Vision, Video and Graphics (2005)
E. Trucco, M. Chantler (Editors)

Monte Carlo Noise Reduction Using Bayesian Method in Wavelet domain

Ruifeng Xu and Sumanta N. Pattanaik

University of Central Florida

Abstract

A novel post-processing approach for removing Monte Carlo noises in synthetic images is presented in this paper. This paper first presents our findings on the statistical characteristics of the Monte Carlo noise, and then proposes a Bayesian method to remove this noise. The aim of this approach is to efficiently produce high quality synthetic images using Monte Carlo based rendering at low sampling rates.

Categories and Subject Descriptors (according to ACM CCS): I.4.3 [Image Processing and Computer Vision]: Enhancement; I.3.3 [Computer Graphics]: Picture/Image Generation

1. Introduction

It is still a great challenge to accurately compute the lighting for rendering a synthetic image in an acceptable time period. A lot of work have been carried out in this field, from the early simple basic ray tracing [Whi80], through radiosity [GCT86], radiance [War94], bi-directional path tracing [LW93], to photon mapping [Jen96], metropolis light transport [VG97], and various hybrid solutions [LTG92], together with a variety of rendering acceleration methods, like level of detail [FC93], scene simplification [LE97], and perception based acceleration methods [BM98, RPG99]. All the modern rendering methods are based on solving the radiance or rendering equation for global illumination [Kaj86].

These various methods for global illumination computation can be classified into two categories: object space methods, and image space methods [McC99]. Finite element methods were developed to solve the radiosity problem [GCT86, LTG92] in object space by exploiting the fact that diffuse light transport results in illumination that is relatively coherent and smooth. But this method introduces error by incorrectly smoothing over discontinuities in true solution. On the other hand, most image space methods [Kaj86, LW93, War98] compute the radiance value for each pixel using Monte Carlo methods. These methods are more general in that they can handle variety of surface geometry, variety of surface properties. They are able to identify discontinuities in the scene without much problem. However, they have their own drawback: extremely low convergence speed, especially around illumination discontinuities such as light source edges, penumbrae, fuzzy specula reflections, and caustics spread. According to the noise analysis in [McC99], plenty of samples are needed for each pixel around such image regions to obtain estimates under some threshold error [Pur87, RW94]. This greatly impedes the rendering speed for a high quality synthetic image. As a result, noise free computation of reasonably complex images may take minutes to hours [SWS96, War98]. So much of computation time is generally not affordable in practice. So when using a Monte Carlo based global illumination method for image synthesis, one trades off rendering time for noisy images. The amount of noise in the image varies with an inverse square relationship with the rendering time. [RW94] provides a time measure in terms of the number of samples required to get a given accuracy. The expression for this time measure is given in Equation (1).

$$M_t = \left(4\bar{S}_{p\_16} \cdot \Delta L_{tvis}{}^{-1}\right)^2 \qquad (1)$$

where, M_t denotes the number of estimate samples required to obtain a result with accuracy or error tolerance of ΔL_{tvis}. $\bar{S}_{p\_16}$ is the mean error obtained using 16 samples.

Thus, to increase the accuracy two times, *i.e.* to reduce ΔL_{tvis} by half of its original value, M_t should be increased four times. In conclusion, it is expensive to get high-quality synthetic images simply by increasing sampling rate. In other words, images computed within a reasonable time period using a Monte Carlo radiance computation method, will invariably have noises (more or less).

To remove these noises, researches have devised various methods, which can be generally categorized into two classes: post-processing and importance sampling. The former removes the Monte Carlo noise after rendering, while the latter suppresses the Monte Carlo noise during rendering. The approach proposed in this paper is a post-processing approach, which builds a statistical model of Monte Carlo noise, and removes the noises using Bayesian method based on this model.

The goal for Monte Carlo noise reduction is to suppress as much noise as possible while keeping true information

© The Eurographics Association 2005.

delivered by EUROGRAPHICS DIGITAL LIBRARY www.eg.org diglib.eg.org

undisturbed. The image processing and computer vision community has invented a bunch of noise reduction techniques, like uniform filtering, Wiener filtering, wavelet thresholding [Mal89, Pae90, PS00, WH94]. All these models take advantage of the intrinsic spatial coherence property of natural images. These methods make assumptions about the noise type.

This paper handles Monte Carlo noises in synthetic images by modeling the noise and proposing a framework to remove the noise to create visually pleasing images.

Three keen observations inspire our work. First, most of the noise is in the indirect inter-reflection component [JC95]. Second, if we assume mostly diffuse environment then the diffuse inter-reflection tends to be of low frequency [WRC88]. Third, we find most of the noise tends to be high frequency. In this paper we explore the idea of Monte Carlo noise reduction using Bayesian method. Bayesian denoising is a successful image denoising technique that favorably suppresses the higher frequency information where noise concentrates.

The paper presents this approach in three sections: analysis of Monte Carlo noise, a framework to remove Monte Carlo noise, experimental results.

This paper has two contributions: (1) it proposes a general model of the Monte Carlo noise; (2) it is the first attempt of applying Bayesian method to Monte Carlo noise reduction.

2. Related work

Extensive work has been carried out to reduce noise in images created by Monte Carlo methods. Of the different methods, we will confine our discussion to post processing related methods.

The early work of [LR90] revealed that the Monte Carlo noise (for sensory data) comes from frequencies above sampling frequency being folded into frequencies that can be displayed, and presents itself as noisy spikes. He proposed median and alpha-trimmed mean filters to reduce the noise, which assumes that the noise comes from unwanted secondary input, such as bit errors in transmission, and just throws away those extraneous samples. However, it is not the same case for synthetic images, because the noise in synthetic images comes from insufficient sampling, not secondary input, and the so-called noise carries information about the true value of the image [Pur87, RW94].

Rushmeier et al [RW94] first realized that the Monte Carlo noises in a synthetic image carry meaningful information and proposed an energy preserving non-linear filter to make use of the information implicit in the noise by distributing them to neighboring pixels. However, this method assumes a simple linear tone reproduction operator when deriving the average number of samples to approximate the rendering equation. As we know, High Dynamic Range image is a natural outcome of most renders and non-linear tone reduction operator are mostly used to display the final image on common display devices. This invalidates the assumption, and hence, invalidates the derivation of the formulation for the number of samples (Equation (1)).

Jensen et al. [JC95] presented an alternative perspective to Monte Carlo noise in synthetic image. They assumed that the noise is mainly from diffusely reflected indirect illumination. They separated indirect illumination and removed the noise from this indirect component using various filters, such as low pass filter, mean filter. However, they failed to consider the "outliers" in areas with sharp luminance changes, like light source edges [RW94]. And the filters used blurred the edges.

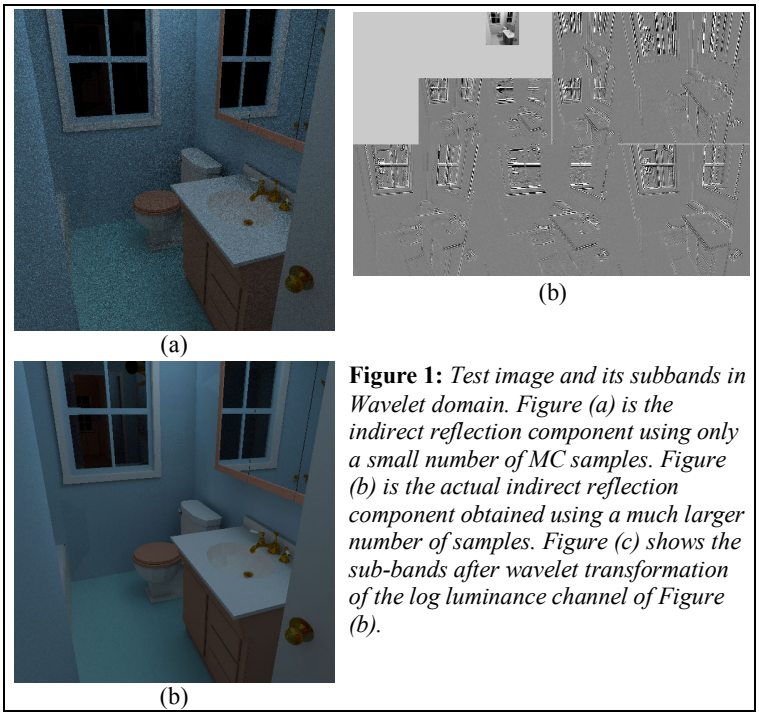


Figure 1: *Test image and its subbands in Wavelet domain. Figure (a) is the indirect reflection component using only a small number of MC samples. Figure (b) is the actual indirect reflection component obtained using a much larger number of samples. Figure (c) shows the sub-bands after wavelet transformation of the log luminance channel of Figure (b).*

Anisotropic diffusion has been successfully applied in Monte Carlo noise reduction to keep edges, and remove noise within regions [McC99]. This method is sensitive to outliers.

The following list summarizes the various facts about Monte Carlo noises from all the work done so far on dealing with Monte Carlo noise.

- The Monte Carlo noise comes from insufficient sampling in high variance regions.
- The noise can be classified into two types: outliers [RW94] and inter-pixel incoherence [JC95, TJ97].
  - Outliers often occur where luminance changes abruptly, like edges.
  - Outliers carry useful luminance information.
  - Inter-pixel incoherence often occurs inside high coherent regions.
- More noises come from computation of indirect diffuse inter-reflection.

Based on the previous work about Monte Carlo noise reduction and on our new observation, a Bayesian noise reduction method is presented in the following sections. Our method makes use of the statistical characteristics of Monte Carlo noise for noise reduction.

3. Monte Carlo noise analysis

Given an image created at a low sampling rate using a Monte Carlo method, there are plenty of visible noises. These noises are generally present in two forms: outliers and inter-pixel incoherence. We try to build a general statistical model to handle both types of Monte Carlo noise. This model addresses two problems: the way noise is combined with true pixel values; and the distribution of the noise.

From our experiments with a number of images generated using Monte Carlo methods, we find that the coefficients of the wavelet band of the Monte Carlo noise map in log domain approximately follow a regular distribution. We use parameterized Laplacian function[1] shown in Equation (2) to model the distribution of these coefficients of the actual image and of noise map in wavelet domain.

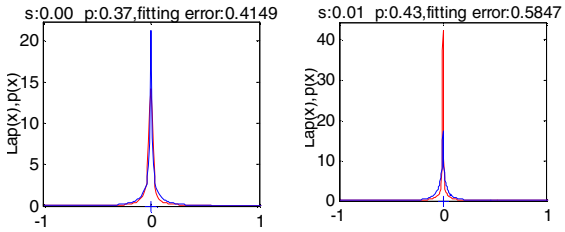
$$f(x;s,p)=\frac{1}{Z}e^{-|x/s|^p},\ -\infty<x<\infty$$
$$Z=2\frac{s}{p}\Gamma(\frac{1}{p}) \qquad (2)$$

where, s, p are parameters of the distributions, and Z is normalization constant. s specifies the heaviness of the noise, and p specifies the shape of the distribution function.

To verify the correctness of our model, we give examples as shown in figures 1 and 2. We use the measure shown in Equation (3) to estimate the fitting error.

$$fitting\quad error=\sqrt{\frac{1}{N}\sum_{n=1}^{N}(F(x)-f(x))^2/f(x)^2} \qquad (3)$$

where, $F(x)$ is the true distribution density of the function at point x, $f(x)$ is the modeled distribution density and N the number of bins used in the discrete summation. The smaller the *fitting error*, the greater the fitting accuracy. The two images Figure 1(a) and Figure 1(b) are rendered using RADIANCE [War04]. In Figure 1(a) indirect reflection component is estimated using 10 samples per bounce, and Figure 1(b) is indirect component estimated using 300 samples per bounce. Figure 1(a) takes less than 5 minutes on a Celeron 2.0G running Window2000, while figure 1(b) takes 1.67 hours on the same platform. Figure 1(b) is used as accurate image. Figure 1(c) is the wavelet subband transformation of the log luminance of Figure 1(b) using steerable filters [SA96, Sim99]. The noise in the image is extracted by dividing Figure 1(a) with Figure 1(b). The logarithm of the noise map is similarly converted into wavelet domain. We compute the high pass band, as well as 4 bands for 2 levels. The coefficient distribution and the fitting to Laplacian function for the high-pass band and one of the other bands are shown in Figures 2(a), 2(b) and Figures 3(a) and 3(b). The blue curves are the fitted Laplacian functions, and the red curves are the actual distribution density. The title line on the figures show the s, p values and the error in fitting the Laplacian function computed using Equation (3).



(a) Distribution of high pass band

[s, p]=[.003, .373]

(b) Distribution of band 1 level 1

[s, p]=[.006, .433]

Figure 2: *The coefficient distribution of two sub-bands vs. fitted Laplacian distribution.*

	s	p	fitting error
High pass band	0.0033	0.3732	0.4149
Level 1, band 1	0.0060	0.4329	0.5847
Level 1, band 2	0.0059	0.4654	0.4269
Level 1, band 3	0.0022	0.3833	0.4673
Level 1, band 4	0.0036	0.4294	0.3991
Level 2, band 1	0.0416	0.5293	0.7867
Level 2, band 2	0.0387	0.5916	0.7241
Level 2, band 3	0.0096	0.4294	0.6356
Level 2, band 4	0.0246	0.5369	0.6610

Table 1: *s, p coefficients of the Laplacian distribution of the various bands in Figure 1(c) and the fitting errors*

The parameters of fitting Laplacian function for the bands in are shown in Table 1 and Table 2.

The results from various experiments show that for most scenes:

- p often lies in the range $[0.5, 2.0]$, and the p values for all bands are generally similar.
- s often lies in the range $[0.0, 1.0]$, and the s values for bands except for the high pass band are generally similar, and is usually one quarter to one half of the value for the high pass band.

Based on these two observations we use two parameters, (s_n, p_n) for all bands except for high pass band, and use $(2s_n, p_n)$ for high pass band. So, only two parameters are used to model the noise, (s_n, p_n). For heavier noise, we use a larger s in $[0.0, 1.0]$, and for more complex scenes we use a smaller p in $[0.5, 2.0]$. We would like to point out that the above selection rules are based solely on our experimental observation.

[1] The Laplacian function has been used in [11] to model the distribution of wavelet coefficients of natural images.

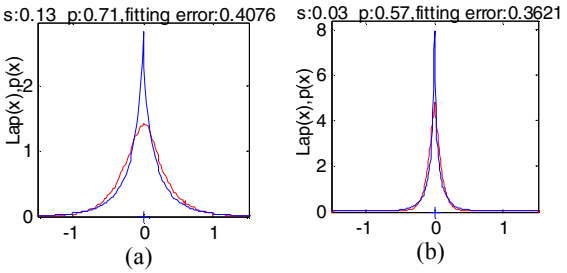


Figure 3: *The coefficients distribution of the wavelet sub-bands of Monte Carlo noise in the sample image shown in Figure 1(a). Figure (a) and (b) are the distribution functions for high pass band, and for bands 1, level 1), respectively.*

	s	p	fitting error
High pass band	0.1345	0.7111	0.4076
Level 1, band 1	0.0288	0.5658	0.3621
Lebel 1, band 2	0.0479	0.7140	0.3381
Level 1, band 3	0.0212	0.5478	0.5338
Level 1, band 4	0.0454	0.6998	0.3537
Level 2, band 1	0.0385	0.5015	02179
Level 2, band 2	0.0369	0.5654	0.2471
Level 2, band 3	0.0114	0.4327	0.6035
Level 2, band 4	0.0370	0.5648	0.3199

Table 2: *s, p coefficients of the Laplacian distribution of the various noise bands.*

4. Bayesian Monte Carlo noise reduction

4.1. Denoising framework

Based on the Monte Carlo noise model given in the previous section, a general Bayesian denoising framework is described in this subsection. The framework has been shown in figure 4. Following [JC95] we assume that most Monte Carlo noise comes from indirect inter-reflection. So, we first separate the rendering result into direct component (direct illumination + specular illumination) and indirect component (diffuse inter-reflection), which is easy to implement by adding a few lines of code into the renderer source code to separately record the indirect and direct components.

Indirect component is denoised, and then combined with direct component to generate the final denoised image. The next subsection presents the Bayesian denoising method, which makes use of Monte Carlo noise model shown in the previous section.

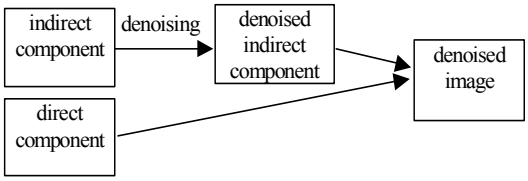


Figure 4: Monte Carlo denoising framework.

4.2. Bayesian denoising

We apply Bayesian denoising in the wavelet domain [SA96, Sim99] to estimate the true image values from the values. The image is first transformed into logarithm domain, and then transformed into wavelet domain. Bayesian method is then applied to remove image noise by adjusting the transformed wavelet coefficients. The method is based on the assumption that the noise is independent of the true value. We model the MC noise to be multiplicative in nature. Hence, the noisy image value can be written as multiplication of true value and noise value.

$$Y = C * N \tag{4}$$

where, Y is noisy image value, C is true image value, and N is noise value.

Wavelet transformation of logarithm of Equation (4) gives us Equation (5).

$$y = c + n \tag{5}$$

In Equation (5), the lower case symbols y,c,n denote the wavelet band coefficients of log$Y$, log$C$ and log$N$, respectively. We model these wavelet band coefficients in wavelet domain using Laplacian distribution. The distribution of wavelet band coefficients, C, is as follows.

$$P_c(c; s_c, p_c) = \frac{1}{Z_c} e^{-|c/s_c|^{p_c}}$$
$$Z_c = 2\frac{s_c}{p_c}\Gamma(\frac{1}{p_c}) \tag{6}$$

where, s_c, p_c are parameters of the distributions, and Z_c is the normalization constant.

The Bayesian estimation of the true sub-band coefficient is then given by Equation (7).

$$\hat{c}(y) = \int p_{c|y}(c \mid y) c dc \tag{7}$$

where, $p_{c|y}(c \mid y)$ the posterior probability, is the probability of the actual coefficient c given the observed coefficient y. Using Bayesian rule, it is possible to express $p_{c|y}(c \mid y)$ in terms of the components that are known in advance. (Refer to Appendix A for a short discussion on Bayesian rule.)

$$\begin{aligned} p_{c|y}(c \mid y) &= \frac{p_{y|c}(y \mid c) p_c(c)}{p_y(y)} \\ &= \frac{p_{y|c}(y \mid c) p_c(c)}{\int p_{y|c}(y \mid c) p_c(c) dc} \\ &= \frac{p_{y-c|c}(y-c \mid c) p_c(c)}{\int p_{y-c|c}(y-c \mid c) p_c(c) dc} \\ &\overset{n=y-c}{=} \frac{p_{n|c}(y-c \mid c) p_c(c)}{\int p_{n|c}(y-c \mid c) p_c(c) dc} \end{aligned} \tag{8}$$

In Equation (8), $p_{y|c}(y \mid c)$ is the posterior probability of y given c. $p_c(c), p_y(y)$ are a prior probability of c and y.

$p_{y-c|c}(y-c\,|\,c)$ is the posterior probability of noise (*y-c)* given *c*. Because the Monte Carlo noise is assumed to be additive in logarithm domain, and independent of the true band coefficient value *c* the posterior probability of noise *n* given *c*, is simply the probability of the noise itself, *i.e.*

$$p_{n|c}(y-c\,|\,c) = p_n(y-c) \tag{9}$$

Equation (8) can now be rewritten as Equation (10) by plugging in Equation (9).

$$p_{c|y}(c\,|\,y) = \frac{p_n(y-c)p_c(c)}{\int p_n(y-c)p_c(c)dc} \tag{10}$$

Using Equation (10), the estimator of true band wavelet coefficient in Equation (7) can be written as Equation (11) [Sim99].

$$\hat{c}(y) = \frac{\int p_n(y-c)p_c(c)cdc}{\int p_n(y-c)p_c(c)dc} \tag{11}$$

$p_n(y-c)$, $p_c(c)$, the distributions of the noise and the true wavelet sub-band coefficients respectively, are both modeled as Laplacian distributions, as shown in Monte Carlo noise analysis in last section and image wavelet sub-band coefficients analysis in this section. Given the values of these distributions, Equation (11) can be calculated using simple discrete integration method.

Both of the distributions are determined by two parameters s and p . For the noise, the parameters (s_n, p_n) are provided as input to the denoising program. And, for the image wavelet sub-band coefficients, the parameters are recovered from second and fourth moments of noise wavelet band coefficients by solving Equation (12). (For details about the derivation of Equation (12), refer to the next subsection). In Equation (12), (s_n, p_n) are parameters of the noise distribution, and (s_c, p_c) are parameters for the accurate sub-band coefficients, σ_y^2, m_y^4 are variance and fourth moments of the noise sub-band coefficients respectively, and Γ is the gamma function. (s_n, p_n) is provided as program parameters. Since there are two unknowns, s_c, p_c in two equations, the parameters are uniquely determined from the equation pair.

$$\begin{aligned} \sigma_y^2 &= \frac{s_c^2\Gamma(3/p_c)}{\Gamma(1/p_c)} + \frac{s_n^2\Gamma(3/p_n)}{\Gamma(1/p_n)} \\ m_y^4 &= \left(\begin{array}{l} \frac{s_c^4\Gamma(5/p_c)}{\Gamma(1/p_c)} + \frac{s_n^2\Gamma(5/p_n)}{\Gamma(1/p_n)} \\ + \frac{6s_c^2s_n^2\Gamma(3/p_c)\Gamma(3/p_n)}{\Gamma(1/p_c)\Gamma(1/p_n)} \end{array} \right) \end{aligned} \tag{12}$$

With the recovered parameters of the wavelet sub-band coefficients distribution, the estimator of accurate sub-band coefficient can be easily computed through Equation (11). The integral is computed using discrete summation of the integrand. Finally, the denoised image is recovered by transforming the denoised wavelet sub-band coefficients into spatial domain.

4.3. Derivation of Second and Fourth Moments of Wavelet Subband Coefficients

Let's write the band coefficient c contaminated by noise as y :

$$y = c + n \tag{13}$$

where, the accurate coefficient c and noise n are assumed to be independent.

Assuming both of the accurate value c and noise n follow Laplacian distributions:

$$\begin{aligned} Lap(x) &= \frac{p}{2s\Gamma(1/p)}\exp(-|x/s|^p), \\ &-\infty < x < \infty \end{aligned} \tag{14}$$

Both the distributions of noise and accurate coefficients are determined by two parameters respectively:

$$(s_n, p_n), (s_c, p_c) \tag{15}$$

The problem is to calculate the second moment and fourth moment of the noise coefficients in light of their independence.

The second and fourth moments for Laplacian distributions are [SA96]:

$$\sigma^2 = \frac{s^2\Gamma(3/p)}{\Gamma(1/p)}, m^4 = \frac{s^4\Gamma(5/p)}{\Gamma(1/p)} \tag{16}$$

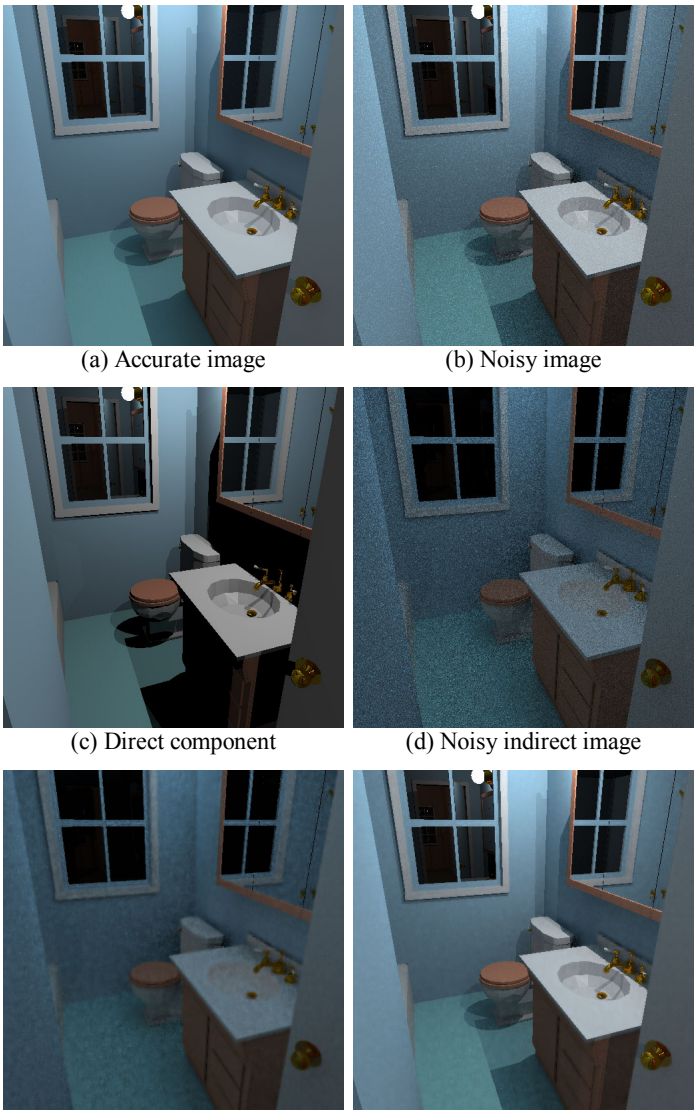
But for the noise coefficients, the second moment can be calculated as:

$$\begin{aligned} \sigma_y^2 &= E[y^2] \\ &= E[(c+n)^2] \\ &= E[c^2 + n^2 + 2cn] \\ &= E[c^2] + E[n^2] + 2E[cn] \\ &= \sigma_c^2 + \sigma_n^2 + 2E[c]E[n] \\ &= \frac{s_c^2\Gamma(3/p_c)}{\Gamma(1/p_c)} + \frac{s_n^2\Gamma(3/p_n)}{\Gamma(1/p_n)} \end{aligned} \tag{17}$$

And, the fourth moment should be calculated as:

$$\begin{aligned} m_y^4 &= E[y^4] \\ &= E[(c+n)^4] \\ &= E\left[\begin{array}{l} c^4 + n^4 + 4c^3n \\ + 4cn^3 + 6c^2n^2 \end{array} \right] \\ &= E[c^4] + E[n^4] + 6E[c^2]E[n^2] \\ &= \frac{s_c^4\Gamma(5/p_c)}{\Gamma(1/p_c)} + \frac{s_n^4\Gamma(5/p_n)}{\Gamma(1/p_n)} \\ &\quad + \frac{6s_c^4s_n^4\Gamma(3/p_c)\Gamma(3/p_n)}{\Gamma(1/p_c)\Gamma(1/p_n)} \end{aligned} \tag{18}$$

So, the parameters for true value can be solved according to Equations (17-18), if the parameters of the noise are known.



(a) Accurate image (b) Noisy image

(c) Direct component (d) Noisy indirect image

(e) Denoised indirect image (f) Denoised image

Figure 5. *Denoising of "cabin".*

5. Experimental results

To show the denoising effects of our approach, two denoised images using our method are shown in this section. We used "RADIANCE" [War04] to create the noisy and accurate images. For the "cabin" image, as shown in Figure 5, computation of the noisy image took 286 secs. The denoising took 37 secs. Thus the total time spend was 323 secs, compared to 3602 secs taken to compute an equivalent image. The denoising was carried out on a 2.0GHz Celeron running Window2000. We implemented our approach using Matlab 6.0.

The noisy cabin image in Figure 5 is composed of direct component (c) and indirect component (d), which are rendered using 10 indirect samples. Accurate image using 300 samples is shown in Figure 5(a), noisy image is shown in figure 6(b) for comparison with denoised results. The denoised indirect component is shown in Figure 5(e), and the final denoised image is shown in Figure 5(f). The noise parameters used in denoising Figure 5(d) are $(s_n = 0.19, p_n = 1.5)$.

Note that the direct component in Figure 5(c) carries little noise.

Figure 6 illustrates another denoising example. And the used parameters are $s_n = 0.6, p_n = 1.5$.

We can see from these experimental results that the edges are well preserved, and the noises are suppressed with little blurring of the image.

The experimental results also verify our findings about Monte Carlo noise. Based on the assumption that most noises gives rise to smaller wavelet coefficients Bayesian denoising method works by suppressing smaller band coefficients, but keeping larger band coefficients. Its successful application to Monte Carlo noise verifies that most Monte Carlo noises are actually concentrated around smaller values. Our Laplacian modeling of Monte Carlo noise also has the greatest density around the smallest values.

6. Conclusions

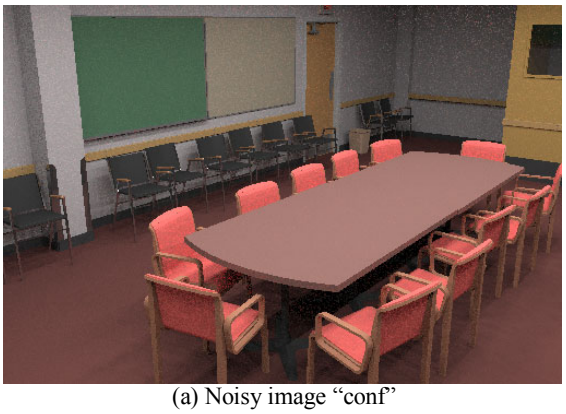
We presented a general framework of the Monte Carlo noise removal. We developed a novel model of Monte Carlo noise. Bayesian method effectively exploits this model for noise reduction. Nice looking images can be synthesized using a combination of low sample rendering and the noise removal technique proposed in this paper. Based on our Monte Carlo noise model, more advanced denoising methods are expected to be developed for better denoising effects.

Compared to the other Monte Carlo denoising method [JC95, McC99, RW94], we take a statistical perspective to the Monte Carlo noise, and introduce a Bayesian method to remove Monte Carlo noise under this perspective. This is the first trial to reduce Monte Carlo noise by modeling its statistical characteristics. Our experimental results prove its feasibility, and we believe more work can be done along this new direction for Monte Carlo noise reduction problem.

So far, we have only applied our technique on the most commonly encountered Monte Carlo noise, that is, Monte Carlo noise generated using a path tracer. We have not tried on other special Monte Carlo noise, like Metropolis noise [VG97], which is presented as streaks.

However, proof of the assumption in our approach remains another problem yet to be studied. We derive our approach by assuming the independence of Monte Carlo noise and its contaminated true value. And our experiments show successful results under this assumption. Because the emphasis of this paper is to present the successful

application of Bayesian method on Monte Carlo noise reduction with a noise statistics model, we leave this problem for later study.



(a) Noisy image "conf"

(b) Denoised image "conf"

Figure 6. *More Denoising Examples.*

Acknowledgement

This work was partially supported by ATI Research and I-4 Corridor Matching Fund.

References

[BM98] Bolin M. R., Meyer G. W.: A Perceptually Based Adaptive Sampling Algorithm. *Proceedings of SIGGRAPH 98*, 299-310, 1998.

[FC93] Funkhouser T. A., Carlo H. S.: Adaptive Display Algorithm for Interactive Frame Rates During Visualization of Complex Virtual Environments. *Proceedings of SIGGRAPH 93*, 247-254, 1993.

[GCT86] Greenberg D., Cohen M., Torrance K.: Radiosity: A Method for Computing Global Illumination. *Visual Computing*. 2(5), 291-297, 1986.

[JC95] Jensen H. W., Christensen N. J.: Optimizing Path Tracing using Noise Reduction Filters. *Proceedings of the WSCG 1995*, 134-142, February, 1995.

[Jen96] Jensen H. W.: Global Illumination using Photon Maps. *Rendering Techniques'96*, Springer-Verlag, 21-30, 1996.

[Kaj86] Kajiya J.: The Rendering Equation. *Proceedings of SIGGRAPH 86*, 143-150, August, 1986.

[LW93] Lafortune E. P., Willems Y. D.: Bi-directional Path Tracing. *Proceedings of 3th International Conference on Computational Graphics and Visualization Techniques (Compugraphics'93)*, 145-153, December, 1993.

[LR90] Lee M. E., Redner R. A.: Filtering: A note on the Use of Nonlinear Filtering in Computer Graphics. *Computer Graphics*, 10(3), 23-29, May/June, 1990.

[LTG92] Lischinski D., Tampieri F., Greenberg D.: Discontinuity Meshing for Accurate Radiosity. *IEEE Computer Graphics and Applications*, 12(6), 25-39, November, 1992.

[LE97] Luebke D., Erikson C.: View-Dependent Simplification of Arbitrary Polygonal Environments. *Proceedings of SIGGRAPH 97*, 199-208,1997.

[Mal89] Mallat S. G.: A Theory for Multiresolution Signal Decomposition: The Wavelet Representation. *IEEE Trans. Pat Anal Mach Intell*, 11, 676-693, July, 1989.

[MS97] Malladi R., Sethian J. A.: *Image Processing: Flows under Min/Max Curvature and Mean Curvature*. LBL, December, 1997.

[McC99] McCOOL M. D.: Anisotropic Diffusion for Monte Carlo Noise Reduction. *ACM Trans. On Graphics*, 18(2), 171-194, April, 1999.

[Pae90] Paeth A.W.: Median Finding of a 3x3 Grid. *Graphics Gems I*, Academic Press, 171-175, 1990.

[PS00] Portilla J., Simoncelli E. P.: Image Denoising Via Adjustment of Wavelet Coefficient Magnitude Correlation. *7th IEEE Int'l Conference on Image Processing*, Vancouver, 10-13, September, 2000.

[Pur87] Purgathofer W.: A Statistical Method for Adaptive Stochastic Sampling. *Computers and Graphics*, 2(2), 157-162, 1987.

[RPG99] Ramasubramanian M., Pattanaik, S. N., Greenberg D. P.: A Perceptually Based Physical Error Metric for Realistic Image Synthesis. *Proceedings of SIGGRAPH'99*, 73-82, Los Angeles, 8-13 August 1999.

[RW94] Rushmeier H., Ward G.: Energy Preserving Non-linear filters. *Proceedings of SIGGRAPH 94*, 131-138, July, 1994.

[SWS96] Shirley P., Wang C., Simmerman K.: Monte Carlo methods for direct lighting calculations. ACM Transactions on Graphics, 15(1), 1-36, January, 1996.

[SA96] Simoncelli E. P., Adelson E. H.: Noise Removal via Bayesian Wavelet Coring. *Proceedings of 3rd IEEE Int'l Conference on Image Processing*, 1, 379-382, Lausanne, Switzerland, September, 1996.

[Sim99] Simoncelli E. P.: Bayesian Denoising of Visual Images in the Wavelet Domain. *Lecture Notes in Statistics*, 141, Springer-Verlag, New York, 1999.

[TJ97] Tamstorf R., Jensen H. W.: Adaptive Sampling and Bias Estimation in Path Tracing. *Rendering Techniques 97*. Eds. Dorsey, J. and Slusallek, P.H., Springer-Verlag, 285-295, 1997.

[VG97] Veach E., Guibas L.: Metropolis Light Transport. *Proceedings of SIGGRAPH'97*, 65-76, July, 1997.

[WRC88] Ward G. J., Rubinstein F. M., Clear R. D.: A Ray Tracing Solution for Diffuse Interreflection. *Computer Graphics*, 22(4), pp85-92, August, 1988.

[War94] Ward G. L.: The RADIANCE Lighting Simulation and Rendering System. *Computer Graphics* (*Proceedings of SIGGRAPH94 conference*), 459-472, July 1994.

[War98] Ward G. L.: Rendering with Radiance: A Practical Tool for Global Illumination. *ACM SIGGRAPH'98 Course #33*, July, 1998.

[Whi80] Whitted T.: An improved Illumination Model for Shaded Display. *Comm. ACM*, 23(6), 1980.

[WH94] Wolberg G., Henry M.: Fast Convolution with Packed Lookup Tables. *Graphics Gems IV*, Academic Press, 447-464, 1994.

[War04] Ward G. L.: *RADIANCE*. http://radsite.lbl.gov/. 2004.

Appendix A: Bayesian rule

The Bayesian rule states the posterior probability of an event A to another event B is equal to the probability of event A and B divided by the prior probability of event B. It is described as Equation (A.1).

$$P(A \mid B) = \frac{P(AB)}{P(B)} \tag{A.1}$$

In other word, the joint probability of events A and B is equal to the posterior probability of A given B times the prior probability of the event B, and it is also equal to the posterior probability of B given A times the prior probability of event A. And it can be written as Equations (A.2-3).

$$P(AB) = P(A \mid B)P(B) \tag{A.2}$$

$$P(AB) = P(B \mid A)P(A) \tag{A.3}$$

Then, the Bayesian rule can be finally written as Equation (A.4), which is derived from Equations (A.3-4).

$$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)} \tag{A.4}$$

So, the Bayesian rule converts the calculation of posterior probability of event A given event B as the calculations of the posterior probability of event B given event A and prior probabilities of event A and B, which are available in many cases.

© The Eurographics Association 2005.