Pacific Graphics 2010
P. Alliez, K. Bala, and K. Zhou
(Guest Editors)

Volume 29 (2010), Number 7

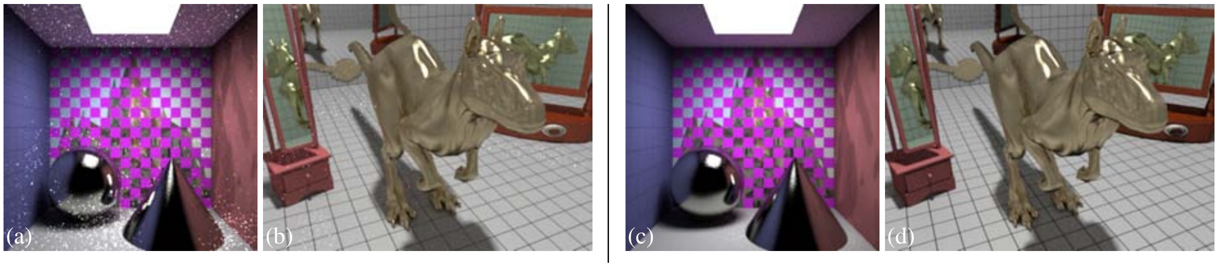
Density-based Outlier Rejection in Monte Carlo Rendering

Christopher DeCoro[†‡] Tim Weyrich[§] Szymon Rusinkiewicz[†]

[†]Princeton University [‡]Yale Law School [§]University College London

Abstract

The problem of noise in Monte-Carlo rendering arising from estimator variance is well-known and well-studied. In this work, we concentrate on identifying individual light paths as outliers that lead to significant spikes of noise and represent a challenge for existing filtering methods. Most noise-reduction methods, such as importance sampling and stratification, attempt to generate samples that are expected a priori to have lower variance, but do not take into account actual sample values. While these methods are essential to decrease overall noise, we show that filtering samples a posteriori allows for greater reduction of spiked noise. In particular, given evaluated sample values, outliers can be identified and removed. Conforming with conventions in statistics, we emphasize that the term "outlier" should not be taken as synonymous with "incorrect", but as referring to samples that distort the empirically-observed distribution of energy relative to the true underlying distribution. By expressing a path distribution in joint image and color space, we show how outliers can be characterized by their density across the set of all nearby paths in this space. We show that removing these outliers leads to significant improvements in rendering quality.



Original Renderings Renderings after Outlier Rejection, $k = 50$

Figure 1: *The standard linear reconstruction of these two renderings leads to significant peaks of noise, a result of outlying samples. We propose a method to identify and remove these outliers, leading to a significant reduction in perceptual noise. Importantly, just the noise samples are targeted by the filtering, while other salient features are left unmodified.*

1. Introduction

When simulating light transport using Monte Carlo methods such as path tracing [Kaj86], finite sampling rates produce the familiar noise artifacts as seen in Figures 1a and 1b. We draw attention to two distinct phenomena of noise: the subtle but globally-distributed high-frequency noise resulting from variance between correlated estimators, as well has the highly-localized, distinctive "speckling" resulting from statistical outliers. These outliers result from low-probability yet high-energy light paths; their error is compounded by importance sampling, which is otherwise an effective method to reduce noise. By removing these outliers in a principled manner, we significantly decrease the perceptual error, and subsequently make the task of smoothing inter-pixel variance amenable to existing filtering methods. Subsequent to outlier rejection and filtering, the same sampling rate results in significantly more plausible renderings, as shown in the two images on the right. The notion of plausibility is key: the filtered renderings are biased, but acceptable to the eye and

Published by Blackwell Publishing, 9600 Garsington Road, Oxford OX4 2DQ, UK and
350 Main Street, Malden, MA 02148, USA.

consistent with the known data, in contrast with the unbiased renderings which contain unacceptable perceptual defects.

This type of peaked noise is highly scene-dependent. In Figure 2 we show a rendering of a scene in which the specular surfaces in Figure 1a are made Lambertian. Both have been rendered with the same number of samples. The specular surfaces, however, induce a significant increase in noise. In particular, the small percentage of paths that undergo specular reflection towards a light have significantly higher energy than diffuse reflection paths, resulting in high variance. This is not dissimilar to the situation frequently encountered in direct lighting: the energetic paths occupy only a small percentage of total paths. However, since the distribution of light sources is known *a priori*, this information can be used to reduce variance through the method of importance sampling. Samples are disproportionately drawn in directions according to their known energy; resulting in most samples making an equal contribution (after dividing their energy by their probability of being selected), and thus reducing variance. Similar high-variance integrands result from peaks in the BRDF, in addition to those in incident illumination, as with highly specular reflectance. Multiple importance sampling [VG95] can combine multiple such sources of *a priori* information to reduce noise.

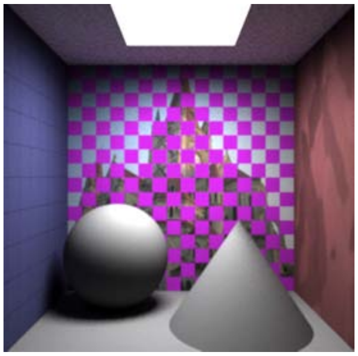


Figure 2: All-diffuse version of Figure 1a. *Note the absence of peaked noise.*

So long as low-likelihood events can be expected (such as the specular peak of a shiny BRDF) this can be accounted for using multiple importance sampling. The problem arises when the empirical distribution of energy differs widely from the expected distribution (i.e. the one from which samples are drawn). In particular, as indirect illumination is not known in advance—it is the quantity which we are using path tracing to compute—it is accordingly more difficult to compensate for variance. In fact, importance sampling can compound the problem of outliers. When a region with low expected energy density is sampled accordingly, importance sampling will weight such values higher to produce an unbiased estimator. However, when such samples in fact have high energy, this produces a significantly larger value. Techniques including defensive importance sampling [Hes95] and others [OZ00] have been developed in response to this phenomenon.

We term such samples as *outliers*. The speckling that results from outliers is a common effect in rendering that has frequently been commented on in the related literature. See for example the discussion of exactly this point in several general introductions to rendering; see, e.g. [Jen01b, Figure 9.1, Pg. 139], and note the similar artifacts in the close-ups in [PH04, Pg. 669]. We propose a definition of such outliers that allows them to be identified and removed, significantly decreasing noise.

One may employ image-space filters designed to remove noise, such as the median filter and the bilateral filter [TM98]. The non-linear median filter in particular has often been recommended for removing the sort of speckled noise as often seen in Monte Carlo renderings [Jen01b]. However, because image-space filters act solely on reconstructed pixels, significant data have already been lost by combining samples, and the existing data have been corrupted by outliers. For example, application of an image-space median filter to the example in Figure 1a results in Figure 3. In many regions of the image, enough pixels have been corrupted by outliers with the result that an accurate approximation of the original cannot be determined. Bilateral filtering performs worse; the range term *preserves* the noise, rather than discarding it. For example, Figure 6a is bilaterally filtered to produce (b), while applying our outlier filter produces (c). While bilateral filtering has removed much of the noise artifacts, many of the speckling artifacts remain; note the side of the knight, as well as the light-colored floor tiles to its left. In addition, pixels on the edge of a high-frequency feature in the underlying scene, such as at an occlusion or texture boundary, are poorly resolved (note the second blowup). These artifacts remain despite a significant range kernel width that is high enough to cause perceptible color banding (note knight's face in the first blowup, and the right-foreground rook). Compare these to (c), which cleanly removes speckling by approximating the density of samples of the rendering equation, in joint image-color space. By assuming that samples in low-density regions are outliers, and rejecting them, we produce a result that leaves only low-level noise, which is subsequently removed using a narrower bilateral filter in (d).

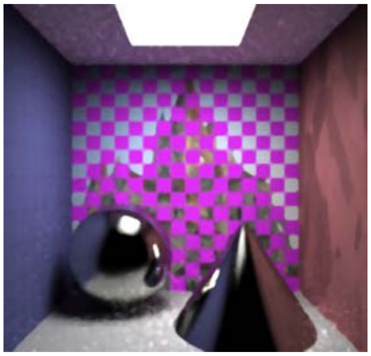


Figure 3: Median Filter in image-space*, which performs poorly due to loss of spatial detail.*

2. Background

In contrast to methods such as importance sampling, other filtering-based methods use the values of the evaluated samples to reduce noise by using an alternate reconstruction method for the final image pixels. The common approach to reconstruction of an image $I(x)$ applies a linear filter W (representing the response function of an image sensor) to the continuous light transport equation

$$I(x) = W * L = \lim_{n\to\infty} \frac{1}{C} \sum_{i}^{n} W(||x - x_i||)L(x_i) \; . \qquad (1)$$

In words, the pixel $I(x)$ is a weighted mean of points $L(x_i)$ on the image plane, where the weight W is determined entirely by the distance between x and x_i, and is independent of the radiance L. Being linear in L, any sample can have an unbounded influence on I by increasing its value; a single unrepresentative sample, when using a finite sampling rate, can significantly affect the reconstructed pixel value.

The percentage of values that, if modified, can force an arbitrary change in the estimate is known as the *breakdown point* [HRRS86]. As an estimator, the sample mean has a breakdown point of 0, but is only one example of an estimator for a *location parameter*; a value that gives the translation of a statistical distribution (see [Tuk77] for an overview). Alternate estimators have higher breakdown points. The median is an example of a commonly used resistant estimator; it has a breakdown point of 50%, that is, the estimate is resistant to arbitrary error until half of the data samples are corrupted. However, while the median has a high breakdown point, it has poor *efficiency*, measured as the variance of the estimator relative to the mean estimator for a normal distribution. The practical effect of this is shown in Figure 5a. While the outlier noise is removed, the transitions in smooth regions are made sharp, and the total level of energy in certain regions, such as the ceiling, is significantly reduced.

A generalization of both the median and the mean is the α-trimmed mean, which has been previously proposed as a robust filtering method for reconstruction in rendering [LR90]. For a value of $\alpha \in [0, .5)$, the upper and lower α order statistics of samples are discarded, and the location parameter estimated as the mean of the remaining samples. Note that the median is equivalent to the 0.5-trimmed mean. While the breakdown point is reduced from 50% to α, efficiency is also increased; as shown in Figure 5b, the 10%-trimmed mean significantly reduces variance in transition regions as compared to the mean, while continuing to remove noise. However, the energy levels of the ceiling and specular highlights are reduced, which continues to differ as compared to the reference even for a smaller α of 1% (Figure 5c), at which point noise from outliers becomes noticeable. Furthermore, the trimmed-mean estimates have an absolute error comparable to or worse than the noisy original, which differs from the reference by an RMS error of .0519. In contrast, we will see that our method performs better both quantitatively (.041 RMS error) and perceptually.

This problem can be characterized as the effect of *heteroskedastic* data, in which the variance is non-constant across the image plane. As a result, a single choice of α is inappropriate across the entire image. A potential solution is to first estimate α for each pixel, but this adds an additional level of complication. Other methods, such as the anisotropic diffusion filter [McC99] and the non-linear energy-preserving filter [RW94] attempt to adapt locally to the amount of noise in each region, and spread the excess energy out to neighboring pixels.

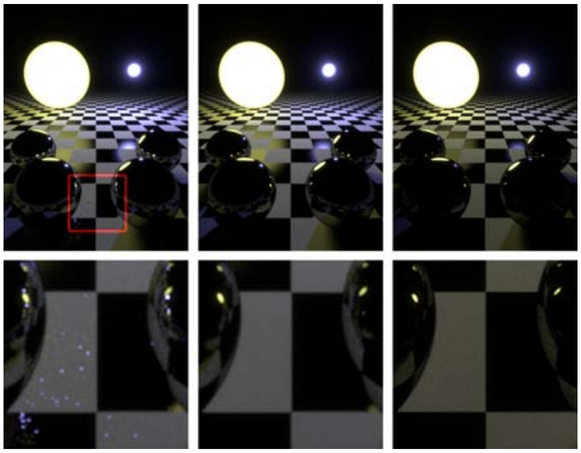

(a) Original (b) Our Method (c) 5% Trimmed

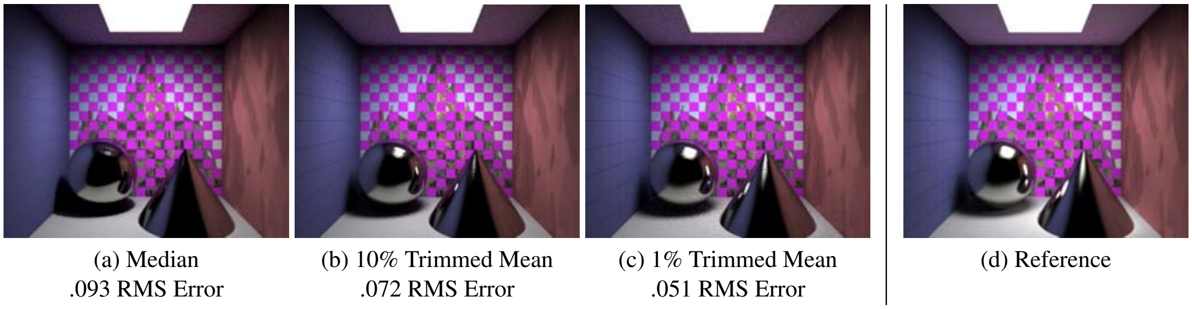
Figure 4: Multi-modal Distributions: *In this scene with two lights, the presence of specular objects cause outliers resulting in spiked noise. While a trimmed mean will reject these outliers, samples from the (significantly smaller and more intense) blue light are rejected as well. Our method rejects only the clear outliers, viewing the blue light samples as consistent with the data as a whole.*

What these methods do not address, however, is the larger problem of multimodal data. Estimators of location parameters, by their nature, find a single mode; the empirical distributions encountered in computer graphics have many modes of energy, induced by multiple lights (or partial occlusion of a single light), multiple modes of a BRDF, indirect illumination, and so on. Take for example the scene shown in Figure 4. While a trimmed mean removes the speckled noise, it also causes a significant shift in color, as rays from the smaller (but more powerful) blue light are rejected as outliers to the larger yellow light. Our method, presented in the next section, can remove noise while preserving coloration.

3. Algorithm Description

Our algorithm operates in the multidimensional space consisting of both image space coordinates (independent variables) and color coordinates (dependent variables) and considers the density of each sample in this joint space, its *joint density*. The algorithm rejects samples with low joint density as outliers. We hence term this as a *path-density filter*.

Note that the joint image and intensity space bears similarities to the bilateral grid [CPD07], which regularly samples this joint space to accelerate edge-aware image processing. In our work, however, we use the joint domain to classify



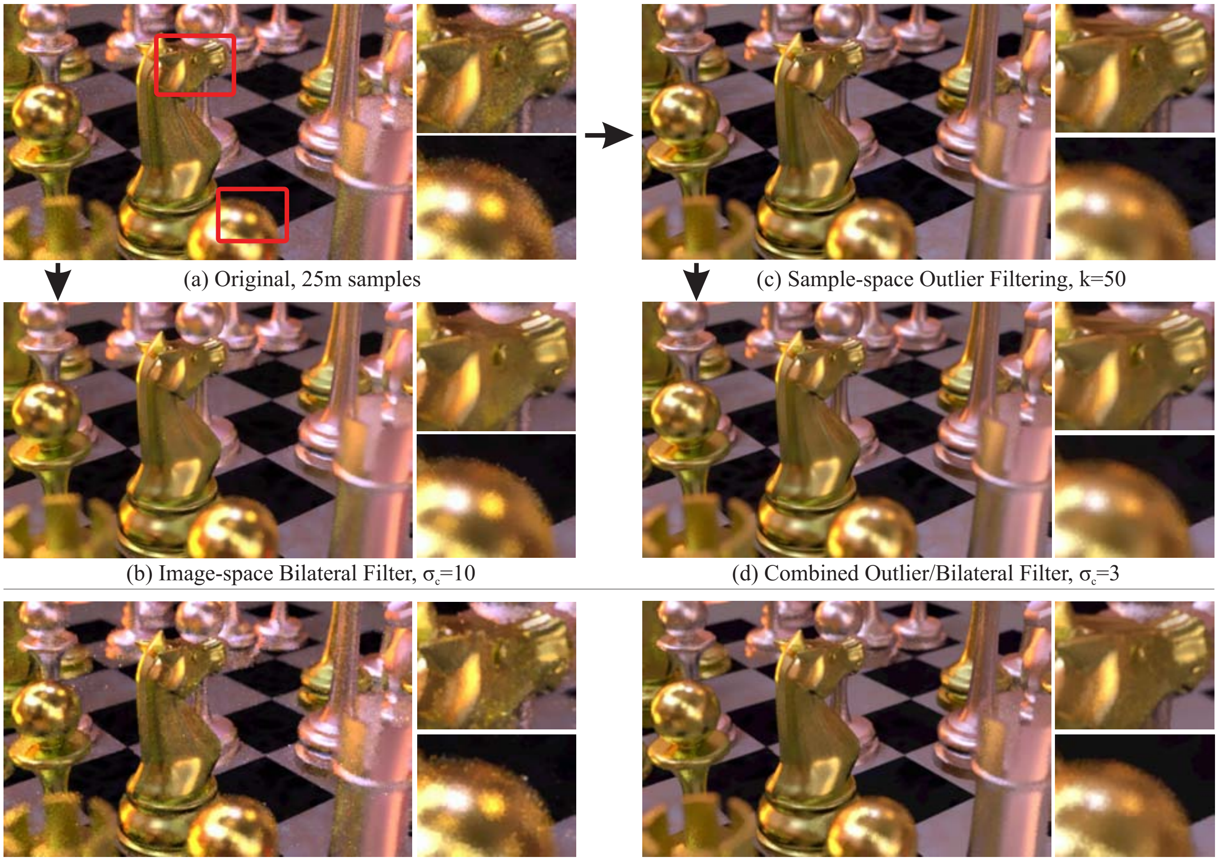
(a) Median
.093 RMS Error

(b) 10% Trimmed Mean
.072 RMS Error

(c) 1% Trimmed Mean
.051 RMS Error

(d) Reference

Figure 5: Alpha-trimmed Means: *Previous methods for outlier rejection using alpha-trimmed means remove noise at the cost of significant decrease in total energy (note especially the roof and specular highlights compared to the reference) and also in adding sharpness into previously smooth transitions (see the soft shadows). As a result, these reconstructions have significant root-mean-squared (RMS) error compared to the reference (computed with 1024 samples/pixel) that does not improve on the .0519 error of the noisy original (Figure 1a). Our method (Figure 1c) removes noise while maintaining .041 RMS error.*



(a) Original, 25m samples

(c) Sample-space Outlier Filtering, k=50

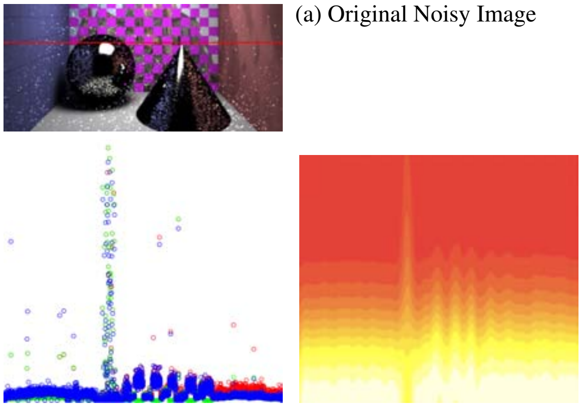
(b) Image-space Bilateral Filter, σ_c=10

(d) Combined Outlier/Bilateral Filter, σ_c=3

(e) Image-space Bilateral Filter, 5m samples, σ_c=10

(f) Combined Outlier/Bilateral Filter, 5m samples, σ_c=10
Rendering Time Approximatly 25% of (a)

Figure 6: Comparison with Bilateral Filtering. *For a noisy rendering of the Chessboard scene with 25M samples (a), a direct application of image-space bilateral filtering removes low-intensity noise but specifically preserves spike noise, even with a relatively large σ_c (b). In contrast, spike noise is removed by application of the sample-space outlier filter (c), producing a result that is amenable to removal of noise through subsequent bilateral filtering (d). At lower sample rates, the contrast is even more apparent: consider the significantly noisy bilateral filtered (e), with the result of combined outlier/bilateral filtering (f) for 5M samples. The reduction in noise comes at the cost of bias (reduced energy in noisy areas); this has been corrected in the filtered images by boosting brightness by 20%.*



(a) Original Noisy Image

(b) Sample x-location vs. intensity

(c) Joint Density

Figure 5: Joint Density Visualization. *Shown is a slice of the noisy input image (a, indicated by red line) where samples have been plotted on a graph of their x-component versus intensity (b, note that red, green and blue color components are plotted separately). Outliers are characterized by low sampling density in the joint space, rather than by actual their values. By computing density explicitly (c, colors indicate lines of equal density) we can identify and exclude these samples from image reconstruction (Figure 1c).*

irregular samples before image formation. In that sense, our method is also related to multi-dimensional adaptive sampling [HJW*08], where an irregular spatio-temporal sampling space is used to reconstruct features in joint sampling space. In contrast, our analysis focuses on the concept of outliers, which is possible by including the color domain.

To motivate our method, Figure 5a shows a close-up of the noisy image previously shown, where a single line of samples (corresponding to those along the red line) has been plotted independently in Figure 5b. The samples are plotted as image x-coordinate vs. radiance (with independent red, green, and blue channels overlaid). One can see clear outliers in the plot, which correspond to the sharp noise peaks seen in the reconstruction. Note how the density of samples is important for identification of outliers: similar to outliers, specular highlights peak in magnitude, but are confirmed by high density. In overview, our algorithm is as follows:

```
1. Initialize space-partitioning tree $T$ and image $I$
2. For each Monte Carlo–rendered sample $x$:
    a. Use $T$ to find the $k$ nearest neighbors in joint image-color space
    b. Compute $\sigma$, the average distance to neighbors
    c. If $\sigma < 1$, then splat $x$ into $I$
    d. Otherwise, add $x$ to $T$
3. For each remaining $x \in T$, repeat Steps 2a–2c.
```

We trace paths through the scene using standard methods. For each sample x we use a space-partitioning tree to find the k-nearest neighbors, where k is a user-specified parameter. As a measure of local density σ, we compute the mean absolute distance from x to its k-nearest neighbors. As the sampling rate increases, the neighbors of x increasingly converge to x itself, and σ approaches 0. If σ is large, x is considered an outlier; σ thereby measures the quality of an individual sample, as opposed to pixel variance, which measures the quality of an entire pixel. Note that our fixed choice of 1 as a threshold value for σ does not lose generality, as a varying threshold would be equivalent to scaling the weight factors of the distance metric.

Non-outliers are composited into the output image as usual. Outliers are not composited into the image, but stored in the space-partitioning tree. At first, all samples will be outliers, and will be added to the tree. However, as the number of samples increases, σ for new samples will tend to decrease. Once an area in joint image-color space has a high density of samples, additional samples will not be stored.

After rendering is finished, we perform a final pass through the samples in the tree to determine if any of them, given the addition of later samples, now has a value of σ that would classify them as a non-outlier. If so, they are also splatted into the image. (Note that for large sample counts, the relatively small number of samples remaining in the tree will have an insignificant effect on the final image).

Figure 5c displays a visualization of the density function of the samples previously shown. Distinct colors represent isolines of equal density. Note that the outliers are located in the low-density region; rejection of all points with low densities removes the outliers and their associated noise (while preserving the higher-density features such as the specular peak). This produces the rendering originally shown in Figure 1c. Figure 6 shows the average σ in our test scenes.

Choosing Neighborhood Size. The choice of the parameter k acts as a smoothing parameter. Larger values of k require samples to have stronger corroboration by its neighbors before being accepted, leading to a removal of samples with low confidence. We demonstrate this in Figure 7. Note that as k increases, the noise in the image decreases.

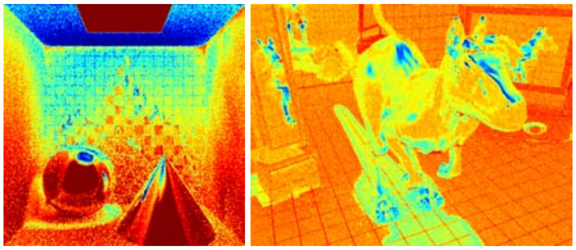
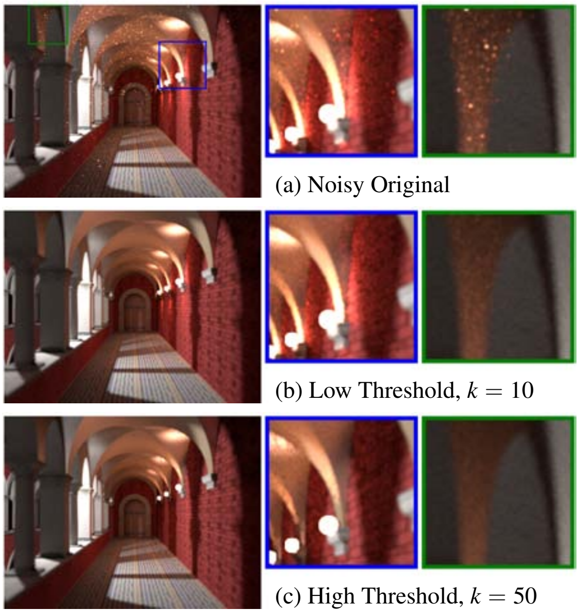


Figure 6: Mean Density Per Pixel *The images visualize the density by displaying mean σ at each pixel, color-mapped such that blue is highest and red is lowest. Note that areas such as the roof in the left image (which is lit entirely by indirect illumination) and shadow penumbrae in both images have large σ compared to their surroundings.*

Scene	Killeroo			Sponza			Chess		
Samples	1M	5M	25M	1M	5M	25M	1M	5M	25M
Standard Rendering	21 sec	39	200	9	32	160	25	120	520
Filtered Rendering, $k = 10$	23 sec / 26k	47 / 30k	220 / 37k	10 / 58k	47 / 67k	200 / 89k	26 / 54k	140 / 83k	610 / 120k
$k = 50$	25 sec / 110k	61 / 117k	260 / 130k	13 / 150k	61 / 220k	270 / 300k	32 / 160k	140 / 250k	690 / 360k
$k = 200$	–	62 / 370k	280 / 410k	–	68 / 560k	270 / 830k	–	160 / 640k	720 / 940k

Table 1: Running times and storage costs for rendering and density filtering, *for varying scenes, sampling rates, and filtering parameters; times over 100 seconds are rounded to the nearest ten seconds. For each filtered result, we show both the total time spent on rendering with filtering, and the number of samples stored in the space-partitioning tree. Time complexity is near-linear with respect to sample size, sublinear with increasing k, and correlated to scene lighting complexity – the area-light Killeroo and Sponza scenes have similar filtering times, both less than that of the environment-light Chess scene. Storage costs are near-linear with k, and roughly logarithmic with total samples. Numbers are given for an 8-thread 2.67GHz Intel Core i7 CPU, with 16GB memory. The algorithm scales nearly linearly with increasing number of cores.*


(a) Noisy Original

(b) Low Threshold, $k = 10$

(c) High Threshold, $k = 50$

Figure 7: Choosing neighborhood size. *We show the Sponza atrium using varying neighborhood sizes. As k is increased, the confidence required to retain a sample also increases. With fewer low-confidence samples in the reconstruction, noise is removed from the image, note especially the glossy archways. The k parameter allows a tradeoff between noise and bias; by increasing k to 50, additional noise is removed, at the cost of slight bias in the illumination. This bias is expressed by the removal of lower-confidence lighting features.*

Distance Metric. For our purpose, distances between two points is defined as the sum of Euclidean distance between their image space coordinates, and their color coordinates in CIELAB space,

$$d(x,y) = \sqrt{\frac{d^2(x_{image}, y_{image})}{\sigma^2_{image}} + \frac{d^2(x_{color}, y_{color})}{\sigma^2_{color}}}. \quad (2)$$

The image-space distance weight is set to approximate the sensor response W (i.e. if W is a Gaussian kernel with $\sigma = 3$ pixels, the image-space distance weight $\sigma_{image} = 3$). Color distance weight is a user parameter; we have fixed it to 100. CIELAB distance is chosen for its perceptual basis, but other color spaces, such as CIE1934 (XYZ) and sRGB are usable for this purpose. We find that it is useful to simplify parameter selection by fixing the weight applied to color distance, and instead use k as the quality-control parameter.

4. Results

As shown in Table 1, our method adds only a slight overhead to standard path tracing. For values of k in the normal range of 10 to 50, filtering adds a time overhead of around 20% to 40%. However, it provides a significantly greater improvement in rendering convergence than would have been achieved if this additional time had been spent rendering additional samples. Consider Figure 6. Subfigure (f) rendered using our method and only 5M samples displays less noise than (a), which is rendered using 25M samples. The time spent to render (f) is about 25% of that spent on (a).

Our filtering method is also memory-efficient. By storing only outlier samples in the space-partitioning tree, our method produces an adaptive representation of density. Storage is most strongly correlated with scene complexity, rather that the total number of samples. Increase in storage is roughly logarithmic with increasing samples. For example, in the Sponza scene with 1M total samples, our algorithm stores 58K samples in the tree; at 25M total samples, this only increases by 53% to 89K samples stored. Each sample is approximately 6 floating point values (X and Y image position, time and color), plus several pointers to maintain the tree. Therefore, additional storage cost is negligible.

Our method generalizes to animations by recording the time value of each sample, and using an appropriately weighted distance term. Analogous to the screen-distance term, the time-distance weight is set proportional to the time reconstruction kernel width, which in our renderings is 2 frames. Outlier noise is particularly noticeable in animations, as they results in speckles that come into view for only a single frame. In the accompanying video, we demonstrate the use of our method on an animation through the Sponza scene, previously shown in Figure 7. The most noticeable outlier noise is eliminated, and the resulting images are a significant improvement.

5. Discussion

Our method invites comparisons to photon mapping, another biased method for noise reduction that relies on density estimation [Jen01a]. Photon mapping does not suffer from the spiked noise artifacts of path tracing that our method addresses. However, it is to path tracing's advantage that it is the more general mechanism, with fewer parameters to control, and is easier to implement. Therefore, while photon mapping converges quickly for many scenes, our method has the benefit of being a very general, yet very simple, improvement of a fundamental rendering technique that performs favorably for a wide range of scenes.

As future work, it may be possible to apply joint-space outlier rejection to photon mapping. As typically implemented, photon mapping casts photons into the scene from light sources, and uses the stored photons to compute the spatial flux density at each rendered location. Analogous to our method, one could instead consider the density in joint spatial-power space. One could then reject photons that have low joint density, potentially reducing rendering artifacts, such as splotching, due to outlier photons.

Moreover, Jensen notes the importance of "ensur[ing] that the stored photons have approximately the same power. This is important for good quality of the radiance estimate...." [Jen01a, Pg. 61]. Therefore, he employs methods such as Russian roulette to avoid disparities in photon energy. It is possible, however, that one could additionally employ outlier rejection methods similar to the one presented here, allowing for wide ranges of photon energy without reduced quality.

In conclusion, we have presented a filtering method by which objects with complex, glossy reflectance can be used in a scene without penalty of introduced noise in other objects. Instead, the resulting noise is withheld from the scene until a sampling rate has been achieved that allows the rendering of lighting phenomena without noise. Importantly, basic lighting features such as direct illumination can be computed relatively quickly, while the rendering filter isolates the low-noise direct illumination from corruption from high-noise indirect illumination. It is intended that this will allow for the use of progressive level-of-detail in realistic renderings, which itself will hopefully lead to greater adaptation of physically-based light transport methods and ray tracing to supplant the heuristic methods most commonly in place today.

References

[AMN*94] ARYA S., MOUNT D. M., NETANYAHU N. S., SILVERMAN R., WU A. Y.: An optimal algorithm for approximate nearest neighbor searching in fixed dimensions. In *Proc. ACM-SIAM Symp. Discrete Algorithms* (1994).

[CPD07] CHEN J., PARIS S., DURAND F.: Real-time edge-aware image processing with the bilateral grid. *ACM Trans. Graphics 26*, 3 (July 2007).

[Hes95] HESTERBERG T.: Weighted average importance sampling and defensive mixture distributions. *Technometrics 37*, 2 (1995).

[HJW*08] HACHISUKA T., JAROSZ W., WEISTROFFER R. P., DALE K., HUMPHREYS G., ZWICKER M., JENSEN H. W.: Multidimensional adaptive sampling and reconstruction for ray tracing. *ACM Trans. Graph. 27*, 3 (2008), 1–10.

[HRRS86] HAMPEL F. R., RONCHETTI E. M., ROUSSEEUW P. J., STAHEL W. A.: *Robust Statistics - The Approach Based on Influence Functions*. Wiley-Interscience, 1986.

[Jen01a] JENSEN H. W.: *Realistic Image Synthesis Using Photon Mapping*. AK Peters, 2001.

[Jen01b] JENSEN H. W.: *State of the Art in Monte Carlo Ray Tracing for Realistic Image Synthesis*. SIGGRAPH Course Notes. 2001.

[Kaj86] KAJIYA J. T.: The rendering equation. *Computer Graphics (Proc. ACM SIGGRAPH) 20*, 4 (1986), 143–150.

[LR90] LEE M. E., REDNER R. A.: A note on the use of nonlinear filtering in computer graphics. *IEEE Computer Graphics and Applications 10*, 3 (1990), 23–29.

[McC99] MCCOOL M. D.: Anisotropic diffusion for monte carlo noise reduction. *ACM Trans. Graphics 18*, 2 (1999), 171–194.

[OZ00] OWEN A., ZHOU Y.: Safe and effective importance sampling. *Journal American Statistical Assosciation* (2000).

[PH04] PHARR M., HUMPHREYS G.: *Physically Based Rendering: From Theory to Implementation*. Morgan Kaufmann, 2004.

[RW94] RUSHMEIER H., WARD G.: Energy-preserving nonlinear filters. *Computer Graphics (Proc. SIGGRAPH)* (1994).

[TM98] TOMASI C., MANDUCHI R.: Bilateral filtering for gray and color images. In *Proc. Intl. Conf. Computer Vision* (1998), pp. 839–845.

[Tuk77] TUKEY J. W.: *Exploratory Data Analysis*. Addison-Wesley, 1977.

[VG95] VEACH E., GUIBAS L. J.: Optimally combining sampling techniques for monte carlo rendering. In *Proc. SIGGRAPH* (1995), pp. 419–428.