Optimizing Path Tracing using Noise Reduction Filters

Henrik Wann Jensen
Niels Jørgen Christensen
Dept. of Graphical Communication
Technical University of Denmark
fax: +45 45 93 83 17 phone: +45 45 93 41 66
e-mail: igkhwj@unidhp.uni-c.dk or igknjc@unidhp.uni-c.dk

November 13, 1994

Abstract

The problem of global illumination can be solved using path tracing. Unfortunately path tracing gives very noisy images. This noise is mostly caused by the indirect illumination reflected diffusely. The common way to reduce the noise is to use more samples/rays pr. pixel. However, the convergence speed in path tracing makes this strategy very costly. In this paper we propose a technique in which light reflected diffusely two times is separated from the final solution. This light is filtered by different noise reduction filters and then added to the remaining solution. In this way we avoid blurring the image and at the same time we are able to reduce the noise level significantly.

Key Words: Global illumination, path tracing, noise reduction, filtering.

1 Introduction

In 1986 Kajiya [Kaji86] introduced the rendering equation which is now well established as the theoretical foundation of global illumination.

$$L_o(\mathbf{x},\theta_o,\phi_o) = L_e(\mathbf{x},\theta_o,\phi_o) + \int_0^{2\pi}\int_0^{\pi} L_i(\mathbf{x},\theta_i,\phi_i) f_s(\mathbf{x},\theta_i,\phi_i;\theta_o,\phi_o)|\cos\theta_i|\sin\theta_i\, d\theta_i\, d\phi_i \tag{1}$$

where
(θ_i,ϕ_i) is the incoming direction
(θ_o,ϕ_o) is the outgoing direction
$L_o(\mathbf{x},\theta_r,\phi_r)$ is the outgoing radiance
$L_e(\mathbf{x},\theta_r,\phi_r)$ is the emitted radiance
$L_i(\mathbf{x},\theta_r,\phi_r)$ is the incident radiance
$f_s(\mathbf{x},\theta_i,\phi_i;\theta_r,\phi_r)$ is the bidirectional reflectance-transmittance distribution function

At the same time Kajiya introduced path tracing which is a Monte Carlo based ray tracing solution to the rendering equation. Path tracing is however a very costly method. In order to sample the indirect light properly it is often necessary to use more than 100 samples/rays pr. pixel. If fewer samples are used the result is disturbing noise in the final image. The noise can be avoided by checking the variance of the samples at each pixel. If this variance is not below some threshold then more samples has to computed. This method ensures that we do not use "to many" samples but it does not reduce the substantial amount of rays required to get an acceptable image.

To solve this problem Ward et al. [Ward88, Ward92a] introduced a caching scheme in which the indirect illumination is stored within the model. In this way the indirect illumination does not have to be sampled at every pixel. Instead it can be either interpolated or extrapolated from previously calculated values. Using this strategy it is possible to reduce the number of samples needed significantly. The interpolation strategy can however consume quite a lot of memory and the exchange of global information makes the method difficult to parallelize to a larger scale.

In [Lafo94] a constant ambient term is used in the Monte Carlo integral. This technique does however only affect the result if it is used in the termination of the light rays e.g. as a part of Russian roulette. Furthermore the constant ambient term must not differ to much from the ambient light in the model since this raises the noise level rather than reducing it. The method is therefore less useful in scenes with varying indirect light. [Veac94] uses bidirectional path tracing in which a light path is constructed by connecting a ray from the eye with a ray emitted from the light sources. This technique is especially useful in scenes with a lot of indirect light. The usage of this method is however quite subtle. It is difficult to decide how many reflections the rays traced from the light source and the eye must undergo.

Within the field of image processing the removal of noise from images is often done using noise reduction filters [Gonz92]. This technique has also been applied to images calculated using Monte Carlo ray tracing. Unfortunately filtering usually results in blurring of the image. Rushmeier et al. [Rush94] compensates for this problem by using more complex energy preserving non-linear filters with varying kernel sizes. They detect noisy pixels as pixels that have a variance above some threshold after an estimated necessary number of trials. These noisy pixels are then subjected to filtering. This is, however, problematic in scenes which contains a Perlin like noise-based texture [Perl85]. If the texture does have some high frequent "noise" then it will be seen as a candidate for filtering and this filtering operation will blur the texture unnecessarily.

In this paper we investigate how noise reduction filters can be used to reduce the noise/error in path traced images without increasing the number of samples. In order to avoid blurring we only filter the indirect diffuse illumination, which is often the noisy part of an image and in general slowly varying. These properties makes it a good candidate for noise reduction filters.

We do not discuss caustics which cannot be calculated very well with the standard path tracing technique. This problem has been investigated in several other papers e.g. [Arvo86, Ward92b] and [Jens94].

2 Sampling the Radiance

A path tracing solution consists of an image in which a radiance value is calculated for each pixel. The pixel radiance is computed by averaging a number of sample estimates.

Each sample consists of tracing a ray from the eye through the pixel based on some filtering function. The radiance for each ray is determined from the surface that the ray hits. It equals the radiance, $L_o(\mathbf{x}, \theta_o, \phi_o)$, leaving the intersection point, $\mathbf{x}$, between the surface and the ray, in the direction, (θ_o, ϕ_o), of the ray. $L_o(\mathbf{x}, \theta_o, \phi_o)$ can be found as the sum of the emitted, the transmitted and the reflected radiance:

$$L_o(\mathbf{x}, \theta_o, \phi_o) = L_e(\mathbf{x}, \theta_o, \phi_o) + L_t(\mathbf{x}, \theta_o, \phi_o) + L_r(\mathbf{x}, \theta_o, \phi_o)$$

where L_e is the emitted radiance, L_t is the transmitted radiance and L_r is the reflected radiance. In the following discussion we omit transmission which is treated analogously to reflection. Likewise we also omit the direction parameters following each radiance value.

The term L_r is computed using the rendering equation (1). The bidirectional reflectance-transmittance distribution function f_s can be separated into two components: A specular-like part and a diffuse-like part. The calculation of L_r is thus separated into the calculation of a specular part $L_{r,s}$ and a diffuse part $L_{r,d}$.

$$L_r = L_{r,s} + L_{r,d}$$

The calculation of $L_{r,d}$ can be separated further into three components.

$$L_{r,d} = \underbrace{L_{r,d,l}}_{\text{direct}} + \underbrace{L_{r,d,s}}_{\text{caustics}} + \underbrace{L_{r,d,d}}_{\text{diffuse}}$$

where $L_{r,d,l}$ is the diffusely reflected radiance due to direct illumination, $L_{r,d,s}$ is diffusely reflected radiance due to light reflected specularly from the light source, this is the term giving rise to caustics. Finally the term $L_{r,d,d}$ is diffusely reflected light that has been reflected diffusely at least one time since it left the light source.

$L_{r,d,d}$ contains almost no directional information and it is very costly to calculated using path tracing. It is necessary to use several hundred rays pr. pixel in order to get a decent estimate of $L_{r,d,d}$ that is not to noisy.

Most of the noise in path traced images is a result of an inadequate sampling of $L_{r,d,d}$. That is, the term $L_{r,d,d}$, is responsible for most of the noise in the image and since we are only interested in removing the noise from the image we only have to filter $L_{r,d,d}$. Fortunately, $L_{r,d,d}$ is often varying slowly and this makes it a good candidate for noise reduction filters. Most noise reduction filters do not only reduce the noise they also remove sharpness from the image. This is not crucial for $L_{r,d,d}$ since it has a low gradient in most parts of the image.

3 Noise Reduction Filters

The noise introduced by path tracing has a high frequency. At every pixel the noise is uncorrelated and it has zero average value. This allows us to perform an efficient reduction of the noise by low pass filtering the image. We can do this by performing a convolution of the image $f(x, y)$ with a spatial filter $h(x, y)$ resulting in an image $g(x, y)$, that is

$$g(x, y) = h(x, y) * f(x, y) \tag{2}$$

The spatial filter $h(x, y)$ determines the properties of the filtering-operation and in order to keep the low frequencies the filter must capture the slow variations in the image. A

$$\frac{1}{9} \times \begin{array}{|c|c|c|}\hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array} \qquad \frac{1}{25} \times \begin{array}{|c|c|c|c|c|}\hline 1 & 1 & 1 & 1 & 1 \\ \hline 1 & 1 & 1 & 1 & 1 \\ \hline 1 & 1 & 1 & 1 & 1 \\ \hline 1 & 1 & 1 & 1 & 1 \\ \hline 1 & 1 & 1 & 1 & 1 \\ \hline \end{array} \qquad \frac{1}{20} \times \begin{array}{|c|c|c|}\hline 1 & 2 & 1 \\ \hline 2 & 4 & 2 \\ \hline 1 & 2 & 1 \\ \hline \end{array}$$

(a) (b) (c)

Figure 1: Typical low pass filters

number of filters with this property are shown in figure 1. The linear filtering performed using equation (2) does unfortunately not only remove the noise from the image. It also removes all the high frequencies which results in blurring. This is especially the case with the filter in fig. 1(b). A large filter is better at removing noise but it also results in more blurring. The filter in fig. 1(c) puts more emphasis to the center pixel which helps avoiding blurring but also reduces the effect of the low pass filter.

The loss of sharp detail due to blurring is an unwanted effect of the linear low pass filters. Our objective is to achieve noise reduction rather than blurring and for this purpose there exists a nonlinear filtering method known as median filtering. Using this filter each pixel is replaced by the median of the pixel values in the neighbourhood of the pixel. This filter is particularly effective when the noise is spiky.

The noise reduction filters presented do not process the pixels on the edge of the image. This problem can be solved by either replicating the edge or simply by just copying edge pixels from the source image to the filtered version. Replicating the edges is probably preferable since this ensures some noise reduction on the edges whereas copying does not give any reduction at all.

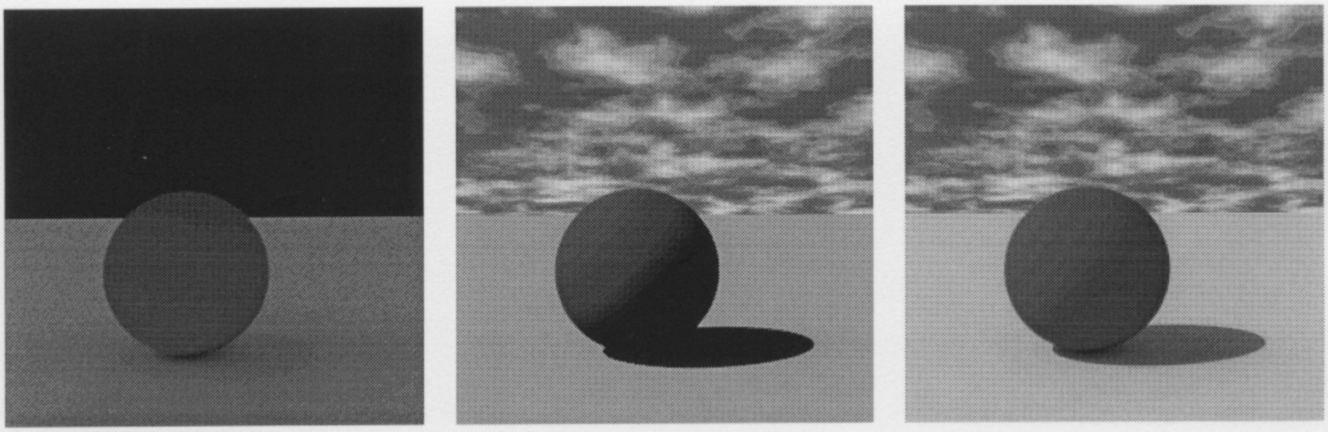
The implementation of these noise reduction filters is very simple. They are standard tools in most image processing programs and source code is easy to find. Furthermore a fast implementation of the median filter can be found in [Paet90] and a fast implementation of convolution in [Wolb94].

4 Results and Discussion

The path tracing algorithm has been implemented in a program called `MIRO` written in C++. The program has been tested under MS-DOS and UNIX. The test-results has been produced using a parallelized version of `MIRO` distributed on 31 Silicon Graphics Indigo workstations. All the images produced have the resolution 256x256 and each pixel is represented using Wards real pixels [Ward91].

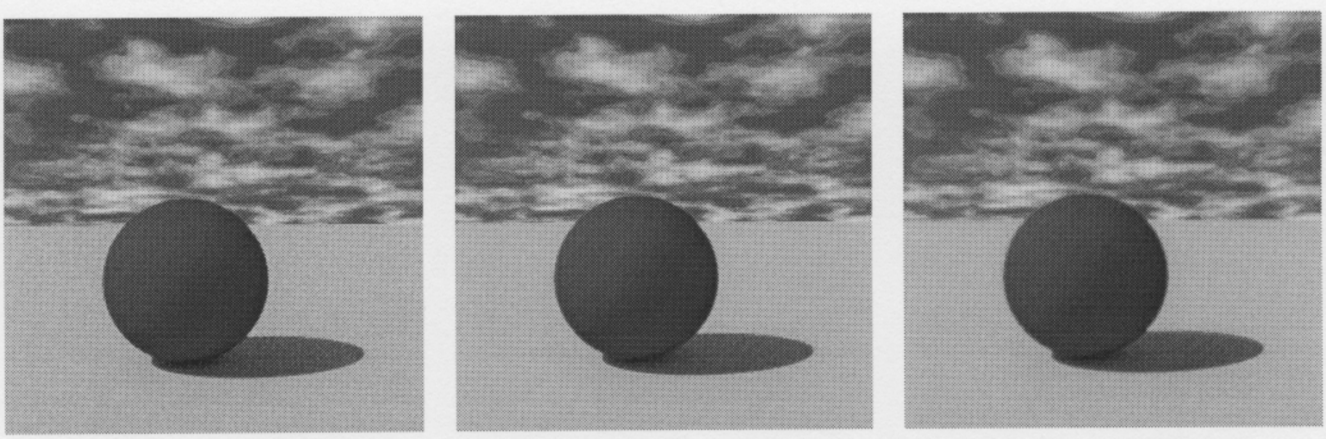
Two test scenes has been constructed to test the filtering techniques presented. The first scene contains a red sphere on a white plane both illuminated by a sun and a cloudy sky. The illumination from the cloudy sky is considered diffuse and it gives the scene a lot of indirect light.

We have calculated a reference image of the first test scene using 1000 rays pr. pixel. The reference image is shown in figure 2(c). In figure 2(a) only the $L_{r,d,d}$ part of the reference image has been shown. In this case $L_{r,d,d}$ is all indirect illumination in the model and we



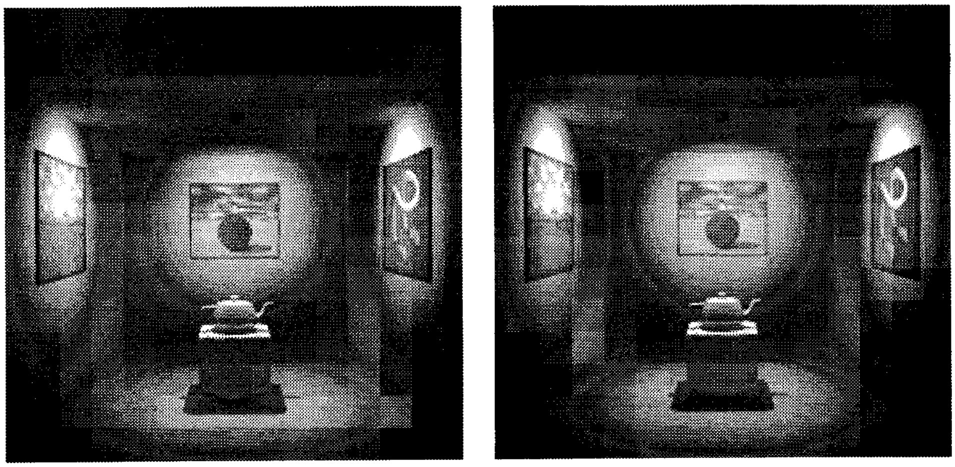
(a) (b) (c)

Figure 2: The reference image (c), the $L_{r,d,d}$ part of it (a) and the remaining light (b). (a)+(b) = (c).



(a) (b) (c)

Figure 3: Test scene 1 sampled with 50 rays pr. pixel. (a) unprocessed version. (b) filtering applied to $L_{r,d,d}$. (c) filtering applied to the completed image.



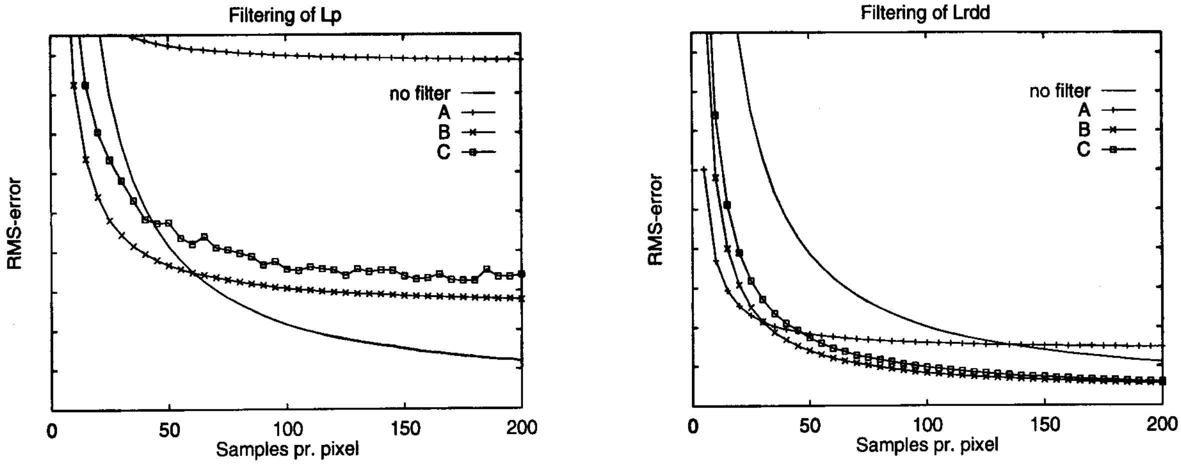
(a) (b)

Figure 4: Test scene 2 sampled with 512 rays pr. pixel.
(a) unprocessed version. (b) filtering applied to $L_{r,d,d}$.

can see that it is only varying slowly. Subtracting fig. 2(a) from fig. 2(c) leaves all the light that is not indirect and the resulting image is shown in fig. 2(b). This image corresponds to the solution produced by simple ray tracing and it can be sampled adequately using only a few rays pr. pixel.

Figure 3(a) shows the resulting image when test scene 1 is sampled with 50 rays pr. pixel. This image contains some very disturbing noise. If, however, we separate $L_{r,d,d}$ from the image and apply noise reduction filter B (discussed later) and then add it to the rest of the solution then we get the image shown in figure 3(b). This image clearly contains less noise than the original image and the edges are not blurred. If we filter the complete image using the same filter then we get the result shown in figure 3(c) and in this image the clouds and the edges are clearly blurred.

Our second test scene is a gallery containing three pictures. In the middle of the room there is a marble pedestal on which a teapot is placed. The floor is textured using a carpet texture which has been created via bump-mapping and modification of the colour (this means that the floor is supposed to look a little noisy). The gallery is illuminated by 4 spotlights, using one spotlight on each image and a spotlight illuminating the teapot. The corners of the gallery and the ceiling is only illuminated by indirect lighting and this makes the calculation of the overall lighting very costly to calculate using pure path tracing. In order to get appropriate estimates of the light in the scene we had to use 512 samples pr. pixel and still we got a very noisy image. The result is shown in figure 4(a). By filtering the image using noise reduction filter B we get the image shown in figure 4(b). This image clearly has less high frequent noise but it still has a little noise of a lower frequency on the ceiling. This noise makes the ceiling look a little dirty. In this test scene the corners in the image and the spotlight in the ceiling are blurred. The reason is that they are only illuminated by $L_{r,d,d}$ and therefore they are blurred by the filtering. However since the pictures on the walls and the teapot in the middle are much more bright and not subjected to any serious blurring the image looks a lot better than the unprocessed version. It is also important to notice that the noise in the carpet texture is not filtered and thereby blurred.



(a) (b)

Figure 5: RMS-error of path traced image with and without filtering of (a) the complete image or (b) $L_{r,d,d}$ only

Instead the carpet keeps the noise which makes it look like a carpet.

Visual comparison of two images is nice but it is not satisfactory when we want to know which of the two images is the "most correct" one. The most correct image is the image that differs the least from the reference image which have been calculated using a large number of samples so that we can have confidence in the radiance values estimated at each pixel. We measure this difference or error by using the following error metric:

$$\text{RMS-error} = \sum_{\text{all pixels } p} (L_{p.ref} - L_p)^2$$

where $L_{p.ref}$ is the pixel value in reference image and L_p is the pixel value in the test image. The measured value is the square sum of the pixel values in the difference image obtained by subtracting the test image from the reference image. This value is directly related to the variance of our estimate (the test image) and as such it is a good representation of the noise in the image. The value is dependent on the intensity in the image, that is the error grows if we make the image brighter. However, this can be ignored, since we are only interested in examining the relative effect introduced via the noise reduction filters.

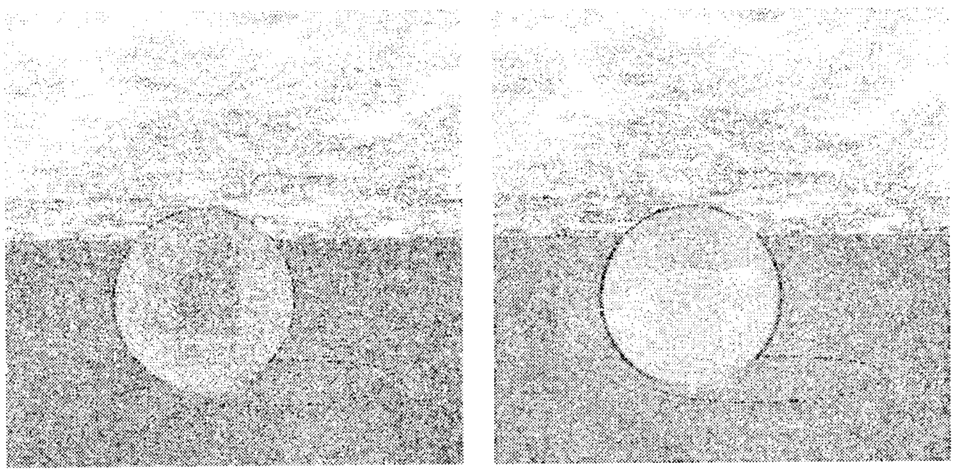
Our test is performed using test scene 1 and we have tested the following 3 filters:

A is a simple low pass filter of size 3x3 as shown in figure 1(a).

B is the low pass filter shown in figure 1(c).

C is a median filter of size 3x3.

In figure 5(a) we have shown the effect of filtering when the filters are applied to the complete image. We can see that we only get an improvement at low sampling rates where the noise is large. Beyond approx. 50 samples filtering makes the image less correct and this is naturally due to blurring. The errors around the edges are rather large and they constitute a lower bound on the error when filtering. It is interesting to notice that median filtering is worse than one of the convolution filters even though this filter blurs the image.



(a) (b)

Figure 6: Bright versions of the difference images using test scene 1. Darker areas indicate larger differences. (a) difference between reference image and image sampled using 50 rays pr. pixel. (b) same as (a) but with filter B applied to $L_{r,d,d}$.

Filtering $L_{r,d,d}$ separately results in the graph shown in figure 5(b). The results have clearly improved compared to the previous graph. We have removed a lot of the blurring error and we are able to reduce the error with more than 100 % even at 200 rays pr. pixel and this is a significant improvement since the convergence speed with path tracing is slow (in general one has to use 4 times as many samples in order to halve the RMS-error).

Having seen that the error is reduced using filtering it could be interesting to know what kind of error is left in the image. We have tried to visualize that by brightening up the difference images corresponding to 50 rays pr. pixel on test scene 1. These bright difference images are shown in figure 6. The image in fig. 6(a) is the difference image between the reference image and the unprocessed version of the test image. In this image the error is spread throughout the image. In fig. 6(b) the $L_{r,d,d}$ part of the test image has been filtered using filter B and as we can see the error is now concentrated around the edges in the image. The error in the rest of the image has been reduced and the number of dark spots (the large differences) has been reduced a lot — the error is more constant and therefore less visible. The errors around the clouds are due to the fact that we only sample these using 50 rays pr. pixel — this difference is, however, not visible in the complete image. In the areas around the edges $L_{r,d,d}$ is mixed from more objects during filtering and this gives rise to the error seen. One might solve the problem by adding information to the image saying which object is visible a each pixel. In this way it is possible to avoid filtering along the edge of an object by just filtering pixels which shows the same object as all their neighbours.

We have only investigated a few filters all of them were 3x3 pixels. Better filters could probably be found. For very noisy images larger filters could be used. Larger filters are better at reducing the noise but they also result in more blurring so it clearly depends on the image. The use of more complex non linear energy preserving filters [Rush94] should also be investigated. One of the arguments in [Rush94] for using more complex filters was to avoid the removal of important details like high lights and caustics. Since we only filter the indirect diffuse illumination we do not risk losing these details. The filters are still interesting since they preserve the energy in the image. The median filter that we use does not have this property. Furthermore more complex filters might help reducing the visible blurring effects that still remain in the areas that are only illuminated by indirect diffuse light.

The noise can also be reduced further by supplementing the filtering method with other variance reduction techniques like [Veac94] and [Lafo94]. Filtering is especially efficient at low sampling rates and this could be very useful in a progressive path tracing algorithm in which intermediate results are visualized. An example of such a method is Chen et al.'s multi pass method [Chen91]. For the irradiance gradient method in [Ward92a] the filtering technique cannot be used since this method does not produce high frequent noise when the sampling is inadequate. Instead of noise the indirect illumination contains artifacts like edges and a wrong lighting level and this kind of error cannot be corrected properly with a noise reduction filter.

5 Conclusion

The use of noise reduction filters on images produced using path tracing can clearly reduce the noise in these images. We have shown that by filtering only the part of the light that is mostly subjected to noise, the light reflected diffusely two times ($L_{r,d,d}$), it is possible to avoid most of the blurring artifacts normally introduced by low pass filtering. As indicated by one of our test scenes we can reduce the RMS-error of the image by filtering only $L_{r,d,d}$ even when the number of samples is high. This is not the case when filtering is applied to the complete image, in this case the RMS-error is increased unless the number of samples is low.

Our experiments showed that a low pass filter, which gives higher weight to the center pixel and the nearest pixel and in this way reduces blurring artifacts, gives the best results at higher sampling rates. The median filter is almost as good whereas a simple low pass filter, that gives equal weight to all pixels, results in to much blurring.

Filtering $L_{r,d,d}$ does however introduce error in the image. This error is mainly due to the slight blurring around the border between to different objects. Along this area the use of low pass filters causes the ambient light to be mixed between two objects. The error is not very visible but it affects the RMS-error of the image.

The filtering technique can easily be combined with other variance reduction techniques in order to make Monte Carlo methods more tractable. It could also be useful in a progressive path tracing scheme in which intermediate results are displayed. These results could be filtered before being displayed in order to improve the image shown.

References

[Arvo86] Arvo, James: "Backward Ray Tracing. Developments in Ray Tracing". *ACM Siggraph Course Notes* **12**, 259-263 (1986).

[Chen91] Chen, Shenchang Eric, Holly E. Rushmeier, Gavin Miller and Douglas Turner, "A Progressive Multi-Pass Method for Global Illumination". *Computer Graphics* **25** (4), 165-174 (1991).

[Gonz92] Gonzales, Rafael C.; Richard E. Woods: *"Digital Image Processing"*. Addison-Wesley Publishing Company, 1992.

[Heck92] Heckbert, Paul S.: "Introduction to Global Illumination". Global Illumination, *SIGGRAPH Course Notes*, **18**, 1992

[Jens94] Jensen, Henrik Wann and Niels Jørgen Christensen: "Photon maps in Bidirectional Monte Carlo Ray Tracing of Complex Objects". To appear in Comp. & Graph.

[Kaji86] Kajiya, James T.: "The Rendering Equation". *Computer Graphics* **20** (4), 143-149 (1986).

[Lafo94] Lafortune, Eric P. and Yves D. Wilems: "The Ambient Term as a Variance Reducing Technique for Monte Carlo Ray Tracing". *Eurographics Workshop on Rendering* **5**, (1994).

[Paet90] Paeth, Alan W.: "Median Finding of a 3x3 Grid". In *Graphics Gems I*, Andrew S. Glassner (ed.), Academic Press, 171-175 (1990).

[Perl85] Perlin, Ken: "An Image Synthesizer". *Computer Graphics* **19** (3), 287-296 (1985).

[Rush94] Rushmeier, Holly E. and Gregory J. Ward: "Energy Preserving Non-Linear Filters". *Computer Graphics* **28** (4), (1986).

[Veac94] Veach, Eric and Leonidas Guibas: "Bidirectional Estimators for Light Transport". *Eurographics Workshop on Rendering* **5**, (1994).

[Ward88] Ward, Gregory J.; Francis M. Rubinstein and Robert D. Clear, A Ray Tracing Solution for Diffuse Interreflection. *Computer Graphics* **22** (4), 85-92 (1988).

[Ward91] Ward, Greg: "Real pixels". In *Graphics Gems II*, James Arvo (ed.), Academic Press, 80-83 (1991).

[Ward92a] Ward, Gregory J. and Paul S. Heckbert: "Irradiance Gradients". *Global Illumination. ACM SIGGRAPH Course Notes* **18** (1992).

[Ward92b] Ward Gregory J.: "The RADIANCE Lighting Simulation System". *Global Illumination. ACM SIGGRAPH Course Notes* **18** (1992).

[Wolb94] Wolberg, George and Henry Massalin: "Fast Convolution with Packed Lookup Tables". In *Graphics Gems IV*, Paul S. Heckbert (ed.), Academic Press, 447-464 (1994).