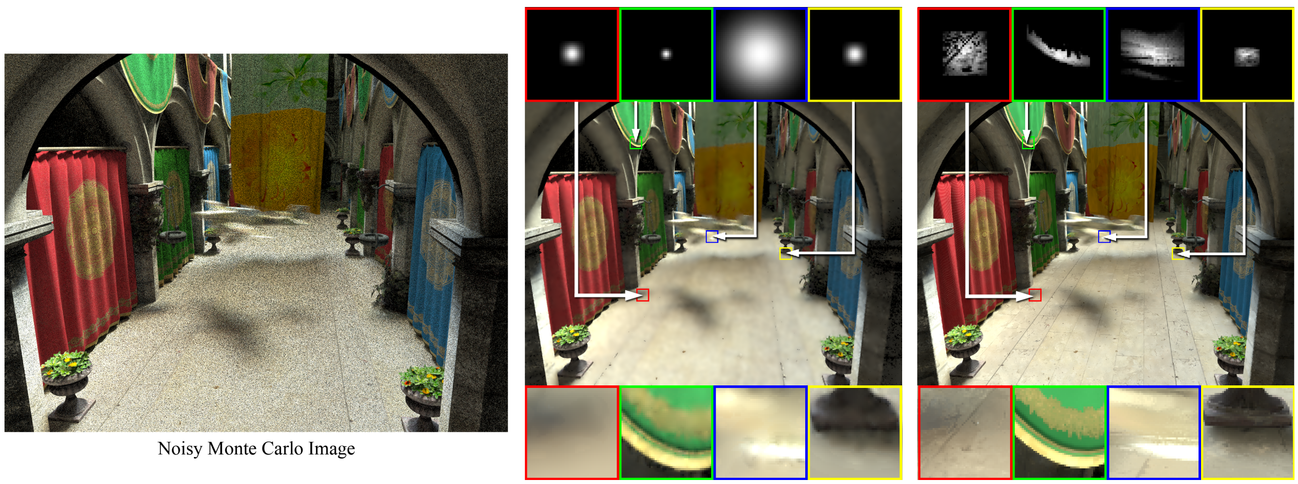
SURE-based Optimization for Adaptive Sampling and Reconstruction

Tzu-Mao Li Yu-Ting Wu Yung-Yu Chuang

National Taiwan University



Noisy Monte Carlo Image

Greedy Error Minimization [Rousselle et al., 2011]

Our Approach

Figure 1: *Comparisons between greedy error minimization (GEM) [Rousselle et al. 2011] and our SURE-based filtering. With SURE, we are able to use kernels (cross bilateral filters in this case) that are more effective than GEM's isotropic Gassians. Thus, our approach better adapts to anisotropic features (such as the motion blur pattern due to the motion of the airplane) and preserves scene details (such as the textures on the floor and curtains). The kernels of both methods are visualized for comparison.*

Abstract

We apply Stein's Unbiased Risk Estimator (SURE) to adaptive sampling and reconstruction to reduce noise in Monte Carlo rendering. SURE is a general unbiased estimator for mean squared error (MSE) in statistics. With SURE, we are able to estimate error for an arbitrary reconstruction kernel, enabling us to use more effective kernels rather than being restricted to the symmetric ones used in previous work. It also allows us to allocate more samples to areas with higher estimated MSE. Adaptive sampling and reconstruction can therefore be processed within an optimization framework. We also propose an efficient and memory-friendly approach to reduce the impact of noisy geometry features where there is depth of field or motion blur. Experiments show that our method produces images with less noise and crisper details than previous methods.

CR Categories: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—RayTracing.

Keywords: Sampling, reconstruction, ray tracing, cross bilateral filter, Stein's unbiased risk estimator (SURE).

Links: DL PDF WEB

1 Introduction

Monte Carlo (MC) integration is a common technique for rendering images with distributed effects such as antialiasing, depth of field, motion blur, and global illumination. It simulates a variety of sophisticated light transport paths in a unified manner; it estimates pixel values by using stochastic point samples in the integral domain. Despite its generality and simplicity, however, the MC approach converges slowly. A complex scene with multiple distributed effects usually requires several thousand expensive samples per pixel to produce a noise-free image.

Adaptive sampling and reconstruction (or filtering, used interchangeably in the paper) are two effective techniques for reducing noise. Given a fixed budget of samples, adaptive sampling determines the optimal sample distribution by concentrating more samples on difficult regions. To decide which pixels are worth more effort, we require a robust criterion for measuring errors. Accurate estimation of errors is challenging in our application because the ground truth is not available. Reconstruction algorithms, in contrast, properly construct smooth results from the discrete samples at hand. One key issue that reconstruction must resolve is how to select the filters for each pixel, as the optimal reconstruction kernels are usually spatially-varying and anisotropic. Recently, approaches have been developed to address the challenge of spatially-varying filters [Chen et al. 2011; Rousselle et al. 2011], producing better results than those that use a single filter across the whole image. However, these methods are limited to symmetric filters and do not work well for scenes with anisotropic features such as high-frequency textures on the floor and curtains in Figure 1.

We here propose an adaptive sampling and reconstruction algorithm to improve the efficiency of Monte Carlo ray tracing. The core idea is to adopt Stein's Unbiased Risk Estimator (SURE) [Stein 1981], a general unbiased estimator for mean squared error (MSE) in statistics, to determine the optimal sample density and per-pixel reconstruction kernels. The advantages of using SURE are twofold. For one, it provides a means by which to measure the quality of arbitrary reconstruction kernels – not just those that are symmetric (e.g. isotropic Gaussians used in greedy error minimization (GEM) [Rousselle et al. 2011]). As such, it allows for the use of more effective filters such as cross bilateral filters and cross non-local means filters. Another advantage is that the per-pixel errors estimated by SURE can be used to guide further sample distribution. Thus more samples can be allocated to difficult regions. In addition to applying SURE, we propose an efficient and memory-friendly approach to maintain the quality of cross bilateral filtering in the presence of noisy geometric features when rendering depth-of-field or motion blur effects. We propose a normalized distance to alleviate the impact of noisy features, making the proposed method more robust to all types of distributed effects. Experiments show the proposed method provides significant improvements over previous techniques for adaptive sampling and reconstruction.

2 Related work

Adaptive sampling and reconstruction. The seminal work of Mitchell [1987; 1991] over twenty years ago laid the foundation for methods for adaptive sampling and reconstruction. We categorize these techniques as image space methods, multidimensional methods, and adaptive filtering.

Image space methods estimate per-pixel errors with various criteria and allocate additional samples to difficult regions. These approaches usually are more general and can be used for various types of effects. Bala *et al.* [2003] combine edge information and sparse point samples to perform edge-aware interpolation. The quality of their results is highly dependent on the accuracy of edge detection. Overbeck *et al.* [2009] proposed adaptive wavelet rendering (AWR), a general wavelet-based adaptive sampling and reconstruction framework. They distinguish the sources of variances as the coarse level for distributed effects and the fine level for edges. These two types of noise are addressed separately by sampling the hierarchical wavelet coefficients adaptively; smooth images are reconstructed by thresholding wavelet coefficients. Recently, several approaches [Chen et al. 2011; Rousselle et al. 2011] have been proposed to smooth out noise using multi-scale filters. Chen *et al.* [2011] focus on depth of field and describe a criterion to select spatially-varying Gaussian filters from a predefined filterbank based on the depth map. Rousselle *et al.* [2011] also use Gaussian filters to form a filterbank. Although their method uses an error minimization framework for adaptive sampling and reconstruction, and is more general, the framework can be used only for symmetric filters. We apply SURE to estimate MSE for more general filters and allow the use of more effective filters, which yield significant improvements.

Other approaches perform adaptive sampling and reconstruction in a multidimensional space. Hachisuka *et al.* [2008] distribute more samples to discontinuities in the high-dimensional space and reconstruct them anisotropically using structure tensors. Their approach achieves good quality but becomes less efficient as the number of dimensions increases. Approaches have also been developed that are based on transform domain analysis, focusing on specific distributed effects such as depth of field [Soler et al. 2009], motion blur [Egan et al. 2009], and soft shadows [Egan et al. 2011]. Recently Lehitinen *et al.* proposed a novel method for reconstructing temporal light-field samples [2011], and later extended it to handle the indirect light field for global illumination [2012]. By reprojecting samples along the sample trajectory in the multidimensional space, expensive samples can be reused. In general, to avoid the curse of dimensionality, multidimensional approaches usually focus on only one or two specific effects. These methods are able to generate better results for the effects they focus on because the anisotropy of the integrand is taken into account. Our method, however, is more general, efficient, and memory-friendly.

Some methods focus on reconstruction only. Xu *et al.* [2005] proposed smoothing out Monte Carlo noise with a modified bilateral filter with smoothed range values. Segovia *et al.* [2006], Dammertz *et al.* [2010], and Bauszat *et al.* [2011] focus on interactive global illumination. They exploit geometric properties such as depths and surface normals to identify edges. Shirley *et al.* [2011] use the depth buffer to help filtering, but target defocus and motion blur. Sen and Darabi [2012] proposed a general adaptive filtering approach based on information theory. By identifying the dependencies between random parameters on sample colors and scene features, they reduce the importance of sample values influenced by MC noise.

Denoising using SURE. Stein [1981] proposed an MSE estimator called Stein's Unbiased Risk Estimator (SURE) for estimators on samples with normal distributions. Donoho and Johnstone [1995] incorporate the estimator into a Wavelet shrinkage algorithm to reconstruct functions from noisy inputs. Recently, SURE has received much acclaim from the image denoising community and has been widely used to optimize denoising parameters [Blu and Luisier 2007; Van De Ville and Kocher 2009].

3 Stein's Unbiased Risk Estimator (SURE)

Monte Carlo ray tracing estimates true pixel colors by randomly sampling the integral domain and reconstructing from samples. The unbiased Monte Carlo rendering techniques have a stochastic error bound which can be estimated using the variance of the estimator. According to the central limit theorem, if Y is the pixel color estimated by an unbiased Monte Carlo renderer and x is the true color, as the number of samples n approaches infinity, Y's distribution approximates a normal distribution with mean x and variance σ^2/n:

$$Y \xrightarrow{d} N\left(x, \frac{\sigma^2}{n}\right), \tag{1}$$

where σ^2 is the variance of the Monte Carlo samples. Previous investigations have demonstrated that, for a finite number of samples, this relationship is still a good approximation [Tamstorf and Jensen 1997; Fiorio 2004; Hachisuka et al. 2010].

Since a Monte Carlo renderer is an estimator, it is often necessary to estimate its accuracy for applications such as adaptive sampling. Stein's Unbiased Risk Estimator (SURE) offers a means for estimating the accuracy of a given estimator [Stein 1981; Blu and Luisier 2007]. SURE states that, if y is a measurement on x with a normal distribution $N(x, \sigma_y^2)$, and F is a weakly differentiable function, then the following estimation of error[1],

$$\mathrm{SURE}(F(y)) = \|F(y) - y\|^2 + 2\sigma_y^2 \frac{dF(y)}{dy} - \sigma_y^2, \tag{2}$$

is an unbiased estimator of the mean square error (MSE) of F(y), that is,

$$E[\mathrm{SURE}(F(y))] = \|F(y) - x\|^2. \tag{3}$$

[1]We followed Blu and Luisier's SURE formulation [2007], which was derived from the original SURE [Stein 1981]; their equivalence has been shown. Since we apply SURE to estimate MSE errors for each color channel independently, the filter kernel F is a scalar function. Thus, the dimension is 1 and the divergence in the original SURE formulation becomes a derivative.

Equations 2 and 3 indicate that if we can compute σ_y and $dF(y)/dy$, we can estimate the error of an estimator F without knowing the true underlying value x.

Our goal is to use the above formula to estimate the reconstruction error of an arbitrary kernel F. As mentioned above, rendering samples y follow a normal distribution. Thus, SURE can be used as the estimator for MSE of any filter F if we can compute $dF(y)/dy$ for the reconstruction kernel (σ_y can be directly estimated from Monte Carlo samples). Section 4 describes details of the algorithm, including the definition of F, the derivation of $dF(y)/dy$, and how to use SURE for optimal filter selection and adaptive sampling.

It is worth noting that Rousselle *et al.* [2011] attempted to estimate MSE for the same application. They decomposed the error into variance and bias and exploited the relation between the biases of the two filters to estimate the error. However, they used a quadratic approximation which is valid only for symmetric filter kernels, thus limiting the effectiveness of their approach. In contrast, our SURE-based estimation works for arbitrary reconstruction kernels and provides more flexibility to the choice of kernels.

4 Method

Figure 2 demonstrates the flowchart of the proposed method. Our method starts by rendering a small number of initial samples for each pixel and then iterates between filter selection and adaptive sampling stages until reaching the sample budget. After each sampling phase, we reconstruct a set of images using a filterbank. Each pixel is filtered multiple times with all candidate filters in the filterbank and the error of each filtered pixel color is estimated by computing SURE. For each pixel, the filtered color with the least SURE error is chosen and filled into the reconstructed image. If more sample budget is available, the adaptive sampling stage is invoked and a batch of new samples is distributed to pixels with larger SURE errors. Details are in the following subsections.

4.1 Initial samples

At the beginning, a small number of initial samples (usually 8 or 16 samples per pixel) are taken to explore the scene. The samples are generated by low discrepancy sequences and distributed evenly to each pixel. After rendering with these samples, our method performs the first reconstruction phase with the gathered information.

4.2 Filter selection using SURE

The main advantage of our method over greedy error minimization [Rousselle et al. 2011] is its ability to use arbitrary filters. We have experimented with three different filters: isotropic Gaussian, cross bilateral, and a modified non-local means filter [Buades et al. 2005] with additional scene feature information (we call this a "cross non-local means filter"; see Section 5.4). Here, we use the cross bilateral filter as an example since it offers the best compromise between performance and quality amongst these filters. Algorithms for different filters are the same except that different filters are used for constructing the filterbank; the derivatives in SURE are different also.

Cross bilateral filters. Cross bilateral filters have been shown effective for removing Monte Carlo noise [Dammertz et al. 2010; Sen and Darabi 2012]. Similar to previous work, auxiliary data including surface normals, depths, and texture colors are collected by caching information after tracing rays. We compute the per-pixel mean and variance of each feature and store them as feature vectors for cross bilateral filters. For the cross bilateral filter, the filter

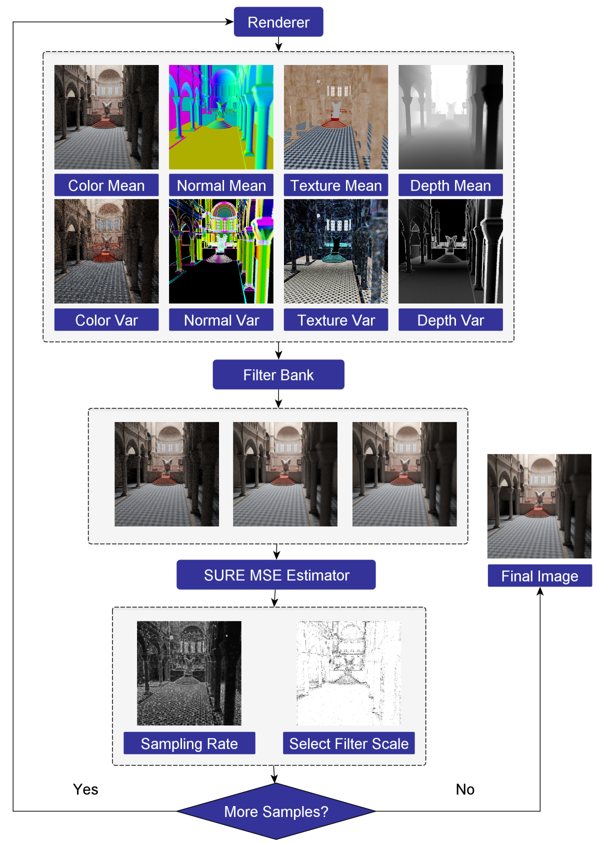


Figure 2: *An overview of our algorithm, which alternates between sampling and reconstruction until reaching the sample budget. During sampling, a set of samples collects colors, normals, textures and depths over the image plane. They are used as the side information for filters in the reconstruction stage, during which, for each pixel, a set of filters are performed and the filtered value with the minimal SURE value is filled into the reconstructed image. In addition, the minimal SURE value of each pixel is recorded to guide adaptive sampling. If there are sample budgets left, more samples are shot for pixels with larger SURE errors.*

weight w_{ij} between a pixel i and its neighbor j is defined as

$$\exp(-\frac{\|p_i - p_j\|^2}{2{\sigma_s}^2}) \exp(-\frac{\|c_i - c_j\|^2}{2{\sigma_r}^2}) \prod_{k=1}^{m} \exp(-\frac{D(\bar{f}_{ik}, \bar{f}_{jk})^2}{2{\sigma_{f_k}}^2}), \tag{4}$$

where $\bar{f}_{ik}$ is the sample mean of the k-th feature for the pixel i; σ_s, σ_r, and σ_{f_k} are the standard deviation parameters of the spatial, range (sample color), and feature terms respectively. D is a distance function used to address noisy scene features for depth of field and motion blur. It will be discussed later in this section. The filtered pixel color $\hat{c}_i$ of the pixel i is computed as the weighted combination of the colors c_j of all neighboring pixels j:

$$\hat{c}_i = \frac{\sum_{j=1}^{n} w_{ij} c_j}{\sum_{j=1}^{n} w_{ij}}. \tag{5}$$

Note that the cross bilateral kernels are spatially-varying due to the range and feature terms. Figure 1 shows examples.

In our current implementation, the filterbank is composed of cross bilateral filters with different σ_s, corresponding to different spatial scales. Other parameters σ_r and σ_{f_k} are fixed and their values are discussed in Section 5.

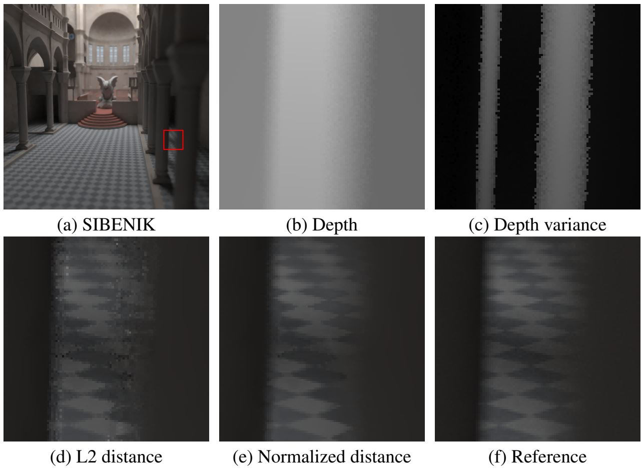


Figure 3: *Comparisons of filtering the SIBENIK scene with depth-of-field effects using L2 distance and our normalized distance. The area with strong depth of field has noisy geometry information, preventing us from filtering when using L2 distance between sample means. By incorporating sample variances, normalized distance allows us to filter these areas even given noisy geometry information. Note that the large depth variances in (c) allow us to filter areas around the pillars, removing the artifacts exhibited in the result with L2 distance (d).*

Depth of field and motion blur. Sen and Darabi [2012] point out that when rendering depth of field and motion blur effects, the geometric features (surface normal and depth) can be noisy due to MC noise. In these situations, as the weighting function is not accurate, using the features dogmatically for filtering can fail to remove the noise. They resolve this problem by computing the functional dependency of the MC random parameters and the scene features and using it to reduce the weight of samples if their features are highly dependent on the random parameters. Although their method successfully handles depth of field and motion blur, it operates at the sample level. Thus, performance and memory consumption become issues since computing pairwise mutual information between samples and parameters is not only time-consuming but also requires considerable memory for storing samples.

We propose a more efficient and memory-friendly approach to prevent cross bilateral filters from being affected by noisy scene features. Each pixel has a set of samples for feature k. Given two pixels i and j, to measure their distance with regard to the feature, the naive metric would be the distance between the sample means, $\bar{f}_{ik}$ and $\bar{f}_{jk}$. This, however, completely ignores sample variances. If we model samples as Gaussians, the distance between two sample sets should be normalized by their variances. Thus we define the normalized distance as

$$D(\bar{f}_{ik}, \bar{f}_{jk}) = \sqrt{\frac{\|\bar{f}_{ik} - \bar{f}_{jk}\|^2}{\sigma_{ik}^2 + \sigma_{jk}^2}}, \tag{6}$$

where σ_{ik}^2 and σ_{jk}^2 are sample variances of the k-th feature of pixels i and j respectively. Intuitively, for a pixel with strong depth of field and motion blur, its samples tends to have a large variance since these samples usually span over a large region in the spatial-temporal domain. Thus, it tends to have smaller distances and larger weights even when the geometric features are far apart. In the extreme case that two feature sets are inseparable due to strong depth of field or fast motion, the cross bilateral filter reduces to a Gaussian filter and does not use the unreliable geometric features. This

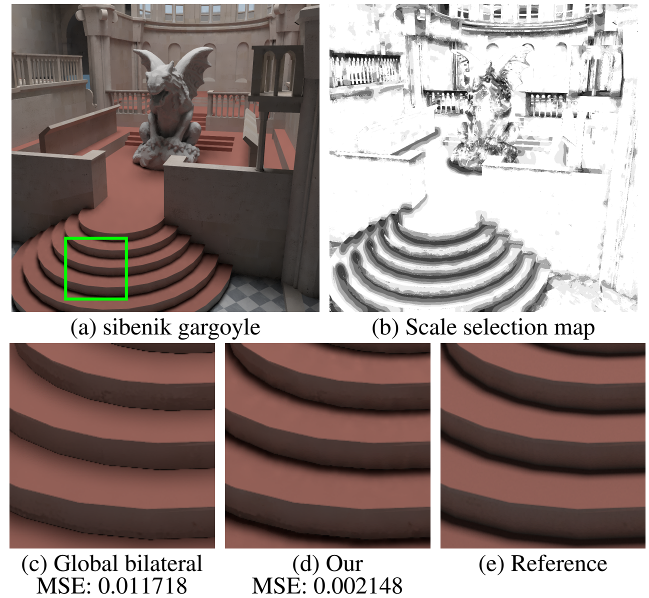


Figure 4: *Visualization of the scale selection map for σ_s of our method. We have also compared our approach to a global cross bilateral filter (which uses the same scale parameter, the largest σ_s in the filterbank, for all pixels). It is clear that the global cross bilateral filter produces large bias in the shadow areas. Our approach adapts better and uses fewer samples in these areas, thus leading to a smaller error. The sampling rate of the noisy input image is about 32 samples per pixel.*

approach allows us to evaluate feature importance at the pixel level and store only the sample mean and variance of features per pixel. Figure 3 shows the effect of the proposed distance metric.

Computing SURE and selecting the per-pixel optimal filter. For each pixel, we need to use the minimal SURE error to determine the optimal scale for the cross bilateral filters in the filterbank. As mentioned in Section 3, in calculating SURE, we need to compute $dF(c_i)/dc_i$ for the cross bilateral filter F defined in Equation 5. We have obtained its analytic form as

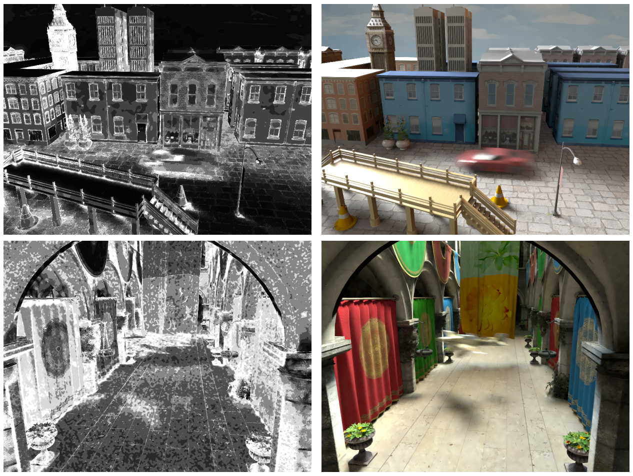
$$\frac{dF(c_i)}{dc_i} = \frac{1}{\sum_{j=1}^{n} w_{ij}} + \frac{1}{\sigma_r^2}(F^2(c_i) - F(c_i)^2), \tag{7}$$

where

$$F^2(c_i) = \frac{\sum_{j=1}^{n} w_{ij} {c_j}^2}{\sum_{j=1}^{n} w_{ij}}. \tag{8}$$

The derivation is in Appendix A. We then compute SURE to estimate the MSE for each filter in the filterbank using Equation 2. For each pixel, the filter with the least SURE error is selected and its filtered color is used to update the pixel.

We have observed that computing SURE using MC samples usually leads to noisy filter selection and thus yields noisy results. This is because SURE is an unbiased estimator of MSE and has its own variances. To reduce variances, one can either add more samples or perform filtering. For the sake of efficiency, we opted to perform filtering to reduce the variances of SURE. To be more concrete, we prefilter the estimated MSE image using a cross bilateral filter with a fixed parameter before SURE optimization. A similar problem was encountered in the previous method [Rousselle et al. 2011]; they smoothed out the selected scales of filters to deal with the variance of their estimator.



Sampling density Reconstructed image

Figure 5: *Visualizations for the sampling density of our approach.*

Figure 4 shows the scale selection map and compares our SURE-based filtering with a global cross bilateral filter (with the same scale for each pixel). It is obvious that spatially-varying scale selection yields better results both visually and quantitatively.

4.3 Adaptive sampling

The MSE estimated using SURE can be taken as feedback to the renderer; the sampling density should be proportional to the estimated MSE. However, since our MSE estimation is not perfect (note that Equation 2 can be negative), a heuristic variance term is included to ensure that regions with higher variances are allocated more samples. In addition, to guide more samples to darker areas, we scale our sampling function with the squared luminance of the filtered color. This strategy was also adopted by a previous approach [Overbeck et al. 2009] because human eyes are more sensitive to error in dark regions. As a result, the sampling function for a pixel i is determined by

$$S(i) = \frac{\text{SURE}(F(c_i)) + \sigma_i^2}{I(F(c_i))^2 + \epsilon}, \tag{9}$$

where σ_i^2 is the variance of samples within the pixel, $I(F(c_i))^2$ is the squared luminance of the filtered pixel color, and ϵ is a small number used to prevent a null denominator (set to 0.001). If the current sampling budget is m, pixel i receives $\lceil mS(i)/\sum_j S(j)\rceil$ samples. Figure 5 visualizes the sampling density for two examples. It is clear that samples concentrate on areas with geometry or texture details, discontinuities, or more noise.

5 Results and discussions

We implemented the algorithm on top of the PBRT2 system [Pharr and Humphreys 2010]. All results were generated on a machine with an Intel dual quad-core Xeon E5420 CPU at 2.5GHz, 32GB of RAM, and using 8 threads. As mentioned, we mainly used cross bilateral filters in the proposed SURE-based framework. In Section 5.4 we discuss results with other filters.

5.1 Parameter setting

There are a number of parameters for the features in Equation 4. They were set as $\sigma_{fk} = 0.4$ for normal, $\sigma_{fk} = 0.125$ for texture color, and $\sigma_{fk} = 0.3$ for depth throughout all experiments. We did not use σ_r in our current implementation since in practice we found the color term in the cross bilateral filter does not help much. We varied the spatial scale parameter σ_s to form the filterbank. We used $\sigma_s = 1, 2, 4$ to construct the filterbank in intermediate iterations and $\sigma_s = 1, \sqrt{2}, 2, 2\sqrt{2}, 4, 4\sqrt{2}, 8$ for the final reconstruction. We used fewer filters for intermediate phases as we found it sufficient and more efficient. Experiments show this setting strikes a good compromise between performance and quality. For the parameters used in prefiltering before SURE computation, we set $\sigma_s = 8$, and the same σ_{fk} as mentioned above. In practice, results are not very sensitive to these parameters and a wide range of parameters work equally well. Although parameters can be fine-tuned for each scene, this yields only marginal improvements.

5.2 Comparisons

We applied our algorithm on rendering four scenes – SIBENIK (1024x1024), TEAPOT (800x800), SPONZA (1600x1200) and TOWN (800x600) – with a variety of effects, including global illumination, motion blur, depth of field, area lighting, and glossy reflection (Figures 6 to 9). We have also compared our method on these scenes with the following methods:

- MC: Uniform sample distribution and per-pixel box filter. This approach is used as the baseline without adaptive sampling and reconstruction.
- GEM: Adaptive sampling and reconstruction using greedy error minimization [Rousselle et al. 2011]. The results were produced by the authors' implementation on the PBRT2 system. For all scenes, we set the γ parameter in their algorithm to 0.2 as the paper suggested.
- RPF: Adaptive filtering using random parameter filtering [Sen and Darabi 2012]. We implemented their approach on the PBRT2 system. The σ^2 in their algorithm is set to 0.002 according to the authors' suggestion.

The number of samples for each method was carefully adjusted to make equal-time comparisons. However, since RPF consumes considerable memory and time compared to other methods, its number of samples was limited to 8 or 16. For very complex scenes, the time for reconstruction could be negligible if taking samples is very expensive. To make fair comparisons under such situations, we also include equal-sample comparisons between RPF and our method. Finally, we also compared all methods quantitatively with the relative MSE proposed by Rousselle *et al.* [2011]. It is defined as the average of $(y - x)^2/(x^2 + \epsilon)$, where y is the estimated pixel color, x is the pixel color in the reference image, and ϵ is set to 0.01 to prevent division by zero.

Figure 6 compares these algorithms on SIBENIK, a scene with global illumination and depth of field. The image produced by MC retains considerable high-frequency noise even in simple areas such as the floor. GEM eliminates floor noise, while at the same time oversmoothing the area with textures due to its use of isotropic filters. Note that, although its relative MSE seems good, GEM tends to yield oversmoothed images. RPF produces a slightly sharper image than GEM but it is still oversmoothed, especially where there are depth-of-field effects. Our approach produces an image with much less noise while faithfully preserving textures.

The TEAPOT scene (Figure 7) demonstrates a challenging case with very high-frequency bump mapping and glossy reflections. None of the four methods preserve the bump map on the floor well. Again, MC produces a very noisy image. It is also worth noting that RPF fails to reproduce the self-reflection on the teapot. Overall, our approach still produces an image that is visually more pleasing and quantitatively more accurate than other methods.

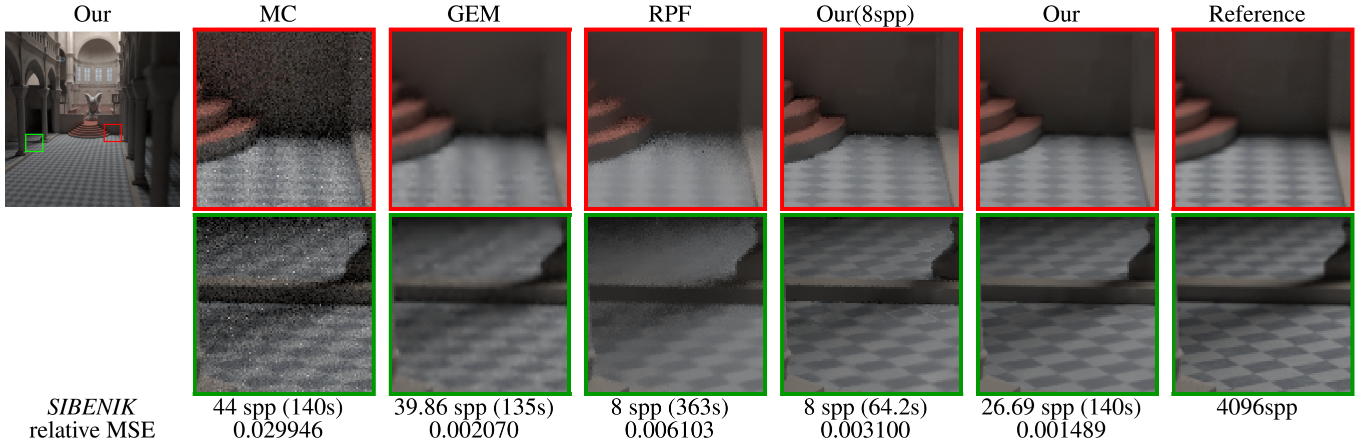


Figure 6: *A comparison on the SIBENIK scene with global illumination and depth of field. GEM adapts poorly to the texture on floor and produces oversmoothed results. RPF detects high dependency between u-v parameters and the color, thus filtering the area heavily and also producing oversmoothed results. The RPF image noise is from the sampling approximation of the bilateral filter.*

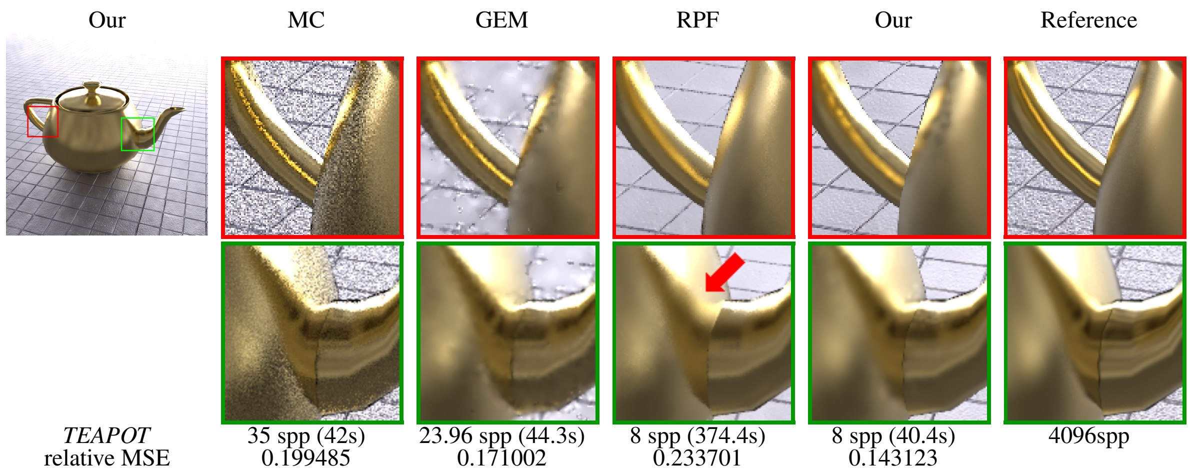


Figure 7: *A scene with a glossy teapot. The floor contains complex texture and bump maps. All methods oversmooth the floor. RPF also oversmooths the glossy self-reflection of the teapot indicated by the arrow.*

The SPONZA scene in Figure 8 contains motion blur effects. As shown in the first row of insets, the anisotropic pattern produced by motion of the wing is more vivid in our result than in the others. In addition, our approach more faithfully preserves the textures on the floor and the curtains.

Finally the TOWN scene shown in Figure 9 was designed to test environment lighting, area lights, and motion blur. The scene is challenging also due to the heavy occlusion between the buildings and skyscrapers. Despite its strong MSE, GEM fails to reconstruct all the textures in the scene, which are preserved well in our results. RPF, on the other hand, produces a very noisy image. This could be related to the sampling procedure in their bilateral filtering computation. Our approach outperforms the others by producing less noise and crisper details.

5.3 Discussions

GEM performs adaptive sampling and selects per-pixel filters in an attempt to minimize MSE. From the results, it does achieve lower relative errors compared to MC and RPF (and comparable to our approach). However, as mentioned, GEM is limited to symmetric filters and does not adapt well to high-frequency textures and detailed scene features. In all our test scenes, the results produced by GEM exhibit obviously oversmoothed artifacts. In addition, the GEM adaptive sampling criterion tends to send very few rays to the regions where most of the samples carry null radiance (for example, the right pillar of SPONZA in Figure 8). Our approach significantly alleviates these problems by using cross-bilateral filters and prefiltering MSE before SURE optimization.

RPF adjusts the weights of cross-bilateral filters by using mutual information and adapts well to scene features in most cases. It also removes the noise produced by few samples when rendering depth-of-field or motion blur effects. However, its multi-pass reconstruction algorithm can produce slightly oversmoothed results, such as the texture on the floor in SIBENIK (Figure 6), the disappeared shadows in SPONZA (Figure 8), and the glossy reflection on the teapot in TEAPOT (Figure 7). Another severe limitation of this approach is that the mutual information must be computed at the sample level, making the computation inefficient in both performance and memory consumption. To render one high-quality image at the 1920x1080 full HD resolution with 64 samples per pixel, it takes up to 13 GB to store the samples (108 bytes per sample as described in the paper). Finally, RPF is designed for reconstruction and does not have a feedback mechanism to the renderer for adaptive sampling.

Figure 8: *Comparisons on a complex scene SPONZA with global illumination and motion blur. The image on the top is our result. Insets show that GEM does not preserve details with symmetric filters, while RPF tends to oversmooth the shadows.*

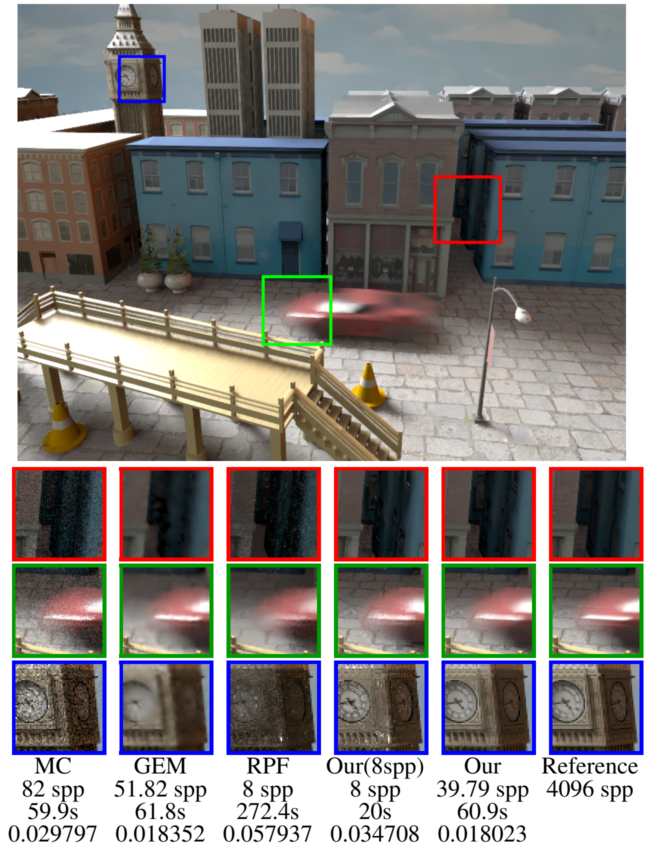


Figure 9: *Comparisons on the TOWN scene with an environment light, an area light, and heavy occlusion. GEM fails to adapt to textures, and RPF does not obtain enough samples to reconstruct the scene within the given time. Also, RPF contains heavy noise due to its sampling bilateral filtering approach. Our method adaptively samples the dark noisy area and preserves details well.*

Our method does away with the limitations of both GEM and RPF. At one end, we adopt SURE to estimate the error of an arbitrary reconstruction kernel. This allows us to optimize over a discrete set of cross bilateral filters for each pixel and determine the optimal sample distribution. Also, we propose a memory-friendly method to detect noisy geometric features when rendering depth of field and motion blur. As a result, our method successfully eliminates MC noise for a wide range of effects while preserving high-frequency textures and fine geometry details.

5.4 Other filters

To demonstrate the flexibility of the proposed framework with respect to different filters, in addition to cross bilateral filters, we have also experimented with isotropic Gaussian filters and cross non-local means filters. For isotropic Gaussians, we compare the results with GEM [2011] which is specifically designed for optimizing over an isotropic Gaussian filterbank. To be fair, we filter the SURE-estimated MSE using an isotropic Gaussian filter without using scene feature information. As shown in Figure 10, results of both methods are comparable and the scale selection maps are similar. This means that our SURE optimization is comparable to the specifically-designed GEM for the isotropic case.

The non-local means filter [Buades et al. 2005] is a popular method for image denoising. It assigns filter weights based on the similarity between pixel neighborhoods. In the context of rendering, we can further utilize scene features for better results. Thus, the cross non-local mean filter assigns the weight w_{ij} between two pixels i and j as

$$\exp\left(-\frac{\sum_{n\in N}\parallel c_{i+n}-c_{j+n}\parallel^2}{2|N|{\sigma_r}^2}\right)\prod_{k=1}^{m}\exp\left(-\frac{D(\bar{f}_{ik},\bar{f}_{jk})^2}{2{\sigma_{f_k}}^2}\right), \tag{10}$$

where N is the neighbourhood ($N = \{(x,y)| -2 <= x, y <= 2\}$ in our implementation). Other symbols are the same as defined in Section 4.2. Note that we use the patch-based distance only for color information, since patch-based distance for scene features tends to smooth out features. The filtered pixel color $\hat{c}_i$ of pixel i is computed as the weighted combination of the colors c_j of all neighboring pixels j within a 41×41 neighborhood.

To demonstrate the utility of SURE-based filter selection, we applied cross non-local means filters in two settings. For the first setting – the global cross non-local means filter – we used the same range parameter σ_r across the whole image. For the second one – the SURE cross non-local means filter – we constructed a cross non-local means filterbank by varying σ_r and used SURE to select best filters and shot samples. Figure 11 shows the comparisons between the two. It is clear that the SURE-based framework significantly alleviates the over-smoothness problem of the global filter, especially in shadows and in the motion blur of the moving car. From our experiments, filtering with cross non-local means filters

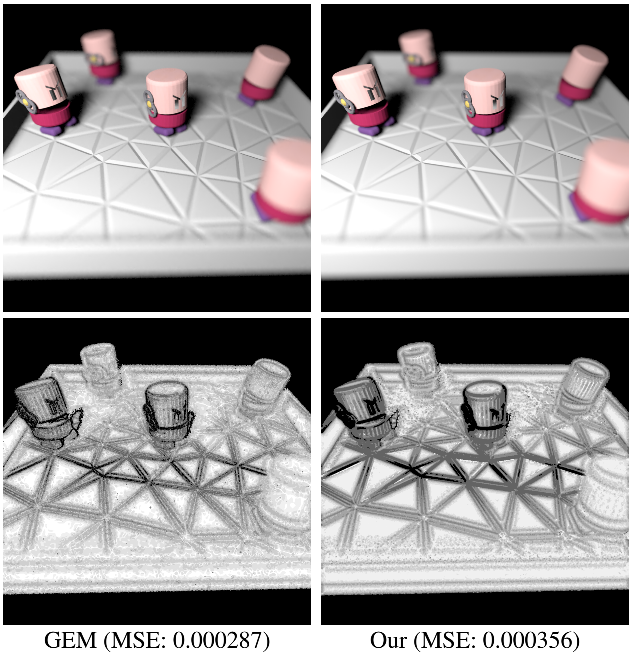


Figure 10: *The proposed SURE-based framework incorporated with an isotropic Gaussian filterbank and compared with GEM [Rousselle et al. 2011]. The results and scale selection maps generated by both methods are similar.*

sometimes generated slightly better results than cross bilateral filters. However, it is about 10 times slower than the cross bilateral filter. As a compromise between quality and performance, we opted to use cross bilateral filters for most results in the paper.

5.5 Limitations

The TEAPOT scene (Figure 7) reveals a limitation of our approach. The bump mapped floor contains a large number of very high-frequency textures. At the same time, it suffers from a large amount of MC noise due to the environment lighting and glossy reflection. With a low sample budget, our approach does not preserve all the details well. In addition, as with most reconstruction approaches, our method was susceptible to oversmoothing.

6 Conclusion and Future Work

We have presented an efficient adaptive sampling and reconstruction algorithm for reducing noise in Monte Carlo rendering by using Stein's Unbiased Risk Estimator (SURE) in the error estimation framework. For reconstruction, the use of SURE enables us to measure the reconstruction quality for arbitrary filter kernels. It does away with the limitation of using only symmetric kernels imposed by previous work. This freedom to use non-symmetric kernels significantly improves the effectiveness of the framework. When performing adaptive sampling, SURE can be used to determine the sampling density. Another contribution of this paper is an efficient and memory-friendly approach to detect noisy geometric features when rendering depth of field and motion blur. As a result, the proposed adaptive sampling and reconstruction method efficiently eliminates MC noise while preserving the vivid details of a scene. Experiments show the proposed method offers significant improvement over state-of-the-art approaches.

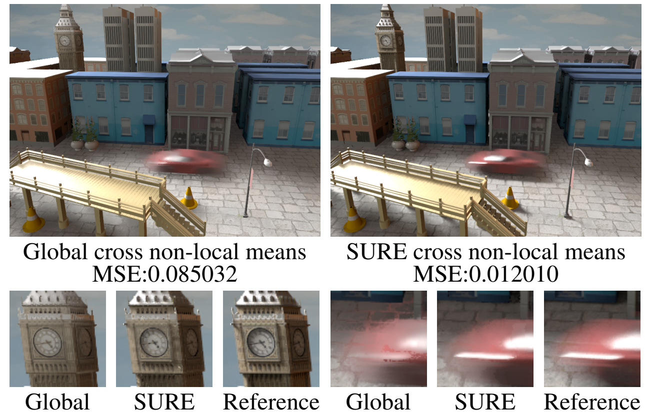


Figure 11: *Comparison of cross non-local means filters without and with SURE-based framework. Compared to the global cross non-local means filter, our SURE-based optimization largely alleviates the oversmoothing problem. The sampling rate of the noisy input image is about 41 samples per pixel.*

One possible future direction is to implement the proposed algorithms on GPUs for interactive applications. We also would like to extend the SURE-based framework to animation rendering. In the current algorithm, as there is no built-in mechanism specifically designed for temporal data, temporal coherence cannot be guaranteed. In practice, we have experimented with a naive approach that renders each frame independently. The results look good enough with only very subtle temporal flicking. However, a better way to handle animation would be to consider temporal samples and perform filtering in the spatial-temporal domain. Finally, it would also be interesting to adapt SURE to other rendering applications that require error estimation.

A Derivatives for filters

To compute SURE for reconstruction filters, we must calculate their derivatives $dF(c_i)/dc_i$ and substitute into Equation 2. For the cross bilateral filter, from Equation 5, we have (note that $w_{ii} = 1$ according to the definition in Equation 4)

$$F(c_i) = \frac{\sum_{j=1}^{n} w_{ij} c_j}{\sum_{j=1}^{n} w_{ij}} = \frac{\sum_{j \neq i} w_{ij} c_j + c_i}{\sum_{j=1}^{n} w_{ij}}. \tag{11}$$

Let $W_i = \sum_{j=1}^{n} w_{ij}$. After applying the quotient rule of derivatives, we have

$$\frac{dF(c_i)}{dc_i} = \frac{\left(\frac{d(\sum_{j \neq i} w_{ij} c_j + 1)}{dc_i}\right) - \frac{dW_i}{dc_i} F(c_i)}{W_i}. \tag{12}$$

After substituting $\frac{dw_{ij}}{dc_i} = \frac{(c_j - c_i)}{\sigma_r^2} w_{ij}$ into Equation 12, and after some manipulations, we obtain

$$\frac{dF(c_i)}{dc_i} = \frac{1}{W_i} + \frac{1}{\sigma_r^2}(F^2(c_i) - F(c_i)^2), \tag{13}$$

where

$$F^2(c_i) = \frac{\sum_{j=1}^{n} w_{ij} {c_j}^2}{\sum_{j=1}^{n} w_{ij}}.$$

For the cross non-local means filter, its $dF(c_i)/dc_i$ is

$$\begin{aligned}\frac{dF(c_i)}{dc_i} &= \frac{1}{W_i} + \frac{1}{|N|\sigma_r^2}(F^2(c_i) - F(c_i)^2) + \\ \frac{1}{|N|\sigma_r^2 W_i} &\sum_{n\in N} w_{i,i-n}(c_i - c_{i+n})(F(c_i) - c_{i-n}),\end{aligned} \tag{14}$$

where W_i and $F^2(c_i)$ are defined similarly as above. Similar derivations can be found in Van De Ville and Kocher's paper [2009].

Acknowledgements. We would like to thank the anonymous reviewers for their valuable comments and the creators of the models used in this paper: airplane (Pedro Caparros), Ferrari (RenderHere), buildings, street cones (UnityDevelopment), western buildings (Tippy), Big Bens (Skipper25), street lamps (jotijoti), downloaded from ShareCG.com; elevated walk (dogbite1066) from ShareAEC.com; plants (Tiago Crisostomo) from Google Sketchup; Crytek Sponza (Marko Dabrovic and Crytek GmbH), Sibenik (Marko Dabrovic and Mihovil Odak), gargoyle (INRIA) via the AIM@SHAPE repository; Toasters scene (Andrew Kensler) via the Utah 3D Animation Repository. This work was partly supported by grants NTU101R7609-5 and NSC101-2628-E-002-031-MY3.

References

BALA, K., WALTER, B., AND GREENBERG, D. P. 2003. Combining edges and points for interactive high-quality rendering. *ACM Trans. Graph. (Proceedings of SIGGRAPH 2003) 22*, 3, 631–640.

BAUSZAT, P., EISEMANN, M., AND MAGNOR, M. 2011. Guided image filtering for interactive high-quality global illumination. *Computer Graphics Forum (Proceedings of EGSR 2011) 30*, 4, 1361–1368.

BLU, T., AND LUISIER, F. 2007. The SURE-LET approach to image denoising. *IEEE Transactions on Image Processing 16*, 11, 2778–2786.

BUADES, A., COLL, B., AND MOREL, J.-M. 2005. A non-local algorithm for image denoising. In *Proceedings of IEEE Computer Vision and Pattern Recognition (CVPR 2005)*, 60–65.

CHEN, J., WANG, B., WANG, Y., OVERBECK, R. S., YONG, J.-H., AND WANG, W. 2011. Efficient depth-of-field rendering with adaptive sampling and multiscale reconstruction. *Computer Graphics Forum 30*, 6, 1667–1680.

DAMMERTZ, H., SEWTZ, D., HANIKA, J., AND LENSCH, H. P. A. 2010. Edge-avoiding Á-Trous wavelet transform for fast global illumination filtering. In *Proceedings of the Conference on High Performance Graphics (HPG 2010)*, 67–75.

DONOHO, D., AND JOHNSTONE, I. M. 1995. Adapting to unknown smoothness via wavelet shrinkage. *Journal of the American Statistical Association 90*, 1200–1224.

EGAN, K., TSENG, Y.-T., HOLZSCHUCH, N., DURAND, F., AND RAMAMOORTHI, R. 2009. Frequency analysis and sheared reconstruction for rendering motion blur. *ACM Trans. Graph. (Proceedings of SIGGRAPH 2009) 28*, 3, Article 93.

EGAN, K., HECHT, F., DURAND, F., AND RAMAMOORTHI, R. 2011. Frequency analysis and sheared filtering for shadow light fields of complex occluders. *ACM Trans. Graph. 30*, 2, Article 9.

FIORIO, C. V. 2004. Confidence intervals for kernel density estimation. *Stata Journal 4*, 2, 168–179.

HACHISUKA, T., JAROSZ, W., WEISTROFFER, R. P., DALE, K., HUMPHREYS, G., ZWICKER, M., AND JENSEN, H. W. 2008. Multidimensional adaptive sampling and reconstruction for ray tracing. *ACM Trans. Graph. (Proceedings of SIGGRAPH 2008) 27*, 3, Article 33.

HACHISUKA, T., JAROSZ, W., AND JENSEN, H. W. 2010. A progressive error estimation framework for photon density estimation. *ACM Trans. Graph. (Proceedings of SIGGRAPH Asia 2010) 29*, 6, Article 144.

LEHTINEN, J., AILA, T., CHEN, J., LAINE, S., AND DURAND, F. 2011. Temporal light field reconstruction for rendering distribution effects. *ACM Trans. Graph. (Proceedings of SIGGRAPH 2011) 30*, 4, Article 55.

LEHTINEN, J., AILA, T., LAINE, S., AND DURAND, F. 2012. Reconstructing the indirect light field for global illumination. *ACM Trans. Graph. (Proceedings of SIGGRAPH 2012) 31*, 4 (July), 51:1–51:10.

MITCHELL, D. P. 1987. Generating antialiased images at low sampling densities. In *Proceedings of SIGGRAPH 1987*, 65–72.

MITCHELL, D. P. 1991. Spectrally optimal sampling for distribution ray tracing. In *Proceedings of SIGGRAPH 1991*, 157–164.

OVERBECK, R. S., DONNER, C., AND RAMAMOORTHI, R. 2009. Adaptive wavelet rendering. *ACM Trans. Graph. (Proceedings of SIGGRAPH Asia 2009) 28*, 5, Article 140.

PHARR, M., AND HUMPHREYS, G. 2010. *Physically Based Rendering: From Theory To Implementation*, 2nd ed. Morgan Kaufmann Publishers Inc.

ROUSSELLE, F., KNAUS, C., AND ZWICKER, M. 2011. Adaptive sampling and reconstruction using greedy error minimization. *ACM Trans. Graph. (Proceedings of SIGGRAPH Asia 2011) 30*, 6, Article 159.

SEGOVIA, B., IEHL, J. C., MITANCHEY, R., AND PÉROCHE, B. 2006. Non-interleaved deferred shading of interleaved sample patterns. In *Proceedings of the 21st ACM SIGGRAPH/EUROGRAPHICS Symposium on Graphics Hardware*, 53–60.

SEN, P., AND DARABI, S. 2012. On filtering the noise from the random parameters in Monte Carlo rendering. *ACM Trans. Graph. 31*, 3, Article 18.

SHIRLEY, P., AILA, T., COHEN, J., ENDERTON, E., LAINE, S., LUEBKE, D., AND MCGUIRE, M. 2011. A local image reconstruction algorithm for stochastic rendering. In *Proceedings of Symposium on Interactive 3D Graphics and Games*, 9–14.

SOLER, C., SUBR, K., DURAND, F., HOLZSCHUCH, N., AND SILLION, F. 2009. Fourier depth of field. *ACM Trans. Graph. 28*, 2, Article 18.

STEIN, C. M. 1981. Estimation of the mean of a multivariate normal distribution. *Annals of Statistics 9*, 6, 1135–1151.

TAMSTORF, R., AND JENSEN, H. W. 1997. Adaptive sampling and bias estimation in path tracing. In *Proceedings of Eurographics Rendering Workshop*, 285–295.

VAN DE VILLE, D., AND KOCHER, M. 2009. SURE-based non-local means. *IEEE Signal Processing Letters 16*, 11, 973–976.

XU, R., AND PATTANAIK, S. N. 2005. A novel Monte Carlo noise reduction operator. *IEEE Computer Graphics and Applications 25*, 2, 31–35.