Institut für Visual Computing &
Human-Centered Technology

Technische Universität Wien

Favoritenstr. 9-11 / E193-02
A-1040 Wien
AUSTRIA

Tel: +43 (1) 58801-193201
Fax: +43 (1) 58801-193209

Institute of Visual Computing &
Human-Centered Technology

TU Wien

email:
technical-report@cg.tuwien.ac.at

other services:
http://www.cg.tuwien.ac.at/

TECHNICAL REPORT

Exploiting A Priori Information for Filtering Monte Carlo Renderings

Hiroyuki Sakai   Károly Zsolnai[1]   Thomas Auzinger   Michael Wimmer

TR-193-02-2021-1
May 2021

Keywords: Monte Carlo noise filtering, photorealistic rendering

Exploiting A Priori Information for Filtering Monte Carlo Renderings

Hiroyuki Sakai*† Károly Zsolnai*‡ Thomas Auzinger§ Michael Wimmer¶

Figure 1: Our noise filtering framework is capable of rapidly denoising images rendered with a low number of samples (2 spp in this example), even in the presence of high-resolution geometry and multiple refractive materials. Our techniques are compatible with all common high-dimensional filters.

Abstract

Monte Carlo rendering techniques are capable of rendering photorealistic images by performing exhaustive stochastic sampling for each pixel but suffer from objectionable noise at low sampling rates. A possible way to mitigate this problem is to perform high-dimensional filtering on the rendered image. The effectiveness of this approach is highly dependent on secondary information regarding both the image structure (given as so-called feature buffers) and the recorded noise. Previous approaches commonly use positions and normals as secondary information, and determine the local noise empirically based on the obtained samples. In this work, we propose to take a priori information in the form of scene material descriptions into consideration. We introduce a noise-estimation technique and a novel feature buffer, based on surface albedos to assist the noise filtering process based on this a priori information. We present an implementation of our method as an extension of the adaptive manifold filter and demonstrate the capabilities of our system by effectively denoising highly undersampled scenes with multiple refractive and reflective materials as well as high-resolution geometry and textures in only a few seconds.

1 Introduction

The synthesis of photorealistic images has always been a major challenge in computer graphics, and rendering methods based on Monte Carlo (MC) integration have proven to be particularly useful for this task. However, this approach is susceptible to objectionable noise inherent to stochastic sampling and consequently, a substantial amount of samples and computation time is necessary to generate satisfactory results. A large body of research exists on how to mitigate MC noise and thus to speed up the rendering process by harnessing the power of filtering algorithms. Rendering methods that take samples in a per-pixel fashion, such as path tracing [Kajiya 1986b] or bidirectional path tracing [Lafortune

*Joint first authors
†e-mail:hiroyuki.sakai@alumni.tuwien.ac.at
‡e-mail:zsolnai@cg.tuwien.ac.at
§e-mail:auzinger@cg.tuwien.ac.at
¶e-mail:wimmer@cg.tuwien.ac.at

and Willems 1993], can profit greatly from the vast amount of noise-reduction algorithms used in the image processing and computer vision communities.

The most promising approaches incorporate auxiliary information from the renderer in the form of *feature buffers* to perform high-dimensional filtering. Common choices for feature buffers are 3D positions, shading normals, texture colors or depths saved during the rendering process. They are suitable for guiding the filter to avoid the oversmoothing of legitimate features, as they generally exhibit a high correlation to the true output image. However, most auxiliary information sources rely on ad hoc decisions whereas we present a novel feature buffer based on a mathematical analysis of the filtering process (Section 4). This analysis enables insights about desirable feature buffer properties, leading to the utilization of surface albedos and a recursive evaluation strategy.

Accurate noise estimates are required for this purpose, but cannot be trivially attained. The reason is that not only the per-pixel radiance samples are subject to noise, but also their variance, and thus sample variance is not a reliable estimate for noise. Still, many previous works rely solely on empirical information obtained by sampling, whereas our approach proposes the inclusion of a priori information (Section 6). This enables more reliable noise estimates at low sampling rates and our approach allows certain aspects of the radiance variance to be precomputed.

Our main contributions are the following:

- A novel **feature buffer** deduced from a mathematical investigation of desirable feature-buffer properties (Section 4). We sample albedo values from the scene materials during path tracing. For each path, we store the albedo value of the first encountered material which is expected to introduce noise. This provides additional edge information for the filtering process, which can be used to significantly reduce the oversmoothing effect of all common high-dimensional filters. We show how to incorporate empirical information of the radiance sample to achieve consistency of our approach. Additionally, different filter capabilities are supported (Section 5).

- A new **noise-estimation technique** that incorporates a priori knowledge on the scene materials (Section 6) and builds the conceptual basis for the aforementioned contributions. We demonstrate that this approach outperforms purely empirical techniques at low sampling rates and generates only a minimal runtime overhead. This is accomplished by means of *variance fusion*, which combines the merits of both our a priori and an empirical variance estimate.

- A practical implementation of the aforementioned methods in a **noise-reduction framework** and its evaluation on a variety of noise filtering scenarios, including MC renderings of a range of different materials with very low per-pixel sample counts in the range of 2-8 samples (Figure 1 and Section 7).

2 Related Work

We present a brief overview of existing noise-removal techniques for photorealistic rendering, mainly focusing on filters using auxiliary information and noise estimation. A general overview can be found in a recent survey on MC noise filtering [Zwicker et al. 2015]. We also provide a brief overview on the basics of high-dimensional filtering in Section 1.1 of the supplementary document.

Early efforts by McCool [1999] led to an approach based on anisotropic diffusion, where filtering is adapted to depth and normal information. More recently, Dammertz et al. [2010] proposed the usage of an edge-avoiding *À-Trous wavelet transform* incorporating positions and normals. Their approach is susceptible to ringing artifacts and the filter support is inherently constrained, making it difficult to filter intricate details adequately. Bauszat et al. [2011] developed an approach where only the indirect illumination component is filtered by the *guided image filter* [He et al. 2010]. This method incorporates depth and normal information. The mentioned approaches solely rely on geometric information for filtering, which ultimately leads to oversmoothed features in complex scenes since geometric information fails to capture many light transport effects, such as varying reflectance behaviors, shadows or caustics.

Sen and Darabi's *random parameter filtering* (RPF) [2012] respects the random numbers used for MC integration to determine filter weights and incorporates positions, shading normals and texture colors. The approach is constrained by high memory consumption as each sample needs to be stored individually. Park et al. [2013] proposed an extension to RPF whose complexity is independent from the number of samples.

Li et al. [2012] and Rousselle et al. [2013] presented techniques where a MC image undergoes *multiple* filtering processes with varying parameters incorporating depth, normal and texture information. The approach by Rousselle et al. additionally utilizes a prefiltered direct illumination buffer and a caustic buffer. Both approaches generate the final image by estimating the error for each pixel in each filtered image using SURE [Stein 1981] and combining the pixels with the lowest estimated noise.

Moon et al. [2013] presented an approach based on non-local means filtering and a *virtual flash image* (VFL), much in the spirit of noise reduction methods for photographs using flash images [Eisemann and Durand 2004; Petschnigg et al. 2004]. The VFL is supposed to be a replacement for all common feature buffer choices, however, it might be devoid of particular edges under specific circumstances. In a later publication, Moon et al. [2014] proposed an approach where locally optimal filtering parameters are automatically chosen by considering reconstruction errors for each pixel. The filtering is accomplished through weighted local regression incorporating depth, normal and texture information. Similarly, Kalantari et al. [2015] propose the utilization of machine-learning techniques to find optimal filtering parameters.

RPF techniques [Sen and Darabi 2012; Park et al. 2013] and SURE-based filtering [Li et al. 2012] require a substantial amount of computation time, i.e., in the order of minutes, to denoise high-resolution (full HD) images. For parallelizable techniques implemented on the GPU [Rousselle et al. 2013; Moon et al. 2014; Kalantari et al. 2015], improved denoising times in the order of tens of seconds were reported. Even faster GPU techniques include the approach by Moon et al. [2013], which requires a few seconds to denoise a 1280×960 image, and the techniques by Dammertz et al. [2010] and Bauszat et al. [2011], which are reported as interactive. They suffer from the aforementioned qualitative limitations, however.

Noise-estimation approaches are generally based on empirically evaluating the MC sample variance [Rushmeier and Ward 1994; McCool 1999; Rousselle et al. 2012; Kalantari and Sen 2013]. An inherent problem of such approaches is that the variance estimates themselves suffer from noise at low sampling rates, just like the MC estimate itself. In this paper, we present a novel alternative to purely empirical techniques to provide reliable estimates at low sample counts. We note that estimating the error introduced by the filtering process [Li et al. 2012; Rousselle et al. 2013; Moon et al. 2014] is orthogonal to estimating the variance of MC samples.

All mentioned techniques, including the state-of-the-art, are primarily concerned with denoising quality with computational efficiency being a secondary goal. In this paper, our foremost concern is the applicability for rendering scenarios, where most of the resources should be available for the light transport simulation, e.g., interactive Monte Carlo rendering. Thus, we provide lightweight techniques that integrate into the path tracing process and cause only a minimal overhead when compared to common high-dimensional filter strategies.

3 Overview

General high-dimensional MC noise filtering takes the per-pixel samples at each image location and its neighborhood as well as secondary information into account and tries to infer from them *i)* how much noise is present at the given location; *ii)* which samples are good estimates for the pixel color; and, finally, *iii)* a filtered color value that exhibits as little noise as possible.

We follow the same scheme and give an overview of our denoising framework in Figure 2: the *filter* takes as input a potentially undersampled and consequently *noisy rendering* from the *renderer* and relies on additional information to evaluate the aforementioned items *i)* and *ii)*. A novel feature buffer augments well-known *noise-free feature buffers* such as depths and normals by material-specific information and determines which samples are suitable candidates for inclusion in the filtering process (see Section 4). Depending on the capabilities of the underlying high-dimensional filter, noise estimates are used to either determine spatially varying kernel sizes or per-pixel *blending* weights (see Section 5). As a basis for these methods, accurate noise estimates are required. For this, we extend conventional methods that empirically determine the *per-tile variance* of the sample in a local tile by a novel *per-pixel BSDF variance estimate* (see Section 6) that takes the properties of the materials along the respective light paths into account, which yields a

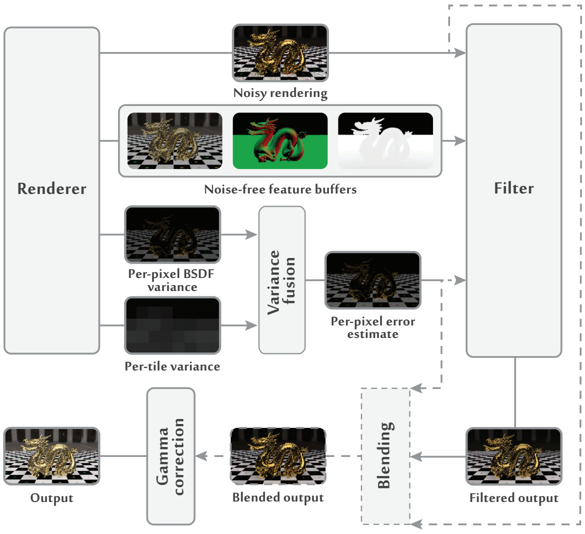
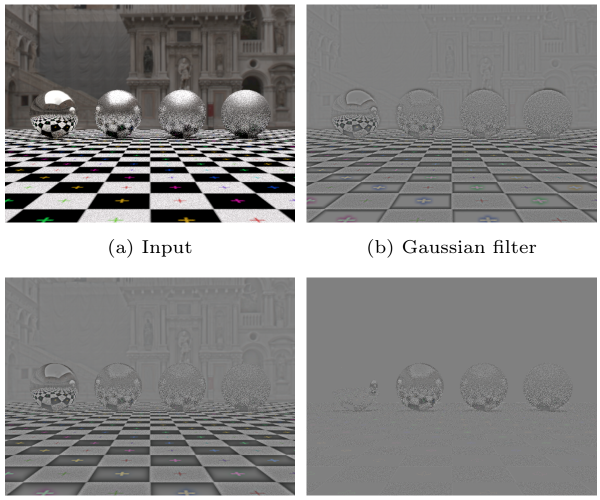


Figure 2: An overview of our noise reduction framework. The renderer generates the noisy rendering which is filtered with respect to the noise-free feature buffers and our novel per-pixel error estimate. The blending stage combines unfiltered and filtered pixels according to the error estimate. It is, however, superfluous if the filter is capable of adaptive kernel sizes.



(a) Input (b) Gaussian filter

(c) [Gastal and Oliveira 2012] (d) Ours

Figure 3: Ideally, the method noise for a particular noise reduction technique should only contain unwanted noise while being devoid of any legitimate image structures. The method noise images above demonstrate that our approach is particularly effective in preserving legitimate image information.

more faithful *per-pixel error estimate*. The gamma-corrected filtered image constitutes our *output*, and an evaluation of our framework is given in Section 7.

4 A Priori Information for Feature Buffers

Most high-dimensional filters rely on feature buffers to guide their filtering behavior. Common choices are 3D positions [Kalantari et al. 2015; Sen and Darabi 2012] or depths, shading normals, and textures [Moon et al. 2014; Rousselle et al. 2013]. Other more complex variants include direct illumination and caustic information [Rousselle et al. 2013], which need to be filtered themselves. This secondary information gives vital cues on where edges in the final images are potentially located (e.g., at object silhouettes) and prevents the filter from smoothing across image regions with significantly different characteristics.

While most of the aforementioned works design such feature buffers in an ad hoc fashion and validate them experimentally, we take the opposite route. In the following section, we mathematically evaluate the denoising capabilities of general feature buffers and derive a description of what constitutes promising candidates. Based on this knowledge, we then introduce the albedo map as a novel feature buffer that also takes material properties along reflected and refracted light paths into account (see Section 4.1).

4.1 Method Noise of Feature Buffers

In this section, we investigate the denoising properties of general feature buffers by looking at the *method noise* q^{F} [Buades et al. 2005] of a filter F applied on an image g given as

$$q^{\mathrm{F}}(\mathbf{x}) = g(\mathbf{x}) - \mathrm{F}g(\mathbf{x}), \qquad (1)$$

where $\mathrm{F}g(\mathbf{x})$ denotes the result of applying the filter F to g at a position $\mathbf{x} \in \mathbb{R}^d$. The method noise describes the local image changes caused by the filtering process. For a perfect (and generally unobtainable) filter, it would contain only the noise that should be removed from the image g. Note that the calculations in this section are valid for arbitrary signals g and conducted in the continuous domain for simplicity.

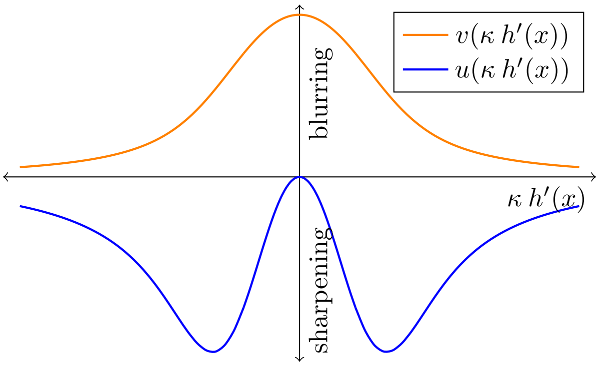


Figure 4: The blurring and sharpening properties of a cross bilateral filter as a function of the gradient magnitudes of the feature buffer given by (3).

For realistic images and filters, it is generally not possible to develop a closed-form expression of the corresponding method noise. However, a description based on a local Taylor-series expansion is sufficient for our purposes and we give as an introductory example the method noise of the well-known Gaussian filter G as [Lindenbaum et al. 1994]

$$q^{\mathrm{G}}(\mathbf{x}) = g(\mathbf{x}) - \mathrm{G}g(\mathbf{x}) = -\sigma^2\,\nabla^2 g(\mathbf{x}) + o(\sigma^2), \quad (2)$$

where σ denotes the width of the Gaussian filter in terms of its standard deviation, and σ is assumed to be small enough. The Gaussian method noise is thus zero on harmonic (i.e., flat) parts of the image and very large near edges or textures, where the Laplacian is expected to be large. This behavior leads to significant blurring of important features, which is evident from the presence of structured image information in the method noise examples in Figure 3.

Method noise of cross bilateral filter. Using a more refined filter definition, our goal in the following is to match the actual noise in an image as faithfully as possible with the filter's method noise. To this end, we develop a feature buffer that shows this desired characteristic when used with a common filtering approach. Like most high-dimensional filtering approaches [He et al. 2010; Petschnigg et al. 2004], we use a cross bilateral filter CBL as the basic filtering methodology and denote its spatial width as τ. For simplicity, we calculate the method noise of the CBL with a feature buffer h in 1D:

$$\begin{aligned} q^{\mathrm{CBL}}(x) &= g(x) - \mathrm{CBL}g(x) \simeq \\ &- \tau^2 \Big(\underbrace{u\big(\kappa\, h'(x)\big)\frac{g'(x)}{h'(x)}h''(x)}_{I} \\ &+ \underbrace{v\big(\kappa\, h'(x)\big)g''(x)}_{II} \Big) + o(\tau^2) \quad (3) \end{aligned}$$

with a constant κ depending on the filter parameters. A derivation of this expression can be found in Appendix A, and an illustration of the function u and v is provided in Figure 4. Several insights into the nature of the sought feature buffer can be obtained from this expression.

Analysis. Similar to the behavior of the Gaussian filter (2), the second term (II) contains the image Laplacian $g''(x)$ (which is the second derivate in 1D). Relevant image information should be preserved by the filtering process, and thus we need to ensure that the term containing $g''(x)$ is suppressed at the corresponding image locations. This can be achieved by a large $|h'(x)|$, which leads to small values of $v(\kappa\, h'(x))$. The contrary also holds true, as regions devoid of relevant image information should be smoothed more prominently, which requires small gradient magnitudes $|h'(x)|$ causing a large $v(\kappa\, h'(x))$.

For vanishing and large feature-buffer gradient magnitudes $|h'(x)|$, the function $u(\kappa\, h'(x))$ in the first term (I) tends to zero, and the corresponding term does not influence the method noise considerably. In between, however, the negative values of u lead to an amplification of the Laplacian $h''(x)$ of the feature buffer, which has a sharpening effect in the corresponding image regions—a characteristic property of advanced filters [Buades et al. 2006]. If h mirrors relevant image features, such a sharpening behavior does not pose a problem per se, but if the feature buffer is contaminated with noise, the strength of sharpening is noisy as well, leading to further noise in the image. This is especially true if the feature buffer matches the noise of the input image, which will lead to $g'(x)/h'(x) \approx 1$.

Requirements for feature buffers. The analysis of the method noise of a CBL implies two requirements for a good feature buffer: *(i)* it should exhibit high gradient magnitudes in the presence of relevant image features, *(ii)* it should be as noise-free as possible to avoid the quality degradation due to the inevitable sharpening behavior of the filter. Additionally, for performance reasons, we add the requirement that *(iii)* it should be possible to evaluate it significantly faster than the rendering step to allow rapid noise reduction.

We note that commonly used feature buffers like positions and normals meet these requirements in general: they exhibit large gradients at legitimate image features such as object silhouettes, they are essentially noise-free and trivially obtained. However, they do not meet *(i)* in all cases: in particular, image features that are entirely due to the materials in the scene are not taken into account at all. If a material is specular, for example, even noise-free scene features visible in the reflection will be treated as noise, since they do not appear in the feature buffer (see Figure 12 for an example).

4.2 Albedo Map

Numerous works have incorporated texture colors in feature buffers [Moon et al. 2014; Rousselle et al. 2013]. However, this cannot account for appearance changes due to the glossyness of the material, for example (i.e., a transition from a glossy to a diffuse part of a surface). Our first contribution to an improved feature buffer is therefore to introduce the *albedo* as a new attribute for feature buffers. Albedo is defined as the hemispherical-directional reflectance ρ_{hd}:

$$\rho_{\mathrm{hd}}(\omega) = \int_{\mathcal{H}^2} f(\mathbf{p}, \omega, \omega')(\omega' \cdot \mathbf{n})\,\mathrm{d}\omega'. \tag{4}$$

This quantity can be intuitively understood as the reflected radiance in direction ω when the surface point $\mathbf{p}$ with BRDF f is uniformly lit across its hemisphere $\mathcal{H}^2$. It depends only on the material model, which allows it to be precomputed.

A feature buffer consisting of the albedo of the surface hit at a pixel (evaluated in the direction towards the viewpoint) adheres to all three aforementioned requirements: it is essentially noise-free *(ii)*, can be efficiently looked up during the rendering process *(iii)*, and it exhibits high gradient magnitudes at the boundaries between different reflectance behaviors, which constitutes legitimate image information with a high probability *(i)*.

Essentially, we use the albedo values as coarse but noiseless approximations for the actual radiance. In this regard, we generalize the albedo ρ_{hd} by including the emission term L_{e}, i.e.,

$$\rho_{\mathrm{hd}}(\omega) = L_{\mathrm{e}}(\mathbf{p}, \omega_o) + \int_{\mathcal{H}^2} f(\mathbf{p}, \omega, \omega')(\omega' \cdot \mathbf{n})\,\mathrm{d}\omega' \tag{5}$$

and we will refer to this modified definition as *albedo* for the remainder of this work.

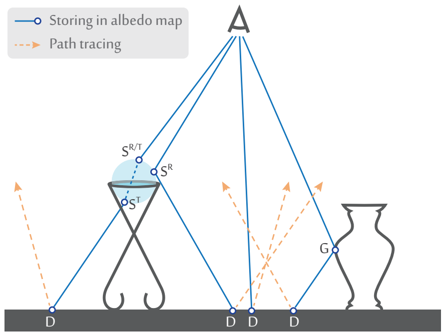


Figure 5: Construction of the albedo map. The estimated error introduced by a given bounce determines the weighting of subsequent bounces to minimize noise.

4.3 Albedo Map Usage

While the described albedo map is more general than, for example, a simple texture, it is still limited to information gathered from the first intersection of viewing rays. However, for specular surfaces, the path tracer follows specular reflections without significant noise, therefore the albedo of the specular surface itself is not relevant.

Recursive evaluation. Instead, we introduce the notion of a noise-limited feature buffer, where we follow an eye path until we reach the first noisy intersection. Conveniently, this is possible as a trivial extension to the path tracing computations that generate the noisy input image. When tracing the path from the camera, we stop at the first noisy surface and instead of exhaustively sampling further paths to obtain the radiance contribution from this surface location, we use the respective albedo value as a *noiseless representative*. This allows us to obtain legitimate gradient information in image space even in the presence of refractive and reflective materials by aggregating per-sample albedo values in the albedo map. An overview of possible light transport scenarios that are covered by this approach is illustrated in Figure 5.

The noisiness of a surface is computed in a precomputation step based its material properties, which are available before rendering. The theory around these a priori noise estimates is developed in Section 6 and the two main results are summarized and used to build our feature buffer in Section 4.4.

Radiance sample integration. While the use of the albedo map provides a noiseless representative for the radiance contributions in the light transport simulation, it is, in some sense, inconsistent, since the information gained from the rendering process is not incorporated into the feature buffer. As solution, we propose a blending scheme, where empirical information is added to the feature buffer at a rate dependent on the its variance, i.e., radiance contributions that are expected to exhibit a large sample variance are added at a lower rate than samples from nearly noise-free sources. As the recursive evaluation above, this approach utilizes the a priori noise estimates that are developed later in Section 6.

4.4 A Variance-Bound Feature Buffer

To facilitate the description of our feature buffer, we briefly summarize the main results of our a priori noise estimation theory that is presented in Section 6. Since our estimates are generated as byproduct of the path tracing process, we assume that N samples were already used to estimate the radiance of a pixel. Furthermore, it is assumed that each path has up to n light bounces and that the radiance throughout of the i-th path segment of path j is given by t_{ij}. The a priori approximation $\tilde{s}^2_{\text{pixel}}$ of the associated radiance variance is given by (14) as

$$\tilde{s}^2_{\text{pixel}} = \frac{1}{N} \sum_{j=1}^{N} \sum_{i=1}^{n} T^2_{ij} \tilde{s}^2_{ij}$$

where $T_i = \prod_{k=1}^{i-1} t_k$ denotes the cumulative throughput along the path. The estimate builds upon the a priori variance estimates $\tilde{s}^2_{ij}$ of the respective light-material interactions at each bounce. Paths that terminate before reaching n bounces contribute no variance through these non-existent bounces, i.e., $\tilde{s}^2_{ij} = 0$ is set for these. Given the variance $\tilde{s}^2_{\text{pixel}}$ of the samples, the variance of the sample *mean* can be approximated with $\tilde{s}^2_{\text{pixel}}/N$.

Empirical information blending. Our goal is now to find the blending weights w_{ij} at each bounce such that the amount of empirical information is maximized without increasing the variance of sample mean above a certain user-defined threshold δ. Choosing $\delta = 0$, as extremal case would enforce the use of only the albedo map, since it is free of noise. The blending at a bounce i of path j with outgoing direction ω and incoming direction ω' is given by

$$w_{ij} f(\mathbf{p}, \omega, \omega') \mathring{L}(p, \omega')(\omega' \cdot \mathbf{n}) + (1 - w_{ij}) \rho_{\text{hd}}(\omega)$$

where $\mathring{L}$ denotes the incoming radiance, which itself is a blended depending on weights of the subsequent light bounces. Note that similar to the throughputs at each path segments, blending is a cumulative effect and the variance contribution of a given bounce i of path j is determined cumulative weight $W_{ij} = \prod_{k=1}^{i} w_{kj}$. As a shorthand, we use $\mathbf{w}$ to denote the vector of all weights.

Given the user-defined error threshold δ, we aim to compute a set of weights $\hat{w}_{ij}$ as solution to the optimization problem

$$\hat{\mathbf{w}} = \arg\max_{\mathbf{w}} a(\mathbf{w}) \quad \text{such that} \quad \begin{cases} a(\mathbf{w}) \leq \delta^2 \\ \mathbf{0} \leq \mathbf{w} \leq \mathbf{1} \end{cases} \tag{6}$$

where $a(\mathbf{w})$ denotes the a priori approximation of the variance of the associated sample mean, i.e.,

$$a(\mathbf{w}) = \frac{1}{N^2} \sum_{j=1}^{N} \sum_{i=1}^{n} T^2_{ij} W^2_{ij} \tilde{s}^2_{ij}.$$

The intuitive meaning of the optimization problem is that we seek to incorporate as much empirical information as permissible.

Blend weight computation. Solving the full nonlinear problem (6) to obtain the desired weights $\hat{\mathbf{w}}$ would severely degrade the computational performance of our filtering approach and reduce its applicability. We found, however, that a potentially suboptimal approximation is already sufficient for our purposes and we leave more sophisticated approaches to solve (6) as future work.

When omitting the constraint $\mathbf{w} \leq 1$, an asymptotically optimal solution $\hat{w}^+_{kj}$ that attains the maximal value at $n \to \infty$, i.e., $a(\hat{\mathbf{w}}^+) = \delta^2 - 2^{-n}$, is given by

$$\hat{w}^+_{kj} = \frac{1}{t_{kj}\sqrt{2}} \left(\delta \sqrt{\frac{2j-1}{\tilde{s}^2_{ij}}} \right)^{1/i}. \tag{7}$$

However, it diverges for vanishing variance estimates $\tilde{s}^2_{ij}$ or throughputs t_{kj} and thus we perform a hard clamping to 1 to obtain the blending weights $\hat{\mathbf{w}}$, i.e., $\hat{w}_{kj} = \min\left(\hat{w}^+_{kj}, 1\right)$. While the final weights respect both constraints of (6), they constitute a suboptimal solution that does not attain the maximum of the objective. However, we found it to deliver excellent results for all our test cases.

Using these blending weights, we are able to craft a feature buffer by aggregating a variance-bounded representative color value at each pixel, which combines contributions of the actual radiance samples and the albedo map. With growing sample count N, the weights are increasing and even for a small user-defined target variances δ^2, the amount of empirical radiance information, which is incorporated into this buffer, grows and eventually dominates the albedo map contribution. In this sense, our feature buffer is biased but consistent and an illustration of this property can be found in Figure 13.

5 Noise-Adaptive Filtering

Apart from the construction of a suitable feature buffer, our a priori noise estimates can also be used in the filtering process itself. Depending on the amount of local noise, more or less smoothing can be performed by adjusting additional filtering parameters. We present two approaches in this section, one for filtering methods that permit spatially varying kernel sizes and a second as a replacement strategy based on blending for filters that lack this capability. Note that both methods can be used independently of the proposed feature buffers, i.e., they can be applied to existing filtering methods.

5.1 Noise-Adaptive Kernel Sizes

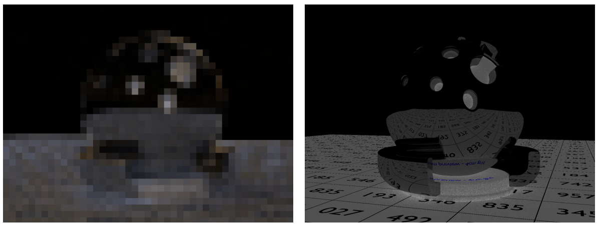
As we will show in Section 6, it is possible to derive an estimate of the variance $\tilde{s}^2_{\text{pixel}}$ that can be expected for a certain pixel location. It is based on a priori knowledge of the variance that can be expected when sampling the various scene materials and is described by (14). The standard error of the radiance sample mean of a typical Monte Carlo rendering process decreases with $1/\sqrt{N}$ in the number of samples N. This allows us to compute a desired number of samples M that is necessary to obtain a user-defined target error ϵ using the expected variance $\tilde{s}^2_{\text{pixel}}$, i.e., $M = \tilde{s}^2_{\text{pixel}}/\epsilon^2$. If the samples per pixel are fixed to N and we assume that the variance estimate also holds in the vicinity of the current pixel, we can alternatively compute the number of pixels m that is necessary to achieve the same goal by $m = \tilde{s}^2_{\text{pixel}}/(N\,\epsilon^2)$.

By adjusting the local filter size, we aim to include at least m pixels in the filter support to limit the influence of the local noise to a user-provided upper error bound ϵ. Such an approach is beneficial for filters which are capable of adaptive filter kernels, such as the cross bilateral filter [Eisemann and Durand 2004; Petschnigg et al. 2004]. The specifics of this process depend closely on the actual filter kernel. A general approach would be to set the internal parameters in such a way that most of the kernel's density is found in an area with the size of m pixels. For a univariate Gaussian, which is used, for example, in the work on adaptive manifold filtering [Gastal and Oliveira 2012], its standard deviation σ can be chosen as $\sqrt{m}/3$ to ensure that the majority of the desired number of pixels is contained within the filter support. This is based on the consideration that the number of pixels contained within the filter support given by σ can be approximated as $(3\sigma)^2$.

5.2 Noise-Adaptive Blending

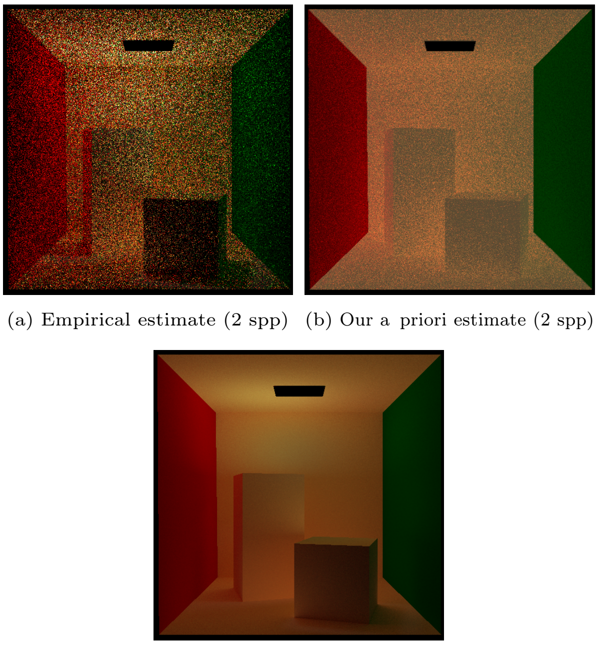
In our experience, none of the approximate high-dimensional filters, such as the adaptive manifold filter [Gastal and Oliveira 2012] or the guided filter [He et al. 2010], are capable of spatially varying kernel sizes without introducing artifacts. For this reason, we propose a simple blending-based approach to adapt the filtering to the local noise level and limit the oversmoothing of low-variance regions.

First, we compute a global filter-kernel size by taking a user-chosen percentile α_{b} of the local kernel sizes, which are computed based on the methods described in the previous section. This allows the user to decide which ratio of image pixels is treated with a fitting or larger kernel size. The result of the subsequent filtering at each pixel g^{F}, which suffers from local oversmoothing, is then blended to recover fine details from the original input g of the filtering. Using a heuristic to compute the blend weights w^{B} based on the user-defined target error ϵ and the local a priori variance estimate $\tilde{s}^2_{\text{pixel}}$, given by $w^{\text{B}} = \min(\epsilon/\sqrt{\tilde{s}^2_{\text{pixel}}}, 1)$, we compute the final output g^{B} as a convex combination, i.e., $g^{\text{B}} = w^{\text{B}}g + (1 - w^{\text{B}})g^{\text{F}}$. If the local noise



(a) (b)

Figure 6: We scale our per-pixel variance estimate (b) according to the empirical per-tile variance (a), combining the pixel accuracy of our estimate with the actuality of empirical information. These images have been multiplied by four for better visibility.



(a) Empirical estimate (2 spp) (b) Our a priori estimate (2 spp)

(c) True error

Figure 7: Our per-pixel error estimate (b) provides more reliable approximations of the true error (c) than empirical estimates (a) at low sampling rates. The images have been multiplied with a factor of two for better visibility.

estimate $\tilde{s}^2_{\mathrm{pixel}}$ is significantly larger than the target error ϵ allows, the corresponding weight w^{B} will be small and the filtered version of the image will dominate the final output g^{B}. Low variance regions, on the contrary, will contribute more of their initial values to the final image.

Together with Section 4 on a suitable feature buffer, this section on noise-adaptive filtering concludes our presentation on how a priori information can be used in the context of high-dimensional filtering. The next section will provide a detailed explanation on how these a priori variance estimates can be obtained.

6 A Priori Information for Noise Estimation

Since filtering affects both the noise and the legitimate image content, the knowledge of the actual noise level at a given image location can significantly enhance the performance of the denoising procedure. Converged image regions only need to be filtered slightly or not at all, whereas high-variance regions benefit from smoothing. In this section, we show how a priori information on the behavior of scene materials can be leveraged for a novel noise estimation method. We consider this approach to be orthogonal to the filtering of higher-order effects such as motion blur or depth of field and leave a combination of both approaches for future work.

The key observation of our method is the fact that specular materials can be sampled deterministically while diffuse materials introduce noise into the MC rendering. This can be visualized with the respective importance function, which is a Dirac delta distribution for specular materials but has a support on the whole hemisphere for diffuse materials. Along each sampled path, the material encountered at each bounce thus plays an important role in the final noise that will be present at the corresponding pixel.

We perform a decomposition of the total per-pixel radiance variance (Section 6.1), which subdivides it into two qualitatively different parts—the explained and unexplained variance. As our main contribution in this context, we present an approximation of the unexplained variance terms that can be precomputed before the rendering process. Furthermore, we show how to sample their expected values at runtime with a negligible overhead to the MC rendering process. We introduce the approximation of the variance caused by a single light-material interaction (see Section 6.2) and a sampling scheme to estimate the expected value of this variance considering multiple light bounces and paths (see Section 6.3). Finally, we fuse these estimates with the empirical sample variance of the tonemapped per-pixel radiance samples to obtain the final radiance estimate (Section 6.4).

6.1 Variance Decomposition

To determine the magnitude of spatially varying noise in the rendered image, sample variances have been used to great effect [McCool 1999; Rousselle et al. 2012; Kalantari and Sen 2013]. Unfortunately, the sample variance is subject to noise, which is prohibitive for reliable noise estimates at low sample counts. In contrast to that, our approach relies on preprocessing the variance induced by light-material interactions (i.e., light bounces) and a subsequent aggregation thereof, which ultimately leads to more reliable noise estimates. In the following, we will present the concepts related to the aggregation procedure.

Law of total variance. Ultimately, we are interested in the variance $\mathrm{V}[\cdot]$ of the radiance L—given as a random variable—over a region of the image plane, i.e., a pixel in our case. Assuming geometrical optics, the radiance that arrives at the camera through an image location X can be parametrized by a sequence of light interactions with surfaces at the locations $P_1, P_2, \ldots, P_n$, which effectively describes a light path with n bounces, where n has to be chosen sufficiently large. Considering these locations as random variables themselves, we have $L = L(X, P_1, \ldots, P_n)$, which allows us to decompose the variance of L into its constituent contributions given by the *law of total variance* [Bowsher and Swain 2012], which states that

$$\begin{aligned}\mathrm{V}[L] = {} & \mathrm{E}\big[\mathrm{V}[L \mid X, P_1, \ldots, P_n]\big] \\ & + \sum_{i=1}^{n} \bigg(\mathrm{E}\big[\mathrm{V}[L \mid X, P_1, \ldots, P_{i-1}] \\ & \qquad - \mathrm{V}[L \mid X, P_1, \ldots, P_i]\big] \bigg) \\ & + \mathrm{V}\big[\mathrm{E}[L \mid X]\big],\end{aligned} \tag{8}$$

where $\mathrm{E}[A \mid B]$ (resp. $\mathrm{V}[A \mid B]$) denote the expected value (resp. variance) of random variable A when conditioned with random variable B. Section 1.2 of the supplementary document also provides an intuition of this law in our context.

We first consider a simplified case to obtain an intuition on the various terms in this expression. If we are not interested in the specifics of the bounces, we can omit their explicit representation and assume the radiance as a function of the image plane location X, i.e., $L = L(X)$. In this case, the variance decomposition yields

$$\mathrm{V}[L] = \mathrm{E}\big[\mathrm{V}[L \mid X]\big] + \mathrm{V}\big[\mathrm{E}[L \mid X]\big].$$

The last term can be thought of as the *variance of the radiance means* and describes the variance among the expected radiances distributed over the image domain. In statistical terms, this is also referred to as the *explained variance*, since it can be obtained by evaluating the average radiance at different image locations and looking at their variance. The other summand gives the *average radiance variance* calculated based on the variance of the radiance at each image location. Often called *unexplained variance*, it gives an indication of the average variability of the radiance at constant locations. Intuitively, the explained variance can be understood as the radiance variance across a pixel, while the unexplained variance abstracts the variances that could arise at fixed image locations due to stochastic sampling of the light bounces.

Unexplained variance estimation. Common noise estimates use the radiance samples that are produced by the MC rendering method to compute their variance as an estimator of $\mathrm{V}[L]$. As such, both the explained and unexplained variance are estimated simultaneously and at low sampling rates, the associated error can become prohibitively high. Our goal is to utilize a priori information on the scene materials to improve the estimate of the unexplained variance considerably. Thus, we are concerned with the first two summands in the general variance decomposition given by (23). Since we account for all light-material interactions with the light bounce locations $P_1, \ldots, P_n$, different path geometries are accounted for in the first summand $\mathrm{E}\big[\mathrm{V}[L \mid X, P_1, \ldots, P_n]\big]$, and only the variance generated due to stochastic behavior of the materials is captured with this term (under fixed in- and outgoing directions). The second summand, given as a sum, aggregates the variance contributions of all light bounces of a given order, i.e., the i-th summand represents the average radiance variances between all light paths, where the first $i-1$ bounces are kept fixed, and those where the first i bounces are fixed. Note that the outer expected value $\mathrm{E}[\ldots]$ is taken over all possible arrangements of fixed-bounce locations and in this sense, the second summand of (23) aggregates the variance contributions of *all possible* bounces of a given order.

6.2 Per-Bounce Variance Estimation

In this section, we develop an approximation of the variance terms for each summand of

$$\sum_{i=1}^{n} \Bigg(\mathrm{E}\big[\underbrace{\mathrm{V}[L \mid X, P_1, ..., P_{i-1}] - \mathrm{V}[L \mid X, P_1, ..., P_i]}_{\Delta_i}\big] \Bigg) \tag{9}$$

in the variance decomposition given by (23). Since each difference Δ_i describes the additional variance that is introduced by adding an i-th light bounce (with all previous bounces being fixed), we investigate in the following the variance contributions of light-material interactions.

Light-material interactions. As we include refractive and reflective materials in our setting, we consider the full-sphere formulation of the rendering equation [Kajiya 1986a] given as

$$L_{\mathrm{o}}(\mathbf{p}, \omega) = L_{\mathrm{e}}(\mathbf{p}, \omega) + \int_{\mathbb{S}^2} f(\mathbf{p}, \omega, \omega') L_{\mathrm{i}}(\mathbf{p}, \omega')(\omega' \cdot \mathbf{n}) \,\mathrm{d}\omega', \tag{10}$$

where $L_{\mathrm{o}}(\mathbf{p}, \omega)$ is the radiance leaving a particular surface with normal $\mathbf{n}$ at the location $\mathbf{p}$ in the outgoing direction ω. L_{e} denotes the radiance contribution through emission, f is the *bidirectional scattering distribution function* (BSDF) describing the scattering properties of the material at hand and $L_{\mathrm{i}}(\mathbf{p}, \omega')$ is the incident radiance incoming from direction $-\omega'$.

A simple MC estimator for the integral is given by

$$\frac{1}{n} \sum_{i=1}^{n} f(\mathbf{p}, \omega, \omega_i') L_{\mathrm{i}}(\mathbf{p}, \omega_i')(\omega_i' \cdot \mathbf{n}), \tag{11}$$

where ω_i' is the sampling direction chosen based on the sampling scheme and n denotes the sample count. It can be regarded as an estimator for the expected value of the product of two random variables F and L that represent all possible samples from $f(\mathbf{p}, \omega, \omega_i')(\omega_i' \cdot \mathbf{n})$ and $L_{\mathrm{i}}(\mathbf{p}, \omega_i')$ respectively. To compute the variance of the samples, an exhaustive sampling of the product is required. While the behavior of the scene materials that determine the respective F is generally known beforehand, L has to be estimated without a light transport simulation. Thus, we approximate the incident radiance with a suitable function that emphasizes the material-specific characteristic and obtain a precomputable approximation of the variance of (11).

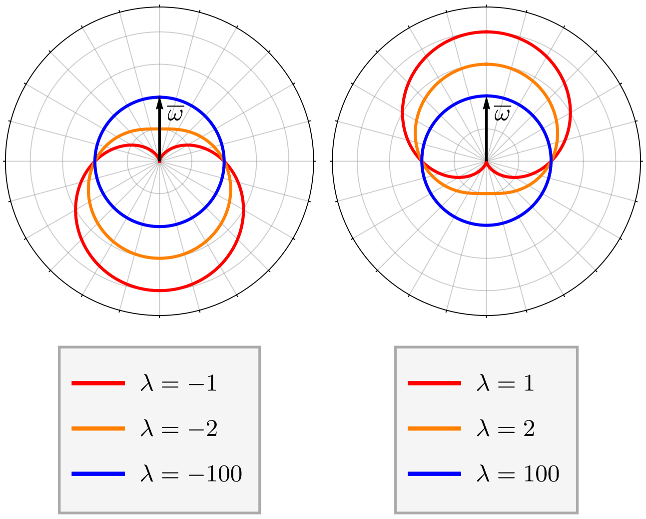


Figure 8: Polar plots showing how our approximation $\tilde{L}_{\mathrm{i}}$ changes with respect to θ for different λ values.

A priori approximation of L. For highly specular materials, only a narrow cone of sampling directions will be relevant, whereas diffuse materials reflect incoming radiance from a larger region of the hemisphere. This immediately translates over to the possible variances of the incoming radiance, i.e., the narrower the cone of relevant directions, the more similar the radiance contribution in this cone will be on average. We therefore use the function ℓ as a heuristic approximation of the random variable L given by

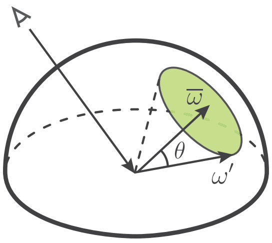
$$\ell(\mathbf{p}, \omega, \omega') = \frac{\lambda + (\omega' \cdot \overline{\omega})}{\lambda} \tag{12}$$

with a parameter λ and with the term $(\omega' \cdot \overline{\omega})$ that ensures that the function changes relative to the *representative sampling direction* $\overline{\omega}$. In Figure 8 we demonstrate the effect of the λ parameter. We use the weighted mean to determine $\overline{\omega}$, i.e.,

$$\overline{\omega} = \frac{\int_{\mathbb{S}^2} f(\mathbf{p}, \omega, \omega')\omega'}{\left|\int_{\mathbb{S}^2} f(\mathbf{p}, \omega, \omega')\omega'\right|},$$

however, other quantities, such as the spherical median [Fisher 1985], are possible as well. While the latter would be robust to outliers, we did not notice any considerable improvement in practice and use the weighted mean due to its reduced computational complexity.

Our approximation is based on the heuristic assumption that, on average, incident radiance changes proportional to $\cos(\theta)$, where θ is the angle between ω' and ϖ. Thus, the values are proportional to the area of a spherical cap of a unit sphere constituted by a cone whose cross-section subtends 2θ. In the case of a uniformly sampled spherical cap (i.e., $\cos(\theta)$ is uniformly distributed) the random variable $\tilde{L}$, representing all possible samples of (12) would be uniformly distributed as well.



Since we desire to capture the material-specific contributions to MC noise due to the incident radiance over the whole sphere, we require ℓ to be non-zero on its whole support, which leads us to a choice of $\lambda = 2$. Note that any $\lambda > 1$ is permissible, but exceedingly large values cause undesired flattening of ℓ, while values close to 1 underemphasize contributions due to transmitted radiance incident from the opposite representative direction $-\varpi$. In any case, we found $\lambda = 2$ to work well on all our test cases. To adapt our heuristic approximation to the actual scene radiance, a scaling based on radiance estimates is performed, which is presented in Section 6.4.

Variance estimate computation. To estimate the variance of the MC samples used for the estimator in (11), we take a set of samples $\hat{f}$ (resp. $\hat{l}$) from the random variables and compute their *sample variance* s^2 given by

$$s^2(\hat{f}, \hat{l}) = \frac{1}{n-1} \sum_{i=1}^{n} \left(\hat{f}_i \hat{l}_i - \overline{\hat{f}\,\hat{l}} \right)^2, \qquad (13)$$

where $\hat{f}_i$ (resp. $\hat{l}_i$) denote the individual samples and $\overline{\hat{f}\,\hat{l}}$ the mean over the resulting set of their element-wise multiplication.

Replacing the radiance samples with a set $\hat{\ell}$ of samples from our approximation given in (12), we can compute an *a priori variance estimate* $\tilde{s}^2 = s^2(\hat{f}, \hat{\ell})$ of the MC estimator. Since this quantity does not depend on the light transport simulation, it can be computed accurately in a preprocessing step and we give a detailed description of this process in Section 2.1 to 2.2 of the supplemental document.

6.3 Variance Estimate Aggregation

Having computed an a priori estimate of difference Δ_i of variances in each summand of (9), we are still left with the computation of the expected value $\mathrm{E}[\Delta_i]$ of these per-bounce variance contributions over *all* bounces of a given order i, i.e., the i-th summand characterizes the additional radiance variances caused by *all* i-th bounces. For a given camera position, these quantities can be computed by exhaustive sampling similar to the light transport simulation itself. This would approximately double the required computational effort, which is clearly unacceptable.

Unexplained variance aggregation. To remedy this, we reuse the sampling that is already performed during the light transport simulation. For each path, we not only compute the corresponding radiance sample, but also query the a priori variances at each light-material interaction. Note that this imposes a sampling scheme that takes radiance values into account in contrast to our approximated radiance. This is beneficial for the accuracy of our variance estimate, as only relevant light-material interactions are considered. The knowledge about which materials are encountered during light transport simulation cannot be sensibly determined a priori, which necessitates this approach. It provides us with samples of our a priori variance for the first m bounces, if the path has m vertices.

To compute the i-th summand of (9), we collect the variances $\tilde{s}_i^2$ at bounces of the same order i for all paths and estimate their expected value by averaging them. Before that, we have to account for geometrical relationships between subsequent bounces, which influence the radiance throughput of each path segment. If the incident radiance at a light-material interaction is scaled by a factor a, the variance of the corresponding estimator is scaled with a^2. Thus, we have to scale each sampled a priori variance $\tilde{s}_i^2$ with a cumulative throughput factor T_i^2, which is a product of all throughputs t_k from the camera to the corresponding bounce, i.e., $T_i = \prod_{k=1}^{i-1} t_k$. Finally, we obtain an approximation $\tilde{s}^2_{\text{pixel}}$ of (9)—the main contribution to the unexplained variance—by

$$\tilde{s}^2_{\text{pixel}} = \sum_{i=1}^{n} \frac{1}{N} \sum_{j=1}^{N} T_{ij}^2 \tilde{s}_{ij}^2 = \frac{1}{N} \sum_{j=1}^{N} \sum_{i=1}^{n} T_{ij}^2 \tilde{s}_{ij}^2 \qquad (14)$$

where the n bounce orders are indexed by i, and j enumerates the N paths which are used for the variance estimate of the given pixel. The reordered form on the right side is suited for an effortless integration into existing MC rendering systems and is used in our implementation. Note that terminated paths, which have less than n bounces, do not contribute to higher bounces and for these, $\tilde{s}^2_{ij} = 0$.

The remainder of the unexplained variance, i.e., the term $\mathrm{E}\big[\mathrm{V}[L \,|\, X, P_1, \ldots, P_n]\big]$, denotes the expected value of the variances that arise if paths are considered fixed. As such, it encodes the remainder of unaccounted stochastic material effects if the directions of both in- and outgoing radiance at each light bounce is kept fixed. Since we generally respect all stochastic material effects by means of our a priori variance, this term can be omitted, i.e., considered zero.

Explained variance aggregation. As the last term in (23) that is unaccounted for, the explained variance $\mathrm{V}\big[\mathrm{E}[L \,|\, X]\big]$ plays only a minor role in our setting. Since we desire per-pixel noise estimates, the random variable X that describes the image location only varies over the extent of the footprint of a single pixel. As such, the expected value $\mathrm{E}[L \,|\, X]$ of the radiance at each location will have limited variance $\mathrm{V}\big[\mathrm{E}[L \,|\, X]\big]$ over the pixel and we choose to simply omit this term. We also note that an effective evaluation of the expected radiance requires exhaustive sampling of light bounces of paths starting at each fixed image location X. Undersampling, especially at low sampling rates as in our context, would degrade the estimate considerably and would effectively reintroduce the unexplained variance that is already accounted for.

6.4 Variance Fusion

Our heuristic approximations of the incident radiance at each light bounce do not take the actual scene radiance into account (see Section 6.2). In this sense, they are decoupled from the light simulation and would yield the same result for all possible lighting configurations. In this section, we present a method on how to fuse the information obtained from radiance samples to perform an adequate scaling of the radiance approximations given by Equation 12.

As basis serves an empirical variance estimate for which we use the approach by Dmitriev and Seidel [2004], who use a tile-based estimate of the form

$$s^2_{\mathrm{tile}} = \frac{1}{m} \sum_{j=1}^{m} \Big(T\big(L_{\mathrm{all}}(j)\big) - T\big(L_{\mathrm{e|o}}(j)\big) \Big)^2 \qquad (15)$$

where the m pixels in the tile are enumerated by index j. A tone mapping operator T is used to transform the variance estimate based using all the radiance samples L_{all} of each pixel as well as the estimate using only the even or odd subset $L_{\mathrm{e|o}}$ of it. A main concern in the use of such an approach is the tile size. While small tiles give a noisy but detailed variance estimate, large tiles significantly reduce the noise on the expense of reduced locality of the estimate. When basing the variance estimates solely on empirical data, small tiles need to be used (e.g., 4×4 pixels in the case of Dmitriev and Seidel [2004]), whereas our a priori variance estimates already provide noiseless local information. Thus, we use significantly larger tiles, i.e., 32×32, to get more accurate estimates with a lot less noise and use them to scale our local estimates, which is performed in three main steps:

Tonemapping. We perform approximative tonemapping, which effectively clamps the radiance variances to a certain range.

Downsampling. To allow comparison, our per-pixel estimates $\tilde{s}^2_{\mathrm{pixel}}$ are coarsened to tile granularity $\tilde{s}^2_{\mathrm{tile}}$ by averaging them over the footprint of each tile.

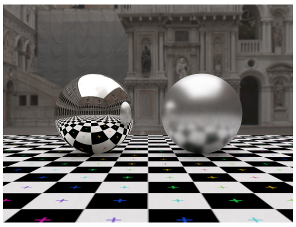
Scaling. A global scaling factor is derived from the per-tile scaling factors $s^2_{\mathrm{tile}}/\tilde{s}^2_{\mathrm{tile}}$ with the help of a user-chosen percentile α_{v}, which allows a trade-off between how much of the rendering's noise is under- or overestimated. Note that this is done for each color channel and the maximum among those global scaling factors is used for the whole image and all channels. This factor effectively scales all approximated radiances evenly and adapts them to the observed radiance in the scene.

This completes the derivation of the scaled per-pixel variance estimates that are build the basis of the albedo map and the filtering strategies in Section 4.

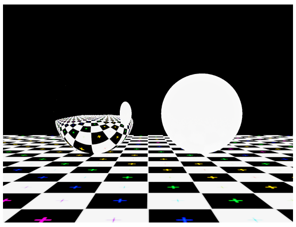
7 Results

We have incorporated our albedo rendering and variance estimation techniques in LuxRender. All the results shown throughout this paper were rendered with unidirectional path tracing on the CPU with MIS combining BSDF and light source sampling. For filtering, we have utilized the MATLAB

(a) Filtering without blending mask



(b) Filtering with blending mask



(c) Blending mask visualization

Figure 9: A noisy input image with 4 spp filtered with and without our blending mask on objects of varying glossyness.

reference implementation of the adaptive manifold filter by Gastal et al. [2012] and a C++ implementation thereof found in OpenCV [Bradski 2000]. The results were obtained using a 2.2 GHz quad-core Intel Core i7 machine with 8 GB of memory.

For the sake of fairness, we compare our method to the approach described by Gastal et al. [2012] using feature buffers containing depth and normal information. Their approach is the only one that we consider comparable to ours in computational overhead. We consider the Classroom scene to be the most representative for a practical case to demonstrate the performance of our approach (Figure 11).

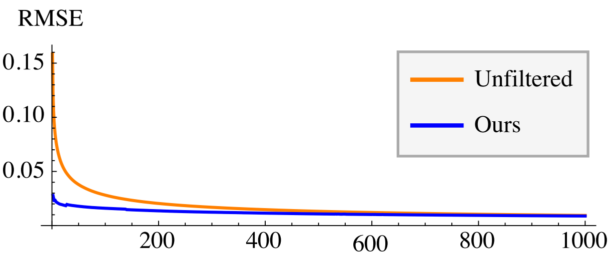


Figure 10: The output of our filtering framework asymptotically converges to the ground truth and retains delicate features obtained later in the rendering process. The measured results correspond to the Splash scene in Figure 1. We show a side-by-side comparison of the evolution of the noisy and the filtered image in the supplementary video.

Step	Time [ms]
Rendering (1 spp)	5,793
Gamma correction	19
Adaptive Manifold Filtering	2,826
Total	**8,639**
Rendering (1 spp)	6,399
Albedo map	410
BSDF variance estimate	138
Tile variance	53
Variance fusion	55
Adaptive filter kernel	36
Filtering	3,990
Blending	85
Gamma correction	19
Total	**10,585**

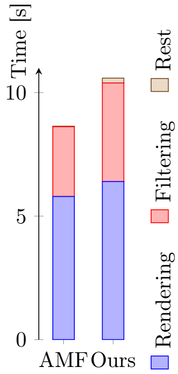


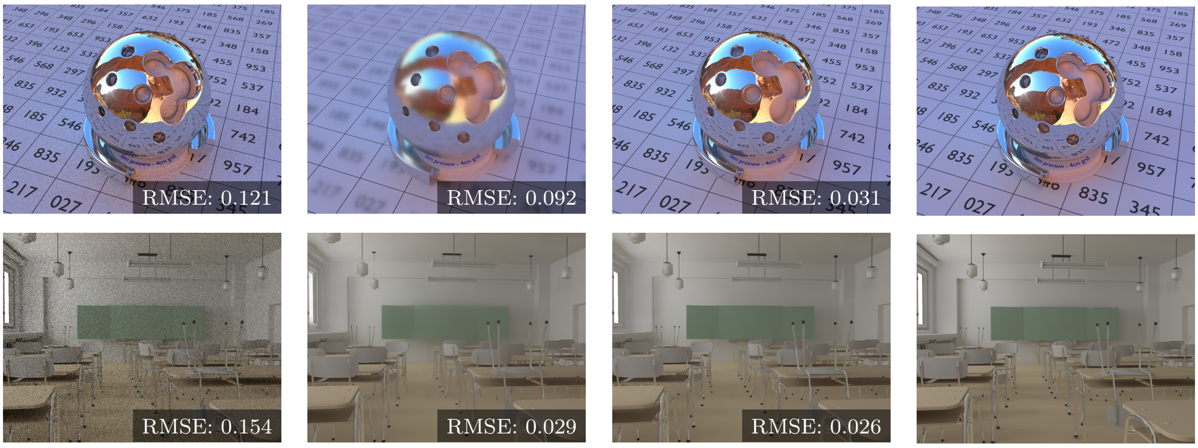
Figure 11: Detailed breakdown of the individual steps of the proposed framework. Our techniques use mainly a priori knowledge known before starting the rendering process, therefore they introduce a minimal additional overhead over the adaptive manifold filter of Gastal and Oliveira [2012]. The timings were recorded during the rendering of the Classroom scene seen in Figure 12.

Albedo map. Our albedo map effectively helps to discern many legitimate edges to keep them from being oversmoothed by the filtering process. The merits of this technique are apparent in our comparison (Figure 12).

The albedo map adheres to the rules stated in Section 4.1 as it provides high-quality gradient information in a noiseless and inexpensive manner. As the rendering progresses, more and more gradient information is included in the albedo map while the noise level is kept below a specified threshold. Moreover, the albedo map is a *consistent* estimator of the actual radiance, which is demonstrated in Figure 13.

Refractive materials can be accounted for by deterministically tracing the reflective and refractive paths at the cost of additional overhead to the rendering process depending on the scene description in exchange for filtering quality (Figure 14).

For scenes primarily containing non-refractive materials, the computational overhead implied by the albedo map is minuscule. This can bee seen in Figure 11.



Noisy input | [Gastal and Oliveira 2012] | Our technique | Ground truth

Figure 12: Comparison between our technique and the adaptive manifold filter using the same set of parameters. In both scenes, the adaptive manifold filter blurs many important details, leaving only the sharpest features intact. Both scenes were rendered with 4 spp. The LuxBall scene was filtered with a spatial standard deviation of 15.81, 15 manifolds the following parameters: $\delta = 0.001, \epsilon = 0.0055, \alpha_v = 50, \alpha_b = 50, \sigma_d = 0.2, \sigma_n = 0.2, \sigma_a = 0.1$. The Classroom scene was filtered with a spatial standard deviation of 30.43, 31 manifolds and the following parameters: $\delta = 0.001, \epsilon = 0.00275, \alpha_v = \alpha_b = 87.5, \sigma_d = 0.2, \sigma_n = 0.2, \sigma_a = 0.05$. The parameters σ_d, σ_n and σ_a correspond to entries in the covariance matrix $\mathbf{\Sigma}$ (described in Section 1.1 of the supplemental document) for the depth, normal and albedo map feature buffers respectively.

A priori noise estimate. Our variance estimation heuristic described in Section 6 accounts for the noise introduced by different material behaviors. It relies on an a priori approximation of the variance introduced by light-material interactions and a subsequent aggregation thereof. We show a visualization of our variance estimate compared to the empirical per-tile variance in Figure 6. Figure 7 demonstrates how the variance estimate increases in the vicinity of multiple diffuse bounces which is due to the aggregation of multiple bounces for a given path (Section 6.3).

The prime advantage of our approach is that it considers at least the first high-variance bounce along a path, thereby limiting the room for underestimation significantly, even at low sampling rates. In contrast to that, empirical estimates frequently underestimate variance. This is especially severe at low sampling rates, which can be observed in Figure 7a.

Our variance-fusion technique ensures that our variance estimate is close to the actual variance of the rendering and that it decreases as more samples are taken. Using a filter which is adapted to the estimated variance therefore results in a consistent output image. This fact is demonstrated by the RMSE plot given in Figure 10. Moreover, Figure 9 shows how our variance estimate can be utilized to derive per-pixel blending factors to blend unfiltered and filtered pixels.

Our variance-estimation scheme poses minimal integration, run-time and memory overhead due to its a priori nature. The remainder of the necessary computations can be inexpensively conducted during the light transport simulation. Figure 11 demonstrates the minimal run-time implications of our variance-estimation method.

8 Limitations and Future Work

In the following, we discuss the limitations of our techniques as well as possible directions for future work.

Higher-order effects. Obtaining a noiseless albedo map while rendering the initial samples requires that there is no integration for higher-order effects such as motion blur or depth of field. These effects introduce additional stochastic processes,

(a) 4 spp (b) 64 spp (c) 1024 spp

Figure 13: A series of albedo maps showing its convergence to the MC estimate with an increasing amount of used samples while keeping noise below a user-defined threshold. As the rendering process matures, the feature buffer contains more and more noiseless gradient information, and eventually converges to the ground truth (disregarding the tone mapping operation).

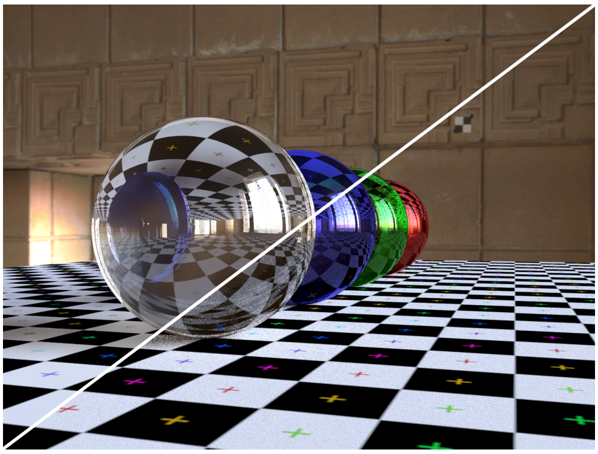


Figure 14: Deterministically tracing refractive objects for the albedo map effectively preserves legitimate information in the filtered output seen above. The image was rendered using 16 spp.

which ultimately causes noise in the feature buffers. Possible remedies for these problems are either the usage of a higher number of samples or the utilization of specific noise-reduction techniques tailored to these effects.

Our BSDF variance estimate suffers from noise due to higher-order effects in a similar manner. An interesting future direction in this regard would be the consideration of an a priori estimate parametrized by depth and lens parameters to estimate the variance induced by depth of field effects.

Consistent variance estimation. Our BSDF variance estimate, given by (14), is reliable at low sampling rates as underestimations, inherent to purely empirical methods, are constrained. By increasing the sample count, however, the empirical sample variance asymptotically converges to the true variance, since the sample variance is a consistent estimator of the population variance. Therefore, it is more accurate than our a priori estimate at high sample counts.

In this regard, we presume that our a priori estimate can be considerably improved by increasingly incorporating the sample variance accordingly as the number of samples grows. We note that this is orthogonal to our variance fusion technique, where our a priori estimates are uniformly scaled according to the sample variance of the sample mean.

Non-linear tone mapping. Our BSDF variance estimate neglects the effects of non-linear tone mapping operations such as gamma correction. This is due to the fact that usually, tone mapping is performed according to the brightness of the scene, which means that the tone mapping parameters are oftentimes not known beforehand and can not be considered during the evaluation of the BSDF variance estimate. Linear tone mapping operations are naturally supported, as a linear scaling of the radiance values results in a linear scaling of the corresponding variances, which is accounted for with our variance-fusion technique.

Non-linear operations are more difficult to consider, as the resulting variance depends on the location of the radiances in the value range. For ranges where radiance differences are compressed, the variance tends to be low, whereas for ranges where differences are expanded, the variance is accordingly high. This is the reason why gamma correction is performed at the end of our rendering pipeline, i.e., filtering is performed on the linearly tone mapped values.

9 Conclusion

We have presented a novel feature buffer and a per-pixel variance estimation heuristic to facilitate the usage of high-dimensional filters for MC noise reduction. Our theoretical treatment of the method noise of high-dimensional filters reveals that a high-quality feature buffer should follow the magnitude of gradients in the presence of relevant images features and should be as noise-free as possible. Our albedo map is a practical implementation of these observations. In particular, it is based on a recursive evaluation and can therefore deal with specular materials. It also supports all commonly used material models without any explicit classification rule. Since the albedo can be precomputed, the whole filter process merely takes a few seconds to compute on full HD images for practical scenes. We also show how to merge in actual radiance samples in more converged areas. This new feature buffer can be used as an extension to improve the quality of any technique relying on a high-dimensional filter to denoise MC renderings.

Furthermore, we propose a novel heuristic to estimate the per-pixel variance reliably. It is based on a priori predictions about the variance introduced by light-material interactions and an aggregation thereof. Moreover, it is adapted to the actual variance levels by means of variance fusion and consequently enables *consistent filtering*, i.e., the spatial width of the filter decreases as more samples are added. Both the albedo map and the BSDF variance estimate can be used to enhance a variety of different techniques not directly related to our work. The variance estimate could be used to drive an adaptive sampling scheme, for instance, whereas the albedo map can be used in any scenario where noiseless gradient information is desired.

References

Adams, A., Baek, J., and Davis, M. A. 2010. Fast high-dimensional filtering using the permutohedral lattice. *Computer Graphics Forum 29*, 2, 753–762.

Bauszat, P., Eisemann, M., and Magnor, M. 2011. Guided image filtering for interactive high-quality global illumination. *Computer Graphics Forum 30*, 4, 1361–1368.

Bowsher, C. G., and Swain, P. S. 2012. Identifying sources of variation and the flow of information in biochemical networks. *Proceedings of the National Academy of Sciences 109*, 20, E1320–E1328.

Bradski, G. 2000. The opencv library. *Doctor Dobbs Journal 25*, 11, 120–126.

Buades, A., Coll, B., and Morel, J.-M. 2005. A non-local algorithm for image denoising. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 2, 60–65 vol. 2.

Buades, A., Coll, B., and Morel, J.-M. 2006. Neighborhood filters and PDE?s. *Numerische Mathematik 105*, 1, 1–34.

Chen, J., Paris, S., and Durand, F. 2007. Real-time edge-aware image processing with the bilateral grid. In *ACM SIGGRAPH 2007 Papers*, ACM, New York, NY, USA, SIGGRAPH '07.

Dammertz, H., Sewtz, D., Hanika, J., and Lensch, H. P. A. 2010. Edge-avoiding à-trous wavelet transform for fast global illumination filtering. In *Proceedings of the Conference on High Performance Graphics*, Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, HPG '10, 67–75.

Dmitriev, K., and Seidel, H.-P. 2004. Progressive path tracing with lightweight local error estimation. In *VMV*, 249–253.

Durand, F., and Dorsey, J. 2002. Fast bilateral filtering for the display of high-dynamic-range images. In *Proceedings of the 29th Annual Conference on Computer Graphics and Interactive Techniques*, ACM, New York, NY, USA, SIGGRAPH '02, 257–266.

Eisemann, E., and Durand, F. 2004. Flash photography enhancement via intrinsic relighting. *ACM Trans. Graph. 23*, 3 (Aug.), 673–678.

Fisher, N. I. 1985. Spherical medians. *Journal of the Royal Statistical Society. Series B (Methodological) 47*, 2, pp. 342–348.

GASTAL, E. S. L., AND OLIVEIRA, M. M. 2012. Adaptive manifolds for real-time high-dimensional filtering. *ACM Trans. Graph. 31*, 4 (July), 33:1–33:13.

HE, K., SUN, J., AND TANG, X. 2010. Guided image filtering. In *Computer Vision –ECCV 2010*, K. Daniilidis, P. Maragos, and N. Paragios, Eds., vol. 6311 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 1–14.

KAJIYA, J. T. 1986. The rendering equation. In *Computer Graphics (SIGGRAPH '86 Proceedings)*, 143–150.

KAJIYA, J. T. 1986. The rendering equation. In *Proceedings of the 13th Annual Conference on Computer Graphics and Interactive Techniques*, ACM, New York, NY, USA, SIGGRAPH '86, 143–150.

KALANTARI, N. K., AND SEN, P. 2013. Removing the noise in Monte Carlo rendering with general image denoising algorithms. *Computer Graphics Forum 32*, 2pt1, 93–102.

KALANTARI, N. K., BAKO, S., AND SEN, P. 2015. A Machine Learning Approach for Filtering Monte Carlo Noise. *ACM Transactions on Graphics (TOG) (Proceedings of SIGGRAPH 2015) 34*, 4.

LAFORTUNE, E. P., AND WILLEMS, Y. D. 1993. Bi-directional path tracing. In *Proceedings of CompuGraphics*, vol. 93, 145–153.

LI, T.-M., WU, Y.-T., AND CHUANG, Y.-Y. 2012. Sure-based optimization for adaptive sampling and reconstruction. *ACM Trans. Graph. 31*, 6 (Nov.), 194:1–194:9.

LINDENBAUM, M., FISCHER, M., AND BRUCKSTEIN, A. 1994. On gabor's contribution to image enhancement. *Pattern Recognition 27*, 1, 1–8.

MCCOOL, M. D. 1999. Anisotropic diffusion for Monte Carlo noise reduction. *ACM Trans. Graph. 18*, 2 (Apr.), 171–194.

MOON, B., JUN, J. Y., LEE, J., KIM, K., HACHISUKA, T., AND YOON, S.-E. 2013. Robust image denoising using a virtual flash image for Monte Carlo ray tracing. *Computer Graphics Forum 32*, 1, 139–151.

MOON, B., CARR, N., AND YOON, S.-E. 2014. Adaptive rendering based on weighted local regression. *ACM Trans. Graph. 33*, 5 (Sept.), 170:1–170:14.

PARK, H., MOON, B., KIM, S., AND YOON, S.-E. 2013. P-rpf: Pixel-based random parameter filtering for Monte Carlo rendering. 123–130.

PETSCHNIGG, G., SZELISKI, R., AGRAWALA, M., COHEN, M., HOPPE, H., AND TOYAMA, K. 2004. Digital photography with flash and no-flash image pairs. *ACM Trans. Graph. 23*, 3 (Aug.), 664–672.

ROUSSELLE, F., KNAUS, C., AND ZWICKER, M. 2012. Adaptive rendering with non-local means filtering. *ACM Trans. Graph. 31*, 6 (Nov.), 195:1–195:11.

ROUSSELLE, F., MANZI, M., AND ZWICKER, M. 2013. Robust denoising using feature and color information. *Computer Graphics Forum 32*, 7, 121–130.

RUSHMEIER, H. E., AND WARD, G. J. 1994. Energy preserving non-linear filters. In *Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques*, ACM, New York, NY, USA, SIGGRAPH '94, 131–138.

SEN, P., AND DARABI, S. 2012. On filtering the noise from the random parameters in Monte Carlo rendering. *ACM Trans. Graph. 31*, 3 (June), 18:1–18:15.

STEIN, C. M. 1981. Estimation of the mean of a multivariate normal distribution. *The annals of Statistics*, 1135–1151.

TOMASI, C., AND MANDUCHI, R. 1998. Bilateral filtering for gray and color images. In *Computer Vision, 1998. Sixth International Conference on*, 839–846.

YAROSLAVSKY, L. 1985. *Digital Picture Processing : An Introduction*, vol. 9 of *Springer Series in Information Sciences*. Springer-Verlag.

ZWICKER, M., JAROSZ, W., LEHTINEN, J., MOON, B., RAMAMOORTHI, R., ROUSSELLE, F., SEN, P., SOLER, C., AND YOON, S.-E. 2015. Recent advances in adaptive sampling and reconstruction for monte carlo rendering. *Computer Graphics Forum (Proceedings of Eurographics) 34*, 2 (May), 667–681.

A Method Noise for Feature Buffers

To evaluate the influence of feature buffer choices on the denoising capabilities of bilateral filtering operations, we evaluate the corresponding method noise. As shown in the work of Buades et al. [2006], two convenient simplifications can be used without compromising the validity of our final conclusions: *i)* the spatial weight function of the bilateral filter can be substituted with a step function, yielding the Yaroslavsky filter YF [Yaroslavsky 1985], and *ii)* the computation can be conducted in 1D. Using a one-dimensional feature buffer $h(x)$, the filter of interest on a signal $g(x)$ is given as

$$\mathrm{YF}g(x) = \frac{1}{C(x)} \int_{x-\tau}^{x+\tau} g(y)\, e^{-\frac{|h(y)-h(x)|^2}{\zeta^2}}\, dy \quad (16)$$

where the normalization is given as $C(x) = \int_{x-\tau}^{x+\tau} e^{-\frac{|h(y)-h(x)|^2}{\zeta^2}} dy$ and the filter parameters for the spatial domain (resp. feature buffer) are given by τ (resp. ζ). The local effects of the filter are given by the method noise $g(x) - \mathrm{YF}g(x)$ for small τ and ζ, where we assume that $\tau/\zeta = \text{const}$. By applying Taylor expansion with $y = x + t$ for a small t on the terms of (16), i.e.,

$$\begin{aligned} g(x+t) - g(x) &= tg'(x) + \tfrac{t^2}{2} g''(x) + \dots \\ e^{-\frac{|h(x+t)-h(x)|^2}{\zeta^2}} &= \\ e^{-\frac{t^2 h'(x)^2}{\zeta^2}} &\left(1 - \tfrac{t^3}{\zeta^2} h'(x) h''(x) + \dots\right) \end{aligned} \quad (17)$$

we obtain for the method noise

$$\begin{aligned} &g(x) - \mathrm{YF}g(x) = \\ &\tau^2 \left(u\left(\frac{\tau}{\zeta} h'(x)\right) \frac{g'(x)}{h'(x)} h''(x) + v\left(\frac{\tau}{\zeta} h'(x)\right) g''(x) \right) \\ &\qquad + O(\tau^3) \end{aligned} \quad (18)$$

with $u(s) = s\, z(s) - 3v(s)$, $v(s) = \frac{1-2s\, z(s)}{4s^2}$, $z(s) = \frac{e^{-s^2}}{\sqrt{\pi}\, \mathrm{Erf}(s)}$.

B Background

B.1 High-Dimensional Filtering

Filtering is a fundamental operation where values from a given input signal are combined in a meaningful way to form the values of the resulting output. In the case of discrete *linear filtering*, the output values g_i^{F} result from the linear combination of all input values g_j with given *filter weights* w_{ij}^{F}:

$$g_i^{\mathrm{F}} = \sum_{j \in \mathcal{I}} w_{ij}^{\mathrm{F}} g_j. \quad (19)$$

Here, $\mathcal{I}$ denotes the spatial domain of the input signal and i, j denote positions in $\mathcal{I}$. For example, in the case of digital images, the spatial domain is two-dimensional ($d_{\mathcal{I}} = 2$) and points in this domain are denoted by positions $i = (x_i, y_i)^T$. In this paper, we concentrate on digital image filters, noting that most of our proposed concepts can easily be applied to different spatial dimensionalities.

The filter weights are defined by the *filter kernel* ϕ which determines the influence of each input value at position j on a particular output value at i by evaluating their *distance*. The notion of distance is constituted by a d-dimensional signal h:

$$w_{ij}^{\mathrm{F}} = \frac{\phi(h_i - h_j)}{\sum_{j \in \mathcal{I}} \phi(h_i - h_j)}. \quad (20)$$

The normalization factor ensures that the weights to filter a particular pixel add up to one, i.e., $\sum_{j \in \mathcal{I}} w_{ij}^{\mathrm{F}} = 1$. Here, h_i is constituted by concatenated values in d-dimensional space: $h_i = (x_i, y_i, \dots)^T$.

A commonly chosen filter kernel is a Gaussian function, since it allows the differentiation of inliers from outliers and is thus *robust* [Durand and Dorsey 2002]. The Gaussian filter kernel can be written as

$$\phi(h_i - h_j) = e^{-\frac{1}{2}(h_i - h_j)^T \mathbf{\Sigma}^{-1} (h_i - h_j)}. \quad (21)$$

where $\mathbf{\Sigma}$ denotes a $d \times d$ covariance matrix determining how the weights decrease with distance. Our notation used in this regard is inspired by previous work [Gastal and Oliveira 2012].

Invariance under rigid transformations of h. The filter weights used to evaluate the influence of each input pixel are solely dependent on the inter-pixel distances constituted by the d-dimensional signal h. Accordingly, the filtered result is invariant under rigid transformations of h. Rigid transformations include translations as well as orthogonal transformations such as reflections and rotations. This can be formally defined as

$$\mathbf{R} h_i + \mathbf{t} \quad \forall i \in \mathcal{I}, \quad (22)$$

where $\mathbf{R}$ denotes an orthogonal transformation matrix and $\mathbf{t}$ a translation vector. This invariance is the reason why flash images can be used to successfully filter a photograph, even if the flash image is comprised of different colors [Petschnigg et al. 2004; Moon et al. 2013].

Notable high-dimensional filters. The *bilateral filter* [Tomasi and Manduchi 1998] is among the most widely used filtering algorithms. It utilizes a Gaussian kernel and the spatial x, y coordinates as well as the r, g, b intensities of the input image as an edge-stopping function, i.e., $h_i = (x_i, y_i, r_i, g_i, b_i)^T$. The bilateral filter heeds the distances in image-space as well as the distances constituted by the values of the input and therefore smoothes the homogeneous regions while preserving the strong discontinuities (edges) in the image accordingly.

The bilateral filter can be generalized in a way that it does not draw the distances from the original input, but from a separate, arbitrary buffer. These filters are termed *joint* [Petschnigg et al. 2004] or *cross bilateral filters* [Eisemann and Durand 2004] and are proven to be useful in situations where the input image exhibits undesirable properties regarding the distance calculation, such as excessive noise. Evaluating the distances based on another signal with more desirable properties can therefore lead to superior results. Petschnigg et al. successfully applied joint bilateral filtering for the noise reduction of photographs taken in low-light conditions by using corresponding flash images as edge-stopping functions.

While the bilateral filter reconstructs a pixel by averaging its neighboring pixels, *non-local* filtering methods consider all pixels in the given input image. A well-known method, the *non-local means filter* [Buades et al. 2005] calculates distances based on the comparison of the neighborhoods of the pixels instead the pixels themselves; regions with similar neighborhoods are weighted more heavily than those with dissimilar ones.

Approximations. The naïve evaluation of (19) is expensive as it requires $O(dm^2)$ operations, where m denotes the number of pixels. For this reason, several approximative techniques have been suggested, such as *bilateral grids* [Chen et al. 2007], *permutohedral lattices* [Adams et al. 2010] or *adaptive manifolds* [Gastal and Oliveira 2012] to accelerate the process. We refer the interested readers to previous work [Gastal and Oliveira 2012] for a detailed survey of acceleration methods.

B.2 An Intuition for the Law of Total Variance

The general law of total variance for an arbitrary number of conditioning random variables $P_1, \ldots, P_n$ is given by (and (8) in the paper):

$$\begin{aligned} \mathrm{V}[L] = {} & \mathrm{E}\big[\mathrm{V}[L \mid X, P_1, \ldots, P_n]\big] \\ & + \sum_{i=1}^{n} \Big(\mathrm{E}\big[\mathrm{V}[L \mid X, P_1, \ldots, P_{i-1}] \\ & \qquad - \mathrm{V}[L \mid X, P_1, \ldots, P_i]\big] \Big) \\ & + \mathrm{V}\big[\mathrm{E}[L \mid X]\big]. \end{aligned} \tag{23}$$

To convey our intuition, we start with the case of one conditioning random variable:

$$\mathrm{V}[L] = \mathrm{E}\big[\mathrm{V}[L \mid X]\big] + \mathrm{V}\big[\mathrm{E}[L \mid X]\big]. \tag{24}$$

As stated in the paper, the first summand denotes the *average variance of the radiances* for locations in image space given by X, where the second summand is the *variance of the radiance means.*

In the case of two or more conditioning random variables the *same* equation holds true:

$$\mathrm{V}[L] = \mathrm{E}\Big[\underbrace{\mathrm{V}[L \mid X]}_{\textit{total}\ \text{var. for a given}\ X}\Big] + \mathrm{V}\big[\mathrm{E}[L \mid X]\big]. \tag{25}$$

In this context, the underbraced term denotes the *total variance of* L *for a given* X. As the conditioning variables are not explicitly considered, it represents the whole unexplained variance for an *arbitrary number of conditioning random variables.*

This insight helps us to understand the general law of total variance by considering the following transformations, starting with the case with one conditioning variable:

$$\begin{aligned}
\mathrm{V}[L] &= \mathrm{E}\big[\mathrm{V}[L\,|\,X]\big] + \mathrm{V}\big[\mathrm{E}[L\,|\,X]\big] \\[1em]
&= \mathrm{E}\big[\mathrm{V}[L\,|\,X]\big] + \mathrm{V}\big[\mathrm{E}[L\,|\,X]\big] \\
&\quad + \mathrm{E}\big[\mathrm{V}[L\,|\,X, P_1]\big] - \mathrm{E}\big[\mathrm{V}[L\,|\,X, P_1]\big] \\[1em]
&= \mathrm{E}\big[\mathrm{V}[L\,|\,X, P_1]\big] \\
&\quad + \mathrm{E}\big[\mathrm{V}[L\,|\,X]\big] - \mathrm{E}\big[\mathrm{V}[L\,|\,X, P_1]\big] \\
&\quad + \mathrm{V}\big[\mathrm{E}[L\,|\,X]\big] \\[1em]
&= \mathrm{E}\big[\mathrm{V}[L\,|\,X, P_1]\big] \\
&\quad + \mathrm{E}\big[\mathrm{V}[L\,|\,X] - \mathrm{V}[L\,|\,X, P_1]\big] \\
&\quad + \mathrm{V}\big[\mathrm{E}[L\,|\,X]\big].
\end{aligned}$$

The last three lines correspond with the general law of total variance. In this regard, the intermediate (center) term can be understood as the expected value of the total variance for a given X without the variance components introduced by the subsequent conditioning variable(s).

Considering our a priori estimates. Our per-bounce a priori variance estimates corresponds to the aforementioned center term as the variance is modeled as if the incident radiance is exactly known, i.e., our a priori estimate does not represent the variance of a bounce and all following bounces. Instead, it represents the variance introduced by a single bounce disregarding all following bounces.

C Implementation Details

In the following, we will outline specific details which we have omitted previously for brevity's sake.

C.1 Multiple Importance Sampling

Multiple importance sampling (MIS) is a powerful noise suppression technique combining the advantages of multiple MC estimators to approximate a particular integral. The combination is based on weights which are calculated according to the *probability distribution functions* (PDFs) of the respective sampling schemes.

To account for differences in variance introduced by MIS, our per-bounce variance estimate, given by (13) in the paper, can be extended with the inclusion of the MIS weights for each sample:

$$s^2(\hat{f}, \hat{\ell}, \hat{\varpi}) = \frac{1}{n-1} \sum_{i=1}^{n} \left(\hat{f}_i \hat{\ell}_i \hat{\varpi}_i - \overline{\hat{f}\,\hat{\ell}\,\hat{\varpi}} \right)^2, \quad (26)$$

where ϖ is the random variable corresponding to the weights. Due to the additivity property of variance, the combination of the variance estimates of the different estimators is a matter of a simple addition.

We have evaluated our variance estimate for BSDF sampling combined with light source sampling through MIS. The sampling of light sources requires the PDF of $\tilde{L}_\mathrm{i}(\mathbf{p}, \omega, \omega')$ to calculate the weight. It can be trivially evaluated through normalization, ensuring that the integration over the sphere yields one:

$$p_\ell(\mathbf{p}, \omega, \omega') = \frac{\ell(\mathbf{p}, \omega, \omega')}{4\pi}. \quad (27)$$

Generating sampling directions according to $p_\ell(\mathbf{p}, \omega, \omega')$ is an additional necessity for the light source sampling MC estimator. This task is detailed in the following.

Importance sampling ℓ. We describe a simple approach to generate sampling directions according to the PDF given by (27). First, we derive a PDF with respect to θ and ϕ from it, which is defined with respect to solid angle:

$$p_\ell(\mathbf{p}, \theta, \phi) = \frac{\lambda + \cos(\theta)}{\lambda} \sin(\theta). \quad (28)$$

Based on this, we can easily generate the needed spherical coordinates θ and ϕ through rejection sampling:

Algorithm 1 Rejection sampling $\tilde{L}_\mathrm{i}$

loop
- $\theta \leftarrow \pi\xi([0,1])$
- $\phi \leftarrow 2\pi\xi([0,1])$
- **if** $\xi([0,1]) < \dfrac{p_\ell(\mathbf{p}, \theta, \phi)}{\max(p_\ell(\mathbf{p}, \theta, \phi) : \theta \in (0, 2\pi))}$ **then**
  - **return** $\mathbf{\Omega}$ **dir**
- **end if**
- $\mathrm{dir}_x \leftarrow \sin(\theta)\cos(\phi)$
- $\mathrm{dir}_y \leftarrow \sin(\theta)\sin(\phi)$
- $\mathrm{dir}_z \leftarrow \cos(\theta)$

end loop

Here, $\xi([0,1])$ is a random number generator generating uniformly distributed random numbers over $[0,1]$. θ is the angle between ω' and $\overline{\omega}$. Therefore, we transform the spherical coordinates to Cartesian ones to ultimately rotate the vector with the rotation matrix $\mathbf{\Omega}$ to transform the sampling direction into the hemisphere's coordinate system, i.e., $\mathbf{\Omega}$ is constituted by an orthonormal basis where $\overline{\omega}$ represents the z axis. We note that random samples can be drawn efficiently and more than half of them are expected to fall inside the function.

C.2 Look-Up Tables (LUTs)

The following points should be considered regarding the generation of LUTs for the albedos and per-bounce variance estimates.

- Depending on the complexity and the number of parameters needed to control the scattering properties of a given material model, the corresponding LUT has to be parametrized accordingly, which can result in significant memory usage. Fortunately, the parametrization of each color channel can usually be avoided by parametrizing only one color channel instead, as material models usually perform the same set of transformations for each color channel separately. The retrieval of the respective quantity during the rendering process is then a simple matter of performing three lookups for each color channel (in the case of RGB triplets).

- Besides the material model parameters, the outgoing direction needs to be parametrized as well. Here, the parametrization according to $\cos(\theta)$ is preferred over the one based on θ considering the smaller footprint on the rendering. If we use θ the resulting higher resolution for grazing angles tends to be wasteful.

- Singular behaviors of material models should be considered carefully. Some material models might yield exceptionally high variance values at grazing angles where $(\omega_\mathrm{o} \cdot \mathbf{n})$ is near zero. Using such values for the LUT would compromise interpolations where the respective value is used. To alleviate this problem, we simply discard the problematic values and use extrapolations of the two following values instead.

C.3 Specular Refractive Materials

For the evaluation of the albedo map for specular refractive material models, the sampling manifold affords deterministic sampling as there are only two possible outcomes. Depending on the scene this approach might create a considerable overhead. However, the ultimate noise reduction performance for such surfaces can be greatly improved.

C.4 A Word on User-Chosen Parameters

Our noise filtering framework relies on the following parameters which can be chosen by the user:

δ The **albedo map error threshold** defines how much error is permissible during the computation of the albedo map. A large threshold raises the blend weights, given by (8) in the paper, in favor of the radiance samples, whereas small values cause the albedo values to dominate. In practice, a fixed small value can be used for a variety of cases. We used a value of 0.001 for all of our scenes.

ϵ Similarly, the **error bound** defines how much error is permissible during the filtering step. It is used to derive a filter kernel size as well as the blending weights between the unfiltered and filtered pixels. We have generated satisfactory results with values between 0.0025 and 0.01.

α_v The **variance fusion percentile** determines how much of the noise of the rendering is under- or overestimated by our BSDF variance estimate. A percentile of 0 causes the BSDF variance estimate to be matched to the lowest noise level in the empirical noise estimate. Conversely, a percentile of 100 causes the BSDF variance estimate to be matched to the highest empirical noise level.

α_b The **filter kernel percentile** determines the global size of the filter kernel for filters which are not capable of adaptive kernel sizes. The percentile is calculated based on the local kernel sizes described in Section 5.1 of the paper.