d}(\mathbf{x}_k^\perp, \mathbf{o}_k) \circ \pi_{\mathcal{L}}^{-1}$. A decisive difference to the definitions in Prop. 6 is that the full measurement contribution of *projected* endpoints $\pi_{\mathcal{L}}(\mathbf{x}_k, \mathbf{o}_k)$ requires separate light emission terms $L_e^{\perp\mathcal{L}}$ and $L_{e|d}^{\perp\mathcal{L}}$ that are also weighted marginals depending on the respective distribution of light transport paths [5]. In practice, the respective emission terms still cancel out in the limit, App. A.1 shows that we can use:

$$r_g \leq \max_{(\mathbf{x}_k, \mathbf{o}_k) \in \mathcal{L}} \frac{F_d^{\perp\mathcal{L}}(\pi_{\mathcal{L}}(\mathbf{x}_k, \mathbf{o}_k))}{F^{\perp\mathcal{L}}(\pi_{\mathcal{L}}(\mathbf{x}_k, \mathbf{o}_k))} \underbrace{\frac{L_{e|d}^{\perp\mathcal{L}}(\pi_{\mathcal{L}}(\mathbf{x}_k, \mathbf{o}_k))}{L_e^{\perp\mathcal{L}}(\pi_{\mathcal{L}}(\mathbf{x}_k, \mathbf{o}_k))}}_{\longrightarrow 1 \ (\hat{g} \to \delta)}. \tag{19}$$

[5] For $L_{e[|d]}^{\perp\mathcal{L}}$, we require: $\int F_{[d]}^{\perp\mathcal{L}} L_{e[|d]}^{\perp\mathcal{L}}\, \mathrm{d}\pi_{\mathcal{L}}(\mathbf{x}_k, \mathbf{o}_k) \overset{!}{=} \int F_{[d]} L_e\, \mathrm{d}(\mathbf{x}_k^\perp, \mathbf{o}_k) \circ \pi_{\mathcal{L}}^{-1}$.

To facilitate computing the projected marginal Gaussian distributions $F^{\perp\mathcal{L}}$ and $F_d^{\perp\mathcal{L}}$, we choose an axis-aligned projection $\pi_{\mathcal{L}}(\mathbf{x}_k, \mathbf{o}_k)$ to $\mathcal{L}$, allowing extraction of the respective 2×2-covariance matrices by simple rotation and truncation of the full 4×4-covariance matrices before projection. As to choosing the axes of projection, we note that the best worst-case variance bounds are achieved when projecting onto the plane of widest projected covariance, which is obtained by PCA of the covariance for $F(\mathbf{x}_k^{\perp}, \mathbf{o}_k)$. In our experiments, this approach was important to ensure robustness and prevent any uncaught outliers.

6.4 Accurate Path Differentials and Variance Ratios

Applying Prop. 6 for variance bounding, we arrive at a ratio $\tilde{r}_X$ based on a sampled path X by incrementally propagating the three 2×2 matrix blocks that make up the (symmetric) covariance matrix $\Sigma[F(\mathbf{x}_k^{\perp}, \mathbf{o}_k)]$ in Eq. (18). For the locally altered deterministic light transport as per $F_d(\mathbf{x}_k^{\perp}, \mathbf{o}_k)$, we track differentials with respect to only the random variables of the camera. The resulting two covariance matrices are projected onto the plane determined by the principal components $\mathbf{v}_1, \mathbf{v}_2$ of scattering in $\Sigma[F(\mathbf{x}_k^{\perp}, \mathbf{o}_k)]$:

$$\boxed{\tilde{r}_X := \frac{\left|\mathbf{V}\ \sum_{i=0}^{k-1}\left[\frac{\mathrm{d}(\mathbf{x}_k^{\perp},\mathbf{o}_k)}{\mathrm{d}\mathbf{u}_i}\ \frac{\mathrm{d}(\mathbf{x}_k^{\perp},\mathbf{o}_k)}{\mathrm{d}\mathbf{u}_i}^T\right]\mathbf{V}^T\right|^{1/2}}{\left|\mathbf{V}\left[\frac{\mathrm{d}(\mathbf{x}_k^{\perp},\mathbf{o}_k)}{\mathrm{d}(\mathbf{u}_0,\mathbf{u}_1)}\ \frac{\mathrm{d}(\mathbf{x}_k^{\perp},\mathbf{o}_k)}{\mathrm{d}(\mathbf{u}_0,\mathbf{u}_1)}^T\right]\mathbf{V}^T\right|^{1/2}}, \quad \mathbf{V} := \begin{pmatrix}\mathbf{v}_1^T\\ \mathbf{v}_2^T\end{pmatrix}.} \tag{20}$$

Practical Halfvector Derivatives. In current renderers, derivatives of all BSDFs with respect to primary sample space (PSS) are not generally available. However, our subfilters $g(\mathrm{U})$ for the random variables $\mathbf{u}_i$ of each vertex $\mathbf{x}_i$ are not required to be isotropic in PSS. Thus, we replace differentials $\mathrm{d}\mathbf{u}$ with halfvector differentials $\mathrm{d}\mathbf{h}$ more easily obtained [Hanika et al. 2015; Jakob and Marschner 2012; Kaplanyan et al. 2014], preserving densities (see App. B.3):

$$\frac{\mathrm{d}\mathbf{x}_k}{\mathrm{d}\mathbf{u}_i} \approx \frac{\mathrm{d}\mathbf{x}_k}{\mathrm{d}\mathbf{h}_i}\frac{\mathbf{I}}{\sqrt{p(\mathbf{h}_i)}} = \frac{\mathrm{d}\mathbf{x}_k}{\mathrm{d}\mathbf{h}_i}\left|\frac{\mathrm{d}\mathbf{h}_i}{\mathrm{d}\mathbf{i}_i}\right|^{1/2}\frac{\mathbf{I}}{\sqrt{p(\mathbf{i}_i)}} \quad (\mathbf{x}_i \text{ inner vertex}),$$

$$\frac{\mathrm{d}\mathbf{x}_k}{\mathrm{d}\mathbf{u}_1} \approx \frac{\mathrm{d}\mathbf{x}_k}{\mathrm{d}\mathbf{i}_1}\frac{\mathbf{I}}{\sqrt{p(\mathbf{i}_1)}}, \quad \frac{\mathrm{d}\mathbf{x}_k}{\mathrm{d}\mathbf{u}_0} \approx \frac{\mathrm{d}\mathbf{x}_k}{\mathrm{d}\mathbf{x}_1}\frac{\mathbf{I}}{\sqrt{p(\mathbf{x}_1)}} \quad (x_1 \text{ camera vertex}).$$

The PDFs $p(\mathbf{i}_i)$ are those of a forward path tracer. Note that path differentials based on halfvectors *can* be approximative, as they require strong BSDF peaks to align with points in halfvector space.

6.5 Derivative-free Conservative Path Differentials

In the following, we present a simplification of our technique that works without any derivative computations, requiring only the local solid-angle PDFs of BSDFs and few floating-point operations. While both approaches require little computation compared to other parts of a renderer, this simplifies implementation in practice.

We simplify our setting by *virtually* replacing the actual scene geometry $\mathcal{M}$ with a converging sequence $(\mathcal{M}_t)$ of increasingly fine piecewise linear approximations. With increasing t, more and more tangent planes in curved regions of $\mathcal{M}$ coincide with an increasing number of flat surfaces in $\mathcal{M}_t$, causing an increasing number of overlaying redirected beams to approach the actual transport on the original geometry. Such constructions are possible and their properties as *tangent duals* of surface triangulations have been studied extensively with application to architecture [Li et al. 2014].

On the planar geometry of $\mathcal{M}_t$, the transformation of sampling densities to endpoints is greatly simplified: Reflections preserve angles, and refractions merely change densities according to Snell's law. In fact, we can either cancel out or bound most of the relevant terms for all axis-aligned 2D projections of our 4D endpoint distributions used to bound variance (please refer to App. C.1). The remaining terms give a variance bound $\tilde{r}_X$ with an approximation factor of 4 and a greatly simplified algorithm:

$$\boxed{\tilde{r}_X := p(\mathbf{i}_1)\left(\frac{1}{p(\mathbf{x}_1)}\left|\frac{\mathrm{d}\mathbf{x}_2^{\perp\mathbf{o}}}{\mathrm{d}\mathbf{i}_1}\right|^{-1} + \sum_{i=1}^{k-1}\frac{1}{p(\mathbf{i}_i)}\left|\frac{\mathrm{d}\mathbf{i}_i}{\mathrm{d}\mathbf{i}_1}\right|^{-1}\right).} \tag{21}$$

After simplification, the required PDFs $p(\mathbf{i}_i)$ are simply the solid-angle PDFs resulting from BSDF sampling, $p(\mathbf{i}_1)$ corresponds to pixel sampling and $p(\mathbf{x}_1)$ to aperture. The term $\left|\frac{\mathrm{d}\mathbf{i}_i}{\mathrm{d}\mathbf{i}_1}\right|^{-1}$ is a product of factors $\left|\frac{\mathrm{d}\mathbf{o}_i}{\mathrm{d}\mathbf{i}_i}\right| = \frac{|\mathbf{i}_i\cdot\mathbf{h}_i|\eta_i}{|\mathbf{o}_i\cdot\mathbf{h}_i|\eta_o}$ per refractive boundary, 1 for reflections. $\left|\frac{\mathrm{d}\mathbf{x}_2^{\perp\mathbf{o}}}{\mathrm{d}\mathbf{i}_1}\right|^{-1}$ is the first geometry term, excluding the cosine at $\mathbf{x}_2$.

6.6 Making use of Multiple Importance Sampling

We can make use of Multiple Importance Sampling (MIS) to reduce the amount of conservative overestimation in variance ratios $\tilde{r}_X$. For example, if our renderer supports next event estimation (NEE), the variance of direct illumination is already controlled by MIS weights. This not only allows us to use sample weights of short direct illumination paths without downweighting, but also allows us to reduce variance bounds for all other paths: Variance does not change, if the direct illumination is seen through additional purely deterministic interactions in-between the penultimate vertex and the camera. The degree to which variance changes for non-deterministic path prefixes can again be measured by applying our variance bounding scheme, but computing marginal distributions for the penultimate vertex rather than the last vertex on the emitter. Thus, we effectively convert the penultimate vertex into a (virtual) light source, under the assumption that the variance of its virtual emitted radiance is sufficiently low.

6.7 Degenerate Light Sources in Specular Light Transport

Making use of MIS already solves all cases of degenerate light sources for our techniques, i.e. (by construction) all cases of paths generated by path tracing with NEE. For other (e.g. bi-directional) techniques, longer specular vertex chains connecting to a *degenerate* emitter may require special treatment. There, the reduced dimensionality allows analytic computation of the local deterministic transport $\int_{\mathcal{U}} g^d f(\mathrm{U})\,\mathrm{dU}$ (or a low-variance MC estimate thereof for non-pinhole cameras, dividing by the aperture PDF) as a substitute for the otherwise unbounded splat weights: This resembles pencil tracing [Shinya et al. 1987], where *pencil* refers to a similar object as the beams resulting from our subfilters. Then, derivations analogous to the bounding in Eq. (32) (with reduced dimensionality) lead to analogous variance bounding ratios.

6.8 Target Function Alteration

We may change the importance function in Metropolis light transport without introducing bias [Hoberock and Hart 2010]. Inserting $\tilde{r}$ into $\tilde{S}$ in Eq. (8), for the acceptance ratio A, we get:

$$A = \min\left\{1, \frac{f(\mathrm{Y})/\tilde{S}(\mathrm{Y})\, p(\mathrm{Y} \to \mathrm{X})}{f(\mathrm{X})/\tilde{S}(\mathrm{X})\, p(\mathrm{X} \to \mathrm{Y})}\right\}. \tag{22}$$

To account for the changed importance function, we then have to splat the following changed weights:

$$\text{Splat for proposal Y:} \quad \tilde{S}(\mathrm{Y}) \cdot A,$$
$$\text{Accumulate for current X:} \quad \tilde{S}(\mathrm{X}) \cdot (1 - A).$$

7 ADAPTIVE MCMC FORWARD PATH SAMPLING

Our variance analyses in Sect. 5 not only apply to stratification, but can also be leveraged to drive adaptive sampling: In Sect. 7.1, we optimize anisotropic adaptive distributions to maximize the explored subsets of path space, while enforcing a controlled variance. We test efficacy by deriving an intentionally simple, yet adaptive MCMC *small-step* perturbation that only relies on forward path sampling. Note that this constraint is realistic in many real-world applications, where path tracing continues to be widely used due to a manageable complexity, relatively uniform efficiency, and robustness. To obtain a complete MCMC algorithm with an equally simple and practical *large-step* mutation, Sect. 7.2 shows how to embed a fully-featured path tracer (PT) with next event estimation (NEE) and shared path prefixes into the MCMC context. We construct our MCMC rendering algorithm with only these two mutation strategies.

7.1 Adaptive Forward Path Exploration

For optimal adaptive sampling, we observe two opposing goals: Our sampling strategy needs to carefully explore difficult-to-discover regions in path space, i.e. propose small steps with a high sampling density where important transport is easily missed. Implicitly, this minimizes the variance of resulting proposal weights, i.e. the targeted distribution divided by the proposal density. However, *overly small* steps *increase* MCMC variance due to increasing covariance, preventing the discovery of other important regions in path space and thus reducing robustness (resp. temporal stability).

7.1.1 Optimal Anisotropic Proposal Densities. The conservative variance analysis in Sect. 5 provides us with a tool to systematically optimize variance with respect to choice of sampled regions, by looking at samples within an according subfilter $g_i(\mathrm{X})$ around any difficult path X_i. However, in contrast to subfilters within individual pixels, we now look at *subfilters of the entire image plane.* Sampling in larger areas than one pixel may increase the variance by the corresponding decrease in probability density, henceforth denoted D_{g_i} (effectively the image-plane area of $g_i(\mathrm{U})$ divided by the area of one pixel). We minimize the correlation of proposal paths by maximizing the sampled (PSS) volume of a corresponding subfilter $g_i(\mathrm{X})$, which we define as a Gaussian function around $g_i(\mathrm{X}_i) = 1$. A volume of 1 would correspond to (close to) perfectly independent sampling, smaller volumes increase the potential for correlation artifacts. The size and shape of our adaptive proposal distribution is determined by a constrained optimization, such that the volume of the corresponding region is maximized while the variance within the corresponding subfilter $g_i(\mathrm{X})$ is constrained to an acceptable R:

$$\max_{g_i(\mathrm{X})} \int_{\mathcal{U}} g_i(\mathrm{U})\,\mathrm{dU}, \quad \text{s.t.} \quad \tilde{r}_{g_i} D_{g_i} \overset{!}{=} R. \tag{23}$$

We set R to one over the square root of the image pixel count, such that after one sample per pixel, the corresponding variance estimate is compensated for by the number of samples in the image plane.

7.1.2 Sampling Algorithm. Adaptively sampling proposal paths becomes straight-forward once we have determined the size and shape of the optimized region in path space: we construct paths from the camera, sampling each next direction according to the distribution obtained for each respective vertex by the optimization problem in Eq. (23). It can be solved using either our differential-free simplification of the variance estimation in Eq. (21), or using path differentials. Both solutions are simple and efficient to implement. The differential-free version has better numerical robustness due to lack of matrix math (circumventing degenerate matrices), while the full-differential version can lead to slightly better decorrelation of samples in some cases, which we attribute to better capturing of anisotropic structure in path space.

Solving for the Optimal Proposal Distribution. The constrained optimization in Eq. (23) is a Karush–Kuhn–Tucker (KKT) system, with the implicit inequalities that our sampled region should be a subset of the full relevant space of paths. For brevity, we discuss the simplified version here. Please refer to App. C.2 for more details on the differential version. Our solution defines $g_i(\mathrm{X})$ as a multivariate Gaussian $\exp\left(-1/2 \sum_{j=0}^{k-1} s_j \|\mathbf{u}_j - \mathbf{u}_j^{(i)}\|^2\right)$ with the same dimensionality as the current path X_i. To shape $g_i(\mathrm{X})$, we find optimal scaling coefficients $s_j \leq 1$ for each dimension. For comparison, we define the area of one pixel in the space of $\mathbf{u}_1$ as s_{px}, such that $D_{g_i} = s_1/s_{\mathrm{px}}$. Using the chain rule, we adapt Eq. (21):

$$\tilde{r}_{g_i} D_{g_i} = \frac{p(\mathbf{i}_1)}{s_1} \left(\frac{s_0}{p(\mathbf{x}_1)} \left| \frac{\mathrm{d}\mathbf{x}_2^{\perp \mathbf{o}}}{\mathrm{d}\mathbf{i}_1} \right|^{-1} + \sum_{j=1}^{k-1} \frac{s_j}{p(\mathbf{i}_j)} \left| \frac{\mathrm{d}\mathbf{i}_j}{\mathrm{d}\mathbf{i}_1} \right|^{-1} \right) \frac{s_1}{s_{\mathrm{px}}}. \tag{24}$$

In Listing 1, lines 1–11 show pseudocode to compute the corresponding factors. The subsequent lines determine optimal coefficients s_i using a Lagrange multiplier (λ) and applying the KKT conditions. Deriving w.r.t. s_j, we obtain the necessary condition:

$$\frac{\lambda}{s_{\mathrm{px}}} \frac{p(\mathbf{i}_1)}{p(\mathbf{i}_j)} \left| \frac{\mathrm{d}\mathbf{i}_j}{\mathrm{d}\mathbf{i}_1} \right|^{-1} \overset{!}{=} \frac{\prod_{l=0}^{k} s_l}{s_j}. \tag{25}$$

(Analogous for $j = 0$). We note that each s_j is reciprocal to its coefficient in Eq. (24), which simplifies the constraint to:

$$(21) = \sum_{l=0}^{k} \underbrace{\frac{\prod_{l=0}^{k} s_l}{\lambda}}_{=:\lambda'} \overset{!}{=} R, \quad \lambda' = R/k. \quad s_j = \frac{R}{k} p(\mathbf{i}_j) \left| \frac{\mathrm{d}\mathbf{i}_j}{\mathrm{d}\mathbf{i}_1} \right| \frac{s_{\mathrm{px}}}{p(\mathbf{i}_1)}. \tag{26}$$

Therefore, the optimal distributions depend mostly on the number of optimized coefficients, but interestingly cancel out the original scattering behavior of inner vertices. Line 13 in Listing 1 accounts for $p(\mathbf{i}_1)$ in Eq. (25) and R in Eq. (26). Line 17 accounts for λ'. The optimal proposal density in primary sample space is computed in

Listing 1. Pseudocode for our adaptive forward path sampling. Vertices $\mathbf{x}_0$ and $\mathbf{x}_1$ refer to the same camera vertex with positional and directional attributes, respectively. The last vertex $\mathbf{x}_k$ refers to a point on an emitter.

in	PDFs $p(\mathbf{x}_i)$ of a plain path tracer for the current path X
out	proposal distribution around X, i.e. $\mathcal{N}(\mathbf{x}_{i+1}, (2\pi\,\mathtt{pdfs}_i)^{-1}\,\mathbf{I})$

```
1  // projected per-vertex footprints
2  float[k] footprints
3  float    angle_compression = 1
4  for i = 0 .. k-1:
5    angle_compression *= |di_i / do_i| if i > 1 else 1
6    if i >= 1:
7      pdf = p(x_{i+1}) |dx_{i+1} / di_i| * angle_compression
8    else:
9      pdf = p(x_1) |dx_2^⊥o / di_1|
10   footprints_i = 1 / pdf
11
12 // constrained optimization
13 target_diff = footprints_1 / √pixel count
14 bool [k] active = [ x_i is non-specular for i = 0..k-1 ]
15 float[k] amplifiers
16 do:
17   target_footprint = target_diff / count(active)
18   retry = false
19   for i = 0 .. k-1:
20     if active_i:
21       amplifiers_i = footprints_i / target_footprint
22       if not amplifiers_i > 1:
23         active_i = false
24         target_diff -= footprints_i
25         retry = true
26 while retry
27
28 // per-vertex adaptive proposal distribution
29 float[k] pdfs
30 for i = 0 .. k-1:
31   pdfs_i = p(x_{i+1}) * (amplifiers_i if active_i else 1)
```

Listing 2. Pseudocode adapting a full path tracer to a large-step mutation.

in	*iff* computing the reverse sampling weight, the `proposal`
out	selected `proposal` path & its weight in the acceptance ratio

```
float totalSplat = 0
Path  candidate, proposal

// for reverse weights, proposal is already given
if proposal:
  if rand() < proposal.mis_weight(NEE):
    t_rev = NEE, candidate = proposal.trimmed(-1)
  else:
    t_rev = Hit, candidate = proposal
  Spectrum contrib = proposal.generic_contrib_mis()
  totalSplat += contrib.luminance()

// sample new candidates or additional candidates
while candidate.sample_or_keep_next_vertex():
  for t in [NEE, Hit]:
    // skip technique that (supposedly) sampled prop.
    if t_rev==t and candidate.k_for(t)==proposal.k():
      continue
    Spectrum contrib = 0
    if t == Hit and candidate.hit_emitter():
      contrib = candidate.hit_contrib()
    if t == NEE and candidate.sample_nee():
      contrib = candidate.nee_contrib()
    contrib *= candidate.mis_weight(t)

    float intermSplat = totalSplat
    totalSplat += contrib.luminance()
    // sample proposal among new candidates
    if not t_rev and contrib:
      // proportional to their contributions
      if rand() >= intermSplat / totalSplat:
        proposal = candidate.full_path(t)

// in practice: for reverse weights, we omit proposal
return proposal, totalSplat
```

line 21, as per scaling coefficients s_j. So far, we neglected the inequalities $s_j \leq 1$, which are checked in line 22: In case a dimension does not require increased sampling density, we remove its coefficient from the optimization, as suggested by the complementary slackness condition for KKT systems. As a result, the density of other dimensions may be decreased. Therefore, in this case, we repeat the optimization with the new set of constraints.

7.1.3 Path Extension and Truncation. We leverage the reversible jump framework [Geyer 2003; Green 2003] that allows adding and discarding of path vertices by dimension matching. Thus, we can accept shorter proposal paths that hit emitters early, and can extend proposals by random walks (with Russian Roulette termination) in case they do not hit an emitter at the current path length.

7.2 Efficient Path Tracing in an MCMC mutation

With the exception of PSSMLT approaches that implicitly mutate collections of paths according to their marginal contribution to the image [Kelemen et al. 2002], previous MCMC algorithms in single-path state spaces fail to compete with uncorrelated MC methods in terms of path discovery efficiency: Common MCMC mutation strategies construct individual path samples, while classic MC techniques commonly create many paths at once, reusing previously sampled subpaths as shared prefixes in multiple importance sampling techniques. In our new large-step mutation, we make use of the reversible jump (RJ) framework [Geyer 2003; Green 2003] to embed a full-featured path tracer with next event estimation into a Markov Chain running on a single-path state space. Thus, we obtain a complete adaptive MCMC rendering algorithm based on purely forward path tracing. (Note that recent works [Bitterli et al. 2017; Otsu et al. 2017; Pantaleoni 2017] show that primary sample space and path space are in principle interchangeable, however, both state spaces are single-path state spaces in these works).

Sampling Many Candidates. Our path sampling technique builds on a generalized perspective on the Multiple Try (MT) method [Craiu and Lemieux 2007; Liu et al. 2000] using the RJ framework. In our background section, we already provided an overview of recent advancements using both frameworks. The RJ framework provides a way to determine valid proposal acceptance ratios for mutations

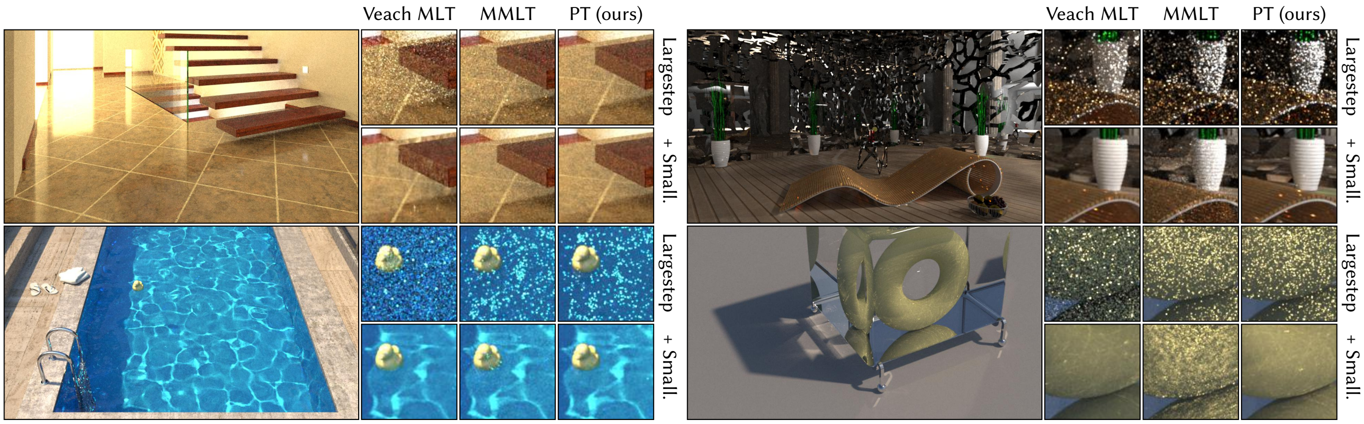


Fig. 4. Equal-time comparisons (50s on 64 cores) of our PT large step to the bi-directional mutation and to MMLT, both in isolation and combined with the remaining standard Veach MLT, resp. MMLT, small steps. Our large step improves the path discovery comparably to MMLT, but in contrast to MMLT allows the full arsenal of path-space small-step perturbations. These perturbations can improve results, as shown here for the Veach MLT family of small steps.

that add or remove dimensions to the proposal path, compared to the source path, and, as we show in App. D, even for the construction of many candidate paths at once, by a stochastic candidate selection similar to MT. The resulting algorithm requires few modifications to a standard path tracer, as we show in the following. We focus on the algorithmic part in this section, please refer to App. D for details on the self-inverse mapping that realizes the RJ, and the derivation of the resulting Jacobian determinant in the acceptance ratio.

7.2.1 Proposal Path Sampling. To obtain the next proposal path, we begin by sampling candidate paths in exactly the same way as a path tracer does: Listing 2, lines 14 – 27, recursively traces rays starting at a random location on the screen, then samples scattering (BSDF) and shadow rays (NEE) for every interaction. The MIS-weighted sample value $w(\mathrm{X}_i)$ of each full path X_i along the way is a good choice for its candidate sampling weight [Bitterli 2015; Martino and Louzada 2017]. We avoid storing all candidate paths by replacing candidate paths incrementally (line 32), i.e. using weighted reservoir sampling [Chao 1982]: We construct a Russian Roulette (RR) (line 31) such that the final selected path is sampled with probability $P(\mathrm{X}_i)$ proportionally to its candidate weight, divided by the sum of all other candidate weights: For this, in every step of the path tracing recursion, the chance $P_s(\mathrm{X}_i)$ of a candidate X_i (replacing the previous tentative proposal) is:

$$P_s(\mathrm{X}_i) = \frac{w(\mathrm{X}_i)}{\sum_{j=1}^{i} w(\mathrm{X}_j)}, \quad w(\mathrm{X}_i) = w_{\mathrm{MIS}}(\mathrm{X}_i)\frac{f(\mathrm{X}_i)}{p(\mathrm{X}_i)}, \tag{27}$$

$$\Rightarrow P(\mathrm{X}_i) = P_s(\mathrm{X}_i)\prod_{j=i+1}\left(1 - P_s(\mathrm{X}_j)\right) = \frac{w(\mathrm{X}_i)}{\sum_{j=1} w(\mathrm{X}_j)}. \tag{28}$$

Improving Mixture. In plain Multiple-Try Metropolis, the weight of the resulting proposal in the acceptance ratio would be $\frac{f(\mathrm{X}_i)}{p(\mathrm{X}_i)\,P(\mathrm{X}_i)}$. This is still suboptimal as it fails to apply variance reduction using MIS to the actual proposal weight. By modification of the dimension-matched mapping that describes our mutation in the reversible jump framework (App. D), adding an auxiliary random encoding of the active sampling technique (similarly to BSDF layers in previous work [Bitterli et al. 2017]), we can instead use the MIS'ed sample value. The resulting proposal weight becomes exactly the variance-reduced weight of the corresponding uncorrelated MC algorithm:

$$\frac{f(\mathrm{X}_i)}{p(\mathrm{X}_i)\,P(\mathrm{X}_i)} = \frac{w(\mathrm{X}_i)}{P(\mathrm{X}_i)} = \sum_j w(\mathrm{X}_j), \quad \forall i. \tag{29}$$

7.2.2 Reverse Sampling Weights. In order to maintain detailed balance, both MT and RJ require that we also sample alternative candidates for the source path. These candidates decide the final source weight in the MCMC acceptance ratio. Correctly simulating the sampling of the source path requires knowledge whether it was sampled by NEE or forward BSDF sampling: If a technique constructed the source path, it cannot provide a second alternative candidate (Listing 2, line 17). Our construction of the dimension-matched mutation mapping (App. D) uses an auxiliary variable that is distributed according to the MIS weights of the current path for each technique, requiring the decision, which candidates to simulate, to be randomly chosen based on exactly these MIS weights (lines 6 – 10).

8 RESULTS AND DISCUSSION

We implemented our approaches in the Mitsuba renderer [Jakob 2010], which comes with path space Veach MLT [Veach and Guibas 1997] and primary sample space MLT (PSSMLT) frameworks. We added implementations of Multiplexed MLT (MMLT) [Hachisuka et al. 2014], multi-stage MLT [Hoberock and Hart 2010] for both MLT and PSS(M)MLT, and the adaptive sampling strategy by Szirmay-Kalos and Szécsi [2017] for comparison to our adaptive forward path tracing (APT). We also imported the public implementation of Halfvector Space Light Transport (HSLT) [Hanika et al. 2015; Kaplanyan et al. 2014]. The source code for all techniques is made available on the project website. All comparisons shown in the paper are equal time comparisons, rendered in 50s on a 64 core machine. Our supplementary materials provide comparisons for more scenes and techniques, with relative RMSE values.

8.1 Efficiency of the Stratification in MLT methods

We implemented our approach to stratification for both path space MLT and primary sample space (M)MLT variants. Fig. 5 shows

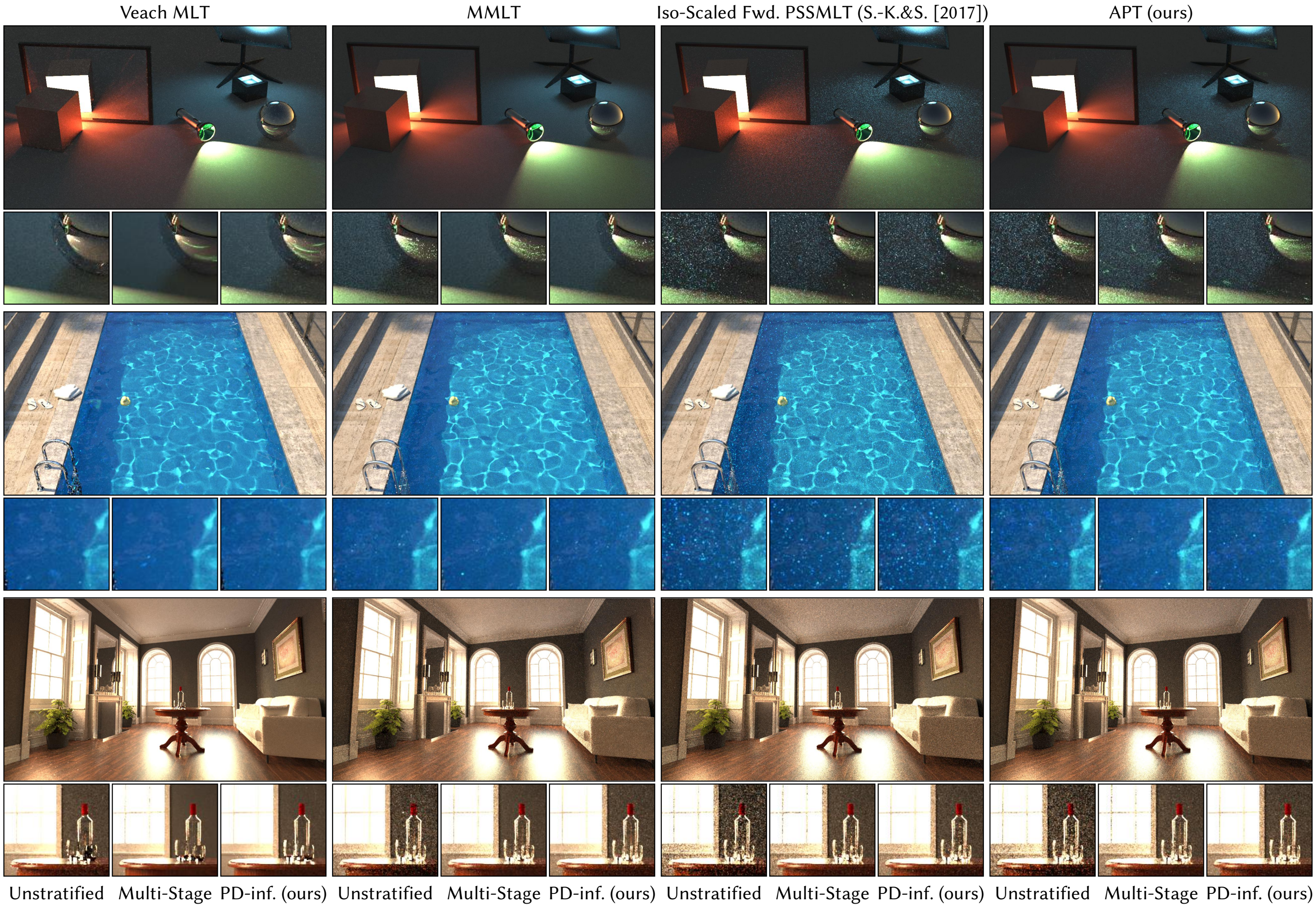


Fig. 5. Equal-time comparisons (50s on 64 cores) of our combined method APT (PT large step & adaptive forward path sampling) with and without stratification, to Veach MLT, MMLT, and unidirectional PSSMLT with isotropic adaptive sampling [Szirmay-Kalos and Szécsi 2017]. All methods can benefit from stratification. Multi-Stage MLT benefits particularly from the good noise reduction of Veach MLT and MMLT, and can outperform our stratification (in still images). Among the unidirectional techniques, our anisotropic APT adapts more consistently to difficult light transport than isotropically rescaled proposal distributions.

equal-time renderings for our simplified stratification scheme (Sect. 6.5) with respective comparisons to multi-stage MLT [Hoberock and Hart 2010]. Fig. 6 evaluates the effectiveness of stratification, both using multi-stage MLT and our technique, in terms of effective mutations per pixel. We chose to show the simplified, derivative-free variants of our algorithms, as they usually give similar or better results due to their increased numerical robustness and lower computational overhead (around 5% faster). The alternative variants using precise path differentials (Sect. 6.4) are compared in our supplementary materials. Particularly for renderings with higher contrast, noise levels are drastically reduced, as for the flashlight and the living room scene. Notably, this is orthogonal to the chosen mutations and similar improvement can be seen for all shown variations of the MLT algorithm (Veach MLT, MMLT, rescaled PSSMLT, and our APT). For the living room scene, Fig. 7 shows that this also manifests in faster convergence.

The computational overhead of our stratification over unstratified rendering depends on the chosen mutation strategies: The bi-directional mutation in Veach MLT generates many proposals with small modifications, thus the impact is higher, reducing mutation counts by about 25%. For our APT, which performs fewer larger steps, the overhead is around 5%.

Comparison to multi-stage MLT. As is to be expected by design, we do not generally outperform multi-stage MLT in all areas: However, we do achieve significant variance reduction in similar cases, while improving on the aspects of control and predictability: Relying only on local information, our approach is less susceptible to artifacts due to correlated outliers and high-frequency brightness changes (see Fig. 8). This is particularly noticeable for multiple reruns in motion (see Sect. 8.3), where it shows that comparably lower quality improvements in still frames are compensated for by increased robustness. Sect. 8.4 details limitations and trade-offs. Our approach allows for standard progressive rendering with live observation of convergence, whereas multi-stage MLT requires the budgeting and balancing of lower-resolution 'learning' phases in advance.

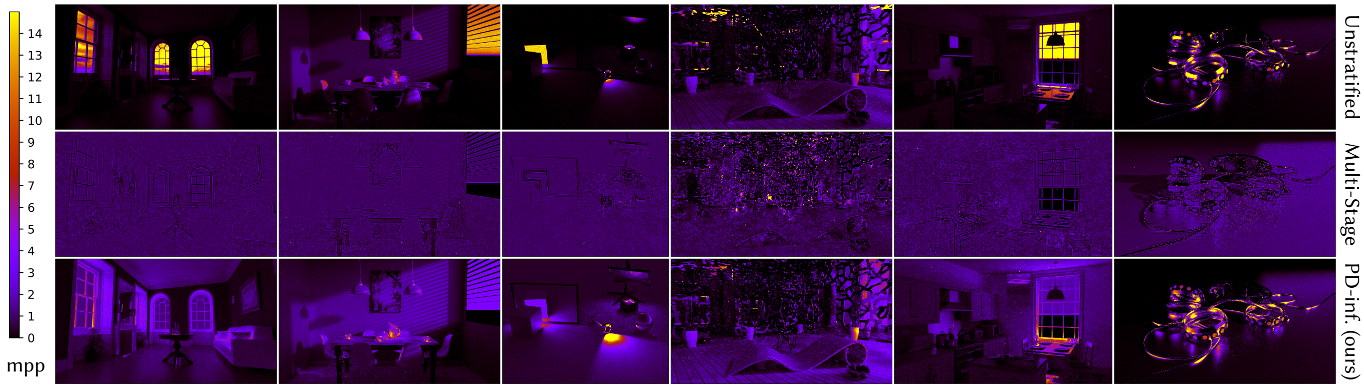


Fig. 6. False color visualizations of the relative distribution of mutations (samples) per pixel for unstratified, multi-stage MLT, and path differential-informed (PD-inf.) stratified MCMC (all shown for our APT technique). Multi-Stage MLT samples the most uniformly, achieving almost perfect stratification. Our method is a trade-off, stratifying much less strictly, for simplicity and reduced correlation. Still, it recovers samples from more trivial bright image areas.

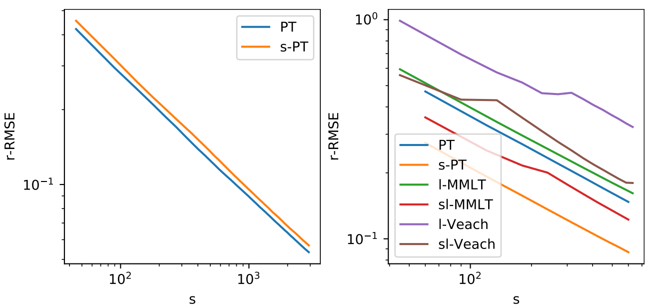


Fig. 7. *Left:* Relative RMSE over time for our RJ-based path tracing mutation (50 minutes on a 128 core machine, on the complex light transport in the torus scene) to verify convergence. The scene is explored well by unstratified MCMC, therefore stratification incurs a small slowdown. *Right:* Convergence of different MLT large steps compared with (s-*) and without stratification in the living room scene. Stratification noticeably reduces the error. Multi-Stage MLT cannot render progressively, we provide relative RMSE values for still images in the supplementary materials.

8.2 Adaptive Forward Path Tracing

Fig. 5 compares our forward path sampling algorithm to challenging competitors: The compared Veach MLT and MMLT both build on bi-directional path construction techniques, which in many configurations of complex light transport work well with less informed sampling decisions, where bi-directional sampling fits these configurations more naturally. This immediately becomes apparent in the comparison to unidirectional PSSMLT (Fig. 9), which uses the same heuristics as the MMLT implementation, but fails to adapt without bi-directional sampling techniques. Yet, the heuristic of bi-directional sampling may also cause inconsistent quality: connectable (directly visible) surfaces may be improved, whereas other reflected or refracted counterparts may remain unaffected. This particularly shows in Fig. 1, where our adaptive forward path tracing outperforms standard bi-directional techniques and approaches the quality of HSLT, which uses a complex iterative solver, specialized to explore paths in these specific scenarios. Like for the stratification, we show results for the simplified adaptive algorithm (Listing 1), as it is up to 10% faster and more robust due to its simplicity. The supplementary materials provide results for both variants.

Optimized Proposal Distributions. To evaluate our optimization of proposal distribution shapes (Sect. 7.1), we compare to previous adaptive unidirectional path sampling in MLT. We show standard PSSMLT [Kelemen et al. 2002] and importance-based isotropic scaling (akin to [Szirmay-Kalos and Szécsi 2017], where an isotropic proposal distribution in primary sample space is scaled to a standard deviation that is inversely proportional to the throughput of the current path). Fig. 9 demonstrates that both heuristics are useful for some light transport configurations, but fail to consistently adapt to more complex high-dimensional light distributions. This is expected, as from the optimization in Sect. 7.1.2, it follows that good proposal distributions are often anisotropic in primary sample space. It is confirmed in that our proposal distributions adapt more consistently to both difficult caustics and multi-bounce indirect diffuse light.

8.3 Temporal Stability

Temporal stability and filterability of unconverged MCMC results are important open problems on the way towards practical MCMC rendering. To this end, we compared our method with other techniques by rendering multiple frames with independent random seeding: In our supplementary materials, we provide looping animations for our adaptive path tracing, MMLT, and unidirectional PSSMLT. In motion, for the dining room, torus, and pool scene, our results come closer to uncorrelated noise than the other methods, while the algorithm still adapts to complex light transport. In the museum scene, where most light comes from caustics, results are more correlated and temporally unstable. As a proof of concept for filtering, we also applied standard DBOR reweighting [Zirr et al. 2018] to the animation frames: With DBOR, our results are more stable than the competition even in the museum scene, at the cost of DBOR applying stronger downweighting and thus bias. Our results indicate that optimizing for minimal correlation is a promising avenue in adaptive sampling, however, more work is needed on reliable filtering of correlated outliers. Comparing multi-stage MLT in motion, it shows strong additional correlation artifacts in the pool

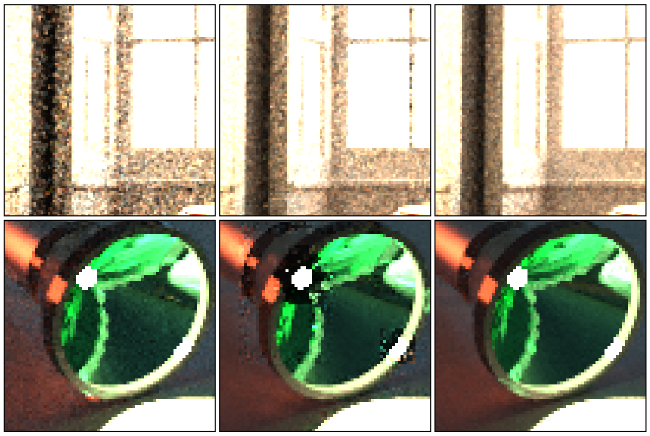


Fig. 8. Stratification by multi-stage MLT struggles with high-frequency brightness changes due to the recursive upsampling (shown for base 4). This manifests in correlation artifacts for correlated techniques like Veach MLT (bottom), or increased noise in our less correlated adaptive path tracer (top).

and the museum scenes, where outliers in low stages misguide the chain in high stages. Our method does not introduce such artifacts.

8.4 Limitations and Trade-Offs of Local Information

While multi-stage MLT leverages aggregate information of many samples throughout the recursion, our approaches can only rely on the local information of individual paths. This limits improvements in some scenarios (e.g. in the staircase and the door scenes in the supplementary materials), particularly when discontinuous visibility counters the locally informed predictions. In contrast, aggregate information can lead to improved decisions, but also to misguiding when confronted with variance, causing correlated artifacts (and flashing under motion). For paths with many diffuse interactions, our variance bounds become increasingly conservative. This complies with Ineq. (15), which may be less tight for smaller subfilters, required for such long scattering paths, so that they can be represented by only the local environment of one sample. Our supplementary results provide insight into the impact of variance bounds per path length, for differing scattering profiles.

8.5 Variance Bounding

Fig. 3 evaluates the practical reliability of our variance bounding scheme, focussing on the splat weights that result from downweighting using $\tilde{r}_X$, computed for each path. These results were computed without MLT, using a plain path tracer with next event estimation. Suspecting that the ratios $\tilde{r}_X$ can be used as conservative estimates of the ratios in density-based outlier reweighting (DBOR) [Zirr et al. 2018], we applied the same reweighting $\frac{1}{\tilde{r}_X} \frac{N}{\kappa}$ (N is the sample count, κ the variance control threshold), and found that it indeed resulted in bounded-variance images that resemble the ones obtained using DBOR (please refer to the supplementary materials for more results). We find this interesting as an instance where our local variance estimates indeed conservatively replace information that is otherwise aggregated from many samples. This could be useful

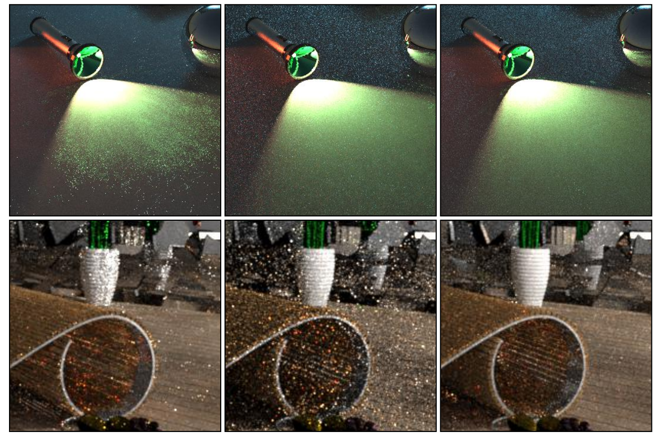


Fig. 9. Adaptive proposal sampling with differently shaped distributions: Unidirectional PSSMLT samples proposals from a fixed isotropic distribution in primary sample space, struggeling to follow the global radiance distribution in all cases. Isotropic rescaling according to importance (akin to [Szirmay-Kalos and Szécsi 2017]) can improve proposals, but does not generalize to all cases, breaking some cases handled by PSSMLT. We optimize the shape of *anisotropic* proposal distributions in line with our variance bounding theory, combining the strengths of both across many cases.

for path guiding techniques, where DBOR estimates have already been applied [Simon et al. 2018] to identify guide paths.

8.6 Path Tracing Embedded as a Large Step

Fig. 4 analyzes the effectiveness of our path tracing (PT) large step mutation (Sect. 7.2) that implements forward PT in single-path state space MLT. Comparing large steps in isolation, our results are better than for the standard bi-directional mutation in many cases, particularly where NEE works well. Even compared to MMLT as an efficient PSSMLT method, our mutation holds up. If we add small steps, our PT large step combined with the classic Veach MLT small steps helps these enter important regions of path space faster. We attribute this to our construction of many paths at once, decreasing the chances of failing to find new candidates in a large step, and thus wasting fewer computations while reducing correlation artifacts.

9 CONCLUSIONS

With the approaches discussed in our paper, we attempt a first step towards simpler and more controllable MCMC algorithms. Our results indicate that targeted control over the amount of correlation in adaptive sampling is a promising avenue for future investigation, in order to reach robust, temporally stable methods in all kinds of correlated sampling techniques. We hope that our analytical approach to stratification and adaptive sampling in the context to MCMC proves a useful tool of variance estimation also in techniques unrelated to MCMC methods, as e.g. in path guiding approaches. Furthermore, we hope that our adaptation of a stock MC path tracer to the MCMC framework serves as an example for more general bridging of MC and MCMC methods. A key challenge to be addressed for wider adoption of correlated techniques in the future remains the filtering and reconstruction of partly converged results. Our approach to minimizing proposal correlation could be a stepping stone, potentially augmented by additional means of correlation tracking.

ACKNOWLEDGMENTS

We thank Christoph Peters, Johannes Hanika, and Lorenzo Tessari for constructive input. For tested scenes, we thank "Jay-Artist" for *kitchen*, Olesya Jakob for *torus*, Johannes Hanika for *necklace*, "Wig42" for *dining room* and *living room* and *staircase*, "NewSee2l035" for *modern stairs*, Ondřej Karlík for *pool*, "NovaZeeke" for *classroom*, and Benedikt Bitterli for *door* and for hosting many of these scenes.

REFERENCES

Sheldon Axler. 2020. *Measure, Integration & Real Analysis.* Springer Nature, page 197.

Kavita Bala, Julie Dorsey, and Seth Teller. 1999. Radiance interpolants for accelerated bounded-error ray tracing. *ACM Transactions on Graphics* 18, 3 (1999), 213–256.

Laurent Belcour, Cyril Soler, Kartic Subr, Nicolas Holzschuch, and Fredo Durand. 2013. 5D Covariance Tracing for Efficient Defocus and Motion Blur. *ACM Transactions on Graphics* 32, 3 (2013), 31:1–31:18.

Laurent Belcour, Ling-Qi Yan, Ravi Ramamoorthi, and Derek Nowrouzezahrai. 2017. Antialiasing complex global illumination effects in path-space. *ACM Transactions on Graphics* 36, 1 (2017), 9:1–9:13.

Benedikt Bitterli. 2015. Informed Choices in Primary Sample Space. M.Sc. Thesis, ETH.

Benedikt Bitterli, Wenzel Jakob, Jan Novák, and Wojciech Jarosz. 2017. Reversible Jump Metropolis Light Transport Using Inverse Mappings. *ACM TOG* 37, 1 (2017), 1:1ff.

Benedikt Bitterli and Wojciech Jarosz. 2019. Selectively Metropolised Monte Carlo light transport simulation. *ACM Trans. Graph.* 38, 6, Article 153 (2019).

Min-Te Chao. 1982. A general purpose unequal probability sampling plan. *Biometrika* 69, 3 (1982), 653–656.

Per Christensen, Julian Fong, Jonathan Shade, Wayne Wooten, Brenden Schubert, Andrew Kensler, Stephen Friedman, Charlie Kilpatrick, Cliff Ramshaw, Marc Bannister, et al. 2018. RenderMan: An Advanced Path-Tracing Architecture for Movie Rendering. *ACM Transactions on Graphics* 37, 3 (2018), 30:1–30:21.

David Cline, Justin Talbot, and Parris K. Egbert. 2005. Energy Redistribution Path Tracing. *ACM Trans. Graph. (Proc. SIGGRAPH)* 24, 3 (2005), 1186–1195.

Radu V. Craiu and Christiane Lemieux. 2007. Acceleration of the Multiple-Try Metropolis algorithm using antithetic and stratified sampling. *Statistics and Computing* 17, 2 (2007), 109–120.

Christopher DeCoro, Tim Weyrich, and Szymon Rusinkiewicz. 2010. Density-based Outlier Rejection in Monte Carlo Rendering. *Computer Graphics Forum (Proc. Pacific Graphics)* 29, 7 (Sept. 2010), 2119–2125.

Oskar Elek, Pablo Bauszat, Tobias Ritschel, Marcus Magnor, and Hans-Peter Seidel. 2014. Spectral Ray Differentials. *Computer Graphics Forum* 33, 4 (2014), 113–122.

Luca Fascione, Johannes Hanika, Mark Leone, Marc Droske, Jorge Schwarzhaupt, Tomáš Davidovič, Andrea Weidlich, and Johannes Meng. 2018. Manuka: A Batch-Shading Architecture for Spectral Path Tracing in Movie Production. *ACM Transactions on Graphics* 37, 3 (2018), 31:1–31:18.

Charles J. Geyer. 1992. Practical Markov Chain Monte Carlo. *Statist. Sci.* 7, 4 (1992), 473–483.

Charles J. Geyer. 2003. The Metropolis-Hastings-Green algorithm. http://www.stat.umn.edu/geyer/f05/8931/bmhg.pdf

Peter J. Green. 1995. Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika* 82, 4 (1995), 711–732.

Peter J Green. 2003. Trans-dimensional markov chain monte carlo. *Oxford Statistical Science Series* (2003), 179–198.

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 Transactions on Graphics* 36, 4 (2016), 4:1–4:13.

Adrien Gruson, Rex West, and Toshiya Hachisuka. 2020. Stratified Markov Chain Monte Carlo Light Transport. *Computer Graphics Forum* 39, 2 (2020), 351–362.

Toshiya Hachisuka and Henrik Wann Jensen. 2011. Robust Adaptive Photon Tracing Using Photon Path Visibility. *ACM Trans. Graph.* 30, 5 (2011), 114:1–114:11.

Toshiya Hachisuka, Anton S. Kaplanyan, and Carsten Dachsbacher. 2014. Multiplexed Metropolis light transport. *ACM Trans. Graph.* 33, 4 (2014), 100:1–100:10.

Johannes Hanika, Anton Kaplanyan, and Carsten Dachsbacher. 2015. Improved Half Vector Space Light Transport. *Computer Graphics Forum* 34, 4 (2015), 65–74.

Jared Hoberock and John C Hart. 2010. Arbitrary importance functions for Metropolis light transport. In *Computer Graphics Forum*, Vol. 29. 1993–2003.

Wenzel Jakob. 2010. Mitsuba Renderer. 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 Trans. Graph. (Proc. SIGGRAPH)* 31, 4 (2012), 58:1–58:13.

Anton S Kaplanyan, Johannes Hanika, and Carsten Dachsbacher. 2014. The natural-constraint representation of the path space for efficient light transport simulation. *ACM Trans. Graph. (Proc. SIGGRAPH)* 33, 4 (2014), 102:1–102:13.

Csaba Kelemen, László Szirmay-Kalos, György Antal, and Ferenc Csonka. 2002. A simple and robust mutation strategy for the Metropolis light transport algorithm. *Computer Graphics Forum* 21, 3 (2002), 531–540.

Jaakko Lehtinen, Tero Karras, Samuli Laine, Miika Aittala, Frédo Durand, and Timo Aila. 2013. Gradient-domain Metropolis light transport. *ACM Trans. Graph. (Proc. SIGGRAPH)* 32, 4 (2013), 95:1–95:12.

Tzu-Mao Li, Miika Aittala, Frédo Durand, and Jaakko Lehtinen. 2018. Differentiable Monte Carlo Ray Tracing through Edge Sampling. *ACM TOG* 37, 6 (2018), 222:1ff.

Tzu-Mao Li, Jaakko Lehtinen, Ravi Ramamoorthi, Wenzel Jakob, and Frédo Durand. 2015. Anisotropic Gaussian mutations for Metropolis light transport through Hessian-Hamiltonian dynamics. *ACM Transactions on Graphics* 34, 6, Article 209 (2015).

Yufei Li, Yang Liu, and Wenping Wang. 2014. Planar hexagonal meshing for architecture. *IEEE Transactions on Visualization and Computer Graphics* 21, 1 (2014), 95–106.

Jun S. Liu, Faming Liang, and Wing Hung Wong. 2000. The Multiple-Try method and local optimization in Metropolis sampling. *J. Amer. Statist. Assoc.* 95, 449 (2000), 121–134.

Fujun Luan, Shuang Zhao, Kavita Bala, and Ioannis Gkioulekas. 2020. Langevin Monte Carlo Rendering with Gradient-based Adaptation. *ACM TOG* 39, 4 (2020), 140:1ff.

Luca Martino and Francisco Louzada. 2017. Issues in the Multiple Try Metropolis mixing. *Computational Statistics* 32, 1 (2017), 239–252.

Thomas Müller, Markus Gross, and Jan Novák. 2017. Practical Path Guiding for Efficient Light-Transport Simulation. *Computer Graphics Forum* 36, 4 (2017), 91–100.

Hisanari Otsu, Johannes Hanika, Toshiya Hachisuka, and Carsten Dachsbacher. 2018. Geometry-Aware Metropolis Light Transport. *ACM Trans. Graph. (Proc. SIGGRAPH)* 37, 6 (2018), 278:1–278:11.

Hisanari Otsu, Anton S Kaplanyan, Johannes Hanika, Carsten Dachsbacher, and Toshiya Hachisuka. 2017. Fusing state spaces for markov chain Monte Carlo rendering. *ACM Transactions on Graphics* 36, 4 (2017), 74:1–74:10.

Silvia Pandolfi, Francesco Bartolucci, and Nial Friel. 2010. A generalization of the Multiple-Try Metropolis algorithm for Bayesian estimation and model selection. In *Proc. of International Conference on Artificial Intelligence and Statistics.* 581–588.

Jacopo Pantaleoni. 2017. Charted Metropolis Light Transport. *ACM Transactions on Graphics* 36, 4, Article 75 (July 2017).

Fabrice Rousselle, Claude Knaus, and Matthias Zwicker. 2012. Adaptive Rendering with Non-Local Means Filtering. *ACM Trans. Graph.* 31, 6, Article 195 (Nov. 2012).

Christoph Schied, Christoph Peters, and Carsten Dachsbacher. 2018. Gradient Estimation for Real-Time Adaptive Temporal Filtering. *Proc. ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games* (2018), 24:1–24:16.

Jorge Schwarzhaupt, Henrik Wann Jensen, and Wojciech Jarosz. 2012. Practical Hessian-based error control for irradiance caching. *ACM Trans. Graph.* 31, 6 (2012), 193:1ff.

Benjamin Segovia, Jean-Claude Iehl, and Bernard Péroche. 2007a. Coherent Metropolis light transport with Multiple-Try mutations. In *LIRIS Technical Report, 2007.*

Benjamin Segovia, Jean Claude Iehl, and Bernard Péroche. 2007b. Metropolis Instant Radiosity. In *Computer Graphics Forum*, Vol. 26. 425–434.

Mikio Shinya, Tomoichi Takahashi, and Seiichiro Naito. 1987. Principles and Applications of Pencil Tracing. In *Computer Graphics (Proc. SIGGRAPH).* 45–54.

Florian Simon, Alisa Jung, Johannes Hanika, and Carsten Dachsbacher. 2018. Selective guided sampling with complete light transport paths. *ACM Trans. Graph. (Proc. SIGGRAPH Asia)* 37, 6, Article 223 (Dec. 2018).

Frank Suykens and Yves D. Willems. 2001. Path differentials and applications. In *Proc. Eurographics Symposium on Rendering.* 257–268.

László Szirmay-Kalos and László Szécsi. 2017. Improved stratification for metropolis light transport. *Computers & Graphics* 68 (2017), 11–20.

Eric Veach. 1998. *Robust Monte Carlo methods for light transport simulation.* Ph.D. Dissertation. Stanford University. AAI9837162.

Eric Veach and Leonidas J. Guibas. 1995. Optimally combining sampling techniques for Monte Carlo rendering. *Proc. ACM SIGGRAPH* (1995), 419–428.

Eric Veach and Leonidas J. Guibas. 1997. Metropolis Light Transport. *Proc. ACM SIGGRAPH* (1997), 65–76.

Jiří Vorba, Ondřej Karlík, Martin Šik, Tobias Ritschel, and Jaroslav Křivánek. 2014. On-line learning of parametric mixture models for light transport simulation. *ACM Trans. Graph. (Proc. SIGGRAPH)* 33, 4 (2014), 101:1–101:11.

Martin Šik and Jaroslav Křivánek. 2018. Survey of Markov Chain Monte Carlo Methods in Light Transport Simulation. *IEEE Transactions on Visualization and Computer Graphics* 26 (2018), 1821–1840.

Martin Šik, Hisanari Otsu, Toshiya Hachisuka, and Jaroslav Křivánek. 2016. Robust Light Transport Simulation via Metropolised Bidirectional Estimators. *ACM Transactions on Graphics* 35, 6 (2016), 245:1–245:12.

Tobias Zirr, Johannes Hanika, and Carsten Dachsbacher. 2018. Re-Weighting Firefly Samples for Improved Finite-Sample Monte Carlo Estimates. In *Computer Graphics Forum*, Vol. 37. 410–421.

Károly Zsolnai and László Szirmay-Kalos. 2013. Automatic Parameter Control for Metropolis light transport. In *Eurographics Short Papers.* 53–56.

A DERIVATIONS OF VARIANCE BOUNDS

Based on the (marginal) contributions $F(\mathbf{x}_k^\perp, \mathbf{o}_k)$ and $F_d(\mathbf{x}_k^\perp, \mathbf{o}_k)$ of endpoints to subfilters $g(\mathrm{U})$ and $g^d(\mathrm{U})$, which exist as Radon-Nikodym derivatives in Prop. 6, we bound r_g by $\tilde{r}_g$ as follows:

$$r_g \overset{(15)}{=} \frac{\int_{\mathcal{M}\times\Omega} F_d(\mathbf{x}_k^\perp, \mathbf{o}_k) L_e(\mathbf{x}_k^\perp, \mathbf{o}_k)\, \mathrm{d}(\mathbf{x}_k^\perp, \mathbf{o}_k)}{\int_{\mathcal{M}\times\Omega} F(\mathbf{x}_k^\perp, \mathbf{o}_k) L_e(\mathbf{x}_k^\perp, \mathbf{o}_k)\, \mathrm{d}(\mathbf{x}_k^\perp, \mathbf{o}_k)} \tag{30}$$

$$= \frac{\int_{\mathcal{M}\times\Omega} \frac{F_d(\mathbf{x}_k^\perp, \mathbf{o}_k)}{F(\mathbf{x}_k^\perp, \mathbf{o}_k)} F(\mathbf{x}_k^\perp, \mathbf{o}_k) L_e(\mathbf{x}_k^\perp, \mathbf{o}_k)\, \mathrm{d}(\mathbf{x}_k^\perp, \mathbf{o}_k)}{\int_{\mathcal{M}\times\Omega} F(\mathbf{x}_k^\perp, \mathbf{o}_k) L_e(\mathbf{x}_k^\perp, \mathbf{o}_k)\, \mathrm{d}(\mathbf{x}_k^\perp, \mathbf{o}_k)} \tag{31}$$

$$\leq \frac{\cancel{\int F(\mathbf{x}_k^\perp, \mathbf{o}_k) L_e(\mathbf{x}_k^\perp, \mathbf{o}_k)\, \mathrm{d}(\mathbf{x}_k^\perp, \mathbf{o}_k)}}{\cancel{\int F(\mathbf{x}_k^\perp, \mathbf{o}_k) L_e(\mathbf{x}_k^\perp, \mathbf{o}_k)\, \mathrm{d}(\mathbf{x}_k^\perp, \mathbf{o}_k)}} \max_{\mathbf{x}_k, \mathbf{o}_k} \frac{F_d(\mathbf{x}_k^\perp, \mathbf{o}_k)}{F(\mathbf{x}_k^\perp, \mathbf{o}_k^\perp)} =: \tilde{r}_g. \tag{32}$$

Note that the exact size of subfilters is irrelevant to ρ_g and r_g w.r.t. isotropic scaling, allowing an arbitrarily localized analysis (the probability volumes of the subfilters cancel out). Anisotropic scaling matters, but since it artificially increases or decreases the amount of (non-)determinism of compared transports, we choose isotropic subfilters for stratification. Sect. 7 makes use of the additional freedom to optimize the anisotropic shape of proposal densities.

A.1 Projection Variance Bounding

The approach taken in Eq. (32) can also be applied to the projected endpoint densities $F^{\perp\mathcal{L}}, L_e^{\perp\mathcal{L}}$ and $F_d^{\perp\mathcal{L}}, L_{e|d}^{\perp\mathcal{L}}$, which exist as Radon-Nikodym derivatives in Sect. 6.3:

$$r_g = \frac{\int_{\mathcal{L}} F_d^{\perp\mathcal{L}}(\mathbf{x}_k, \mathbf{o}_k) L_{e|d}^{\perp\mathcal{L}}(\mathbf{x}_k, \mathbf{o}_k)\, \mathrm{d}\pi_{\mathcal{L}}(\mathbf{x}_k, \mathbf{o}_k)}{\int_{\mathcal{L}} F^{\perp\mathcal{L}}(\mathbf{x}_k, \mathbf{o}_k) L_e^{\perp\mathcal{L}}(\mathbf{x}_k, \mathbf{o}_k)\, \mathrm{d}\pi_{\mathcal{L}}(\mathbf{x}_k, \mathbf{o}_k)} \tag{33}$$

$$\overset{(32)}{\leq} \max_{(\mathbf{x}_k, \mathbf{o}_k)\in\mathcal{L}} \frac{F_d^{\perp\mathcal{L}}(\mathbf{x}_k, \mathbf{o}_k) L_{e|d}^{\perp\mathcal{L}}(\mathbf{x}_k, \mathbf{o}_k)}{F^{\perp\mathcal{L}}(\mathbf{x}_k, \mathbf{o}_k) L_e^{\perp\mathcal{L}}(\mathbf{x}_k, \mathbf{o}_k)} =: \tilde{r}_g^{\mathcal{L}}. \tag{34}$$

Here, the $L_e^{\perp\mathcal{L}}$ and $L_{e|d}^{\perp\mathcal{L}}$ effectively are "precomputed" means, such that the product integrals give the correct contribution: Practically, they are difficult to obtain, but choosing $\mathcal{L}$ such that $L_e(\mathbf{x}_k, \mathbf{o}_k)$ is sufficiently smooth on the preimages $\pi_{\mathcal{L}}{}^{-1}$ of points in $\mathcal{L}$ (as e.g. a point light radiates in many directions, or a directional light covers an area), we expect mean emission in deterministic and non-deterministic subfilters to converge in the limit (for small subfilters).

To find directions of projections that result in the tightest variance bounds, we find the projection giving the lowest upper bound for the light transport within a subfilter $g(\mathrm{U})$, and thus the highest variance for any large sample value:

$$\int_{\mathcal{L}^\varepsilon} F^{\perp\mathcal{L}}(\mathbf{x}_k, \mathbf{o}_k) L_e^{\perp\mathcal{L}}(\mathbf{x}_k, \mathbf{o}_k)\, \mathrm{d}\pi_{\mathcal{L}}(\mathbf{x}_k, \mathbf{o}_k) \tag{35}$$

$$\overset{5,\text{ Sect. }6.3}{\leq} \max_{(\mathbf{x}_k, \mathbf{o}_k)} F^{\perp\mathcal{L}}(\mathbf{x}_k, \mathbf{o}_k) \max_{(\mathbf{x}_k, \mathbf{o}_k)} L_e(\mathbf{x}_k^\perp, \mathbf{o}_k) \int_{\mathcal{L}^\varepsilon} \mathrm{d}\pi_{\mathcal{L}}(\mathbf{x}_k, \mathbf{o}_k). \tag{36}$$

Thus, the lowest worst-case bound (in environments $\mathcal{L}^\varepsilon$ on emitters) is found in the projection plane where $F^{\perp\mathcal{L}}$ spreads the widest.

A.2 Separability of Involved Terms

For convolution, the effects of individual scattering interactions on the endpoint distribution need to be (locally) separable. In our implementation, to avoid relying on differentiable rendering architecture, we compute all Jacobians w.r.t. halfvectors instead (see Sect. 6). These fit common peaky BSDFs well, i.e. keeping halfvectors constant is close to keeping respective $\mathbf{u}_i$ constant. For halfvector representations, Kaplanyan et al. [2014] indeed show a separation of BSDF interactions in their results.

The albedo factor $A(\mathrm{U})$ and its correlations in high-dimensional subfilters $g(\mathrm{U})$ are generally unknown. We assume these factors to be either sufficiently smooth or independently distributed. Effectively, we neglect the change in average albedo $\bar{A}$ from $\bar{A}_d$ to $\bar{A}$, going from deterministic to actual light transport within a local subfilter $g(\mathrm{U})$ with marginal endpoint contribution functions F and F_d:

$$\frac{F_d(\mathbf{x}_k^\perp, \mathbf{o}_k)|_{A(\mathrm{X})=1}}{F(\mathbf{x}_k^\perp, \mathbf{o}_k)|_{A(\mathrm{X})=1}} = \frac{F_d(\mathbf{x}_k^\perp, \mathbf{o}_k)}{F(\mathbf{x}_k^\perp, \mathbf{o}_k)} \overbrace{\frac{F(\mathbf{x}_k^\perp, \mathbf{o}_k)}{F(\mathbf{x}_k^\perp, \mathbf{o}_k)|_{A=1}}}^{:=\bar{A}(\mathbf{x}_k^\perp, \mathbf{o}_k)} \overbrace{\frac{F_d(\mathbf{x}_k^\perp, \mathbf{o}_k)|_{A=1}}{F_d(\mathbf{x}_k^\perp, \mathbf{o}_k)}}^{:=\bar{A}_d(\mathbf{x}_k^\perp, \mathbf{o}_k)^{-1}}. \tag{37}$$

A.3 Bounding the ratio ρ_g

We can verify the inequality (15) as follows:

$$\rho_g = \frac{\|h_j^{|d_i} f\|_1}{\|h_j f\|_1} = \frac{\int_{\mathcal{U}} (h_j(\mathrm{U}) - g(\mathrm{U})) f(\mathrm{U})\, \mathrm{dU} + \int_{\mathcal{U}} g^d(\mathrm{U}) f(\mathrm{U})\, \mathrm{dU}}{\int_{\mathcal{U}} (h_j(\mathrm{U}) - g(\mathrm{U})) f(\mathrm{U})\, \mathrm{dU} + \int_{\mathcal{U}} g(\mathrm{U}) f(\mathrm{U})\, \mathrm{dU}} \tag{38}$$

$$=: \frac{A + B_d}{A + B} =: l(A). \tag{39}$$

The maximal ratio $\rho_g > 1$, i.e. when $B_d \geq B$ is reached for $A = 0$:

$$\max l(A) = \max \frac{A + B_d}{A + B}, \tag{40}$$

$$l'(A) = \frac{1}{A + B} - \frac{A + B_d}{(A + B)^2} = \frac{B - B_d}{(A + B)^2} \leq 0, \tag{41}$$

$$l'(A) \leq 0 \wedge A \geq 0 \Rightarrow A = 0. \tag{42}$$

B FULL DERIVATION OF PATH DIFFERENTIALS

In our experiments with precise differentials, we use a linear-time algorithm (w.r.t. path length) with constant storage to compute endpoint distributions. The following derivations work for a forward path sampling strategy $\mathrm{X}(\mathrm{U})$ constructing paths X from random variables U, i.e. by sampling each next vertex $\mathbf{x}_i$ using random variables $\mathbf{u}_{i-1}$. In practice, we approximate all derivatives w.r.t. U using halfvector space, so to not require differentiability in renderers.

B.1 Positional and Directional Path Derivatives

We use the ideas of Manifold Exploration [Jakob and Marschner 2012]: The endpoint distribution of our locally deterministic light transport is defined by keeping the random variables for all inner vertices fixed, thus they become constraints for a mapping from pixels to the endpoint, characterized by the Implicit Function Theorem. To allow this in the general case, we first have to ensure invertibility of the constraint Jacobian by dimension matching: For example, if the strategy $\mathrm{X}(\mathrm{U})$ uses auxiliary random variables, such as for layer selection, the respective derivatives w.r.t. U will not be meaningful. However, there is a pushforward measure $\mu(A) := \int_A \mathrm{dU} \circ \mathrm{X}^{-1}$ for each path space measure $\nu(A) := \int_A \mathrm{dX}$ that effectively marginalizes out all auxiliary random variables that ultimately result in the same path $X(U)$. Like Otsu et al. [2017], we can construct a space $\mathcal{A}$ very similar to $\mathcal{U}$ by inversion of a CDF corresponding to $\mu(A)$, resulting in a dimension-matched mapping $X(A)$. This mapping effectively compresses auxiliary dimensions into other dimensions (e.g. by allocating a subset of values to each discrete decision for encoding). We thus assume $X(U)$ or an analogous $X(A)$ to be invertible w.l.o.g., and write the Jacobi matrix of $X^{-1}(X)$ as the tridiagonal matrix:

$$\frac{\partial\left(\mathbf{u}_{0\ldots k-1}\right)}{\partial\left(\mathbf{x}_{1\ldots k}\right)}=\begin{pmatrix} \frac{\partial \mathbf{u}_0}{\partial \mathbf{x}_1} & 0 & \cdots & \\ \frac{\partial \mathbf{u}_1}{\partial \mathbf{x}_1} & \frac{\partial \mathbf{u}_1}{\partial \mathbf{x}_2} & 0 & \cdots \\ \ddots & \ddots & \ddots & 0 \\ \ldots 0 & \frac{\partial \mathbf{u}_{k-1}}{\partial \mathbf{x}_{k-2}} & \frac{\partial \mathbf{u}_{k-1}}{\partial \mathbf{x}_{k-1}} & \frac{\partial \mathbf{u}_{k-1}}{\partial \mathbf{x}_k} \end{pmatrix}. \tag{43}$$

We follow the convention of labeling the entries of each row with $\mathbf{C}_i$, $\mathbf{B}_i$, $\mathbf{A}_i$ from the camera ($\mathbf{x}_1$) to the emitter ($\mathbf{x}_k$), i.e. $\mathbf{A}_i = \partial\mathbf{u}_i/\partial\mathbf{x}_{i+1}$ (w.r.t. vertex of incident light), $\mathbf{B}_i = \partial\mathbf{u}_i/\partial\mathbf{x}_i$, and $\mathbf{C}_i = \partial\mathbf{u}_i/\partial\mathbf{x}_{i-1}$ (w.r.t. vertex of exitant light). By the Inverse Function Theorem, we obtain the partial derivatives of $X(U)$ solving the linear system of constraints corresponding to the tridiagonal block matrix. Effectively, we propagate perturbations of random variables through subsequent interactions to the endpoint by the recurrence:

$$\frac{\mathrm{d}\mathbf{x}_{j+1}}{\mathrm{d}\mathbf{u}_i}=-\mathbf{A}_j^{-1}\left(\mathbf{C}_j\frac{\mathrm{d}\mathbf{x}_{j-1}}{\mathrm{d}\mathbf{u}_i}+\mathbf{B}_j\frac{\mathrm{d}\mathbf{x}_j}{\mathrm{d}\mathbf{u}_i}\right),\quad \frac{\mathrm{d}\mathbf{x}_{i+1}}{\mathrm{d}\mathbf{u}_i}=\mathbf{A}_i^{-1},\frac{\mathrm{d}\mathbf{x}_i}{\mathrm{d}\mathbf{u}_i}=0. \tag{44}$$

Note that aperture sampling appears as $\mathbf{A}_0 = \partial\mathbf{x}_1/\partial\mathbf{u}_0$. The positional derivatives of the last two vertices allow us to directly compute the change of outgoing directions at vertex $\mathbf{x}_k$:

$$\frac{\mathrm{d}\mathbf{o}_k}{\mathrm{d}\mathbf{u}_j}=\frac{\mathrm{d}\mathbf{o}_k}{\mathrm{d}\mathbf{x}_{k-1}}\frac{\mathrm{d}\mathbf{x}_{k-1}}{\mathrm{d}\mathbf{u}_j}+\frac{\mathrm{d}\mathbf{o}_k}{\mathrm{d}\mathbf{x}_k}\frac{\mathrm{d}\mathbf{x}_k}{\mathrm{d}\mathbf{u}_j}. \tag{45}$$

B.2 Incremental Path Differentials

Computing the recurrent Eq. (44) for every vertex individually has cost $O(k^2)$ for path length k. Instead, we can unroll the path differential $\Sigma[F(\mathbf{x}_k,\mathbf{o}_k)]$ in Eq. (18) to solve all recurrences in parallel. Incremental construction from the camera requires storing three 2×2-matrices $[\Phi_i^2]$, $[\Phi_{i-1}^2]$, and $[\Phi_i\Phi_{i-1}^T]$:

$$\begin{aligned}
[\Phi_{i+1}\Phi_i^T] &= -\mathbf{A}_i^{-1}\left(\mathbf{C}_i[\Phi_i\Phi_{i-1}^T]^T+\mathbf{B}_i[\Phi_i^2]\right),\\
[\Phi_{i+1}^2] &= \left(\mathbf{A}_i^{-1}\left(\mathbf{C}_i[\Phi_{i-1}^2]+\mathbf{B}_i[\Phi_i\Phi_{i-1}^T]\right)\mathbf{C}_i^T-[\Phi_{i+1}\Phi_i^T]\mathbf{B}_i^T\right)\mathbf{A}_i^{-T}.
\end{aligned}$$

For the emitter directions, we can directly re-use our computations:

$$\begin{aligned}
[\Theta_k^2] = &\frac{\mathrm{d}\mathbf{o}_k}{\mathrm{d}\mathbf{x}_{k-1}}[\Phi_{k-1}^2]\frac{\mathrm{d}\mathbf{o}_k}{\mathrm{d}\mathbf{x}_{k-1}}^T+\frac{\mathrm{d}\mathbf{o}_k}{\mathrm{d}\mathbf{x}_{k-1}}[\Phi_k\Phi_{k-1}^T]^T\frac{\mathrm{d}\mathbf{o}_k}{\mathrm{d}\mathbf{x}_k}^T\\
&+\frac{\mathrm{d}\mathbf{o}_k}{\mathrm{d}\mathbf{x}_k}[\Phi_k\Phi_{k-1}^T]\frac{\mathrm{d}\mathbf{o}_k}{\mathrm{d}\mathbf{x}_{k-1}}^T+\frac{\mathrm{d}\mathbf{o}_k}{\mathrm{d}\mathbf{x}_k}[\Phi_k^2]\frac{\mathrm{d}\mathbf{o}_k}{\mathrm{d}\mathbf{x}_k}^T.
\end{aligned}$$

The path differential $\Sigma[F(\mathbf{x}_k,\mathbf{o}_k)]$ is fully defined by the block triple:

$$\left(\quad [\Phi_k^2],\quad [\Theta_k^2],\quad [\Phi_k\Phi_{k-1}^T]\frac{\mathrm{d}\mathbf{o}_k}{\mathrm{d}\mathbf{x}_{k-1}}^T+[\Phi_k^2]\frac{\mathrm{d}\mathbf{o}_k}{\mathrm{d}\mathbf{x}_k}^T\quad\right).$$

B.3 PSS Differential Approximation using Halfvectors

To avoid requiring differentiability in the renderer, in Sect. 6.4, we use the *Natural Constraint Space Representation* [Hanika et al. 2015; Jakob and Marschner 2012; Kaplanyan et al. 2014] to obtain a good approximation of BSDF differentials at inner vertices, which is easily computed. In the halfvector parameterization, fixed points fit the constraints of microfacet BSDFs well. We preserve density:

$$\left|\frac{\mathrm{d}\mathbf{x}_k}{\mathrm{d}\mathbf{u}_i}\right|=\left|\frac{\mathrm{d}\mathbf{x}_k}{\mathrm{d}\mathbf{i}_i}\right|\left|\frac{\mathrm{d}\mathbf{i}_i}{\mathrm{d}\mathbf{u}_i}\right|=\left|\frac{\mathrm{d}\mathbf{x}_k}{\mathrm{d}\mathbf{i}_i}\right|\Big/p(\mathbf{i}_i)=\left|\frac{\mathrm{d}\mathbf{x}_k}{\mathrm{d}\mathbf{h}_i}\right|\Big/\left(p(\mathbf{i}_i)\left|\frac{\mathrm{d}\mathbf{i}_i}{\mathrm{d}\mathbf{h}_i}\right|\right).$$

C DERIVATIONS BASED ON PATH DIFFERENTIALS

In the following, we provide our derivations that allow us to skip the computation of actual path differentials, and we derive the optimal shape of adaptive proposal distributions. For this, let $\mathbf{P}_i$ be the change of sampled directions $\mathbf{i}_i$ (resp. vertex position $\mathbf{x}_1$ for aperture sampling, $i=0$) w.r.t. random variables $\mathbf{u}_i$. The determinants of $\mathbf{P}_i$ are equal to the path tracing PDFs $p(\mathbf{i}_i)$ (resp. $p(\mathbf{x}_1)$). We do not need to know the full matrix $\mathbf{P}_i$ in our final expressions, removing requirement of differentiability in the renderer's sampling procedures. Let $\mathbf{T}_i$ be transformations from the respective spaces of $\mathbf{P}_i$ to a planar 2-manifold in the 4D space of endpoints. E.g., for the projection $\mathbf{V}$ in Sect. 6.4, we have $\mathbf{T}_i\mathbf{P}_i = \mathbf{V}\frac{\mathrm{d}(\mathbf{x}_k^\perp,\mathbf{o}_k)}{\mathrm{d}\mathbf{u}_i}$ and write $\tilde{r}_X$ as:

$$\tilde{r}_X \overset{(20)}{=} \frac{\left|\sum_{i=0}^{k-1}\left(\mathbf{T}_i\mathbf{P}_i\right)^2\right|^{1/2}}{\left|\left(\mathbf{T}_1\mathbf{P}_1\right)^2\right|^{1/2}}=\left|\sum_{i=0}^{k-1}\left(\mathbf{P}_1^{-1}\mathbf{T}_1^{-1}\mathbf{T}_i\mathbf{P}_i\right)^2\right|^{1/2}, \tag{46}$$

assuming that $\mathbf{u}_1$ is responsible for pixel sampling.

C.1 Path Differentials on Linearized Geometry

We can skip computing path differentials at the cost of slightly approximate estimates: We perform our simplifications using a Rayleigh quotient representation of $\tilde{r}_X$ as written in Eq. (46). Let $\mathbf{v}_1, \mathbf{v}_2$ be eigenvectors of $\sum_{i=0}^{k-1}\left(\mathbf{P}_1^{-1}\mathbf{T}_1^{-1}\mathbf{T}_i\mathbf{P}_i\right)^2$, then we can write the determinant as $\tilde{r}_X = \sqrt{\tilde{r}_X(\mathbf{v}_1)\tilde{r}_X(\mathbf{v}_2)}$ in terms of the quotient $\tilde{r}_X(\mathbf{v})$:

$$\tilde{r}_X(\mathbf{v})=\frac{\mathbf{v}^T\left(\sum_{i=0}^{k-1}\left(\mathbf{P}_1^{-1}\mathbf{T}_1^{-1}\mathbf{T}_i\mathbf{P}_i\right)^2\right)\mathbf{v}}{\mathbf{v}^T\,\mathbf{v}}\le\sum_{i=0}^{k-1}\max_{\mathbf{u}}\frac{\mathbf{u}^T\left(\mathbf{P}_i\right)^2\mathbf{u}}{\mathbf{u}^T\left(\mathbf{T}_i^{-1}\,\mathbf{T}_1\,\mathbf{P}_1\right)^2\mathbf{u}}. \tag{47}$$

The approximation factor comes from the fact that we bound our simplified ratios $\tilde{r}_X$ for axis-aligned projections $\mathbf{V}$ only, as the primary goal of the projection is to reduce the dimensionality while handling cases of degenerate emitters, which we mostly expect in either the positional or directional domain, or an orthogonal combination thereof. To evaluate robustness in arbitrary cases, we ran global optimizations on positive symmetric 4×4-matrices, choosing projection vectors $\mathbf{v}_1, \mathbf{v}_2$ only from two fixed orthogonal subspaces, and experimentally found the peak approximation factor to be 4.

We simplify Eq. (47) for transformations T_i to endpoint positions ($\partial\mathbf{x}_k^\perp/\partial\mathbf{u}_i$) and directions ($\partial\mathbf{o}_k/\partial\mathbf{u}_i$) separately. For directions, noting $\mathbf{o}_i = -\mathbf{i}_{i-1}$, and by the chain rule $\frac{\mathrm{d}\mathbf{i}_i}{\mathrm{d}\mathbf{i}_{i-1}} = \frac{\mathrm{d}\mathbf{i}_i}{\mathrm{d}\mathbf{o}_i}\frac{\mathrm{d}\mathbf{o}_i}{\mathrm{d}\mathbf{i}_{i-1}}$, we find:

$$\begin{aligned} \text{For } \mathbf{T}_i &= \frac{\mathrm{d}\mathbf{o}_k}{\mathrm{d}\mathbf{i}_i} = \frac{\mathrm{d}\mathbf{o}_k}{\mathrm{d}\mathbf{i}_{k-1}} \left(\frac{\mathrm{d}\mathbf{i}_{k-1}}{\mathrm{d}\mathbf{i}_{k-2}} \cdots \frac{\mathrm{d}\mathbf{i}_{i+1}}{\mathrm{d}\mathbf{i}_i} \right) : \\ \mathbf{T}_i^{-1}\,\mathbf{T}_1 &= \left(\frac{\mathrm{d}\mathbf{i}_i}{\mathrm{d}\mathbf{i}_{i-1}} \cdots \frac{\mathrm{d}\mathbf{i}_2}{\mathrm{d}\mathbf{i}_1} \right) = \frac{\mathrm{d}\mathbf{i}_i}{\mathrm{d}\mathbf{i}_1}. \end{aligned} \tag{48}$$

On the linearized geometry introduced in Sect. 6.5, the resulting term merely has to account for angle compressions along the path due to Snell's law. Reflections do not apply any scaling to the ratio.

For transformations $\mathbf{T}_i$ to endpoint positions, we look at 2D projections of vertex positions to surfaces: Let $\mathbf{x}_i^{\perp\mathbf{o}}$ be the vertex $\mathbf{x}_k$ projected orthogonally to the outgoing edge $\mathbf{o}_k$. We recall $\mathbf{o}_k = -\mathbf{i}_{k-1}$, apply the chain rule to $\frac{\mathrm{d}\mathbf{x}_k^{\perp\mathbf{o}}}{\mathrm{d}\mathbf{x}_{k-1}^{\perp\mathbf{o}}} = \frac{\mathrm{d}\mathbf{x}_k^{\perp\mathbf{o}}}{\mathrm{d}\mathbf{x}_{k-1}^{\perp\mathbf{i}}}\frac{\mathrm{d}\mathbf{x}_{k-1}^{\perp\mathbf{i}}}{\mathrm{d}\mathbf{x}_{k-1}^{\perp\mathbf{o}}}$, and find:

$$\begin{aligned} \text{For } \mathbf{T}_i = \frac{\mathrm{d}\mathbf{x}_k^{\perp\mathbf{o}}}{\mathrm{d}\mathbf{i}_i} &= \frac{\mathrm{d}\mathbf{x}_k^{\perp\mathbf{o}}}{\mathrm{d}\mathbf{i}_{k-1}} \frac{\mathrm{d}\mathbf{i}_{k-1}}{\mathrm{d}\mathbf{i}_i} + \frac{\mathrm{d}\mathbf{x}_k^{\perp\mathbf{o}}}{\mathrm{d}\mathbf{x}_{k-1}^{\perp\mathbf{o}}} \left(\frac{\mathrm{d}\mathbf{x}_{k-1}^{\perp\mathbf{o}}}{\mathrm{d}\mathbf{i}_{k-2}} \frac{\mathrm{d}\mathbf{i}_{k-2}}{\mathrm{d}\mathbf{i}_i} + \cdots \right) : \\ \mathbf{u}^T \Big(\mathbf{T}_i^{-1} \underbrace{\mathbf{T}_1}_{\mathbf{T}_1 = \left(\frac{\mathrm{d}\mathbf{x}_k^{\perp\mathbf{o}}}{\mathrm{d}\mathbf{i}_i} \frac{\mathrm{d}\mathbf{i}_i}{\mathrm{d}\mathbf{i}_1} + \frac{\mathrm{d}\mathbf{x}_k^{\perp\mathbf{o}}}{\mathrm{d}\mathbf{x}_i^{\perp\mathbf{i}}} \frac{\mathrm{d}\mathbf{x}_i^{\perp\mathbf{i}}}{\mathrm{d}\mathbf{x}_i^{\perp\mathbf{o}}} \frac{\mathrm{d}\mathbf{x}_i^{\perp\mathbf{o}}}{\mathrm{d}\mathbf{i}_1} \right) = \left(\mathbf{T}_i \frac{\mathrm{d}\mathbf{i}_i}{\mathrm{d}\mathbf{i}_1} + \cdots \right)} \mathbf{P}_1 \Big)^2 \mathbf{u} &\geq \mathbf{u}^T \left(\frac{\mathrm{d}\mathbf{i}_i}{\mathrm{d}\mathbf{i}_1} \mathbf{P}_1 \right)^2 \mathbf{u}. \end{aligned} \tag{49}$$

The inequality requires that the discarded mixed terms of our $\mathbf{T}_1^2$-decomposition are positive semi-definite, so that the corresponding inner products are non-negative. Due to our linearized geometry, we do indeed expect that $\mathbf{i}_1$ moves $\mathbf{i}_i$ and $\mathbf{x}_i^{\perp\mathbf{i}}$ in the same directions, and that this property is preserved after transformation to $\mathbf{x}_k^{\perp\mathbf{o}}$ and $\mathbf{o}_k$.

Thus, we find that so far the terms in our projected ratios can be bounded in the same way for endpoint positions and directions. A special consideration is required for camera movement/aperture sampling, as on our linerized geometry it only affects endpoint positions. We exploit that T_1 and T_0 share many chained terms:

$$\begin{aligned} \text{For } \mathbf{T}_0 &= \frac{\mathrm{d}\mathbf{x}_k^{\perp\mathbf{o}}}{\mathrm{d}\mathbf{x}_1^{\perp\mathbf{i}}} = \left(\frac{\mathrm{d}\mathbf{x}_k^{\perp\mathbf{o}}}{\mathrm{d}\mathbf{x}_{k-1}^{\perp\mathbf{i}}} \frac{\mathrm{d}\mathbf{x}_{k-1}^{\perp\mathbf{i}}}{\mathrm{d}\mathbf{x}_{k-1}^{\perp\mathbf{o}}} \cdots \right) \frac{\mathrm{d}\mathbf{x}_2^{\perp\mathbf{o}}}{\mathrm{d}\mathbf{x}_1^{\perp\mathbf{i}}} : \\ \mathbf{u}^T \left(\mathbf{T}_0^{-1}\,\mathbf{T}_1\,\mathbf{P}_1 \right)^2 \mathbf{u} &= \mathbf{u}^T \left[\mathbf{T}_0^{-1} \left(\mathbf{T}_0 \left(\frac{\mathrm{d}\mathbf{x}_2^{\perp\mathbf{o}}}{\mathrm{d}\mathbf{x}_1^{\perp\mathbf{i}}} \right)^{-1} \frac{\mathrm{d}\mathbf{x}_2^{\perp\mathbf{o}}}{\mathrm{d}\mathbf{i}_1} + \cdots \right) \mathbf{P}_1 \right]^2 \mathbf{u}, \end{aligned} \tag{50}$$

leaving $\frac{\mathrm{d}\mathbf{x}_2^{\perp\mathbf{o}}}{\mathrm{d}\mathbf{i}_1}$ as the only non-identity term in an inequality analogous to Eq. (49). All the terms remaining in Eqs. (48,49,50) end up in the final bounding Eq. (21) in Sect. 6.5. Dropping the $\max_{\mathbf{u}}$ inherited from Eq. (47), we effectively reshape subfilters such that the simplified terms corresponding to each $\mathbf{u}_i$ become isotropic in the respective projection.

C.2 Precise Optimal Proposal Distributions

Additional degrees of freedom are added to the optimization problem in Eq. (23) if we solve it for the ratio bound $\tilde{r}_X$ in Eq. (20) using precise path differentials: Not only can the proposal sampling density be chosen individually for each interaction, but also the (anisotropic) shape of the Gaussian for each $\mathbf{u}_i$ becomes a variable in the optimization. For this purpose, we adapt Eq. (46) by setting $\mathbf{T}_i$ to the full transformation from $\mathbf{u}_i$ to projected endpoints, directly. Our optimization then finds new $\mathbf{P}_i$ in Eq. (46) that fulfill the constraint $\tilde{r}_X D_{g_i} \stackrel{!}{=} R$: Let $\mathbf{v}_1, \mathbf{v}_2$ be the eigenvectors from Sect. 6.4, and

Listing 3. Pseudocode for our small step with differentials. Assignment with ':' indicates tuple distribution across multiple variables. Vertices $\mathbf{x}_0$ and $\mathbf{x}_1$ refer to the same camera vertex with positional and directional characteristics, respectively. The last vertex $\mathbf{x}_k$ refers to a point on an emitter.

in Differentials $\frac{\mathrm{d}\mathbf{u}_i}{\mathrm{d}\mathbf{x}_j}$ of a plain path tracer for a current X(U)
out proposal distribution around U, i.e. $\mathcal{N}(\mathbf{u}_i, \texttt{covariances}_i)$

```
// emitter and last interaction differentials
TridiagDifferential D_emitter(I), D_connect(0)
// per-vertex impact on endpoint distribution
Matrix2x2[k] Δ, Θ
Matrix2x2    ΣΔ^2, ΣΘ^2, ΣΔΘ^T := (0, 0, 0)
// for current path X of length k in MCMC
for i = k-1 .. 0:
  Matrix2x2 A, B, C := (du_i/dx_j | j = i+1 .. i-1)
  Δ_i, Θ_i := proj_area_and_arc(A^-1, D_connect, D_emitter)
  ΣΔ^2, ΣΘ^2, ΣΔΘ^T :+= (Δ_iΔ_i^T, Θ_iΘ_i^T, Δ_iΘ_i^T)
  D_emitter.chain_interaction(C, B, A^-1)
  D_connect.chain_or_init_interaction(C, B, A^-1)

// principal components of convolution
Vector4[2] projV = eigenbase_asc((ΣΔ^2, ΣΔΘ^T; (ΣΔΘ^T)^T, ΣΘ^2))[2:4]
// projected per-vertex footprints
Matrix2x2[k] Φ
float    [k] footprints
for i = 0 .. k-1:
  Φ_i = (projV_0  projV_1)^T (Δ_i; Θ_i)
  footprints_i = |Φ_i|

// constrained optimization
target_diff = footprints_pixel-i / √(pixel count)
bool [k] active = [ x_i is non-specular for i = 0..k-1 ]
float[k] amplifiers
do:
  target_footprint = target_diff / count(active)
  retry = false
  for i = 0 .. k-1:
    if active_i:
      amplifiers_i = footprints_i / target_footprint
      if not amplifiers_i > 1:
        active_i = false
        target_diff -= footprints_i
        retry = true
while retry

// per-vertex adaptive proposal distribution
Matrix2x2[k] covariances
for i = 0 .. k-1:
  // note: inverse eigen-sqrt more robust, see impl.
  covariances_i = Φ_i^-1 Φ_i^-T / amplifiers_i if active_i else I
```

let $T_i = \mathbf{V}\frac{\mathrm{d}(\mathbf{x}_k^{\perp}, \mathbf{o}_k)}{\mathrm{d}\mathbf{u}_i}$. We intend to maximize the probability volume covered by the proposal subfilter $g(\mathrm{U})$. To understand the influence of its anisotropic shape, we apply the Minkowski determinant inequality to Eq.(46):

$$\tilde{r}_X = \left| \sum_{i=0}^{k-1} \left(\mathbf{P}_1^{-1} \mathbf{T}_1^{-1} \mathbf{T}_i \mathbf{P}_i \right)^2 \right|^{1/2} \geq \sum_{i=0}^{k-1} \left| \left(\mathbf{P}_1^{-1} \mathbf{T}_1^{-1} \mathbf{T}_i \mathbf{P}_i \right)^2 \right|^{1/2}. \tag{51}$$

If all matrix terms are isotropic, i.e. $\left(\mathbf{P}_1^{-1} \mathbf{T}_1^{-1} \mathbf{T}_i \mathbf{P}_i \right)^2 = \mathbf{I} \left| \mathbf{P}_1^{-1} \mathbf{T}_1^{-1} \mathbf{T}_i \mathbf{P}_i \right|$, the inequality becomes an equality. The result does not change if the same anisotropic rescaling is applied to all endpoint projections on both sides, i.e. as long as all projections keep their anisotropies aligned. If we change the anisotropy of individual $\mathbf{P}_i$, however, bigger $\tilde{r}_\mathrm{X}$ have to be expected, requiring more contraction in some of the $\mathbf{P}_i$. This in turn would decrease the probability volume of $g(\mathrm{U})$, leading to less optimal exploration.

Following these observations, we choose the anisotropy of each $\mathbf{P}_i$ such that the resulting endpoint projections $\mathbf{T}_i\mathbf{P}_i$ are isotropic. This allows the resulting algorithm in Listing 3 to optimize the distribution solely based on the determinants of each projection (see line 20). Note that the proposal distributions for each interaction may still be anisotropic, as they have to counter any anisotropic rescaling applied by the geometry of the subpath connecting to the endpoint. Compared to the algorithm in Listing 1, the respective anisotropic covariance matrices (line 43) potentially allow faster path space exploration in some directions while respecting the variance bound.

In order to compute path differentials and subsequently projections $\mathbf{T}_i\mathbf{P}_i$ for each interaction, the algorithm computes the required derivatives in parallel: Starting at the emitter, the chain rule is applied successively in lines 7-12, effectively keeping track of $\mathrm{d}\mathbf{x}_k/\mathrm{d}\mathbf{x}_i$ (line 11) and $\mathrm{d}\mathbf{x}_{k-1}/\mathrm{d}\mathbf{x}_i$ (line 12) by solving the tridiagonal constraint system backwards (please refer to the implementation for details). Line 9 computes the projections on endpoint positions and directions, for the former we apply $\mathrm{d}\mathbf{x}_k^\perp/\mathrm{d}\mathbf{x}_k$, for the latter we compute Eq. (45). Line 15 determines the planar projection from four to two dimensions by an eigendecomposition of the path differential.

D CANDIDATES AND MIS IN THE RJ FRAMEWORK

In the RJ framework [Geyer 2003; Green 1995, 2003], proposal construction must conform with a self-inverse mapping $g(\mathrm{X}, \mathrm{U})$ from current states (paths) X and auxiliary random variables U to proposal states Y and matching auxiliary variables. Then, the acceptance ratio is determined by the Jacobian determinant of g and the ratio of the target function evaluated for the proposal path and source path.

g can only be self-inverse if input and output dimensions match: RJ relies on *dimension matching*, such that for each current path length k, auxiliary random variables augment the corresponding space of dimensionality k to a fixed combined dimensionality K, i.e. each path of length k is complemented by $K-k$ random input variables. Note that this is still very flexible, as we can choose any sufficiently large K to accommodate the required random variables for all proposals given *any* bounded path length.

For our large step mutation based on path tracing, we first append all random variables $\mathrm{U_Y}$ that are potentially required by our path tracer to independently sample any *collection* of proposal paths Y (constructing candidates using BSDF sampling and NEE at every interaction). We then emulate Multiple Try Metropolis sampling by adding another random variable C for candidate selection: Depending on this value, our mapping g will output different candidates from the collection. Finally, we need to ensure a correct self-inverse mapping for candidates of the same length, but sampled using different techniques (i.e. BSDF sampling or NEE). For disambiguation, we add another auxiliary variable T that will encode the technique.

So far, we have only concatenated a fixed (albeit arbitrarily large) number of random variables. We yet have to perform the dimension matching, which we do by complementing each current path X with yet to be determined auxiliary variables $\mathrm{U_X^+}$ such that the dimensionality of X and $\mathrm{U_X^+}$ add up to the dimensionality of $\mathrm{U_Y}$. Now, to define our mapping, let $\mathrm{X}(\mathrm{U}, C)$ be our path tracing strategy that samples candidate paths using the random variables U, to then select and output one candidate depending on the auxiliary random variable C. For self-invertibility, we require that:

$$g(\mathrm{X}, \mathrm{U_X^+}, \mathrm{U_Y}, C, T) = \left(\mathrm{Y} = \mathrm{X}(\mathrm{U_Y}, C), \mathrm{U_Y^+}, \mathrm{U_X}, C', T'\right), \tag{52}$$

$$\text{s. t. } g(\mathrm{Y}, \mathrm{U_Y^+}, \mathrm{U_X}, C', T') \stackrel{!}{=} \left(\mathrm{X} = \mathrm{X}(\mathrm{U_X}, C'), \mathrm{U_X^+}, \mathrm{U_Y}, C, T\right). \tag{53}$$

The RJ framework does not require us to actually compute all auxiliary outputs of the mapping g, as long as it fulfills these requirements and we can compute its Jacobian determinant. In order to reduce variance using MIS (as also recommended for MT [Martino and Louzada 2017]), our goal is to incorporate the MIS weights $w_t(\mathrm{X})$ of a path tracer with BSDF sampling ($t = 0$) and NEE ($t = 1$). We therefore select candidates proportionally to their MIS'ed contribution $W_t(\mathrm{X}) = w_t(\mathrm{X}) f(\mathrm{X})/p_t(\mathrm{X})$. The tricky part is the self-invertibility: g needs to deconstruct the current path X into random variables $\mathrm{U_X}, C'$ that reconstruct X on reverse application. Reconstructing $\mathrm{U_X}$ uniquely is possible if we know the technique t that is supposed to sample X, because we can store additional information in $\mathrm{U_X^+}$, as e.g. for layer selection in previous work [Bitterli et al. 2017]. To encode and know t for each Y, we map T' to $[0, w_0(\mathrm{Y})]$ for $t = 0$ and $[w_1(\mathrm{Y}), 1]$, otherwise. However, we also need to select X again among all candidates constructed by $\mathrm{U_X}$ on reversal: For this, we encode the candidate selection by mapping C to C' such that C' always lies in the interval of a CDF corresponding to the PMF $p_t(\mathrm{X}|\mathrm{U_X}) = W_t(\mathrm{X})/(\sum_{\mathrm{X}' \in \mathrm{X}(\mathrm{U_X}, [0,1])} \sum_{t'=0}^{1} W_{t'}(\mathrm{X}'))$, which chooses among all candidates constructed from $\mathrm{U_X}$ proportionally to their respective MIS'ed contributions $W_j(\mathrm{X})$. Now, the only thing left to make g self-inverse, is to reverse both encodings on the respective input variables. This results in the ratios:

$$\frac{\mathrm{d}T'}{\mathrm{d}T} = \frac{w_t(\mathrm{Y})}{w_t(\mathrm{X})}, \quad \frac{\mathrm{d}C'}{\mathrm{d}C} = \frac{p_t(\mathrm{X}|\mathrm{U_X})}{p_t(\mathrm{Y}|\mathrm{U_Y})}. \tag{54}$$

We can reduce the Jacobian of g to a block matrix such that: The determinant of $\mathrm{dY}/\mathrm{dU_Y}$ is the reciprocal path tracing PDF $p_t^{-1}(\mathrm{Y})$. The auxiliary variables used for other hidden decisions such as layer selection and the construction of unselected candidate paths are simply passed on to $\mathrm{U_Y^+}$ with a determinant of 1. The deconstruction of $\mathrm{X}, \mathrm{U_X^+}$ into $\mathrm{U_X}$ has the determinant $p_t(\mathrm{X})$ (applying an inverse mapping as in [Bitterli et al. 2017]). The derived ratios in Eq. (54) make up the last two diagonal entries. Thus, the determinant is exactly the ratio of MIS'ed sample values multiplied by the candidate selection weight, analogously to standard Multiple Try Metropolis. In practice, the RJ framework neither requires us to store or know the auxiliary random variables, but to sample them as needed (as in *lazy sampling* for PSSMLT [Kelemen et al. 2002]). Yet, our construction of the mapping g explains why we sample t proportionally to the MIS weights $w_t(\mathrm{X})$ for the reverse candidates, as we have to reverse the encoding of the technique in T' by our chosen MIS weights. Fig. 7 shows that the approach exhibits the expected convergence behavior of an unbiased independent MC estimator.