

Adaptive Sampling and Reconstruction using Greedy Error Minimization

Fabrice Rousselle*
University of Bern

Claude Knaus †
University of Bern

Matthias Zwicker‡
University of Bern

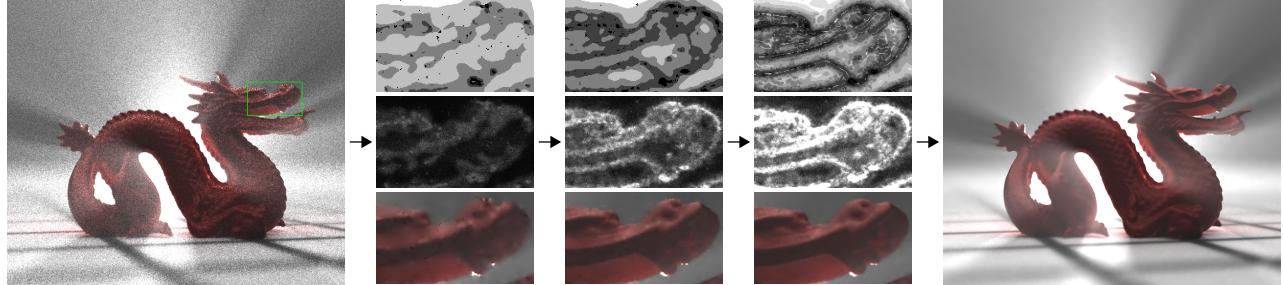


Figure 1: We minimize MSE in Monte Carlo rendering by adaptive sampling and reconstruction in image space. We iterate over two steps: given current samples, optimize over a set of filters at each pixel to minimize MSE; then, given a filter at each pixel, distribute more samples to further reduce MSE. Left: initialization with 4 samples per pixel. Insets: each column is one iteration (top to bottom): filter selection (smooth filters shaded white, sharp ones black), sample density map, reconstruction. Right: result at an average of 32 samples per pixel. This image features single scattering participating media, indirect illumination using photon mapping, depth of field, and area lighting.

Abstract

We introduce a novel approach for image space adaptive sampling and reconstruction in Monte Carlo rendering. We greedily minimize relative mean squared error (MSE) by iterating over two steps. First, given a current sample distribution, we optimize over a discrete set of filters at each pixel and select the filter that minimizes the pixel error. Next, given the current filter selection, we distribute additional samples to further reduce MSE. The success of our approach hinges on a robust technique to select suitable per pixel filters. We develop a novel filter selection procedure that robustly solves this problem even with noisy input data. We evaluate our approach using effects such as motion blur, depth of field, interreflections, etc. We provide a comparison to a state-of-the-art algorithm based on wavelet shrinkage and show that we achieve significant improvements in numerical error and visual image quality. Our approach is simple to implement, requires a single user parameter, and is compatible with standard Monte Carlo rendering.

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

Keywords: adaptive sampling and reconstruction

1 Introduction

Monte Carlo techniques compute pixel colors by (quasi-)randomly sampling an integration domain that covers all light paths transporting light from a source to the camera. The integration domain may include effects such as depth of field, motion blur, and light paths with multiple interreflections. Unless one computes an excessive number of samples, this often leads to high pixel variance and the typical noise artifacts in Monte Carlo rendering. There are two main strategies to address this. The first is to distribute samples in an optimal fashion, with respect to the problem at hand. The second is to smooth out noise by applying suitable filters. Both strategies

can be applied in the high dimensional space of light paths or in the image plane. We focus on strategies that operate in the image plane.

We formulate the problem as follows: given a certain budget of Monte Carlo samples, obtain an image that minimizes the relative mean squared error (MSE) by distributing samples in a suitable fashion in the image plane and by filtering the image with appropriate filters. We can interpret this as an optimization problem over the space of sample distributions and image filters. Our core idea is to make the problem tractable by restricting the space of filters to a discrete set of predetermined filters per pixel. Each pixel may have a different set of filters, but the set is predefined and not itself part of the optimization. We use a simple greedy strategy to obtain an approximate solution to the MSE minimization problem. Starting from an initial set of samples, we iterate over two steps. First, for each pixel we select the filter from its discrete set that minimizes the pixel MSE given the current samples. Second, given the currently chosen pixel filters, we distribute a new batch of samples that try to further reduce MSE as much as possible. This process is repeated until a termination criterion is fulfilled.

To minimize pixel MSE we express it as the sum of the squared bias, i.e., expected error, and variance. We define the set of filters at each pixel such that it provides a trade-off between reducing bias and increasing variance. Then we attempt to minimize pixel MSE by selecting the filter that offers an optimal compromise. The main challenge in practice is that we only have access to noisy data to estimate bias and variance. Therefore, an important component of our algorithm is a robust method to solve this filter selection problem.

We demonstrate and evaluate our framework using Gaussian filters at different scales as the smoothing filters. We describe simple but effective methods to select filter scales at each pixel and to distribute samples in each iteration, always attempting to minimize pixel MSE. We evaluate the performance of our approach and its robustness to noise by comparing it to images rendered using ground truth statistics, i.e., bias, variance, and MSE values. We show that our method to minimize MSE based on noisy data comes reasonably close to the reference, and provides a significant improvement over state-of-the-art adaptive sampling and reconstruction algorithms. Our method is consistent in that it converges to

*e-mail: roussel@iam.unibe.ch

†e-mail:knaus@iam.unibe.ch

‡e-mail:zwicker@iam.unibe.ch

noise and bias free images as the number of samples increases. Bias is only guaranteed to vanish in the limit, however, as the number of samples goes to infinity. Our framework is orthogonal to Monte Carlo ray tracing on a pixel-by-pixel basis, and we implemented it on top of the PBRT renderer.

In summary, we make the following contributions:

- We present a framework for adaptive sampling and reconstruction based on minimizing per pixel MSE. In a greedy error minimization procedure, we iterate over two steps: selecting pixel filters from a set of smoothing filters to minimize pixel MSE, and distributing new samples in an attempt to maximally reduce MSE in each iteration.
- We describe an implementation of our framework with Gaussian smoothing filters at different scales. This includes robust methods to select filter scales and estimate MSE, and a strategy to distribute samples in each iteration.
- We evaluate our implementation by comparing it to results obtained using ground truth statistics. We show that in many cases, our approach comes reasonably close to ground truth. Despite its simplicity, it provides significant improvements over the state-of-the-art in adaptive sampling and reconstruction.

2 Previous Work

We discuss related work in the areas of adaptive sampling and reconstruction and image denoising. We do not discuss advanced Monte Carlo sampling techniques such as importance sampling and its extensions, since these are orthogonal to our approach.

Image Space Adaptive Sampling and Reconstruction. Our framework belongs to the class of image space adaptive sampling and reconstruction methods. The basic idea of these techniques is to estimate some per pixel error criterion and adaptively distribute more samples in pixels with large errors. The pioneering approach by Mitchell [1987] is often used as a reference to compare such techniques. Bala et al. [2003] describe a sophisticated approach that is targeted at interactive rendering. By explicitly representing edges, they obtain high quality image reconstruction. If edge detection fails, however, artifacts may occur. Our approach for filter selection is similar to wavelet-based methods. Bolin and Meyer [1998] use Haar wavelets and develop a perceptual error metric for adaptive sampling. We currently use relative mean squared error, and it would be interesting to include a more perceptually based measure in future work, similarly to Farrugia and Péroche [2004]. Adaptive wavelet rendering (AWR) by Overbeck et al. [2009], which we consider the current state-of-the-art for screen space algorithms, is probably most closely related to our technique. Similar to our approach, their technique is based on iteratively smoothing (or denoising) the image and distributing new samples. Their technique specifically uses wavelets, where smoothing is performed by soft thresholding the wavelet coefficients. Our framework is more general in that it is not restricted to wavelets. We explicitly attempt to minimize MSE, while their approach is based on a heuristic to distribute new samples. We provide a detailed comparison to adaptive wavelet rendering and show that our algorithm consistently leads to lower numerical errors and visually higher quality images. Recently, Chen et al. [2011] proposed adaptive sampling and reconstruction for depth of field effects. They attempt to select an optimal filter from a filterbank on a per pixel basis similarly to our approach. In addition, they combine depth maps with sample variance maps to guide sampling and reconstruction.

In contrast, our method needs no information other than sample values and variances and is not limited to depth of field.

Multidimensional Adaptive Sampling and Reconstruction. A logical extension of image space methods is to perform adaptive sampling and reconstruction directly in the high dimensional space of light paths. A general solution to this problem has been proposed by Hachisuka et al. [2008], where samples are adaptively distributed in the multidimensional space, and the high dimensional signal is reconstructed using anisotropic interpolation. Overbeck et al. [2009] point out that this strategy becomes ineffective for higher dimensions because of the curse of dimensionality. Nonetheless, the approach can be highly successful if restricted to specific effects such as depth of field, motion blur, or soft shadows, where problem specific information is available to guide sampling and reconstruction. In particular, a number of algorithms are based on a frequency analysis of the sheared light field structure generated by these effects. Soler et al. [2009] propose a multidimensional sampling scheme with an appropriate reconstruction filterbank tailored to depth of field. Egan et al. [2009] propose an adaptive scheme with sheared reconstruction filters for motion blur, and a similar scheme for shadow light fields [Egan et al. 2011]. Lehtinen et al. [2011] also exploit the sheared light field structure to render combinations of motion blur, depth of field and soft shadows, although their approach does not rely on frequency analysis. In contrast to these methods, our approach is not restricted to motion blur, depth of field, or soft shadow effects.

Adaptive Filtering. A variety of algorithms attempt to adaptively smooth images suffering from Monte Carlo noise, without coupling the filtering process with adaptive sampling. This strategy has recently become more popular for interactive rendering, where performance and visual quality is more important than theoretical properties such as convergence or bias. A common strategy is to construct edge preserving smoothing filters by including information about the scene geometry, such as normal or depth buffers. For example, Ritschel et al. [2009] use cross bilateral filtering to smooth noisy image data while respecting edges. Dammertz et al. [2010] describe an edge-avoiding à-trous wavelet transform. Shirley et al. [2011] exploit the depth buffer for adaptive filtering. A common shortcoming of these approaches is that they fail to adapt to image features, such as caustics, that cannot be inferred from the auxiliary information that is used. In addition, the objective of these techniques is slightly different from ours, since they prioritize interactive performance. Nonetheless, it would be interesting to extend our framework to operate on these more sophisticated filters rather than the Gaussians that we currently use.

Image Denoising. A vast body of work in the image processing field addresses the problem of image denoising, which is related to our problem. We restrict the discussion to the most prevalent approaches and highlight differences to our setting. Wavelet shrinkage, introduced by Donoho and Johnstone [1994], is particularly widespread because it rests on a very thorough theoretical foundation. The intuition is simple: decompose a noisy signal using a wavelet representation, subtract the noise from the wavelet coefficients, and reconstruct the smoothed signal. Since noise yields small wavelet coefficients and signal yields large coefficients, the signal is effectively smoothed while retaining its sharp features. The reconstruction step of the AWR algorithm by Overbeck et al. actually performs wavelet shrinkage using a conservative estimate of the noise. Another line of work is based on the idea that if the probability distribution of the true image in a neighborhood and the distribution of the noise are known, one can find the most likely true image value by Bayes estimation. Starting from methods based on

Gaussian models as proposed by Lee et al. [1980], this approach has been refined to include non-Gaussian image statistics, operate in the wavelet domain, and take into account correlation between wavelet coefficients [Portilla et al. 2003]. However, these sophisticated methods are computationally expensive. In addition, they usually assume a global noise model, which is not an appropriate assumption for images produced by Monte Carlo rendering. Our work is related to the concept of non-parametric bandwidth selection [Lepski et al. 1997; Goldenshluger and Nemirovski 1997]. The idea is to spatially adapt a filter bandwidth to local signal properties such as to minimize the MSE of the filtered signal. The intersection of confidence interval (ICI) method described by Katkovnik [1999] inspired our own filter selection approach.

3 Algorithm Overview

Ideally, we would like to solve the following problem: given a certain budget of Monte Carlo samples, obtain an image that minimizes the MSE by distributing samples in a suitable fashion in the image plane and by filtering the image with appropriate filters. This problem is probably intractable in general, because the space of potential image filters is too large. A core idea is to make the problem more manageable by restricting the potential filters to a discrete set of predetermined filters per pixel. Each pixel may have a different set of filters, but the set is predefined and not itself part of the optimization.

Our framework uses a simple greedy strategy to solve the MSE minimization problem, as illustrated in Figure 2. Starting from an initial set of samples, we iterate over two steps. First, for each pixel we select the filter from its discrete set that minimizes the pixel MSE given the current samples. Second, given the currently chosen pixel filters, we distribute a new batch of samples that tries to further reduce MSE as much as possible under the current selection of filters. This process is repeated until a termination criterion is met, for example, a given sample budget is reached.

In the following we describe an implementation of this framework that uses the same set of filters for each pixel. In addition, the filters compute a linear function of their input, and they are related by a uniform scale. The smallest scale corresponds to the usual, unbiased pixel filter that is used in standard rendering. Selecting appropriate filters from this set allows us to minimize pixel MSE by making an optimal trade-off between bias and noise. Filters at smaller scales have little bias but much noise, and vice-versa for larger scales. We next describe the two steps of our approach in detail, i.e., filter selection (Section 4) and sample distribution (Section 5).

4 Filter Selection

In this section we describe how, at each pixel of a noisy image, we select a filter from our predefined set that attempts to minimize the pixel MSE, that is, the sum of the squared bias and the variance. In Section 4.1 we formulate an incremental MSE minimization strategy that avoids explicit bias estimation. We then describe in Section 4.2 how we implement this strategy under the assumption that the true image is locally a quadratic function. In Section 4.3 we analyze the behavior of the resulting filter selector under noisy input, and finally we describe a post-processing approach to greatly reduce erroneous filter choices in Section 4.4.

4.1 Incremental MSE Minimization

In theory we could estimate bias and variance for each filter directly to minimize MSE. It is very challenging, however, to guess the true

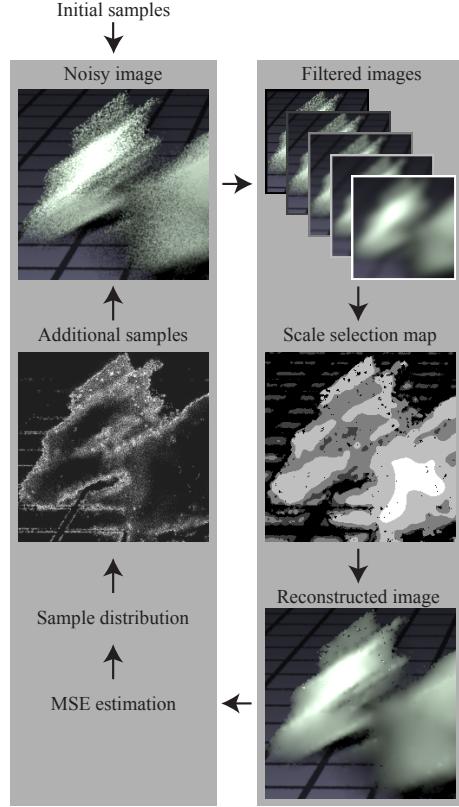


Figure 2: Overview of our framework. We iterate over two steps: filter selection based on the noisy image on the right, and distribution of new samples to greedily reduce MSE on the left.

bias from noisy input. Hence, we avoid explicit bias estimation. Directly estimating the variance is less problematic, because it does not require knowledge of the true pixel value. Considering that the filters in our set are related by uniform scaling, we can order them in a fine to coarse, or sharp to smooth, manner according to their scales. A key observation is that, for most pixels, the filter bias increases and the variance decreases monotonically as we go from finer to coarser scales. Assuming monotonicity, we find the filter with minimum MSE simply by traversing the list of filters from fine to coarse. At each pixel, for each pair of consecutive fine and coarse filters f and c , we compute the change in MSE, $\Delta\text{MSE}[f \rightarrow c]$, and stop when it is positive. Note that the difference between the MSE of the fine and coarse scales is

$$\begin{aligned} \Delta\text{MSE}[f \rightarrow c] &= \text{MSE}[c] - \text{MSE}[f] \\ &= \underbrace{\text{Bias}[c]^2 - \text{Bias}[f]^2}_{\text{bias term}} + \underbrace{\text{Var}[c] - \text{Var}[f]}_{\text{variance term}}, \end{aligned}$$

consisting of a *bias* and *variance* term. The crucial benefit of this approach is that we can well approximate the bias term without knowledge of the true bias $\text{Bias}[c]$ and $\text{Bias}[f]$, as we will show in Section 4.2.

We compare MSE minimization using exhaustive search over the filters to MSE minimization based on the assumption of monotonicity of bias and variance in Figure 3(a) and (b). We use a filter set consisting of five scales of Gaussian filters at dyadic intervals. For the sake of this comparison, we computed ground truth per pixel bias and variance for each filter by empirically gathering ground truth statistics (of course, in practice ground truth statistics

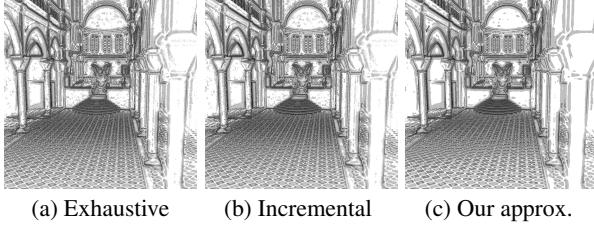


Figure 3: Results of scale selection using (a) minimization of the MSE using exhaustive search, (b) the incremental approach, (c) our incremental approach with bias computation using the quadratic approximation. We used five scales, where white indicates the coarsest and black the finest scale.

are not available; see Section 4.3). We obtain ground truth bias by rendering a reference image and, for each filter, computing the difference between the reference and filtered image. We obtain per pixel variance by rendering a noisy image with a limited number of 16 samples per pixel many times. We then applied the filters to each noisy image and computed empirical pixel variance over all filtered images. In the figure we visualize the selected scales for both methods and observe that they agree for 99.7% of the pixels.

4.2 Quadratic Approximation

We now show how we can compute the bias term without knowledge of the true pixel value. Let us assume that the true image is a quadratic function within the support of the coarse and fine filters f and c at each pixel. In addition, we require the filters to have vanishing first central moments. We denote the scale of the coarse and fine filter by r_c and r_f . It is straightforward to show that in this case, there is a simple relation between the bias of the filters based on their relative scales [Silverman 1986],

$$\text{Bias}[c] = \frac{r_c^2}{r_f^2} \text{Bias}[f].$$

Let us denote the true value of the image by ξ , and, by slight abuse of terminology, the filtered pixel value using the coarse filter by c and the value using the fine filter by f . Then $\text{Bias}[c] = c - \xi$ and $\text{Bias}[f] = f - \xi$. Using the above relation between $\text{Bias}[c]$ and $\text{Bias}[f]$ we get two equations in two unknowns, and we can eliminate ξ . After some more algebraic manipulation, we find that we can express $\text{Bias}[c]^2 - \text{Bias}[f]^2$ in terms of $c - f$ as

$$\text{Bias}[c]^2 - \text{Bias}[f]^2 \approx \frac{r_f^2 + r_c^2}{r_f^2 - r_c^2} (c - f)^2,$$

where the approximation is exact for quadratic image regions. Using this approximation, we get the following expression for the change in MSE,

$$\mathcal{S} \approx \underbrace{\frac{r_f^2 + r_c^2}{r_f^2 - r_c^2} (c - f)^2}_{\text{approximate bias term}} + \underbrace{\text{Var}[c] - \text{Var}[f]}_{\text{variance term}}. \quad (1)$$

We call this our *scale selector* \mathcal{S} using an approximate bias term. If the scale selector is positive, we select the fine scale f ; otherwise, we proceed to the next pair of coarser scales.

We compare MSE minimization using our scale selector \mathcal{S} to the two previous methods in Figure 3(c). Similar as above, we use filtered reference images to evaluate the bias term and empirically

established ground truth pixel variances. For 82.8% of pixels our method agrees with ground truth scale selection in (a), indicating that the quadratic approximation is valid for most pixels. Results for the other scenes of Figure 12 are given in Table 1.

Table 1: Percentage of pixels in agreement with scale selection obtained using exhaustive search, for both the incremental approach (Inc.) and our approximation (Our) for scenes of Figure 12.

Scene	Inc.	Our	Scene	Inc.	Our
killeroos	97.7%	92.8%	toasters	98.7%	91.9%
plants-dusk	95.5%	86.0%	yeahright	98.2%	87.8%

4.3 Estimation from Noisy Data

Of course in practice, we do not have access to ground truth data for the filtered pixel values, c and f , and their variances, $\text{Var}[c]$ and $\text{Var}[f]$. Instead, we need to estimate them from the noisy data available, i.e., the Monte Carlo samples that we acquired so far in the iterative procedure. We denote these samples by $s_i, i = \{1 \dots k\}$. We now express the filtered pixel values and their variances directly using the Monte Carlo samples. Again, our equations are for an individual pixel.

A filtered pixel value for, e.g., the fine filter is simply the weighted average

$$f = \sum_{i \in \{1 \dots k\}} w_i^f s_i,$$

where w_i^f are the filter weights for each sample s_i . The expression for the coarse filter is analogous. The pixel variance is

$$\text{Var}[f] = \sum_{i \in \{1 \dots k\}} (w_i^f)^2 \text{Var}[s_i], \quad (2)$$

where we use the squared filter weights, because the Monte Carlo samples are supposed to be uncorrelated. Again, the expression for $\text{Var}[c]$ is analogous. Unfortunately, $\text{Var}[f]$ and $\text{Var}[c]$ rely on the variances of the Monte Carlo samples $\text{Var}[s_i]$, which are not known. Therefore, for each s_i we use the empirical variance over all samples that are in the same pixel (i.e., square pixel region) as s_i . Let us denote this subset of samples by P . Then

$$\text{Var}[s_i] \approx \frac{1}{|P| - 1} \sum_{j \in \{P\}} (s_j - \bar{s})^2, \quad (3)$$

where \bar{s} is the mean of the samples in P , and $|P|$ is the number of samples in the pixel.

We have now expressed our scale selector \mathcal{S} in Equation 1 directly using the Monte Carlo samples. Interpreting the Monte Carlo samples as random variables, \mathcal{S} is itself a random variable. This opens up the possibility to analyze the behavior of \mathcal{S} under given assumptions about the random samples. Unfortunately, even for normally distributed s_i , its density is highly complex and not easily amenable to analytical investigation.

Empirical Analysis. We analyze the probability density of the scale selector \mathcal{S} using an empirical experiment. We use independent and identically distributed samples s_i that sample a constant 1D function with value zero and additive noise, for simplicity. The 1D function consists of one million “pixels”. We use Gaussian filters for f and c with a fixed relative scale of two. The free input parameters of our experiment are:

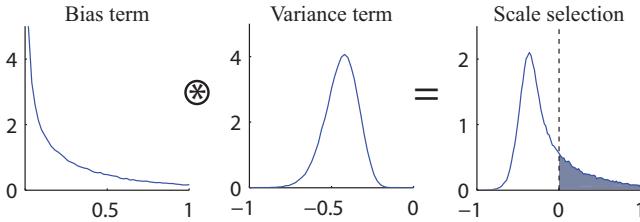


Figure 4: Histograms of bias, variance, and scale selector \mathcal{S} , which is the convolution of the former two. The horizontal axes indicate the values of the random variables, and the vertical axes the relative frequencies of the corresponding values. The parameters are: normally distributed samples with variance 10, number of samples per pixel $|P| = 32$, and a filter scale factor $r = 1$.

- The noise variance of the random samples s_i .
- A radius r for the fine filter, which determines the weights w_i^c and w_i^f .
- The number of samples per pixel $|P|$ used to estimate empirical sample variance $\text{Var}[s_i]$.

For this experiment, the ideal decision of the scale selector \mathcal{S} is to always select the coarse scale (i.e., \mathcal{S} is negative), since the true bias for all filters is zero and going to a coarser scale reduces variance. Because of the noisy s_i , however, the scale selector will occasionally make a wrong decision. We call the probability that the selector is positive and makes a wrong decision the *error rate* of the selector. Our goal now is twofold: First, to understand how the error rate is related to the parameters of the experiment; and second, to derive a single user specified parameter that directly controls the error rate, independent of the input parameters.

Before discussing the results of the experiment, we point out that the bias and variance terms in the scale selector are independent random variables, because a weighted sum of samples (as in the bias term) is independent from the empirical variance of the samples (as in the variance term). Therefore, the probability density of the scale selector is a convolution of the densities of the bias and the variance term, and we can study these separately. In Figure 4 we plot the histograms of the bias and variance terms and the histogram for the scale selector. The histograms are collected from the one million 1D “pixels” in our experiment. The error rate is simply the area under the histogram of the scale selector over the positive part of the horizontal axis, indicated by the area shaded in gray. We now summarize the observations we made in our experiment by varying the input parameters.

Sample Variance. The error rate remains constant under different variances of the random samples s_i . This is because the bias and variance terms are both proportional to the variance of the samples. Consequently, the scale selector, which is the sum of the two, is scaled as well. But scaling a random variable does not change the probability that it is larger (or smaller) than zero, hence the error rate remains constant.

Filter Radius. The error rate depends weakly on the filter radius r . Larger scales lead to slightly lower error rates. We observed that the bias term is scaled inversely proportional to r . The variance term, however, is not scaled exactly the same. Instead, the density of the variance term becomes slightly sharper with larger radii as illustrated in Figure 5(a). Because the error rate is based on the convolution of the densities of the bias and variance terms, it will

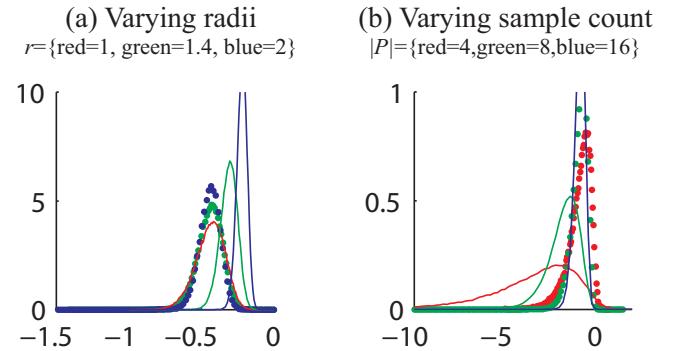


Figure 5: Histograms for the variance term under varying (a) radii r and (b) sample count $|P|$. In (a) we illustrate that the variance distribution is not inversely proportional to the radii. We scale the distributions for radii $r = 1.4$ and $r = 2$ with factors 1.4 and 2 to match the distribution of $r = 1$. We depict the result with dotted lines, showing that the variance distribution becomes slightly sharper for larger filters. In (b) we provide a similar visualization for different sample counts $|P|$. We scale the histograms for $|P| = 4$ and $|P| = 8$ with factors 1/4 and 1/2 to match the histogram for $|P| = 16$. The results (dotted lines) show that the scaled distributions are less sharp than the true distribution.

be slightly reduced for larger filters due to the sharpening of the variance term.

Number of Samples per Pixel. The error rate remains largely constant under different numbers of samples per pixel $|P|$, except for low sample counts, where the error rate increases. We observed that the bias term is again scaled inverse proportionally to $|P|$. The variance term is scaled similarly, but for low sample counts the shape of its density is significantly different as visualized in Figure 5(b). We have observed that this mismatch leads to significantly higher error rates for low sample counts.

Intuitive Parameterization. Based on the above observations, we introduce an intuitive user specified parameter γ that allows the user to directly indicate the desired error rate for constant inputs. This can be achieved easily by weighting (i.e., scaling) the bias and variance terms relative to each other. We apply a weight consisting of two factors ρ and $z(\gamma)$ to the bias term. The factor ρ compensates for low sample counts, while $z(\gamma)$ controls the error rate. Both are empirically determined. We found that a factor $\rho = (1 - 1/|P|)$ works well in practice to compensate for the effect of low sample counts. Using experimentation, we determined the factor $z(\gamma)$ such that the user parameter γ approximately achieves the desired error rate. We manually found an appropriate mapping to be $z(\gamma) = -\log(1 - (1.9\gamma)^{(1/\sqrt{2})})$ valid in the range $\gamma \in 0 < \gamma < 0.4$. Error rates above 0.4 are not interesting in practice. An alternative would be to tabulate the relation between γ and the error rate to provide an intuitive user parameter.

4.4 Post-Processing the Filter Selection

Controlling the error rate is useful to adjust the trade-off between bias and variance, but any non-zero error rate will produce a given percentage of wrong filter selection decisions that will be noticeable as spikes in the reconstructed image. Therefore, we post-process the filter selections to remove these outliers. We illustrate the process in Figure 6, where we filter a 1D signal consisting of two boxes with uniform noise, evaluated at 250 1D pixels. Each pixel received

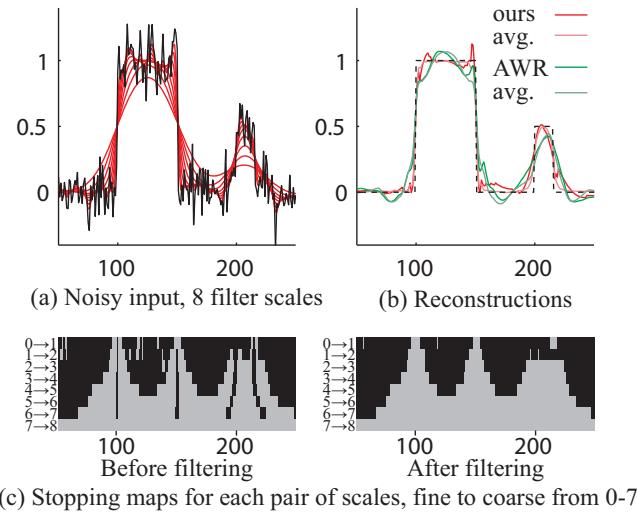


Figure 6: Illustration of outlier removal during post-processing: (a) noisy input with 8 filter scales, (b) reconstructions (including average over 200 runs) of our approach and wavelet soft thresholding, (c) binary stopping maps for each pair of scales.

32 samples that we used to estimate empirical pixel variance. Our filter set consists of Gaussian filters at eight scales related by scaling factors of $\sqrt{2}$. Figure 6(a) depicts the noisy input signal and the 8 filtered inputs. We show the reconstructed signal in Figure 6(b).

We represent the results of our scale selectors as binary *stopping maps* for each pair of scales at each pixel. This is shown as one row per filter pair in Figure 6(c). A value of 1 (shaded gray) means that we should stop and use the fine scale, and 0 (black) indicates that going to the coarse scale is estimated to reduce MSE. An important observation is that outliers appear as isolated clusters of zeros or ones, where the size of the clusters is related to the size of the pair of filter scales. Hence, we can remove outliers by additionally filtering each map (row), where we choose the size of the outlier removal filter according to the size of the pair of scales. In practice we obtained good results with Gaussian filters of the same size as the coarse scale. Then we round the filtered map back to binary. In practice, we ignore the center pixel of the outlier filter, because this dominates the result too much for small outlier removal filters. A more thorough investigation to determine optimal outlier removal filters is an interesting topic for future research. The final filter selection at each pixel is the finest scale with value 1 (gray in the figure). This process works well in practice because the scale selector guarantees a low outlier rate.

Figure 6 also includes a comparison to soft wavelet thresholding using Daubechies wavelets as used by Overbeck et al. [2009]. For both methods we show the reconstruction of the noisy signal on the left, and the average reconstruction by generating the noisy signal 200 times. The average per pixel MSE for wavelet thresholding is 7.6×10^{-3} , and for our method it is 2.5×10^{-3} . The average reconstruction from Wavelet thresholding also contains ringing artifacts, which stem from aliasing in the wavelet decomposition.

Figure 7 visualizes postprocessing using a 2D example, where we show the stopping map from scale 2 to 3. The input data uses 32 noisy samples per pixel. We remove most outliers without losing any features from the unfiltered map. In Figure 8 we compare the performance of our complete filter selection procedure to selection maps obtained by minimizing MSE according to empirical ground truth statistics collected from a large number of samples, similar as

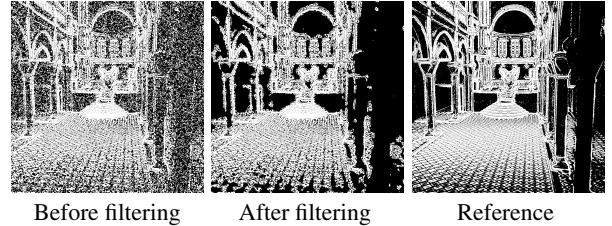


Figure 7: Filtering a binary stopping map from scale 2 to 3 for the “sibenik” scene. Left: before outlier removal; middle: after post-processing; right: reference from ground truth statistics.

in Figure 3. We use five filters at dyadic scales. The coarsest scale is visualized in white, the finest one in black. Note that the black areas in the map of the leftmost scene correspond to completely black regions in the image (see also Figure 1). Our maps were computed using a uniform sample distribution of 32 samples per pixel. We lose some detail compared to ground truth, but reliably retain the main features. All maps were obtained by setting the error rate parameter γ to 0.2 and using outlier removal.

5 Sample Distribution

The goal of the sample distribution is to place new samples in the image plane, such that the *relative* MSE given the current per pixel filter selection is reduced as much as possible. The idea is to select the m pixels, whose relative MSE can be improved the most by distributing n new samples over their selected filters.

We estimate the MSE of a selected filter based on the same approach as in Section 4. We simply accumulate the estimated MSE differences until we reach the selected filter. The MSE of the finest scale is estimated as its variance, since the finest scale is considered unbiased as it uses the pixel filter requested by the user. We compute relative MSE by dividing the estimated MSE by the squared value of the selected scale plus a small constant $\epsilon = 0.001$ to prevent overemphasizing very dark image areas. Because (relative) MSE is inversely proportional to the number of samples contributing to a filter, adding n samples over the support of a filter that received a total of n_s samples from previous iterations reduces (relative) MSE by a factor $n_s/(n + n_s)$. Therefore, after some algebraic manipulation we find that the reduction in relative MSE after adding n samples is $(\text{relative MSE}) \cdot n/(n + n_s)$.

We maintain a priority queue of pixels according to their potential error reduction. In each iteration, we retrieve m pixels from the queue and distribute samples over their filters. The assumption behind our computation of potential MSE improvement is that the filters of these pixels do not overlap, but we do not enforce this. We randomly select n pixels in the support of each filter, where we use importance sampling according to the filter weights. We then draw one (or more, if the pixel was chosen several times) additional sample for each of the selected pixels. This allows us, if desired, to sample pixels using per-pixel low discrepancy sequences. We maintain the total number of samples per pixel to be able to compute the potential MSE improvement as required above.

Figure 9 visualizes the sample distribution generated by our algorithm using an error rate of $\gamma = 0.2$ and with an average of 32 samples per pixel. We compare our sample map to a “ground truth map”, which is obtained using the same algorithm but with ground truth statistics for MSE estimation, as for the comparison in Figures 3 and 8. We also include the distribution obtained from AWR in the comparison. We observe that our approach exhibits more adaptivity. The AWR algorithm also produces density

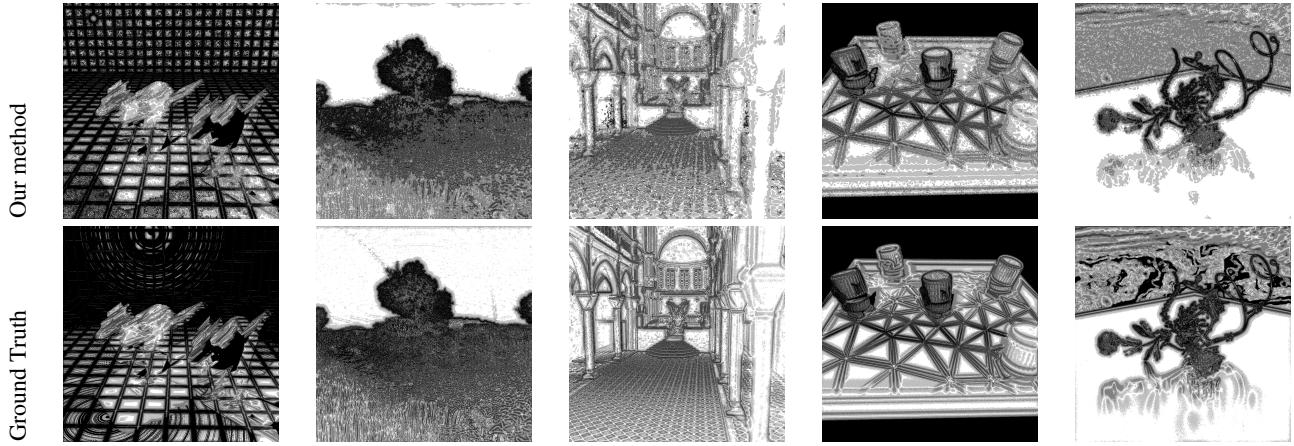


Figure 8: Selection maps for five scales using our method (top) and ground truth statistics (bottom). Our method selects filters using statistics from 32 samples per pixel, while ground truth statistics are collected from a large number of samples. We show final renderings in Figure 12.

peaks aligned with the subsampled wavelet grids. In contrast, our method operates at full resolution for all filter scales, which leads to a smoother sample distribution.

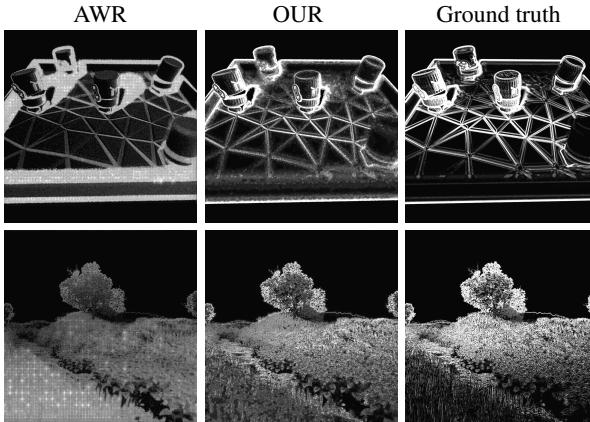


Figure 9: Sample densities with an average of 32 samples per pixel obtained with the AWR algorithm, our scale selection method, and scale selection using ground truth MSE values. The AWR method samples smooth regions more densely. Our algorithm better detects smooth regions and distributes more samples in the high-frequency regions, approximating the ground truth distribution more closely.

6 Implementation

We now detail some important aspects of our implementation: computation of filter scales and their bias and variances, image reconstruction from a sub-pixel grid to accommodate non-uniform sample distributions, additional details concerning final reconstruction, and a brief description how we integrate our method in PBRT.

Computation of Scales and Their Statistics. Computing filter scales and their variances directly using the Monte Carlo samples as described in Section 4.3 would be expensive in terms of computation and storage. Instead, we store only the finest filter scale and the mean of the empirical sample variances in each pixel and update them in each iteration. Updating the finest filter scale is

straightforward. Since variance decreases linearly with the number of samples, using Equation 3 the mean variance of pixel p is

$$\text{Var}[p] \approx \frac{1}{|P|} \frac{1}{|P|-1} \sum_{j \in \{P\}} (s_i - \bar{s})^2.$$

We update pixel mean variances from one iteration step to the next by maintaining the necessary terms separately.

In each iteration we compute the coarser scales and their variances by further filtering the finest scale and the mean pixel variances. For the variances at a given scale we filter the initial pixel mean variances $\text{Var}[p]$ using the *squared* version of that scale filter. This method is valid if the initial pixel mean variances are uncorrelated [Overbeck et al. 2009]. Since we estimate these using samples landing within each individual square pixel, i.e., the estimates do not share any samples, there is no correlation as required.

In our implementation we use Gaussian filters and their squares. Since both are separable, filtering is efficient. We do not perform any local updates but simply filter the complete finest scale and the pixel mean variances after each iteration. As an important detail, our implementation also allows us to use an arbitrary pixel filter employed by the renderer as our finest scale, while only the coarser scales are Gaussian filtered. We compute the bias term for pairs of coarser scales as described in Section 4.2. To compute the bias term for the transition from the pixel filter to the first Gaussian filtered scale, note that the pixel filter is unbiased by definition. Therefore, the bias term here is simply the squared difference between the pixel filter and the first Gaussian filtered scale. In practice, we use the pixel filter employed by the renderer as the finest scale, in addition to four Gaussian filters at dyadic scales.

Filtering Non-Uniform Sample Distributions. Our sample density map has sharp and significant changes across image edges. Therefore, adjacent pixels may receive significantly different numbers of samples, which could severely bias the reconstruction if not handled correctly. Following the work of Mitchell [1987], we store samples on a subpixel grid. Each individual subpixel value is obtained using a subpixel box filter, which is valid provided the sample distribution is uniform over the subpixel. We use a subpixel grid resolution of 4×4 . At low sample counts, the subpixel grid has many holes that we must fill. We use a simple two-step pull-push strategy, where we fill empty sub-pixels with the mean value of the samples in the whole pixel. To minimize the number of holes

in the subpixel grid, we distribute images samples using scrambled low-discrepancy sequences [Kollig and Keller 2002]. Other dimensions are still sampled using pure uniform random distributions. We then apply our filters on the subpixel resolution.

Final Reconstruction. To optimize final image quality we use a slightly modified filter selection procedure in the last iteration of our procedure (Figure 2). First, we use eight Gaussian filters related by a factor of $\sqrt{2}$, as opposed to four filters related by a factor of 2 during the previous iterations. The standard deviation of the finest scale is 1.0 during the adaptive phase and $\sqrt{2}$ during the final reconstruction. The increased number of scales reduces seams that may appear at transitions between scales in the final reconstruction. We found that using more scales during the adaptive process does not reduce error, but comes at a performance penalty. Second, we use a larger Gaussian filter for filtering the binary stopping map during the final reconstruction. While we use the coarser scale Gaussian filter during the adaptive phase, we double this size for the final reconstruction. Using smaller filters during the adaptive process better samples clusters of outliers, while using larger filters during the final reconstruction ensures we smooth out most of the remaining outliers. Third, when filtering the binary stopping maps we use the filtered binary value only if it leads to selecting the coarser scale. By disregarding filtered binary values that switch to stopping at the finer scale we suppress spike noise more effectively.

Integration with PBRT. We have implemented our algorithm using PBRT to demonstrate its compatibility with standard Monte Carlo ray tracers. Our implementation consists of a scale selector and an adaptive sampler. The adaptive sampler implements the existing PBRT *Sampler* interface. It has two states: “initialization” and “adaptive”. Rendering starts with the “initialization” phase, where each pixel is sampled uniformly using 4 samples per pixel. After initial sampling, the scale selector first computes the filtered scales and the corresponding variances as described above. It then uses the scale selector from Section 4.3 and the post-processing algorithm from Section 4.4 to determine the scale for each pixel that minimizes the MSE. During the “adaptive” phase of the sampler, all pixels are first ranked according to their estimated improvement in relative MSE, which is established as described in Section 5. We then select the m pixels with the highest estimated improvement. In our implementation, the user specifies a desired average number of samples per pixel n that should be distributed over the iterations. We performed 8 iterations in all our results. Hence, for an image composed of M pixels, we distribute $N = M(n - 4)/8$ samples per iteration, over $m = N/n$ pixels, and each selected pixel receives n samples. The n samples are distributed over the selected filter scale as described in Section 5.

7 Results

We generated all results on a dual 4-cores XEON system at 2.50GHz, with 8GB of RAM, using 8 threads. We implemented both our approach and the AWR algorithm on top of PBRT [Pharr and Humphreys 2010]. For our method, we set the error rate parameter γ to 0.1 for all scenes. For the AWR algorithm, we use the Daubechies 9/7 wavelets, a smoothing constant $c_s = 1$, and renormalization factor of 1.05 as suggested in the original paper. Due to the limited amount of samples drawn per iteration with the AWR algorithm (from 64 to 2048), our implementation scales poorly to 8 cores, which dramatically decreases its performance.

We present results from five test scenes using a wide range of effects, geometries and materials. The “killeroos” scene showcases the impact of motion blur. The “plants-dusk” scene uses envi-

ronment lighting with a very complex geometry. The “yeahright” scene shows glossy materials, a model with fine geometric details, environment lighting, and one-bounce indirect illumination. The “plants-dusk”, “toaster” and “sibenik” scenes were used in the original AWR paper [Overbeck et al. 2009]. We modified the “sibenik” scene to have an environment light (seen by refraction through the windows) and a single environment light over the gargoyle.

We compare the following four methods:

- NAIVE: Uniform sample distribution and filter using finest kernel. This is the default PBRT behavior.
- AWR: Adaptive sample distribution and reconstruction using wavelet coefficient shrinkage [Overbeck et al. 2009].
- OUR: Our proposed approach with adaptive sampling and filter selection.
- OUR-GRD: Our proposed approach with adaptive sampling and filter selection, but with ground truth bias and variance values. This is to illustrate the best results we could theoretically achieve with our technique.

In Figure 10 we illustrate the impact of γ , the single user parameter of our method, on the “sibenik” scene. Intuitively, γ controls the error rate for filter selection in uniform areas. Lower γ values produces smoother results in uniform areas, but they tend to blur across edges. Larger γ values preserve edges better, but exhibit artifacts due to outliers in uniform areas. The experiment shows that the MSE remains relatively constant for a wide range of γ values from about 0.20 to 0.30, meaning that scale selection and outlier removal operate effectively. For lower γ values, scale selection tends to pick filters that are too smooth. For higher values, the error rate in uniform areas becomes too large, such that we are not able to remove outliers robustly any more.

In Figure 11 we report on the convergence of the four methods measured in terms of relative MSE to a reference image produced with PBRT. We compute relative pixel MSE as $(img - ref)^2/(ref^2 + \epsilon)$, and we report the average error over the images. We set $\epsilon = 0.01$ to prevent over-weighting of errors in very dark regions. Our algorithm explicitly attempts to minimize this error, which AWR does not. We acknowledge that relative mean squared error is not a perfect measure for image quality, but we believe it is still a useful indicator for the convergence rate of a method.

For all scenes, our method consistently improves upon both the NAIVE and AWR approaches. The AWR method usually improves the MSE at lower samples counts, however converges slowly to the right solution. Several factors could cause this behavior. First, the functional variance estimate in AWR only considers the maximum and minimum sample values within a pixel. This essentially maximizes the impact of outliers, both bright and dark. Another source of error are ringing effects in dark regions, emphasized by the relative error metric. Finally, using the standard parameters for AWR tends to overblur images, leading to smooth results, but not necessarily low numerical error. This is apparent as AWR has low error initially for scenes with large uniform, but noisy areas (“killeroos”, “sibenik”). In contrast, the error remains high for the “plants-dusk” scene, which contains a lot of high frequency details.

Figure 11 evaluates the performance of our and AWR’s adaptive filtering techniques used with standard uniform sampling. In the “killeroos” and “plants-dusk” scenes, uniform sampling performs significantly worse than adaptive sampling. Adaptive sampling only slightly improves the MSE of the “sibenik” scene, despite the apparent adaptivity of our sample distribution (see Figure 10). Even with reference statistics we observe a similar behavior. This may

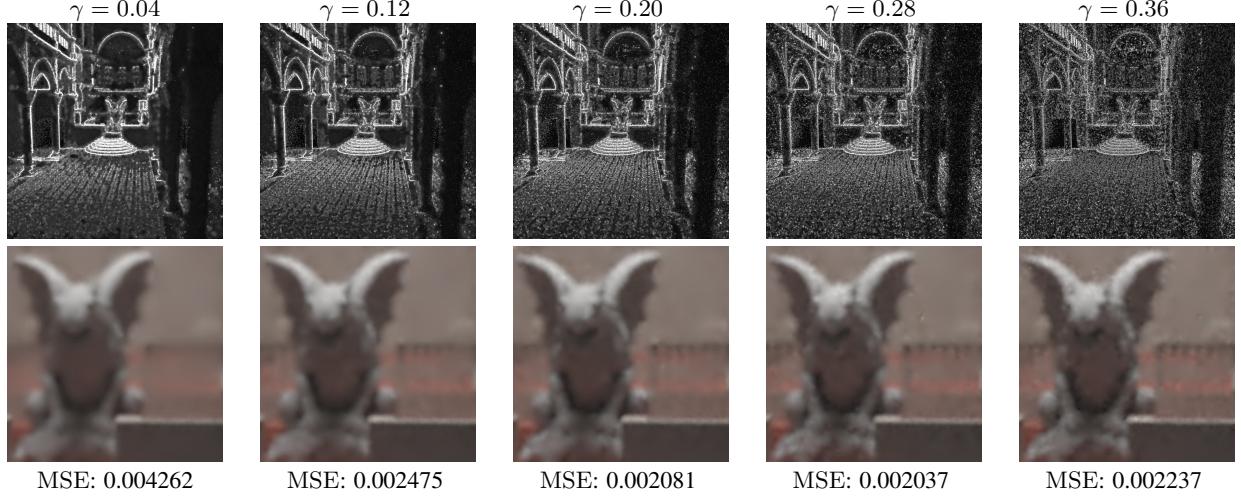


Figure 10: Top: sample densities obtained for the “sibenik” scene by varying γ . Low values yield a better reconstruction of smooth regions, which leaves more samples to resolve edges. High values yields more uniform sample densities, since outliers are draining more samples.

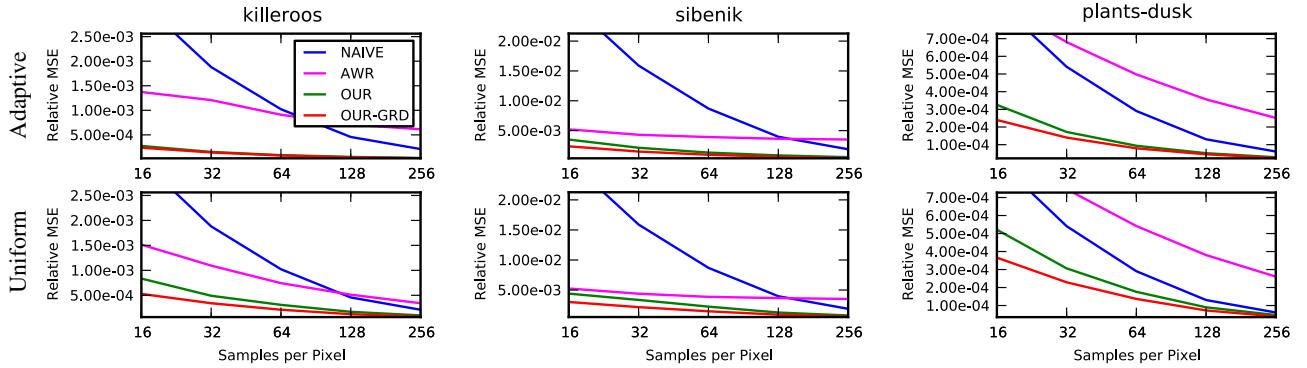


Figure 11: Convergence plots over average number of samples per pixel measured in average per pixel MSE. Top: adaptively distributed samples; bottom: uniformly distributed samples with adaptive reconstruction only. The NAIVE method is shown as a reference. AWR is the algorithm proposed by Overbeck et al. [2009], OUR is our proposed approach, and OUR-GRD is our approach using ground truth statistics. For scenes with a combination of noisy areas and high frequency details, such as the “killeroos” and the “plants-dusk” scenes, adaptive sampling provides a significant improvement. For the “sibenik” scene, where noise is equally distributed across the image and there are large uniform areas, adaptive sampling yields only a slight improvement, even with the reference statistics. AWR behaves similarly, but has consistently lower convergence rates. Results for the “toasters” and “yeahright” scenes are similar but omitted due to space constraints.

be because most of the error is due to variance in uniform regions, which is filtered effectively without adaptive sampling.

In Figure 12 we present results obtained with the naive approach, the AWR algorithm, our method and the ground truth adaptive result obtained by minimizing the ground truth MSE. Results obtained with the AWR algorithm frequently show ringing artifacts caused by aliasing in the Daubechies wavelet decomposition. Ringing is most evident at the edge between the animal ear and the background in the “killeroos” scene. AWR tends to oversmooth images with many details, which is best visible in the “plants-dusk” scene. The algorithm performs extremely well in uniform regions, which are reconstructed virtually noise free.

Our method gives high quality results both visually and numerically, offering a good compromise between noise and sharpness. We filter more conservatively across edges retaining most high frequency information, but still manages to smooth uniform regions effectively. The “toaster” scene is a good example: we maintain the sharp edges of the model, but have slightly noisy smooth shadow

transition on the ground. For all five scenes, our method gives results which are visually close to the solution obtained using ground truth statistics.

We evaluated the performance overhead of our method with the Google CPU Profiler, since this tool supports multi-threaded applications. For our simplest scene (“toasters”), we spend 4.36% of rendering time in our adaptive sampler and filter selection algorithms. Since our algorithm is independent of scene complexity, this is indicative of a worst case scenario. If measured in terms of samples rendered per second, there is often a larger discrepancy between our adaptive rendering performance and the uniform one. We believe the reason for this is that adaptive sampling tends to generate more “difficult” samples in average than uniform sampling. For instance, for the “killeroos” scene, our adaptive sampler targets the animals ($\sim 33K$ primitives each) much more than the ground and back wall, which yields a correspondingly larger cost per sample.

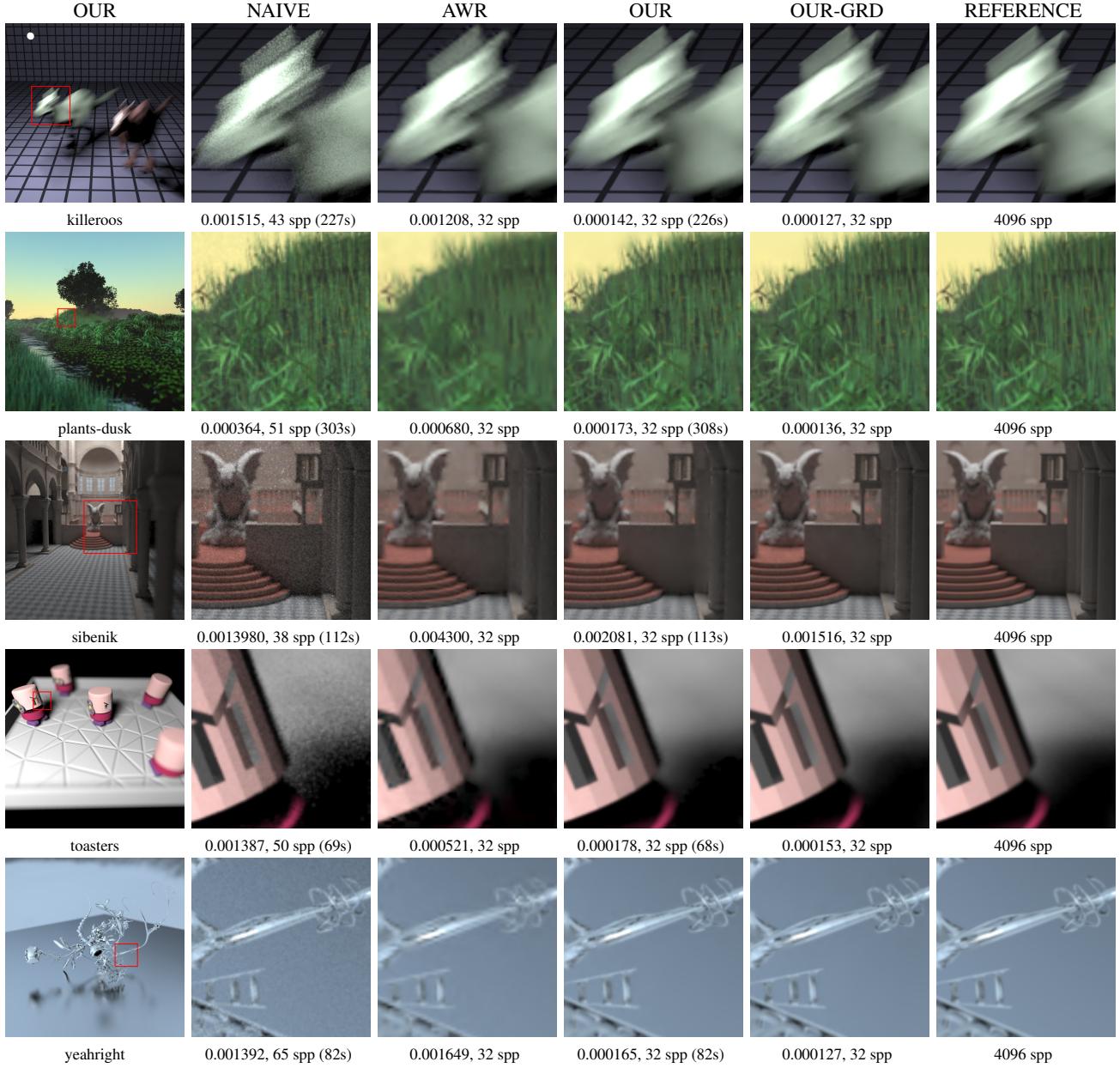


Figure 12: Result images at 1024 × 1024 pixels. The rightmost column “REFERENCE” shows converged renderings. We indicate MSE and rendering times in seconds. Timing data for the AWR is omitted since our implementation does not scale to multiple CPUs. We adjusted the number of samples in the “NAIVE” method to match the rendering times of “OUR” method, providing an equal time comparison.

Discussion and Limitations. Our algorithm relies heavily on the estimated variance to guide the adaptive sampling scheme. In tests using two samples per pixel in the initialization phase, we obtained very similar results to those reported in Figure 12 (an improvement of up to 3% of the MSE, depending on the scene), illustrating our algorithm robustness to noise in the initial variance estimate. Regardless, our algorithm has difficulties handling regions where light paths are unlikely to be found by brute force sampling. There, it tends to systematically underestimate the variance, which leads to undersampling and filtering artifacts. The dark regions of the “sibenik” scene, where more than 95% of the samples carry a null radiance, present such a case. Our method also cannot reconcile the need to filter overlapping elements in the same pixel using different kernels. The “killeroos” scene features sharp

lines which illustrate this fact (Figure 12). One can still faintly see the lines through the blurred animal head in the 4096 spp rendering, while they are mostly blurred out in our result. Also, the soft shadow cast by the animal is filtered heavily, except around the sharp lines, resulting in an increased noise level in our reconstruction. In other words, given contradictory filtering requirements, our algorithm preserves the most prominent feature. Nonetheless, the adaptive process of our method assigns more samples to these difficult regions, mitigating the problem.

8 Conclusions

We described a versatile adaptive sampling and reconstruction algorithm that greedily minimizes MSE in Monte Carlo rendering. The method provides significant improvement in terms of numerical error and image quality over previous work. A key component is a robust filter selection procedure that minimizes pixel MSE over a set of discrete filters. A main limitation of our approach is that, similar as AWR, our variance estimation assumes that the Monte Carlo renderer generates random samples. As a consequence, we will overestimate variances for low discrepancy sequences or stratified samples. One could heuristically reweight our variance term to account for this, but a more thorough solution would be desirable. A weakness of our approach is that we cannot filter noise close to edges because we only use isotropic filters. With oriented anisotropic filters we could potentially further improve the effectiveness of filtering in these situations. An interesting avenue for future work is to extend our approach to more general filters, such as cross-bilateral kernels. This would require a generalization of the differential bias estimation procedure. Finally, it would be interesting to adapt our approach to interactive GPU based ray tracing.

Acknowledgements

This project was funded in part by the Swiss National Science Foundation under grant no. 200021_127166. Implementation of the AWR paper by Marco Manzi. Killeroo model courtesy of headus 3D, toasters scene by Andrew Kensler via the Utah 3D Animation Repository, dragon from the Stanford 3D Scanning Repository, Sibenik cathedral courtesy of Marko Dabrovic and Mihovil Odak, gargoyle courtesy of INRIA via the AIM@SHAPE repository, plants ecosystem from Deussen et al. [1998], and yeahright by Keenan Crane.

References

- BALA, K., WALTER, B., AND GREENBERG, D. P. 2003. Combining edges and points for interactive high-quality rendering. *ACM Trans. Graph.* 22 (July), 631–640.
- BOLIN, M. R., AND MEYER, G. W. 1998. A perceptually based adaptive sampling algorithm. In *SIGGRAPH '98*, 299–309.
- 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*.
- DAMMERTZ, H., SEWTZ, D., HANIKA, J., AND LENSCHE, H. P. A. 2010. Edge-avoiding à-trous wavelet transform for fast global illumination filtering. In *Proceedings of the Conference on High Performance Graphics*, Eurographics Association, Aire-la-Ville, Switzerland, HPG '10, 67–75.
- DEUSSEN, O., HANRAHAN, P., LINTERMANN, B., MĚCH, R., PHARR, M., AND PRUSINKIEWICZ, P. 1998. Realistic modeling and rendering of plant ecosystems. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, ACM, New York, NY, USA, SIGGRAPH '98, 275–286.
- DONOHO, D. L., AND JOHNSTONE, J. M. 1994. Ideal spatial adaptation by wavelet shrinkage. *Biometrika* 81, 3, 425–455.
- 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.* 28 (July), 93:1–93:13.
- EGAN, K., HECHT, F., DURAND, F., AND RAMAMOORTHI, R. 2011. Frequency analysis and sheared filtering for shadow light fields of complex occluders. *ACM Transactions on Graphics* 30, 2 (Apr.), 9:1–9:13.
- FARRUGIA, J., AND PÉROCHE, B. 2004. A progressive rendering algorithm using an adaptive perceptually based image metric. In *Computer Graphics Forum*, vol. 23, Wiley Online Library, 605–614.
- GOLDENSHLUGER, A., AND NEMIROVSKI, A. 1997. On spatial adaptive estimation of nonparametric regression. *Math. Meth. Statistics* 6, 135–170.
- 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.* 27 (August), 33:1–33:10.
- KATKOVNIK, V. 1999. A new method for varying adaptive bandwidth selection. *Signal Processing, IEEE Transactions on* 47, 9, 2567–2571.
- KOLLIG, T., AND KELLER, A. 2002. Efficient multidimensional sampling. *Computer Graphics Forum* 21, 3, 557–563.
- LEE, J.-S. 1980. Digital image enhancement and noise filtering by use of local statistics. *IEEE Trans. PAMI* 2, 2 (march), 165–168.
- LEHTINEN, J., AILA, T., CHEN, J., LAINE, S., AND DURAND, F. 2011. Temporal light field reconstruction for rendering distribution effects. *ACM Trans. Graph.* 30 (August), 55:1–55:12.
- LEPSKI, O. V., MAMMEN, E., AND SPOKOINY, V. G. 1997. Optimal spatial adaptation to inhomogeneous smoothness: An approach based on kernel estimates with variable bandwidth selectors. *The Annals of Statistics* 25, 3, pp. 929–947.
- MITCHELL, D. P. 1987. Generating antialiased images at low sampling densities. *SIGGRAPH Comput. Graph.* 21 (August), 65–72.
- OVERBECK, R. S., DONNER, C., AND RAMAMOORTHI, R. 2009. Adaptive wavelet rendering. *ACM Trans. Graph.* 28 (December), 140:1–140:12.
- PHARR, M., AND HUMPHREYS, G. 2010. *Physically Based Rendering, Second Edition: From Theory To Implementation*, 2nd ed. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.
- PORTILLA, J., STRELA, V., WAINWRIGHT, M., AND SIMONCELLI, E. 2003. Image denoising using scale mixtures of gaussians in the wavelet domain. *Image Processing, IEEE Transactions on* 12, 11, 1338–1351.
- RITSCHEL, T., ENGELHARDT, T., GROSCH, T., SEIDEL, H.-P., KAUTZ, J., AND DACHSBACHER, C. 2009. Micro-rendering for scalable, parallel final gathering. *ACM Trans. Graph.* 28 (December), 132:1–132:8.
- 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 *Symposium on Interactive 3D Graphics and Games*, ACM, New York, NY, USA, I3D '11, 9–14 PAGE@5.
- SILVERMAN, B. 1986. *Density estimation for statistics and data analysis*, vol. 26. Chapman & Hall/CRC.
- SOLER, C., SUBR, K., DURAND, F., HOLZSCHUCH, N., AND SILLION, F. 2009. Fourier depth of field. *ACM Trans. Graph.* 28 (May), 18:1–18:12.