

General and Robust Error Estimation and Reconstruction for Monte Carlo Rendering

Pablo Bauszat¹, Martin Eisemann^{1,2}, Elmar Eisemann² and Marcus Magnor¹

¹Computer Graphics Lab, TU Braunschweig, Germany ²Delft University of Technology, Netherlands

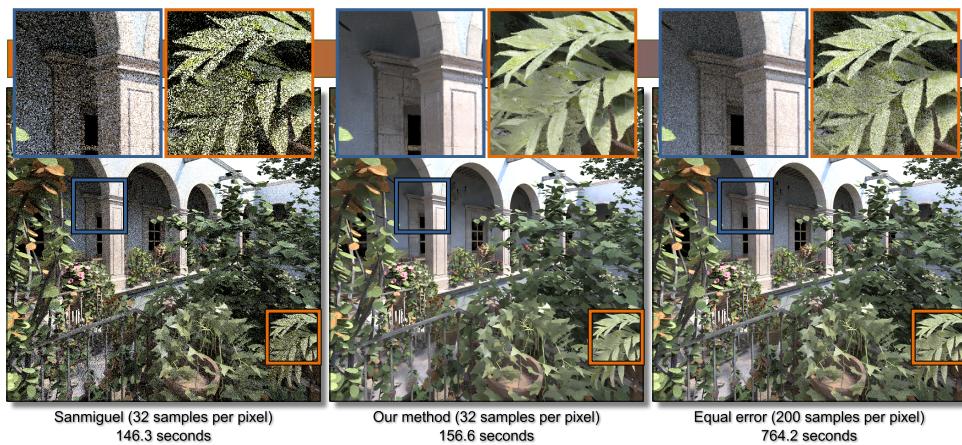


Figure 1: A complex scene with fine details and global illumination. Left: Images rendered with PBRT [PH10] using 32 samples per pixel rendered in 2.5 minutes. Middle: Image reconstructed by our algorithm in 2.6 minutes including rendering and filtering. Right: Equal error image with 200 samples per pixel rendered in 12.7 minutes.

Abstract

Adaptive filtering techniques have proven successful in handling non-uniform noise in Monte-Carlo rendering approaches. A recent trend is to choose an optimal filter per pixel from a selection of non spatially-varying filters. Nonetheless, the best filter choice is difficult to predict in the absence of a reference rendering. Our approach relies on the observation that the reconstruction error is locally smooth for a given filter. Hence, we propose to construct a dense error prediction from a small set of sparse but robust estimates. The filter selection is then formulated as a non-local optimization problem, which we solve via graph cuts, to avoid visual artifacts due to inconsistent filter choices. Our approach does not impose any restrictions on the used filters, outperforms previous state-of-the-art techniques and provides an extensible framework for future reconstruction techniques.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Raytracing

1. Introduction

Monte Carlo (MC) techniques have become an industry standard for high-quality image rendering [Kaj86]. Their robustness and simplicity are attractive properties for simulating various rendering effects including global illumination, depth of field, motion blur, participating media, and others.

Nonetheless, these procedures are computationally costly as thousands of samples of the rendering equation need to be evaluated for each pixel to reach a noise-free estimate of the lighting integral (Fig. 1).

Over the last decades, much research aimed at reducing the rendering cost by reconstructing (or filtering, used inter-

changeably throughout the paper) an image from a smaller number of samples. As the distribution of MC noise is usually spatially-varying (changes in geometry, lighting, or secondary effects), choosing an optimal reconstruction filter becomes a challenging per-pixel task. Recent approaches make use of a *filter bank*, a set of reconstructions using filters with different parameter settings [RKZ11, LWC12, KS13, RMZ13], from which one result is chosen per pixel to emulate spatially-varying and anisotropic kernels. Up to now, this selection has relied on risk estimators which suffer if input variance is high, or general image-noise classifications, which cannot robustly distinguish between noise and high-frequency image content.

Our contribution is a robust, low-variance selector for choosing the best possible reconstruction per pixel from an arbitrary set of general reconstruction techniques, based on two key observations. First, noise distributions within a small local window, although spatially-variant, change rather gently in most parts of a natural image. Thus, the reconstruction error of a filter candidate is in general locally smooth across the image and can be well approximated via an interpolation of sparse precise error estimates at carefully chosen locations within the image plane. Second, a suitable reconstruction for a low sample count is more effective than a mediocre reconstruction for a high sample count. Consequently, an algorithm to choose the best input from a given filter bank can be more important than spending computing time on higher sampling rates. Actually, for many samples, a suitable reconstruction is more important due to the convergence rate of the MC estimator and potential bias introduced in the reconstruction process.

We leverage these observations by reducing the number of samples spent on the noisy MC image and redistribute the remaining samples to create *filter caches*, which are highly sampled sparse image locations with strongly reduced variance. These filter caches serve as a robust, sparse error estimation for *any* reconstruction technique, and we can produce a dense error estimation via interpolation. Related to cross-validation techniques in statistics [Koh95] where samples are removed to validate the model fitness, the filter cache samples do *not* contribute to the filtering process. They are solely used to validate the fitness of the filters in the filter bank. Using fewer samples outside the filter caches leads to more variance in the filter input but the variance reduction due to the improved filter selection largely outweighs this downside.

Nonetheless, selecting filters per pixel solely on their local expected error will result in visible seams and outliers in the final image. Ad-hoc solutions, such as smoothing the filter-selection map, are only possible if the filter bank entries are semantically related, e.g. if they represent different parameter choices of the same filter. However, it becomes a challenge to support arbitrary filters within the filter bank.

Our solution is to solve the filter choice via a graph-cut approach by formulating it as a compositing task.

Our method is completely generic regarding image content and filtering techniques, as long as our two key assumptions hold. Hence, in contrast to previous approaches, we support arbitrary filter banks with no restrictions on differentiability (even non-filter reconstruction techniques are applicable) and most state-of-the-art denoising techniques for image and MC denoising can be utilized. The used filters do not need to be semantically related in any way, e.g. that neighboring indices in the filter bank have to refer to similar filter sizes, etc. The only requirement is that all filters in the filter bank operate on the same input, e.g., some filters may only be applicable to low dynamic range images while others operate on the original high dynamic range radiance values. Our approach requires fewer samples for higher quality reconstruction of MC renderings than many competitors. It is orthogonal to fundamental research on image and MC denoising and will support future reconstruction techniques as well. Finally, any MC effect can be used and the reconstruction quality solely depends on the filter bank.

In summary, our contributions are:

- An adaptive filter-cache selection based on sample and filter bank variances;
- A robust and dense error estimation to best choose from a filter bank;
- A global optimization removing visible seams between different filter selections.

2. Related Work

Denoising Noise removal for MC renderings is often inspired by image denoising. An early approach spread out the radiant energy in the image plane [McC99]. Later, bilateral filtering [XP05], non-local means [RKZ12], wavelet shrinkage [ODR09], guided image filter [BEM11], and $\tilde{\Lambda}$ -Trous wavelets [DSHL10] have been successfully applied. Specialized rendering filters sometimes make use of additional guides such as ray color distributions [DMB^{*}14] or features such as normals, texture, 3D positions [SD12]. While these approaches show impressive noise-reduction capabilities, they differ in their drawbacks, assumptions, number of required initial samples, or other constraints. Some are applicable in general [SD12, KS13, MYRD14], others focus on specific effects [BEM11, LAC^{*}11]. Choosing the *best* technique for every situation is still an open problem, especially, as the choice of an appropriate filter can vary within an image. Also, no robust statistic exists to *evaluate and compare* the reconstruction quality of arbitrary filters to make a good choice without knowledge of a reference. Further, even with such a classifier, severe visible artifacts can arise in form of seams wherever filters are switched. In this work, we introduce a general and robust approach to solve both of these problems.

Filter Selection Spatially-varying filters result in a high-quality reconstruction and are increasingly popular in the scientific domain [CWW^{*}11, RKZ11, LWC12, KS13, RMZ13, MCY14]. They reformulate the MC noise reduction problem into an error estimation and filter-selection problem. The Greedy Error Minimization [RKZ11] restricts the filter bank to Gaussian filters and guides the selection via an approximate bias term and the empirical sample variance. In the work of Li et al. [LWC12] and Rousselle et al. [RMZ13], Stein’s Unbiased Risk Estimator (SURE) [Ste81] is applied to MC denoising by exploiting the fact that the MC estimator itself resembles a normal distribution. SURE can estimate the Mean Squared Error (MSE) of a filter for a signal polluted by additive white Gaussian noise if the standard deviation of the noise is known and the filter is (weakly) differentiable. Unfortunately, computed locally for a single pixel, the error estimate can potentially be highly variant. A wavelet-based noise estimator for local filter parameter selection for a single type and varying parameters was presented in [KS13].

A common drawback of all previously-mentioned approaches is that the variance/MSE estimator is variant in itself, which is why all of them require a subsequent smoothing in a post-process. This heuristic can quickly lead to suboptimal filtering results. In contrast, our error estimation makes no assumption about the applied filter; even reconstruction techniques, which are not image-space filtering techniques in the strict sense [LAC^{*}11, LALD12, SD12] are supported.

Image Compositing Our filter-selection approach supports arbitrary reconstruction techniques, but, in consequence, color shifts (or seams) might appear between two neighboring areas where different filters are applied. Such problems are known from image-compositing tasks, such as panorama stitching [Sze06], digital photomontage [ADA^{*}04] or image-based rendering [BZS^{*}07]. Gradient-domain compositing removes color shifts between input images [PGB03] but come at the cost of a potentially different bias. One very successful approach is to formulate compositing as a labeling problem, which can be solved efficiently using graph cuts [BVZ01]. We adapt these approaches to our context and fuse different filter results in the final output.

3. Motivation

Our algorithm is based on several insights, which we will illustrate in this section. First, a good choice for a reconstruction filter is often more beneficial than increasing the number of samples. Second, it is possible to make coherent filter choices in many regions of the image without introducing a large error. This property is key to interpolating filter error estimates and will allow us to avoid seams due to filter changes. Our approach, described in the next section, will build upon these observations.

	Opt. 16 spp	Opt. 32 spp	SURE 32 spp
CONFERENCE	2.344	1.605	12.327
SIBENIK	0.258	0.157	0.758
TOASTERS	0.156	0.096	0.187
SANMIGUEL	9.831	6.419	16.880

Table 1: The table shows the MSE (10^{-3}) for the Sibenik, Conference, Toasters and Sanmiguel scene for 16 and 32 spp and reconstructed from a set of filters using optimal selection (Opt.) or SURE selection.

Optimal Filter Selection vs. Sampling Rate To show that in many cases it is more beneficial to choose appropriate filter settings instead of using a higher sampling count with mediocre reconstruction, we compare two different reconstruction techniques using the same filter bank. The filter bank consists of four Gaussian filters with $\sigma_{\text{domain}} = [2, 4, 8, 16]$ and four joint-bilateral filters with $\sigma_{\text{domain}} = [1, 2, 4, 8]$. Normals, world-space positions, and texture albedo colors are used as joint guides with $\sigma_{\text{normal}} = 0.8$, $\sigma_{\text{position}} = 0.6$, and $\sigma_{\text{texture}} = 0.25$. As a reference, we employ the SURE estimator from [LWC12] with 32 samples per pixel (spp), which is one of the current top-ranking selection techniques. For comparison, we test a reconstruction with 16 spp, for which we always chose the most optimal filter (determined by comparing to a reference image with 20000 spp). Further, we also tested an optimal reconstruction with 32 spp to determine an upper limit of the reconstruction quality. Table 1 depicts the MSE for several test scenes.

The optimal filter selection with just 16 spp reduces the error up to 81% (51% on average) compared to SURE using twice the number of samples. With an equal number of samples the optimal filter selection reduces the error up to 87% (69% on average). This finding illustrates the potential and importance of a good filter selection procedure.

Coherent Filter Selection Fig. 2 shows color-coded visualizations of the squared error for the Gaussian filter, the joint-bilateral filter [PSA^{*}04, ED04], non-local means filtering [BCM05], the BM3D filter [DFKE06] and the BLS-GSM filtering algorithm [PSWS03] for the SIBENIK scene and 16 spp. Additionally, the squared error for Guided Image Filtering [HST10] using varying radii is shown. We observe that the error of a filter varies rather smoothly for most regions of the image. An additional observation is that in many regions the error for more than one filter is close to the optimum, which is interesting because it shows that the filter selection is not always explicit. Instead, multiple candidates can be considered near optimal.

We examine the impact of a coherent filter selection compared to an optimal filter selection. Here, we deliberately chose non-optimal filters from the filter bank to enforce

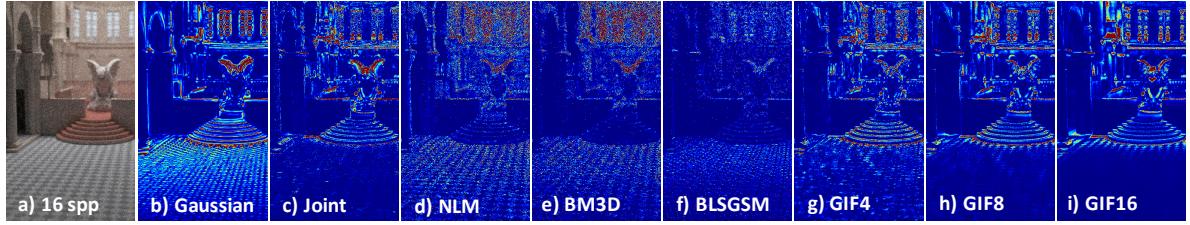


Figure 2: Error visualization for the SIBENIK scene with 16 spp (a) for different types of MC denoising filters and parameters. The first 5 insets show the squared errors for the (b) Gaussian ($\sigma = 8$) filter, (c) the joint-bilateral filter ($\sigma = 8$), (d) non-local means filtering (window size=16), (e) the BM3D filter ($\sigma = 0.8$) and (f) the BLS-GSM filter ($\sigma = 0.06$). The insets (g)–(i) show the squared errors for the Guided Image Filter with varying radii (4, 8, 16).

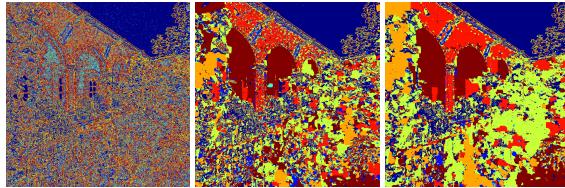


Figure 3: Filter selection maps with increasing coherency for the SANMIGUEL scene for a filter bank with 8 filters. The MSE reduction of the coherent selections is only slightly less compared to the optimal selection.

spatially-consistent filter choices but restricted the overall solution to have a defined maximum error.

Fig. 3 shows three filter selection maps with varying coherency for the SANMIGUEL scene using a similar filter bank as in Table 1. Each color represents one entry in the filter bank used for reconstruction of the final image. The optimal per-pixel selection (on the left) reduces the error to 8.0% of the MSE of the noisy image. Coherent selections still result in a low overall error of 8.4% (middle) and 9.1% (right). Similar observations have been made with other test scenes (see supplemental material for more details).

It shows that the optimal filter selection map is comparably noisy, but in large regions, the filter selection can be made coherent without introducing significant errors. However, for specific regions the filter selection is indeed crucial. These findings imply, that a filter selection focusing only at sparse, but carefully chosen pixel locations can be sufficient for a good filter selection across the whole image.

4. Error estimation and Filter Selection

Based on the insights of the previous section, we introduce a filter-selection process for a general filter bank in order to benefit from a better reconstruction, which uses the following input: A noisy MC rendering \mathbf{N} computed from a user-defined number of samples per pixel, a filter bank \mathbb{F} consisting of the results $\mathbf{F}_0, \dots, \mathbf{F}_m$ of different reconstruction techniques (filtered images) using \mathbf{N} as input, and access to

the renderer itself to compute additional samples to produce filter caches. We stress that any filter bank could be used and refer the interested reader to the according publications for more details. Still, we assume a set of reasonable filters and settings, which means that for most pixels a choice exists, which represents an improvement over the initial MC estimate.

To reach our goal, we estimate the reconstruction error of each entry in \mathbb{F} at a small number of pixels (Sec. 4.1) and show how to optimize their locations (Sec. 4.2). The sparse estimates are interpolated based on a smoothness assumption (Sec. 4.3) to derive per-pixel error estimations. These will then be used as input to a labeling process to choose the optimal entry in \mathbb{F} per pixel. The latter is solved via a graph-cut approach (Sec. 4.4) to avoid visual artifacts due to inconsistent or inappropriate filter choices. An overview of our algorithm is given in Fig. 4.



Figure 4: Algorithm overview

4.1. Filter Caches

Our approach is inspired by ir-radiance caching algorithms where the incident indirect lighting is computed for a small subset of pixels (the caches) and later interpolated. In this spirit, we compute a high-quality radiance estimate for a small subset of pixels. To obtain these so-called *filter caches*, more samples are computed and because the MC error initially decreases quickly [PH10], even a slightly elevated number of samples leads to a significant improvement of the incoming radiance estimate. In consequence, it is possible to obtain a good error estimate Err at a pixel cache location p with value $\mathbf{C}(p)$ for any $\mathbf{F}_i \in \mathbb{F}$

$$Err(p, \mathbf{F}_i) \approx \|\mathbf{F}_i(p) - \mathbf{C}(p)\|.$$

To roughly maintain the overall rendering cost, we create two sample sets out of the sample budget based on user-defined parameters, one part used for uniformly sampling

the image plane to create the filter input \mathbf{N} and the rest to compute the cache entries. Given the initial per-pixel sample budget b , a cache sparsity $s \in [0, 1]$ (a value of 0 places a cache at each pixel and a value of 1 results in no caches at all), and the number of samples per pixel n used to compute \mathbf{N} , the additional samples per cache c are given by $\frac{b-sn}{1-s}$. To acquire a robust radiance estimate close to the reference, as we treat $\mathbf{C}(p)$ as ground truth, the sparsity has to be high; we used between 0.85 and 0.98 of sparsity in our test scenes.

4.2. Filter Cache Placement

As our intermediate goal is to interpolate the error values between the caches for each filter, a good placement is crucial. For an even spread, we can distribute their location according to a blue-noise power spectrum [DW85]. Nonetheless, to better capture error variations, more caches should be placed in regions with varying error, which are unknown.

Instead, we base our adaptive cache placement strategy on three insights. First, the variance within \mathbb{F} for a pixel p indicates how crucial a good filter selection is. If the variance is high, a wrong filter choice will introduce a large error. Vice versa, if all \mathbf{F}_i are the same, the choice is unimportant. Second, pixels with already low variance in the MC estimator for \mathbf{N} are likely to provide a more robust radiance estimate with less residual noise for the cache's sampling rate. Third, the overall image domain Ω should be roughly covered with a maximum distance between the caches.

Following the first two insights, we compute a joint probability distribution function (PDF) $\mathbf{P}_{\mathbb{F}\mathbf{N}}$ to drive the cache positioning:

$$\mathbf{P}_{\mathbb{F}\mathbf{N}}(p) = \frac{\mathbf{P}_{\mathbb{F}}(p) \cdot \mathbf{P}_{\mathbf{N}}(p)}{\sum_{p \in \Omega} \mathbf{P}_{\mathbb{F}}(p) \cdot \mathbf{P}_{\mathbf{N}}(p)}, \quad (1)$$

where $\mathbf{P}_{\mathbb{F}}$ and $\mathbf{P}_{\mathbf{N}}$ are the PDFs for importance sampling based on the filter bank variance and the MC variance of \mathbf{N} . We use the per-pixel filter bank variance directly as PDF and set $\mathbf{P}_{\mathbb{F}}(p) := \mathbf{F}_{\sigma^2}(p) = \frac{1}{m} \sum_{i=1}^m (\mathbf{F}_i(p) - \mathbf{F}_\mu(p))^2$, where \mathbf{F}_μ is the per-pixel mean of the filter bank. For $\mathbf{P}_{\mathbf{N}}$, we use a Gaussian probability model and define

$$\mathbf{P}_{\mathbf{N}}(p) = \frac{1}{\sigma_r \sqrt{2\pi}} e^{-\frac{\mathbf{N}_{\sigma^2}(p)}{2\sigma_r^2}},$$

where \mathbf{N}_{σ^2} is the variance of the MC estimator. Although \mathbf{N}_{σ^2} is unknown, it can be approximated by the empirical sample variance for each pixel using n samples (cf. [RKZ11, LWC12]). Here, σ_r is a global, user-defined parameter, which we set to 0.15 for all our scenes.

Directly sampling the total number of caches m_{total} via $\mathbf{P}_{\mathbb{F}\mathbf{N}}$ leads to cluttered cache locations and potentially larger image regions without caches, which leads us to the third insight. We use our previous importance sampling strategy to produce $m_{\text{importance}} := \kappa \cdot m_{\text{total}}$ caches and $m_{\text{poisson}} := m_{\text{total}} - m_{\text{importance}}$ caches using standard Poisson sampling.

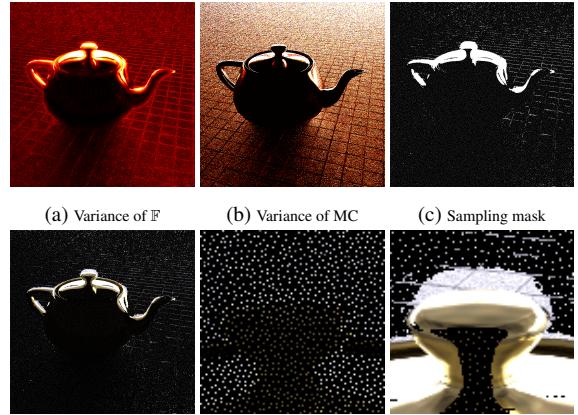


Figure 5: Adaptive cache placement. (a) Magnitude of the variance of the filter bank (b) Magnitude of the variance of the MC estimator (c) Binary mask of the representative pixels using our importance sampling approach and (d) the representative radiance values. (e) Close-up with standard Poisson sampling and (f) with our proposed combination of Poisson and importance sampling.

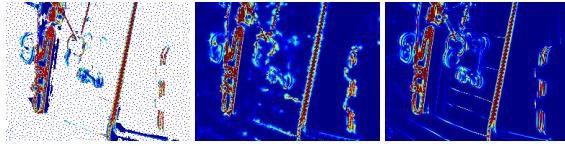
Here, $\kappa \in [0, 1]$ is a user-defined number, which balances between both strategies.

To avoid duplicate caches, we first draw m_{poisson} samples from the Poisson distribution and remove the sampled pixels from the PDF computation of the importance sampling given by Eq. (1). Afterwards, we draw the $m_{\text{importance}}$ samples from the resulting PDF using 2D importance sampling [PH10]. As the parameter κ is set only once, the Poisson distribution can be precomputed. A similar sampling could be achieved with a pure variable density-based Poisson Distribution, e.g. [KS11], but these are often costly to compute. Our approach is cheap as the uniform Poisson samples can be precomputed.

An example of our approach is given in Fig. 5; the samples of our importance sampling approach visibly gather around object boundaries and high frequency edges, as these are often difficult to reconstruct for many filters.

4.3. Dense error reconstruction

Given the error estimation at the cache locations for each $\mathbf{F}_i \in \mathbb{F}$, we want to estimate a dense error for all remaining pixels, which will be the input to our filter selection approach. We compared several reconstruction techniques of sparsely sampled images, including PDE-based inpainting [BSCB00], Total-Variation inpainting [CS02], and ACT [FGS95]. An excellent comparison study for several image-based sparse-reconstruction techniques has been presented in [SD11]. The study shows that Compressed Sensing and Delaunay triangulation are the best choices for sparsely-distributed samples with a high degree of sparsity. As we



(a) Sparse squared error (b) Inpainted squared error (c) Reference squared error
Figure 6: From a sparse estimation of the squared error, we use an inpainting based on Delaunay triangulation to compute a dense estimation, which resembles the reference squared error. Note the uniform areas are sampled with fewer caches by our adaptive cache placement.

have to perform multiple interpolations for each \mathbf{F}_i , we chose Delaunay triangulation because it provides a good tradeoff between quality and performance. Further, once computed, it can be re-used for each filter in the bank.

We use the square of the L2-norm as error distance metric (Fig. 6) and interpolate the error of each reconstruction technique \mathbf{F}_i between the caches to create a dense error map \mathbf{D}_i for each filter.

4.4. Filter Composite

Given the dense error estimate \mathbf{D}_i , a straightforward solution to minimize the MSE of the final result would be to select the filter \mathbf{F}_i with the lowest estimated error per pixel, but this may lead to visual artifacts in form of seams. Instead, we write the problem of selecting the optimal filter per-pixel as a multi-labeling optimization problem. An optimal labeling $\mathbf{L} : \Omega \rightarrow 1 \dots m$ can be found by minimizing the energy

$$E(\mathbf{L}) := E_{\text{Data}}(\mathbf{L}) + \lambda \cdot E_{\text{Smoothness}}(\mathbf{L}), \quad (2)$$

λ allows us to balance between data and smoothness terms, which are defined as follows;

$$E_{\text{Data}}(\mathbf{L}) := \sum_{p \in \Omega} \mathbf{D}_{\mathbf{L}(p)}(p) \quad (3)$$

$$E_{\text{Smoothness}}(\mathbf{L}) := \sum_{\{p,q\} \in \mathcal{N}} V(p, q, \mathbf{L}(p), \mathbf{L}(q)), \quad (4)$$

where \mathcal{N} is the set of interacting pairs of pixels and $V(p, q, \mathbf{L}(p), \mathbf{L}(q))$ is a label cost function. We follow Agarwala et al. [ADA*04] and define this cost function to match color and gradients of neighboring pixels:

$$V(p, q, \mathbf{L}(p), \mathbf{L}(q)) = X + Y$$

with

$$X = \|\mathbf{F}_{\mathbf{L}(p)}(p) - \mathbf{F}_{\mathbf{L}(q)}(p)\| + \|\mathbf{F}_{\mathbf{L}(p)}(q) - \mathbf{F}_{\mathbf{L}(q)}(q)\|$$

$$Y = \|\nabla \mathbf{F}_{\mathbf{L}(p)}(p) - \nabla \mathbf{F}_{\mathbf{L}(q)}(p)\| + \|\nabla \mathbf{F}_{\mathbf{L}(p)}(q) - \nabla \mathbf{F}_{\mathbf{L}(q)}(q)\|$$

where $\nabla \mathbf{F}_i(p)$ is the (horizontal and vertical) gradient of the filtered image \mathbf{F}_i at p . Eq. (2) can be solved efficiently within a known factor of the global minimum using a graph-cut optimization [BVZ01].

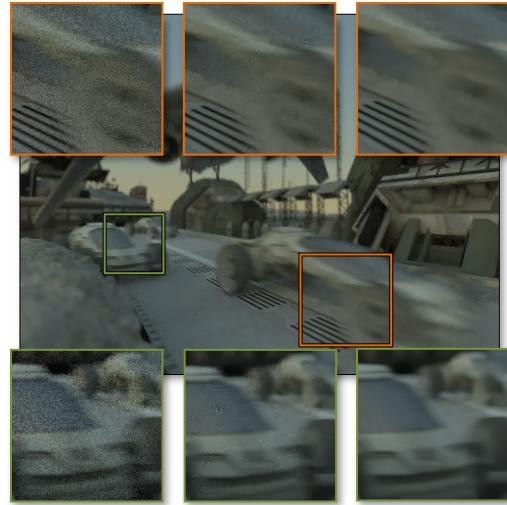


Figure 7: Noisy image with 32 spp (left inset), the result using only an interpolated local error estimation (middle inset) and the graph-cut version (right inset) of the SCIFI scene.

Fig. 7 shows the effect of the global filter optimization. Local per-pixel filter selection leads to small erroneous patches where neighboring labels differ and creates visually-disturbing artifacts, which are robustly removed by our graph-cut approach.

Gradient-Domain Fusion In image-stitching applications [Sze06, ADA*04], it has become common practice to add a final Poisson integration step to adjust colors along the seams of neighboring regions. For very low sampling rates, it also smoothes out juxtaposed filter regions. We use two Jacobi iterations of the Poisson solver to smooth the most visible seams without affecting the overall MSE.

Additionally, the Poisson formulation can be used to enforce the filter-cache radiance as a constraint for the image reconstruction. In practice, for the low sample count that we target, the remaining variance in the caches is usually similar to the reconstruction. Consequently, integrating the caches proved counterproductive.

5. Results

We implemented larger parts of our method in MATLAB R2014b without multi-threading. For the multi-label graph-cut solver, we use the algorithm proposed in [BVZ01] and implemented it in NVIDIA’s CUDA 6.5 on top of the binary graph-cut implementation provided by the CUDA NPP library. We also implemented the joint-bilateral filter with the modified distance function from [LWC12] using CUDA. For the BM3D filter [DFKE06] and the modified non-local means filter by Rousselle et al. [RMZ13], we used the original implementations. All statistics were measured on an In-

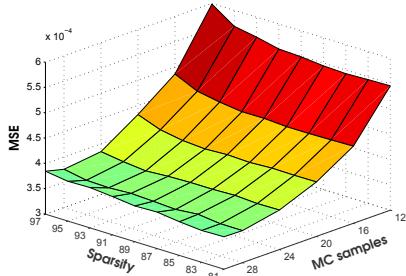


Figure 8: Parameter evaluation. While the overall sampling budget is fixed the sampling rate for the noisy input image and the number of filter caches is varied.

tel Core i7-2600, 3.40 GHz and 16 GB RAM PC with an NVIDIA GeForce 780 GTX running on Windows 7, 64-bit.

The test scenes and the input and reference data were created using the PBRT2 system [PH10]. We used nine test scenes for our evaluations and comparisons (resolution in pixels is given in brackets) - SANMIGUEL(1024x1024), SIBENIK (1024x768), TEAPOT (800x800), TOASTERS (512x512), CHESS (750x1000), POOLBALL (1024x1024), DRAGON (1024x1024), CONFERENCE (1024x1024) and SCIFI (1024x768). The scenes cover a variety of MC effects including global illumination, depth of field, motion blur, glossy materials and participating media. All reference solutions have been computed with 20000 spp.

5.1. Parameter and Error Evaluation

In the following, we evaluate the influence of the different parameters of our approach to derive an optimal setting used in all the following results. Additionally, we investigate the error of our method compared to the optimal filter selection. For all experiments, we set a total sample budget of 32 samples times the number of pixels. The samples used for \mathbf{N} are uniformly distributed among the image in our approach.

Sampling and Sparsity We start by evaluating the influence of the number and quality of filter caches. To find the optimal sample distribution, we vary the number of samples used for \mathbf{N} , as well as the number of filter caches. The results for the SIBENIK scene are shown in Fig. 8. The trade-off between sparsity s and cache sampling rate c has in general a lower impact on the MSE in comparison to varying the number of samples for \mathbf{N} . This shows that the reduction of the interpolation error from lower sparsity is outweighed by the increase in the variance of the cache radiance estimates. Optimal parameters are achieved between three-fourths and seven-eighth samples assigned to \mathbf{N} and a sparsity of approximately 95%, which results in a mean distance of roughly four pixels between the caches.

Approximation Error & Adaptive Cache Placement We evaluate the influence of adaptive cache placement vs. a

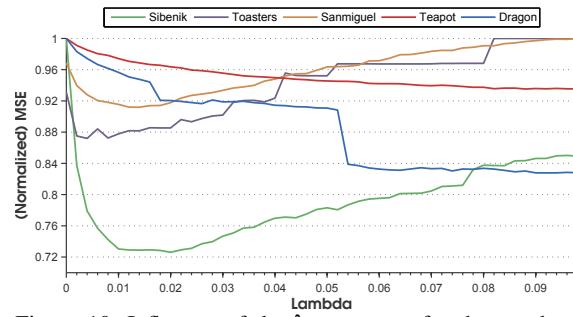


Figure 10: Influence of the λ parameter for the graph-cut filter selection (using the parameters from Fig. 9). Choosing the *optimal* λ reduces the overall MSE for SIBENIK (73%), TOASTERS (87%), SANMIGUEL (91%), TEAPOT (94%) and DRAGON (83 %) in comparison to un-regularized selection.

blue-noise distribution for several test scenes. To this extent, we vary the parameter κ to interpolate between the two extremes as described in Sec. 4.2. In addition, we are interested in the significance of the two possible error sources of our approach; interpolation errors and errors introduced by residual variance in the cache radiance values. To gain further insights into each of them, we also measure the MSE when using ground truth radiance at the caches instead of the variant estimates $\mathbf{C}(p)$.

Adaptive placement consistently outperforms uniform placement and our mixed importance sampling decreases the overall MSE down to 90% for the SIBENIK scene, 84% for the TOASTERS scene, 63% for SANMIGUEL, 86% for the TEAPOT scene and even down to 43% for the DRAGON scene. We used values of 0.5 – 0.7 for κ in our scenes.

As expected, interpolation from sparse caches is our main source of error in most scenes (72% on average) when compared to an optimal filter selection (Fig. 9), while error from residual noise in the caches is comparably small (28% on average). An exception is the DRAGON scene where interpolation works exceedingly well due to large homogeneous image regions and residual cache variance contributes more strongly to the overall error (80% on average).

Regularization We varied the smoothness parameter λ , controlling the influence of the smoothness term in the graph-cut labeling, Eq. (2), and evaluated the MSE. The regularized version decreases the overall error compared to the non-regularized version ($\lambda = 0$) for all scenes up to 27% (Fig. 10). Hereby, small filter patches in the resulting images are removed, which otherwise could appear as visible artifacts (Fig. 7). The graph cut operates on the color values of \mathbb{F} , hence, the optimal choice of λ depends on the dynamic range of the scene radiance. When λ is chosen too large, the error is increased again due to over-smoothing of the final labeling.

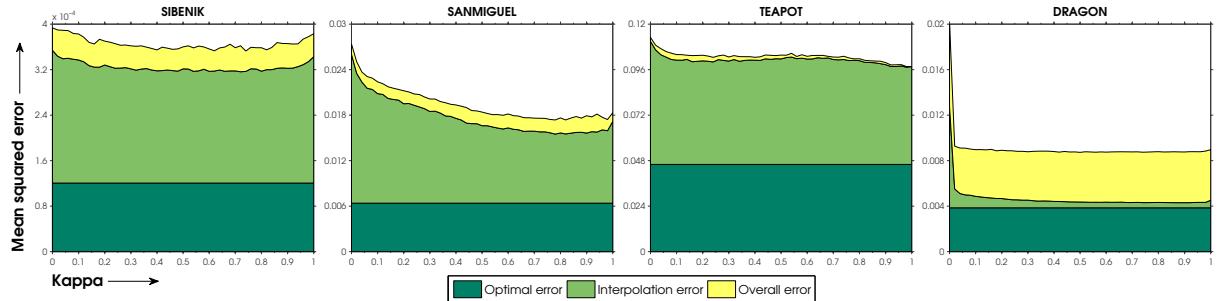


Figure 9: Influence of κ parameter for the adaptive cache sampling strategy with 16 spp used for \mathbf{N} , 95% sparsity, resulting in 176 spp for each cache entry, and the filter bank from Sec. 3. The error from an optimal filter selection (dark green) is compared to the error due to interpolation (using ground truth radiance at the caches, shown in green) and the overall error (including interpolation and residual noise in the caches, shown in yellow). The mixed importance sampling decreases the overall MSE for all test scenes: SIBENIK (90%), SANMIGUEL (63%), TEAPOT (86%), DRAGON (43%).

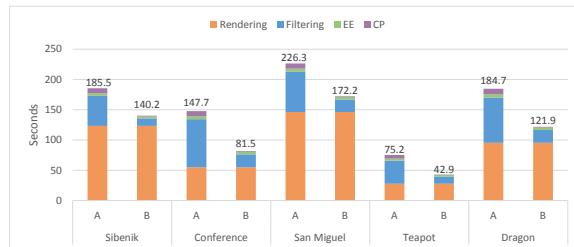


Figure 11: Timings for rendering (with 32 spp), filter bank creation (A uses 8 joint-bilateral filters, B uses 4 BM3D filters), error estimation (EE) including cache location sampling and dense interpolation, and compositing (CP) including graph cut and Poisson integration. All timings are in seconds.

5.2. Timings

We evaluate the runtime of our method in Fig. 11 using 32 spp and 90% sparsity. It can be seen that our method accounts for only a small portion of the total time, 3% - 6% for the SIBENIK scene, 6% - 9% for CONFERENCE, 3% - 6% for SANMIGUEL, 8% - 12% for TEAPOT and 4% - 7% for the DRAGON scene. Most time is consumed by the rendering process and the filter-bank creation. Our method's most costly aspect is the graph-cut solver, which can be seen in the increase of the compositing time between 4 and 8 filters, due to the quadratic complexity in terms of labels.

5.3. Comparisons

We compared our technique to a variety of state-of-the-art MC denoising techniques. For all comparisons, we used the original source codes kindly provided by the respective authors. All parameters of these techniques were set to values proposed in the respective publications and the same overall sample budget was used for a fair comparison.

SURE To evaluate the quality of our filter selection, we compare our technique to SURE-based filter selection which was proposed first for MC denoising in [LWC12]. Because SURE only works for differentiable filters, we use a filter set consisting of four joint-bilateral filters (using similar parameters settings as [LWC12]) for both SURE and our approach for a fair comparison. We use 32 spp for the SURE method to create the filter bank, while our method uses only 28 spp for the noisy estimate \mathbf{N} , from which the filter bank is constructed. Both noisy images are created with uniform sampling. We distribute the rest of the samples to the cache pixels, which we create with 95% sparsity, i.e. each cache is computed from 108 samples. Results for the SIBENIK and CONFERENCE scenes are shown in Fig. 12.

For both scenes our technique has fewer visible artifacts and an overall MSE reduction of up to 54% for the SIBENIK scene and 64% for CONFERENCE compared to the SURE-based selection. Even when choosing the best filter only locally without the graph-cut and Poisson-integration step, we still achieve an improvement in MSE by 43% (SIBENIK) and 47% (CONFERENCE). Potentially, our results could be improved even further by using higher quality filters, including non-differentiable ones, in \mathbb{F} .

GID We compare our method to the General Image Denoising (GID) framework [KS13], a framework for adaptive filtering and variance estimation (Fig. 13). GID uses a wavelet-based noise metric to estimate the standard deviation of the noise per pixel and then selects an optimal filter from a series of high-quality filters (BM3D or BLS-GSM). The BM3D and BLS-GSM are only applicable to low dynamic range images, which is why images have been tone mapped beforehand using a gamma correction. The test sequences are the TOASTERS, CHESS and POOLBALL scene using BM3D filters and the parameters suggested by the authors. For our method, we create a filter set consisting of four BM3D filters with uniformly distributed parameters, while GID even optimized the filter parameters for each scene. Additionally,

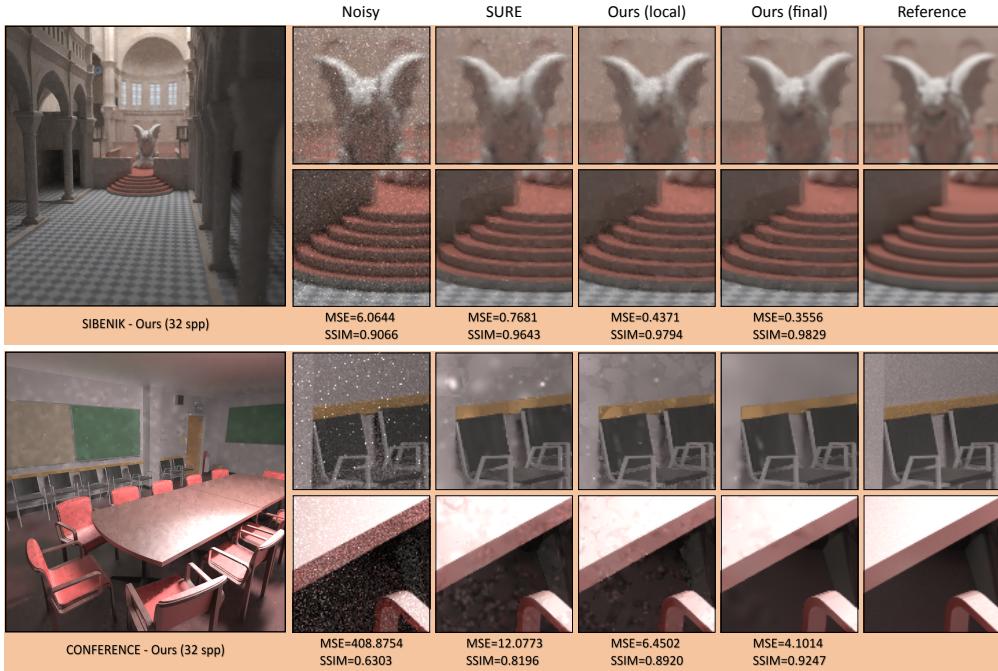


Figure 12: Comparison with SURE [LWC12]. Top: Reconstructed result and associated filter selection map. All reconstruction results, except for the reference solution, use the same sample budget. MSE is scaled by 10^3 .

GID performs adaptive sampling. We tested configurations with 8, 16 and 32 spp on average. For all 3 scenes, we used a sparsity of 95%.

We achieve an MSE reduction of 13% - 64% (51% on average). The GID approach has problems distinguishing between noise and high frequencies in the image signal and smoothes over them. This shortcoming is visible in areas with high-frequency textures (e.g. in the chess scene, second row).

Similar to previous approaches, the GID estimator suffers from variance for low sampling rates which forces the approach to heuristically smooth the noise map before filter selection. The framework supports arbitrary image-denoising filters, however, they only estimate the variance of the noise and not the MSE itself. Therefore, only results for one family of filters at a time have been shown. The GID algorithm has no means to compare the reconstruction quality of different filters directly. Our approach differs in this sense, as we estimate the MSE directly per filter and are able to compare arbitrary reconstruction results.

RD In Fig. 14, we compare our method with the Robust Denoising (RD) framework [RMZ13], which is the currently best-performing reconstruction technique of all tested approaches. RD uses three specialized non-local means filters with different sensitivity to the image colors. Additionally, it uses a tailor-made filter selection algorithm for these three

filters based on SURE. We use the same three filters for our filter bank.

The results shown in Fig. 14 for both approaches are quite similar. For 16 spp (32 spp), our technique shows a MSE reduction of up to 25% (16%) for the DRAGON scene compared to RD. On the TEAPOT scene, our approach performs slightly worse and the MSE increases by 14% (13%). For the CONFERENCE scene, both approaches yield similar results, with an MSE decrease by 9% (7%) using our method. A possible explanation could be that a sparse cache sampling potentially misses peaks in the highly glossy material of the teapot, resulting in worse error estimates in the dense error maps. However, it should be noted that our approach is a general framework for arbitrary reconstruction techniques, whereas RD is a specifically customized approach. Our images show slightly more noise compared to the results presented in [RMZ13] as we omit their final filtering step for a more direct comparison.

Though not tested yet, we could potentially use the final results from the SURE, GID and RD frameworks for our input and locally choose an optimal one, which illustrates the flexibility of our approach.

6. Discussion

Based on our experiments from Sec. 5, using filter caches appears to be a fruitful research direction to develop more

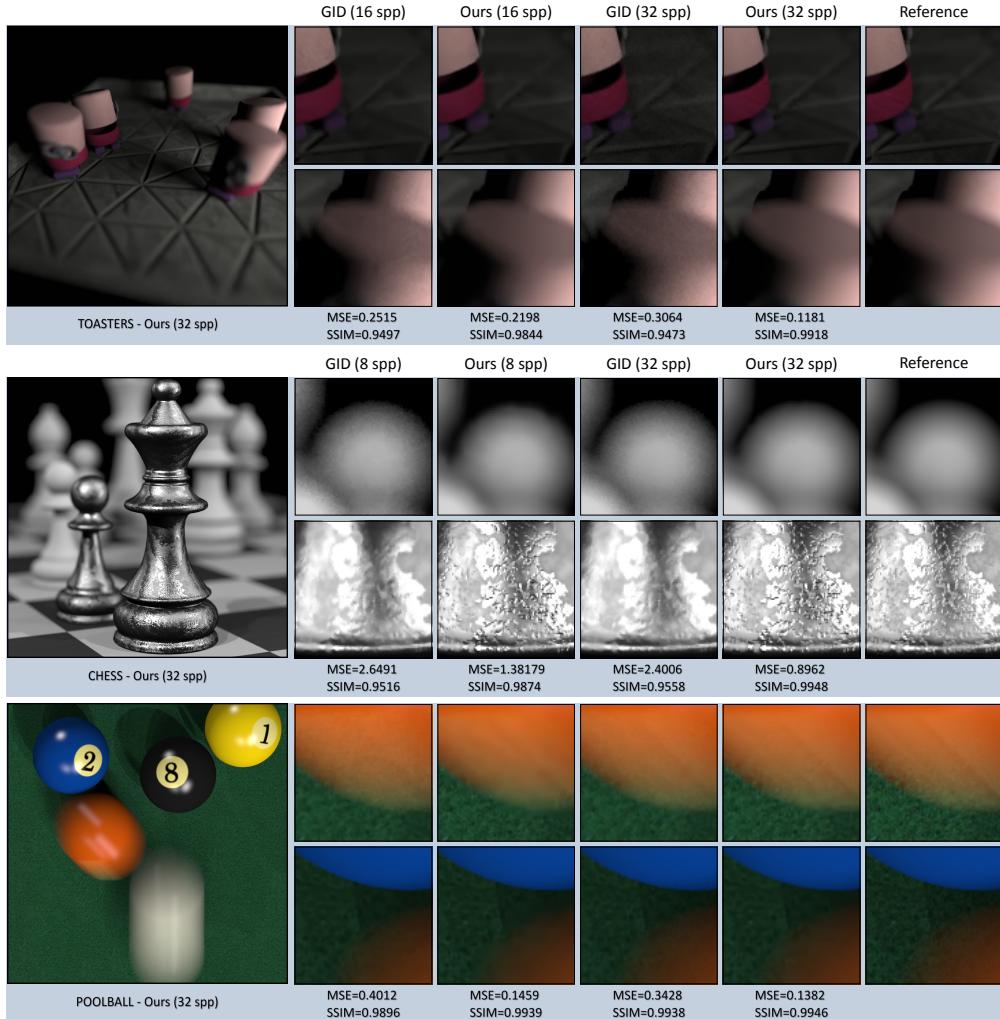


Figure 13: Comparison with the General Image Denoising (GID) framework [KS13] for 8, 16 and 32 spp on average. Our approach uses four different BM3D filters. The MSE is scaled by 10^3 .

robust error estimators as our solution outperforms many competitors. It also shows that sample count alone can be much less crucial than appropriate settings for the reconstruction method. This finding indicates that solutions like ours, which opt at making better filter choices, have great potential to improve image reconstruction in the future.

A limiting factor can be residual noise in the filter caches, or an inferior error interpolation, which can both affect the quality of our filter selection. Both problems could potentially be tackled by compressed-sensing [SD11] in combination with cross-validation [BDB07], but we currently opted for a more computationally efficient solution.

Adaptive MC sampling is often based on intermediate reconstruction results to steer the distribution of samples. Our approach offers many degrees of freedom, as samples can be

added to the caches, be integrated in our noisy estimate \mathbf{N} , or can even be used to create new caches. The gain could be high, but such measures make the problem also substantially more complex.

In the current state, our adaptive placement of caches tends to avoid selecting outliers stemming from the MC rendering process as caches, which are, unfortunately, even preserved by some filters (e.g., BM3D or BLS-GSM). This situation violates our assumption of local error smoothness, although it is more a filter limitation. Further, preprocessing the individually-filtered images with a spike-noise reduction should alleviate these problems.

So far, we did not focus on temporal data, and no coherence is guaranteed. The reason is mostly that the evaluated reconstruction techniques do not explicitly tackle an-

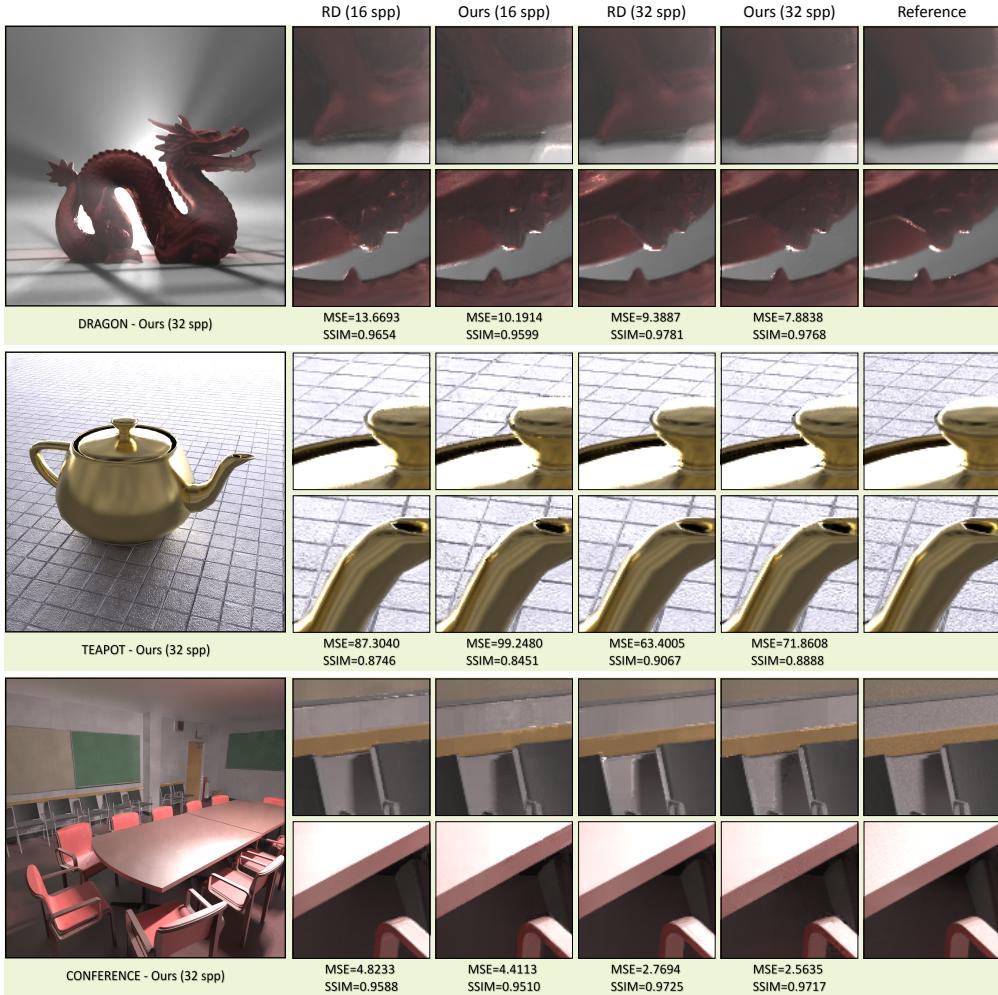


Figure 14: Comparison to the Robust Denoising (RD) technique [RMZ13] for 16 and 32 spp on average. For all scenes three customized non-local means filters are used. MSE is scaled by 10^3 .

imations. An interesting step in this direction might be to extend our graph-cut technique to the time dimension.

It would also be promising to use filter combinations instead of a single filter per pixel. Such an extension would broaden the solution space and could improve reconstruction.

7. Conclusion

We have presented a way to estimate the error of arbitrary filtering and reconstruction techniques for MC renderings and how to exploit this estimate to select a close-to-optimal filter per pixel. We introduced filter caches, sparse high sample locations within the image plane, to robustly estimate the error. Our adaptive cache placement based on the variances within the filter bank and MC estimator conservatively reduces the threat of choosing wrong filters at crucial image positions.

The freedom to basically use any reconstruction techniques, in combination with our robust error estimator, significantly improves the reconstruction results over state-of-the-art approaches.

References

- [ADA*04] AGARWALA A., DONTCHEVA M., AGRAWALA M., DRUCKER S., COLBURN A., CURLESS B., SALESIN D., COHEN M.: Interactive digital photomontage. In *ACM SIGGRAPH 2004 Papers* (2004), SIGGRAPH '04, pp. 294–302. 3, 6
- [BCM05] BUADES A., COLL B., MOREL J.-M.: A non-local algorithm for image denoising. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)* (2005), CVPR '05, pp. 60–65. 3
- [BDB07] BOUFOUNOS P., DUARTE M. F., BARANIUK R. G.: Sparse signal reconstruction from noisy compressive measurements using cross validation. *Statistical Signal Processing, IEEE/SP Workshop on* 0 (2007), 299–303. 10

- [BEM11] BAUSZAT P., EISEMANN M., MAGNOR M.: Guided image filtering for interactive high-quality global illumination. *Computer Graphics Forum (Proc. of Eurographics Symposium on Rendering (EGSR))* 30, 4 (June 2011), 1361–1368. [2](#)
- [BCSB00] BERTALMIO M., SAPIRO G., CASELLES V., BALLESTER C.: Image inpainting. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques* (2000), SIGGRAPH '00, pp. 417–424. [5](#)
- [BVZ01] BOYKOV Y., VEKSLER O., ZABIH R.: Fast approximate energy minimization via graph cuts. *IEEE Trans. Pattern Anal. Mach. Intell.* 23, 11 (2001), 1222–1239. [3, 6](#)
- [BZS*07] BHAT P., ZITNICK C. L., SNAVELY N., AGARWALA A., AGRAWALA M., COHEN M., CURLESS B., KANG S. B.: Using photographs to enhance videos of a static scene. In *Proceedings of the 18th Eurographics Conference on Rendering Techniques* (2007), EGSR'07, pp. 327–338. [3](#)
- [CS02] CHAN T. F., SHEN J.: Mathematical models for local nontexture inpaintings. *SIAM J. Appl. Math.* 62 (2002), 1019–1043. [5](#)
- [CWW*11] CHEN J., WANG B., WANG Y., OVERBECK R. S., YONG J.-H., WANG W.: Efficient depth-of-field rendering with adaptive sampling and multiscale reconstruction. *Comput. Graph. Forum* 30, 6 (2011), 1667–1680. [3](#)
- [DFKE06] DABOV K., FOI A., KATKOVNIK V., EGIAZARIAN K.: Image denoising with block-matching and 3d filtering. In *IN ELECTRONIC IMAGING'06, PROC. SPIE 6064, NO. 6064A-30* (2006). [3, 6](#)
- [DMB*14] DELBRACIO M., MUSÉ P., BUADES A., CHAUVIER J., PHELPS N., MOREL J.-M.: Boosting monte carlo rendering by ray histogram fusion. *ACM Trans. Graph.* 33, 1 (2014), 8:1–8:15. [2](#)
- [DSHL10] DAMMERTZ H., SEWTZ D., HANIKA J., LENSCHE H. P. A.: Edge-avoiding a-trous wavelet transform for fast global illumination filtering. In *Proc. of the Conference on High Performance Graphics* (2010), pp. 67–75. [2](#)
- [DW85] DIPPÉ M. A. Z., WOLD E. H.: Antialiasing through stochastic sampling. *SIGGRAPH Comput. Graph.* 19, 3 (1985), 69–78. [5](#)
- [ED04] EISEMANN E., DURAND F.: Flash photography enhancement via intrinsic relighting. *ACM Transactions on Graphics* 23, 3 (2004), 673–678. [3](#)
- [FGS95] FEICHTINGER H. G., GRÖCHENIG K., STROHMER T.: Efficient numerical methods in non-uniform sampling theory. *Numer. Math.* 69, 4 (Feb. 1995), 423–440. [5](#)
- [HST10] HE K., SUN J., TANG X.: Guided image filtering. In *Proc. of the European Conference on Computer Vision* (2010), vol. 1, pp. 1–14. [3](#)
- [Kaj86] KAIJYA J. T.: The rendering equation. In *Conference on Computer graphics and interactive techniques* (1986), SIGGRAPH, pp. 143–150. [1](#)
- [Koh95] KOHAVI R.: A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence* (1995), IJCAI'95, pp. 1137–1143. [2](#)
- [KS11] KALANTARI N. K., SEN P.: Efficient computation of blue noise point sets through importance sampling. *Computer Graphics Forum* 30, 4 (2011). [5](#)
- [KS13] KALANTARI N. K., SEN P.: Removing the noise in monte carlo rendering with general image denoising algorithms. *Computer Graphics Forum* 32, 2pt1 (2013), 93–102. [2, 3, 8, 10](#)
- [LAC*11] LEHTINEN J., AILA T., CHEN J., LAINE S., DURAND F.: Temporal light field reconstruction for rendering distribution effects. *ACM Trans. Graph.* 30, 4 (2011). [2, 3](#)
- [LALD12] LEHTINEN J., AILA T., LAINE S., DURAND F.: Reconstructing the indirect light field for global illumination. *ACM Trans. Graph.* 31, 4 (2012). [3](#)
- [LWC12] LI T.-M., WU Y.-T., CHUANG Y.-Y.: Sure-based optimization for adaptive sampling and reconstruction. *ACM Trans. Graph.* 31, 6 (Nov. 2012), 194:1–194:9. [2, 3, 5, 6, 8, 9](#)
- [McC99] MCCOOL M. D.: Anisotropic diffusion for monte carlo noise reduction. *ACM Trans. Graph.* 18, 2 (1999), 171–194. [2](#)
- [MCY14] MOON B., CARR N., YOON S.-E.: Adaptive rendering based on weighted local regression. *ACM Trans. Graph.* 33, 5 (Sept. 2014), 170:1–170:14. [3](#)
- [MYRD14] MEHTA S. U., YAO J., RAMAMOORTHI R., DURAND F.: Factored axis-aligned filtering for rendering multiple distribution effects. *ACM Trans. Graph.* 33, 4 (2014), 57:1–57:12. [2](#)
- [ODR09] OVERBECK R. S., DONNER C., RAMAMOORTHI R.: Adaptive Wavelet Rendering. *ACM Transactions on Graphics (SIGGRAPH Asia 09)* 28, 5 (2009), 1–12. [2](#)
- [PGB03] PÉREZ P., GANGNET M., BLAKE A.: Poisson image editing. *ACM Trans. Graph.* 22, 3 (2003), 313–318. [3](#)
- [PH10] PHARR M., HUMPHREYS G.: *Physically Based Rendering, Second Edition: From Theory To Implementation*, 2nd ed. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2010. [1, 4, 5, 7](#)
- [PSA*04] PETSCHNIGG G., SZELISKI R., AGRAWALA M., COHEN M., HOPPE H., TOYAMA K.: Digital photography with flash and no-flash image pairs. *ACM Trans. Graph.* 23, 3 (2004), 664–672. [3](#)
- [PSWS03] PORTILLA J., STRELA V., WAINWRIGHT M. J., SIMONCELLI E. P.: Image denoising using scale mixtures of gaussians in the wavelet domain. *IEEE Trans. Image Process* 12 (2003), 1338–1351. [3](#)
- [RKZ11] ROUSSELLE F., KNAUS C., ZWICKER M.: Adaptive sampling and reconstruction using greedy error minimization. *ACM Trans. Graph.* 30, 6 (2011), 159:1–159:12. [2, 3, 5](#)
- [RKZ12] ROUSSELLE F., KNAUS C., ZWICKER M.: Adaptive rendering with non-local means filtering. *ACM Trans. Graph.* 31, 6 (Nov. 2012), 195:1–195:11. [2](#)
- [RMZ13] ROUSSELLE F., MANZI M., ZWICKER M.: Robust denoising using feature and color information. *Comput. Graph. Forum* 32, 7 (2013), 121–130. [2, 3, 6, 9, 11](#)
- [SD11] SEN P., DARABI S.: Compressive rendering: A rendering application of compressed sensing. *IEEE Transactions on Visualization and Computer Graphics* 17, 4 (2011), 487–499. [5, 10](#)
- [SD12] SEN P., DARABI S.: On filtering the noise from the random parameters in monte carlo rendering. *ACM Trans. Graph.* 31, 3 (June 2012), 18:1–18:15. [2, 3](#)
- [Ste81] STEIN C. M.: Estimation of the mean of a multivariate normal distribution. *The Annals of Statistics* 9, 6 (11 1981), 1135–1151. [3](#)
- [Sze06] SZELISKI R.: Image alignment and stitching: A tutorial. *Found. Trends. Comput. Graph. Vis.* 2, 1 (2006), 1–104. [3, 6](#)
- [XP05] XU R., PATTANAIK S. N.: A novel monte carlo noise reduction operator. *IEEE Comput. Graph. Appl.* 25, 2 (2005), 31–35. [2](#)