



Progressive path tracing with bilateral-filtering-based denoising

Qiwei Xing¹ · Chunyi Chen¹ · Zhihua Li¹

Received: 22 August 2019 / Revised: 12 August 2020 / Accepted: 18 August 2020 /

Published online: 8 September 2020

© Springer Science+Business Media, LLC, part of Springer Nature 2020

Abstract

Path tracing can generate realistic images based on virtual 3D scene models, but the images are prone to be noisy. To solve this problem, we developed a novel denoising algorithm framework. Firstly, according to the relative mean square error of the noisy pixels, we introduced a progressive adaptive sampling strategy to optimize the distribution of samples. Next, to enhance the quality of the final reconstructed images, we designed an improved bilateral filtering algorithm with use of the gradient feature to obtain the noise-free images. Experimental results demonstrate that our framework outperforms the state-of-the-art path tracing denoising methods in terms of the visual quality, numerical error, and time cost.

Keywords Path tracing · Image denoising · Progressive adaptive sampling · Gradient feature · Improved bilateral filtering

1 Introduction

Path Tracing(PT) is among the most effective techniques for generating photorealistic images [10, 16]. And in recent years, PT emerged as rendering algorithm of choice for film and visual effects [31]. For example, most of the high-end, film production pipelines now employ PT, and even real-time pipelines are moving towards physically-based rendering as modern games begun to use PT [10, 31]. Unless an excessive number of ray samples is distributed, PT renderings converge slowly and suffer from noise artifacts, i.e., variance at low sampling rates [19]. As a result, there is increasing demand to obtain the images generated by PT faster and with much fewer samples than before.

✉ Chunyi Chen
chenchunyi@hotmail.com

Qiwei Xing
673659600@qq.com

Zhihua Li
814155716@qq.com

¹ School of Computer Science and Technology, Changchun University of Science and Technology, Changchun 130022, China

Of all the variance reduction techniques proposed during recent years, PT denoising [12, 15, 29, 33], in particular, has helped to fuel the recent, rapid adoption of PT. PT denoisers for high-end images [1, 6, 25, 32, 36] have demonstrated impressive results at low sampling rates for their respective applications. Still, their results could substantially improve if they could make their input images better. Besides, they wish their denoising methods could be employed in real-time rendering systems [6, 25, 32]. Furthermore, their methods require a significant number of samples as the input. Then they denoise the input to generate the noise-free results. However, the more samples, the more time the algorithm takes [15, 25, 31, 32]. In addition, their reconstruction process are most based on the offline rendering systems, such as Physically Based Rendering Technique(PBRT) [3]. So, if we could improve the quality of input images and speed up the convergence, we can obtain the same quality at ever lower sampling rates or be able to denoise more complex scenes with the current number of samples.

Based on the above observations, we propose a novel denoising framework to gain a wide variety of PT rendering effects. Specially, we wanted our denoising framework that could:

- (1) Generate the images as the input of reconstruction with lower sampling rate.
- (2) Denoise the results generated by sampling as quickly as possible.
- (3) Preserve as many details as possible while denoising.

To do this, we observe that adaptive sampling and reconstruction based on filtering could be combined to reconstruct smooth results without noise or quite little noise. By gathering samples adaptively, we can use the relative mean square error(ReLMSE) to estimate the noise level and decide the number of samples per pixel according to the estimated noise level. For this, we leverage a simple greedy strategy to obtain an approximate solution to the ReLMSE minimization problem which appears as visually noise [18]. For each pixel we samples suitable number of rays to achieve the optimal distribution of samples. Furthermore, while comparing with other methods, we found that sampling in our strategy rather than typical sampling strategy(importance sampling or multi-importance sampling) can improve the quality of the input images and thereby impacts the quality of the final results.

By solving the reconstruction problem(i.e., filtering), we can improve the efficiency of the filtering algorithm by adding other information in different spaces to determine the optimal filter parameters [4]. So, we can speed up the convergence of filtering. In this paper, we improved the bilateral filter by adding the gradient information, the normal and Experiments show our proposed method provides significant improvements over the state-of the art of techniques for denoising PT renderings.

To summarize, our denoise framework is shown as Fig. 1. Our main work makes following contributions:

- (1) We present a novel denoising framework for PT rendering. Our framework is able to estimate the noise level per pixel accurately. We directly use the ReLMSE which is unbiased [18] instead of any estimator. Besides, when selecting samples number per pixel, we make use of a greedy ReLMSE minimization strategy. This allows our framework could optimize the samples distribution so that our samples number per pixel(spp) is lower than other samples methods.
- (2) In our framework,we introduce the information in gradient domains to speed up the convergence of bilateral filter method so that our framework can be employed in both real-time rendering system and offline rendering system. Our time cost is less than most PT denoising methods.

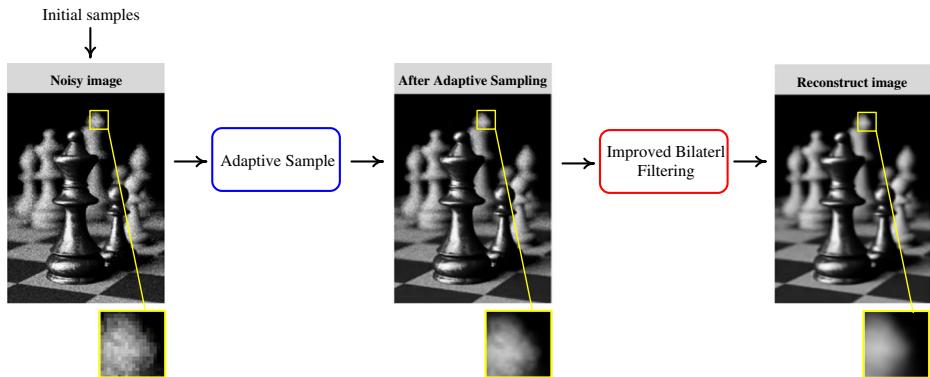


Fig. 1 Overview of our framework. As fig shown, there are two major parts. One is the adaptive sampling. We utilize the greedy strategy for minimizing the RelMSE and distribute new samples per pixel. The other one is that we improved the bilateral filtering by introducing the information in other feature spaces to speed up the convergence of filter process and preserve more details while smoothing the adaptive sampling results. The yellow rectangles indicates the local differences between the initial sample results, the adaptive sampling results, and the reconstruction results

- (3) To predict the better filter parameters, we improve our framework with the neural networks. By comparing with other PT denoising methods, especially those based on neural networks(e.g. NFOR [38], KPCN [1]), our images are with better quality.

2 Related works

Here, we will briefly summarize the most important related work in the following. We refer the reader to Zwicker et al. [38], Sen et al. [34] and Santos et al. [30] for a discussion on PT denoising methods based on adaptive sampling and filtering reconstruction, and to Bako et al. [1] for an introduction to PT denoising methods based on neural networks.

2.1 PT Denoising based on adaptive sampling and filtering

Shortly after the introduction of PT rendering [7], several approaches proposed to reduce the noise by distributing the samples adaptively. The basic idea of these techniques is to estimate the error criterion per pixel. So, several approaches were proposed [5, 9, 11, 14, 17, 20, 21, 24, 35] which used a specific metric(e.g., contrast or variance et.al.). However, these methods suffered from two main drawbacks. First, they typically require the initial image to be sampled with a reasonably large number of samples, and then, produce unsatisfactory results in our framework with extremely low sample counts. Second, these approaches removed the noise just by adaptive sampling stage, and thus, is no optimal. To address the two disadvantages, several approaches have been proposed to perform joint adaptive sampling and reconstruction [22, 23, 27, 28]. Several algorithms estimate the reconstruction error by computing the bias and variance of a Gaussian [27], non-local means [28], local linear regression [22], and polynomial [23] filters and use the error to guide the sample stage. The main drawback of these approaches is that they are not able to reliably estimate the reconstruction error on images with low sampling rates. Finally, Stein unbiased risk estimator(SURE) and was used as no-reference error estimation metric [18]. Then Bauszat

et al. [3] proposed to reconstruct a dense error map from a set of sparse, but robust estimates. Our sampling strategy is inspired by these two approaches for estimating noise level accurately with low sampling rates.

Recently, interest in general PT filtering algorithms has enjoyed a significant revival when Sen and Darabi [12, 33] demonstrated that general, post-process image-space filters could effectively distinguish between noisy scene detail and PT noise. Then the approaches using filtering to generate high-quality images were widely used [4, 22, 29]. Rousselle et al. [29] extended the study of Li et al. [18] to apply the SURE metric to choose the best three candidate filters. Moon et al. [22] estimated the error for discrete sets of filter parameters using a weighted local regression. Bauszat et al. [3] posed the filter parameters selection problem as an optimization and solved it with graph cuts. Bitterli et al. [4] used a collaborative denoising denoising scheme, NL-Means regression weights and feature prefiltering to create an improved first-order regression technique.

2.2 PT Denoising based on neural networks

Kalantari et al. [15] used a multilayer perceptron to compute optimal filter parameters for cross-bilateral and NL-means filters from primal color and auxiliary feature buffers. The reconstruction method whose results is closest to ours, are the Kernel-Predicting Convolutional Network(KPCN) proposed by Bako et al [1]. They used a convolutional neural network to predict a separate smoothing filter kernel per pixel while we predict some filter weight parameters for each pixel. Vogels et al. [36] extended Bako's works to animations and build a modular system for supporting multiple samplers with the same denoising network. Chaitanya et al. [6] designed a feature-based convolutional neural network which uses the multi-resolution U-Net network structure as its base, enhanced with recurrent connections for use in animation. These methods are tailored to extremely low sample counts just because of the absence of necessary sample distribution optimal process.

3 Denoising framework

Given a fixed of a small number of samples n (e.g., 8 spp or less), our goal is to generate an image with better quality and we hope our results could be as close to the ground truth image rendered with many samples as possible. As shown in Fig. 1, we achieve this goal with an adaptive sampling and the improved bilateral filtering. In the following sections, we'll describe our sampling strategy and reconstruction method based on the bilateral filtering.

3.1 Adaptive sampling based on RelMSE minimization

Our adaptive sampling process uses a simple greedy strategy to solve the RelMSE minimization problem, as illustrated in Fig. 2.

3.1.1 Progressive RelMSE minimization

In theory we could estimate bias and variance for each iteration directly to minimize RelMSE. To guess the true bias from noisy input is a challenge for us because we do not have access to ground truth data for the pixel with additional samples. Hence, we avoid explicit bias estimation. Directly estimating the variance is easier than estimating the bias, because it does not require knowledge of the true pixel value. And a key observation is that,

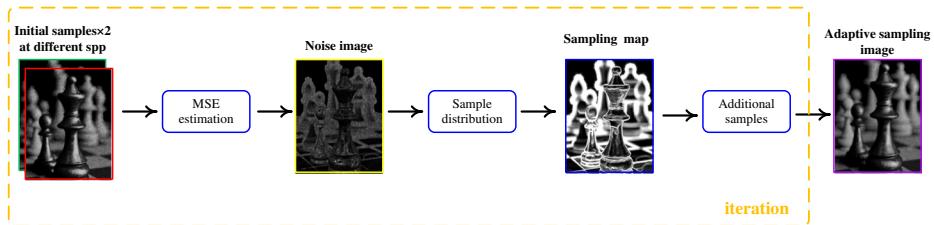


Fig. 2 Overview of our adaptive sampling framework. We iterate over four steps: initial samples, RelMSE estimation, deciding on samples number to greedily reduce RelMSE and distribution of new samples to generate new image used in the next iteration

for most pixels, the samples bias increases a little and the variance decreases a lot monotonically as the samples number goes from less to more [25]. Assuming monotonicity, we find the suitable samples number per pixel with minimum RelMSE simply by increasing the samples number per pixel steadily from less to more. At each pixel, we compute the change of RelMSE at different spp and represent it as $\Delta\text{RelMSE}|f \rightarrow c|$. f is the fine spp and c is the coarse spp. The difference between the RelMSE at different spp term $\Delta\text{RelMSE}|f \rightarrow c|$ can be compute as follows [27]:

$$\begin{aligned}\Delta\text{RelMSE}|f \rightarrow c| &= \text{RelMSE}|c| - \text{RelMSE}|f| \\ &= \text{Bias}|c|^2 - \text{Bias}|f|^2 + \text{Var}|c| - \text{Var}|f|.\end{aligned}\quad (1)$$

There are two terms including *bias* and *variance*. The crucial benefit of our approach is that we can well approximate the bias term without knowledge of the true bias $\text{Bias}[c]$ and $\text{Bias}[f]$. To estimate the bias accuracy, we referred to the quadratic approximation [27]. There is a straightforward simple relation between the bias term $\text{Bias}[c]$, $\text{Bias}[f]$ and their spp [27]:

$$\text{Bias}[c] \approx \frac{s_c^2}{s_f^2} \text{Bias}[f]. \quad (2)$$

By slight abuse of terminology, we denote the fine spp by s_c , and the coarse spp by s_f . Then we can denote the true value of the image by δ , the pixel value at fine spp by f and the pixel value at coarse spp by c . Next, we could compute the $\text{Bias}[c]$ and the $\text{Bias}[f]$ [27]

$$\begin{cases} \text{Bias}[c] = c - \delta \\ \text{Bias}[f] = f - \delta. \end{cases} \quad (3)$$

We express $\text{Bias}[c]^2 - \text{Bias}[f]^2$ in terms of $c - f$ as [27]

$$\text{Bias}[c]^2 - \text{Bias}[f]^2 \approx \frac{s_c^2 + s_f^2}{s_f^2 - s_c^2} (c - f)^2, \quad (4)$$

where the approximation is exact for quadratic image regions. Using this approximation, we could get the following expression for the change in RelMSE [27],

$$D \approx \frac{s_c^2 + s_f^2}{s_f^2 - s_c^2} + \text{Var}[c] - \text{Var}[f]. \quad (5)$$

We call this our sampling selector D using an approximate bias term. If the sampling selector is positive, we will select the fine spp s_f , otherwise, we will terminate the progressive sampling. Notice that, in each iteration, we use the image newly generated in last

iteration as the fine and the image of fine spp in last iteration as the coarse. And we will iterate the sampling process until D turns to negative.

3.1.2 Samples distribution

The goal of our sample distribution is to place new samples in the image plane, such that the RelMSE given the current per pixel spp selection is reduced as much as possible. Our core idea is to select the m pixels, whose RelMSE can be reduced(the selector D is positive) by distributing n addition samples over the support of their selected spp. When iteratively implementing the sampling strategy proposed in Section 3.1.1, we decide the samples number of the selected m pixels as follows:

$$\begin{cases} \text{if } & 0 < D_i \leq \Phi_1 & \Delta s_i = \alpha_1 \\ \text{else if } & \Phi_1 < D_i \leq \Phi_2 & \Delta s_i = \alpha_2 \\ \text{else if } & \Phi_2 < D_i \leq \Phi_3 & \Delta s_i = \alpha_3 \\ \text{else } & & \Delta s_i = \alpha_4 \end{cases} \quad (6)$$

Here, D_i represents the selector of pixel i , and Δs_i represents the number of increasing samples for pixel i . The choice threshold is accumulated as $\Phi = \{\Phi_1, \Phi_2, \Phi_3, \Phi_4\}$. We define four kinds of spatially varying samples number, which denoted as $\alpha = \{\alpha_1, \alpha_2, \alpha_3, \alpha_4\}$ that goes from less to more. Additionally, to prevent discontinuity(or ringing artifact) between two neighbor pixels, we set our varying of samples number as $\{1, 2, 4, 8\}$ or $\{1, 4, 16, 32\}$. And we usually set the threshold values as $\{5 \times 10^{-3}, 10^{-2}, 2 \times 10^{-2}, 5 \times 10^{-2}\}$. Figure 3 shows the results of our adaptive sampling during different iteration process and the image of the RelMSE change.

3.2 Reconstruction based on improved bilateral filtering

Bilateral filter(BF) has been shown effective for removing noise [26]. However, general BF method is not suitable for removing the PT noise. Hence, the improved BF method is proposed [8, 33], which is named as cross bilateral filter. The core idea is to increase the

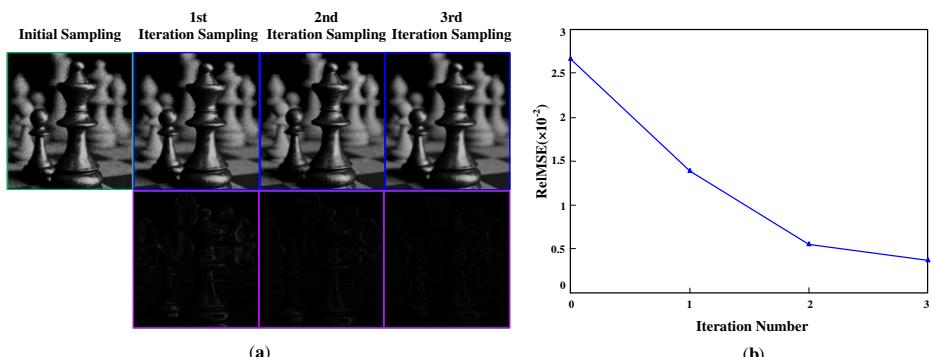


Fig. 3 Illustration of our sampling results during progressive sampling: **a** the sampling results in different iteration and the difference image between the newly generated images and the result in last iteration. **b** The changes of RelMSE(*Chess scene*) in different iterations. Notice that, the resolution of these results are all 512×385 . The average spp is as: 4spp(Initial sampling); 7.9spp(1st Iteration sampling); 9.4spp(2nd Iteration sampling); 10.6spp(3rd Iteration sampling)

auxiliary data based on the color data of pixel. Similar to the previous work, our auxiliary data includes surface normal, depths and color of pixel, which are collected in the adaptive sampling process. Based on these features, we introduce the gradient features to speed up the convergence of improved BF. In this section we describe our reconstruction method based on improved BF. In Section 3.2.1 we formulate the improved BF method. We then describe how we generate the gradient images to speed up the convergence in Section 3.2.2.

3.2.1 Improved BF

According to previous work [33], the improved BF weight $\omega_{i,j}$ between the pixel i and its neighbor j can be defined as

$$\exp\left(-\frac{\|p_i - p_j\|^2}{2\sigma_s^2}\right) \exp\left(-\frac{\|c_i - c_j\|^2}{2\sigma_r^2}\right) \prod_{k=1}^m \exp\left(-\frac{D(\bar{f}_{ik}, \bar{f}_{jk})}{2\sigma_{fk}^2}\right), \quad (7)$$

where \bar{f}_{ik} is the mean of the k -th feature of pixel i ; σ_s , σ_r and σ_{fk} are the standard deviation parameters in the spatial space, color space and feature spaces, respectively. D is a distance function to compute the similarity distance. We define the distance computation function as

$$D(\bar{f}_{ik}, \bar{f}_{jk}) = \sqrt{\frac{\|\bar{f}_{ik} - \bar{f}_{jk}\|^2}{\sigma_{ik}^2 + \sigma_{jk}^2}}, \quad (8)$$

where σ_{ik}^2 and σ_{jk}^2 are variance of the k -th feature of pixel i and j , respectively. Then we can denote the filtered pixel color \hat{c}_i with the weighted color c_j of its all neighboring pixels j :

$$\hat{c}_i = \frac{\sum_{j=1}^n \omega_{ij} c_j}{\sum_{j=1}^n \omega_{ij}} \quad (9)$$

Note that the improved BF kernels are spatially-varying due to the color term and feature term. Hence, it is limited by the low convergence rate just because of the finding optimal filter parameter. To overcome the limitation, we introduce the gradient term to speed up the convergence.

3.2.2 Improved BF with gradient features

Gradient features In theory [19], the texture feature-preserving filtering can use the gradient image, which can be regarded as a special case of weighted linear regression. Compared with other features such as depth or shading normal, the gradient images remains consistent with the ground truth and can extract fine details. So, we can obtain a key observation: if the filter is convergent in gradient space, the improved BF is close to convergency in other spaces.

Here we utilize the Sobel operator to generate the gradient image. The Sobel operator is a discrete differentiation operator, which is widely used in computer vision domains. In this paper, we compute the Sobel gradient as [19]:

$$G(x, y) = \sqrt{(\Delta_x f(x, y))^2 + (\Delta_y f(x, y))^2} \quad (10)$$

where $G(x, y)$ represents the gradient value of the pixel (x, y) . $f(x, y)$ represents the gray value of pixel point (x, y) ; $\Delta_x f(x, y)$ represents the gradient value along the X direction; $\Delta_y f(x, y)$ represents the gradient value along the Y direction. The above two values can be computed by [19]

$$\begin{cases} \Delta_x f(x, y) = -f(x-1, y-1) - 2 \times f(x-1, y) - f(x-1, y+1) \\ \quad + f(x+1, y-1) + 2 \times f(x+1, y) + f(x+1, y+1) \\ \Delta_y f(x, y) = -f(x-1, y-1) + f(x-1, y+1) - 2 \times f(x, y-1) \\ \quad + 2 \times f(x, y+1) - f(x+1, y-1) + f(x+1, y+1) \end{cases} \quad (11)$$

As shown in Fig. 4, taking the Chess scene as an example, after adaptive sampling the noise in the gradient image was less than other feature images while texture details were preserved.

Improved BF with gradient features As mentioned in previous paragraph, finding the optimal parameter in gradient domains is faster than that in other space. We can combine the gradient feature and the BF method to speed up the convergence. As described in Formula(10) and Formula(7), we can modify the improved BF as [26]:

$$\hat{c}_i = \frac{\sum_{j=1}^m \omega_{ij}^{(s)} (\|s_j - s_i\|) \omega_{ij}^{(r)} (\|c_j - c_i\|) \omega_{ij}^{(k)} (\|k_j - k_i\|)}{\sum_{j=1}^m \omega_{ij}^{(s)} \omega_{ij}^{(c)} \prod_{k=1}^m \omega_{ij}^{(k)}} \quad (12)$$

where $\omega_{ij}^{(s)}$ represents the weight in spatial space; $\omega_{ij}^{(c)}$ represents the weight in color space; $\omega_{ij}^{(k)}$ represents the weight in k -th feature space. And by slight abuse of terminology, we denote the s, c, k as the value of the pixel in spatial space, color space and k -th feature space, respectively. It always takes long time to find the optimal $\omega_{ij}^{(s)}$, which mainly decides the quality of reconstruction images. Next, we utilize the gradient feature to speed up this finding process.

According to the mechanism mentioned in previous work [26], the factor in spatial space is close to the factor in gradient domains. Hence, based on this theory, we can approximately

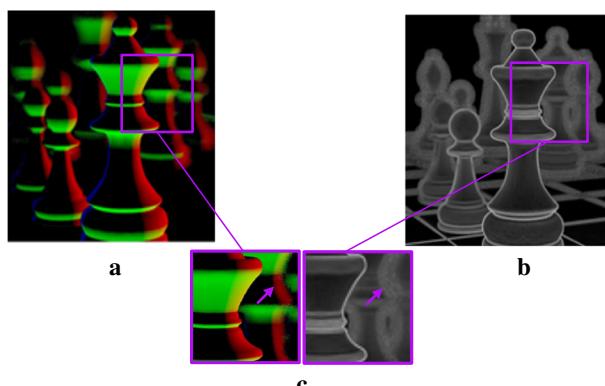


Fig. 4 The comparison between the shading normal and the gradient images after our adaptive sampling. **a** is the shading normal image; **b** is the gradient image; **c** is the detail image of the two, respectively. Notice that, the arrow part denotes the noise. We can find that there is less noise in gradient image

compute the $\omega_{ij}^{(s)}$ by

$$\omega_{ij}^{(s)} \approx \frac{(\|g_j - g_i\|)}{(\|s_j - s_i\|)} \omega_{ij}^{(g)}, \quad (13)$$

where $\omega_{ij}^{(g)}$ represents the weight in gradient domains; g represents the value of the pixel in gradient domains. Then combining the Formula(13) and Formula(7), we can obtain as follows:

$$\omega_{ij}^{(s)} \approx \frac{(\|g_j - g_i\|)}{(\|s_j - s_i\|)} \exp \left(-\frac{\|g_j - g_i\|^2}{2\sigma_s^2} \right). \quad (14)$$

According to Taylor Series Based on Arbitrary Exponential Functions [13], Formula(14) can be rewritten as

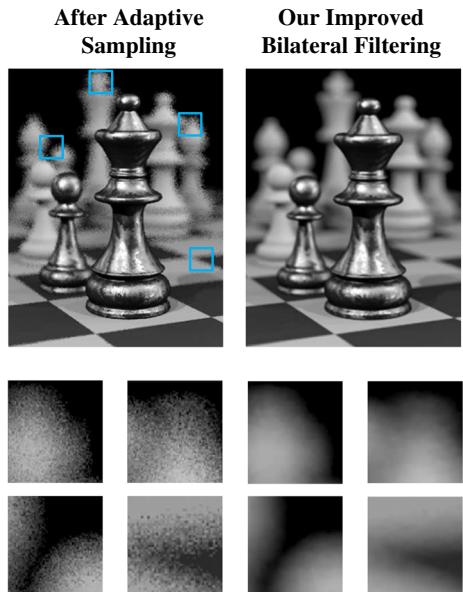
$$\omega_{ij}^{(s)} \approx \frac{(\|g_j - g_i\|)}{(\|s_j - s_i\|)} \exp \left(-\frac{g_j^2 + g_i^2}{2\sigma_s^2} \right) \sum_{n=0}^N \frac{1}{n!} \left(\frac{g_j g_i}{\sigma_s^2} \right)^n. \quad (15)$$

When implementing, we could find the optimal weight $\omega_{ij}^{(s)}$ by adjusting the parameter σ_s . So far, we transform the problem that finding the optimal weight parameter $\omega_{ij}^{(s)}$ into a fast linear convolution operation. Finally, The convergence of the whole method is faster than before. As shown in Fig. 5, we utilized our improved BF to reconstruct the adaptive sampling result. From the details, we can find, after reconstruction, the quality of the image improved a lot.

4 Results and discussion

For all experiments, we used an Intel(R) Core-i5-7400 with 8 GB RAM and a NVIDIA Quadro K5200 GPU. The proposed method was integrated as an extension of PBRT [25] and

Fig. 5 Comparison between the images after our adaptive sampling and after our improved BF



for comparison, we implemented our framework in PBRT, and gradient computing process is written in CUDA to enable GPU acceleration.

4.1 Parameter setting

There are a number of parameters in our framework, both in the sampling process and reconstruction process. As mentioned in previous paragraph, the sampling thresholds $\Phi = \{\Phi_1, \Phi_2, \Phi_3, \Phi_4\}$ were set as $\{5 \times 10^{-3}, 10^{-2}, 2 \times 10^{-2}, 5 \times 10^{-2}\}$; the spatially varying samples number $\alpha = \{\alpha_1, \alpha_2, \alpha_3, \alpha_4\}$ were set as $\{1, 2, 4, 8\}$ or $\{1, 4, 16, 32\}$. The filtering parameter σ_{fk} was set as $\sigma_{fk} = 0.4$ for shading normal, $\sigma_{fk} = 0.125$ for texture color, and $\sigma_{fk} = 0.3$ for depth in all experiments. The σ_r was set as $\sigma_r = 8$ in our current implementation. We used intermediate iteration, whose step size was set as $\{1, \sqrt{2}, 2, 2\sqrt{2}, 4, 4\sqrt{2}\dots\}$, to find the optimal σ_s . Experiments shows this setting strikes a good compromise between performance and quality. In practice, results are not very sensitive to these parameters. Although parameters can be finetuned for each scene, this yields only marginal improvements.

4.2 Results and discussions

To evaluate the performance of our framework, we compared our framework to a variety of state-of-the-art PT denoising techniques. To measure the quality of generated images, the mean relative squared error(MrSE) proposed by Rousselle et al [2]: $MrSe = (\hat{c} - c)^2 / (c^2 + \varepsilon)$ averaged over all pixels, where \hat{c} and c are the reconstructed and ground truth pixel colors, respectively. Here ε is a small number(we set it as 0.001) to prevent division by zero. We also measure the perceptually-based structural similarity index(SSIM) [37] which is a value from 0 to 1, where 1 indicates perfect quality with respect to the ground truth image.

The results of our method were compared with five state-of-the-art methods: SURE-based filter filtering(SBF) [18], non-local means filtering(NLM) [28], random parameter filtering(RPF) [33], robust denoising(RD) [29], and weighted local regression(WLR) [22]. For all algorithms, we used the implementations provided by the authors with their default parameters. Figures 6, 7 and 8 show the comparison results. Although our algorithm is faster than all other methods shown here, we perform as equal sampling number comparisons as possible to ensure fairness. The image of *Sponza scene*, has a resolution of 1600×1200 . The image of *Teapot Room scene* has a resolution of 1200×800 . The resolution of the *Sibenik scene* and *San Miguel Terrace* are 1200×1200 (Fig. 9).

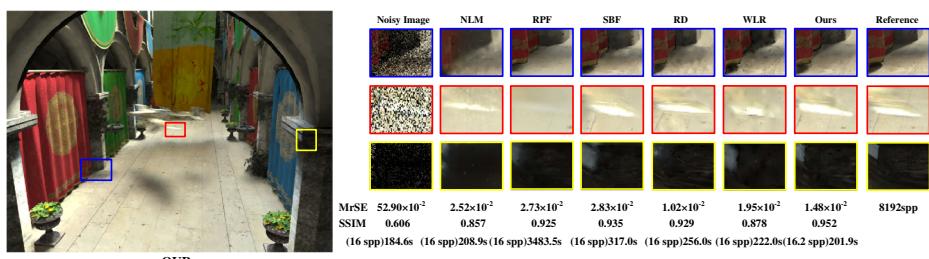


Fig. 6 Comparison result on *Sponza scene*

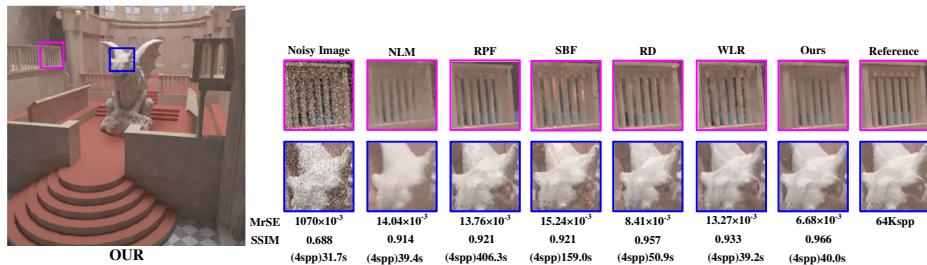


Fig. 7 Comparison result on *Sibenik scene*

First, we examine the path-traced *Sponza scene* with global illumination and motion blur(shown as Fig. 6). NLM used only color information when filtering, so it cannot preserve the geometry and texture details in the scene. Moreover, although the other approaches used additional features, they did not have appropriate filter weights, resulting in either over blurred texture or residual motion blur noise. Our framework could preserve the texture while removing the motion blur noise, resulting in an image with fewer artifacts than the other methods. Although the MrSE of RD was lower than ours, our result is the closest to the reference shown as the SSIM. Note, our framework was the fastest.

The *Sibenik scene* is a path-traced scene that includes global illumination. RPF and NLM removed the noise, but over blurred the geometry of the rail and statue. SBF and WLR do not weight the feature appropriately and thus their results contained residual noise in the texture regions. Although both our framework and RD could handle the texture regions well, RD produced visible artifacts in the smooth regions due to the low sampling rate. Besides, our framework was faster than RD and our result is closer to the reference.

San Miguel Terrace scene is a path-traced scene with severe noise at 8spp. Again, NLM over blurred the textures on the wall. Moreover, SBF and WLR did not weight the features appropriately, and, thus, produced significant splotches on the walls. Moreover, SBF and WLR tended to over blurred images in regions where the variances were exceedingly high. Despite having a slightly higher MrSE than RPF and RD, we produced a relatively noise-free result that is better than the other methods both visually and in terms of SSIM.

Finally, the *Teapot Room scene* is a challenging, path-traced scene containing one glossy and two diffuse teapots with mostly indirect illumination. Unfortunately, none of other methods could effectively remove the strong indirect illumination noise on the back wall without introducing artifacts or over blurring the glossy reflections on the body and spout of the teapot. While we produced a relatively noise-free result.

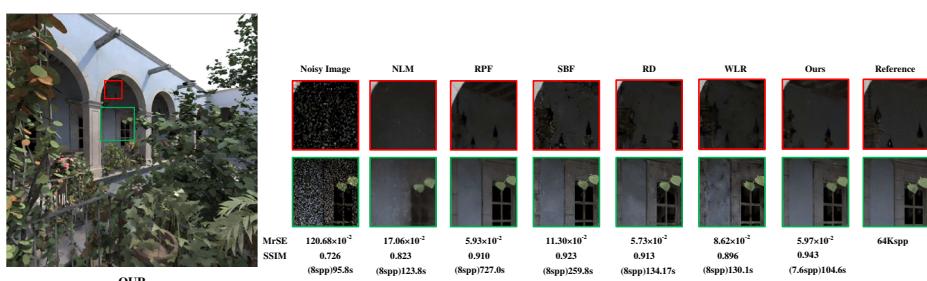


Fig. 8 Comparison result on *San Miguel Terrace scene*

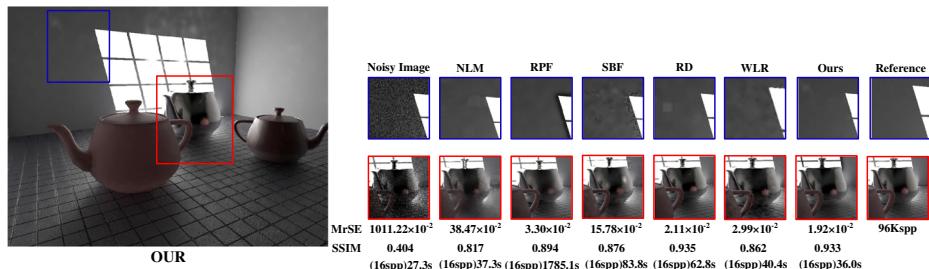


Fig. 9 Comparison result on *Teapot Room scene*

To test our convergence as the sampling rate increases, we provide convergence plots in Fig. 10 using MrSE and SSIM metrics for the *Sponza scene* and provide comparisons against low-discrepancy samples, NLM, SBF, RD, and WLR. Because the time cost of RPF is too expensive, we do not compare with RPF. As shown in Figure, our MrSE consistently decreases even at the sampling rate that the optimal σ_s has not been found, demonstrating the filter's ability to reconstruct the images. Moreover, the perceptual quality of our framework is higher than others based on SSIM index, particularly at lower sampling rates. Our improved BF is faster than others because of the gradient features. Thus, as the sampling rate increases, the variances are reduced and our improve BF could be convergent faster.

Several methods could introduce the gradient feature to improve the quality of reconstruction images, such as the SBF [18]. We also compare our results and the SBF with gradient feature. In Fig. 11, we compare our result with the improved SBF. Improved SBF could remove spike noise and texture noise than before. However, improved SBF tended to be aggressive in its filtering, resulting in a loss of some local details (indicated by green arrow). But our results did not have this problem.

4.3 Extensions and limitations

Extensions As presented in previous paragraph, our framework could generate the noise-free images with high quality, and it takes less time than other methods. There is another

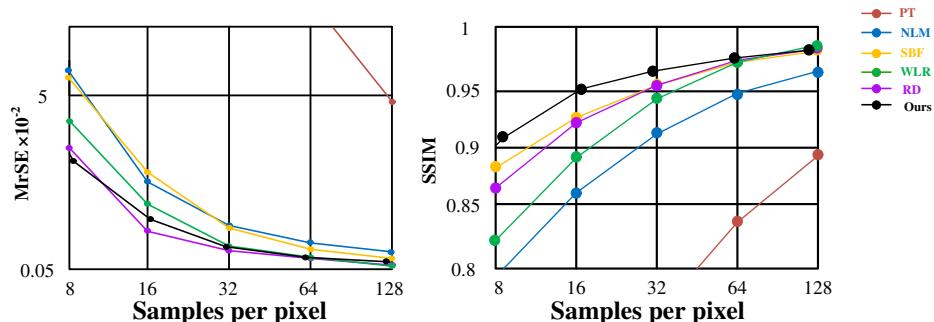


Fig. 10 On the left, we show the convergence plot in terms of MrSE for the *Sponza scene* with our framework and a few state-of-the-art methods. Our framework produces results with higher perceptual quality, particularly at low sampling rates. Note that although WLR have strong MrSE, the perceptual quality of their results are lower than SBF, RD, and our framework because of visible artifacts. Besides, our framework is the fastest, so approximately equal number of samples is not equal time

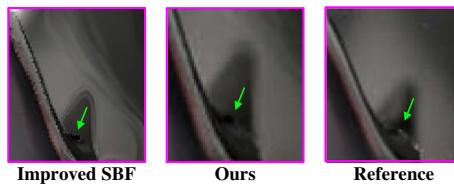


Fig. 11 We show the comparison of the improved SBF(with gradient feature) and our framework on the *Teapot Room scene* shown in Fig. 9. Even with gradient feature, the improved SBF is able to capture the glossy reflections on the teapot. However, the result of our framework has fewer artifacts

advantage of our framework. Our framework is suitable for the PT denoising method based on neural network. Because of the absence of adaptive sampling, most denoising method could not perform well at low sampling rate. Our framework could solve this problem and with the neural network, we can obtain more optimal filter parameters. Inspired by Bako [1], we modified our framework with a deep CNN(shown in Fig. 12). And we compare our results with several popular PT denoising methods, for example, the NFOR [4] and KPCN [1]. The results can be seen in Fig. 13.

Limitations A main limitation of our method is that the estimator may be not efficient at recognizing fine structures. As a result, features could be over blurred, or there is aliasing on the edge of textures. As shown in Fig. 14, results using our method were visually better in regions containing less texture details, but over blurred the edge of textures(indicated by blue arrows and green arrows in Fig). The impact of this problem may be reduced by increasing samples, but this can result in more expensive time cost of the whole framework. Furthermore, we could not find the best feature parameters as a weighted average of the candidates by iterations. And the pixels with different visual effects might have different parameters, such as the σ_{fk} , σ_r and σ_s . To solve this problem, we plan to modify our noise estimator to improve the adaptive sampling and utilize the state-of-the-art techniques of neural network to find better feature parameters for the pixels with different effects in future work.

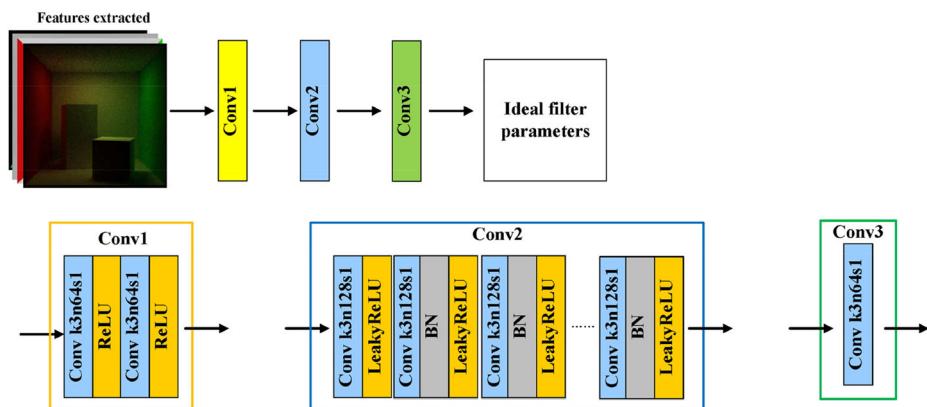


Fig. 12 Structure of the utilized CNN, which has also been used in our previous work. There are three different parts in our CNN: Conv1, Conv 2 and Conv3. Interpretation of network layer annotations: e.g., k3n64s1 indicates that kernel size is 3, number of feature channels is 64 and stride is 1. There are totally 64 Conv k3n128s1 networks in Conv2

Noisy Images	Ours without CNN	NFOR	KPCN	Ours with CNN	Reference
MrSE 0.613	0.025	0.0064	0.0032	0.0031	8Kspp
SSIM 0.695	0.874	0.925	0.940	0.942	
8spp	7.6spp	8spp	8spp	7.6spp	

Fig. 13 Comparison with the methods based on neural network. Results indicate that under the help of CNN, our method produced smooth images while returned more details than before. And with CNN, the generated image of our framework was better quality than the widely-used denoising methods based on neural network

5 Conclusions

We propose a novel path tracing denoising framework based on progressive adaptive sampling and improved bilateral filtering. First, our framework estimate the noise level per pixel with the RelMSE to guide the adaptive sampling. Next, in the pixels of high noise level,

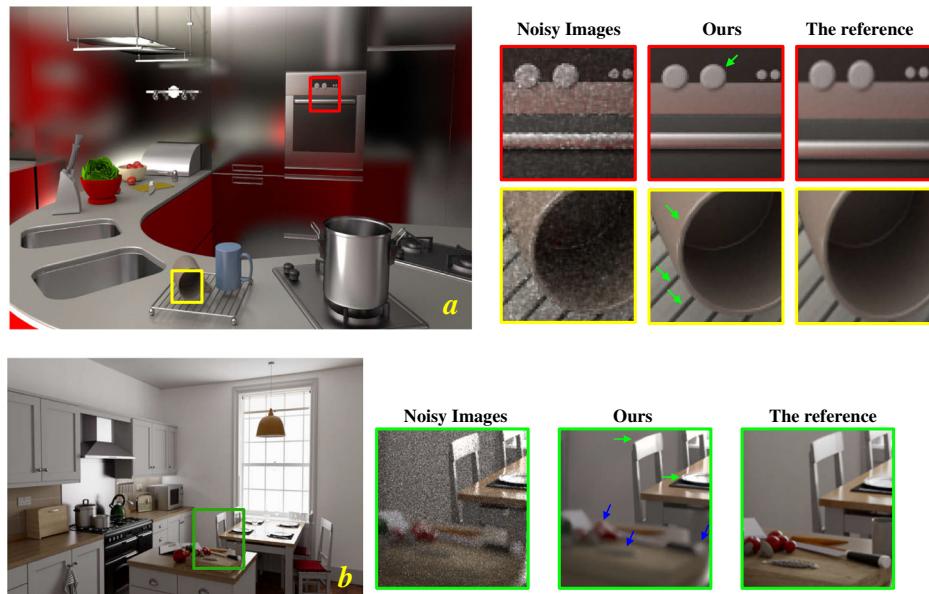


Fig. 14 Failure cases of our framework. The two images about kitchen scenes were generated at approximately 8 spp. Our framework removed significant noise in the motion-blurred regions and the regions with global illumination effects. However, at low sampling rate, there is aliasing(indicated with green arrow) and the image texture details are over blurred(indicated with blue arrow)

we distribute more samples. On the contrary, we distribute less samples in the pixels of low noise level. Then the results of adaptive sampling and their features are used as the input of an improved bilateral filtering. When filtering, we introduce the gradient feature to speed up the convergence of filtering. Finally, after the above two steps, we can obtain the noise-free images with high quality. Experimental results demonstrate the robustness and efficiency of our framework on a set of challenging test scenes and show that our framework outperforms the state-of-the-art path tracing denoising methods in terms of the visual quality, numerical error, and time cost.

Acknowledgements This work was supported partially by the National Natural Science Foundation of China under Grant U19A2063, partially by the Jilin Provincial Science & Technology Development Program of China under Grants 20170101005JC, 20180519012JH and 20190302113GX, partially by the 13th Five-year Science & Technology Research Program of Jilin Provincial Department of Education under Grant JJKH20200792KJ, and partially by Jilin provincial industrial innovation funds under Grant 2016C091.

References

1. Bako S, Vogels T, Mcwilliams B, Meyer M, Novak J, Harvill A, Sen P, Derose T, Rousselle F (2017) Kernel-predicting convolutional networks for denoising Monte Carlo renderings. ACM Transactions on Graphics(TOG) 36(4):14. Article 97
2. Bako S, Meyer M, Derose T, Sen P (2019) Offline deep importance sampling for Monte Carlo path tracing. Computer Graphics Forum 38(7):527–542
3. Bauszat P, Eisemann M, Eisemann E, Magnor M (2015) General and Robust Error Estimation and Reconstruction for Monte Carlo Rendering. Computer Graphics Forum 34(2):597–608
4. Bitterli B, Rousselle F, Moon B, Iglesiasguitian JA, Adler D, Mitchell K, Jarosz W, Novak J (2016) Nonlinearly weighted first-order regression for denoising Monte Carlo renderings. Computer Graphics Forum 35(4):107–117
5. Bolin MR, Meyer GW (1998) A perceptually based adaptive sampling algorithm. In: Proceedings of the 25th annual conference on computer graphics and interactive techniques (SIGGRAPH '98). Association for Computing Machinery, p 299–309
6. Chaitanya CRA, Kaplanyan AS, Schied C, Salvi M, Lefohn A, Nowrouzezahrai D, Aila T (2017) Interactive reconstruction of Monte Carlo image sequences using a recurrent denoising autoencoder. ACM Transactions on Graphics(TOG) 36(4):12. Article 98
7. Cook RL, Porter TK, Carpenter L (1988) Distributed ray tracing. In: Proceedings of the 11th annual conference on Computer graphics and interactive techniques (SIGGRAPH '84). Association for Computing Machinery, p 137–145
8. Dammertz H, Sewitz D, Hanika J, Lensch HP (2010) Edge-avoiding Δ -Trous wavelet transform for fast global illumination filtering. In: Proceedings of the Conference on High Performance Graphics (HPG '10). Eurographics Association, p 67–75
9. Egan K, Tseng Y, Holzschuch N, Durand F, Ramamoorthi R (2009) Frequency analysis and sheared reconstruction for rendering motion blur. In: ACM SIGGRAPH 2009 papers (SIGGRAPH '09). Association for Computing Machinery, Article 93, pp 13
10. Fascione L, Hanika J, Fajardo M, Christensen P, Burley B, Brian G (2017) Path tracing in production - Part 1: Production Renderers. In: ACM SIGGRAPH 2017 Courses (SIGGRAPH '17). Association for Computing Machinery, Article 13, pp 1–39
11. Hachisuka T, Jarosz W, Weistroffer RP, Dale K, Humphreys G, Zwicker M, Jensen HW (2008) Multidimensional adaptive sampling and reconstruction for ray tracing. In: ACM SIGGRAPH 2008 papers (SIGGRAPH '08). Association for Computing Machinery, Article 33, pp 10
12. Han S, Lee K (2017) Implementation of random parameter filtering using OpenMP. In: 2017 4th International Conference on Computer Applications and Information Processing Technology (CAIPT), p 1–4
13. Howard RM (2019) Dual Taylor series, spline based function and integral approximation and applications. Mathematical & Computational Applications 24(2):35
14. Kalantari NK, Sen P (2013) Removing the Noise in Monte Carlo Rendering with General Image Denoising Algorithms. Computer Graphics Forum 32(2):93–102

15. Kalantari NK, Bako S, Sen P (2015) A machine learning approach for filtering Monte Carlo noise. ACM Transactions on Graphics(TOG) 34(4):12. Article 122
16. Keller A, Foscione L, Fajardo M, Georgiev I, Christensen P, Hanika J, Eisenacher C, Nichols G (2015) The path tracing revolution in the movie industry. In: ACM SIGGRAPH 2015 Courses (SIGGRAPH '15). Association for Computing Machinery, Article 24, pp 1–7
17. Lee ME, Render RA, Uselton SP (1985) Statistically optimized sampling for distributed ray tracing. In: Proceedings of the 12th annual conference on Computer graphics and interactive techniques (SIGGRAPH '85). Association for Computing Machinery, p 61–68
18. Li T, Wu Y, Chuang Y (2012) SURE-based optimization for adaptive sampling and reconstruction. ACM Transactions on Graphics(TOG) 31(6):9. Article 194
19. Liu Y, Zheng C, Zheng Q, Yuan H (2018) Removing Monte Carlo noise using a Sobel operator and a guided image filter. The Visual Computer 34(4):589–601
20. Mitchell DP (1987) Generating antialiased images at low sampling densities. In: Proceedings of the 14th annual conference on Computer graphics and interactive techniques (SIGGRAPH '87). Association for Computing Machinery, p 65–72
21. Mitchell DP (1991) Spectrally optimal sampling for distribution ray tracing. In: Proceedings of the 18th annual conference on Computer graphics and interactive techniques (SIGGRAPH '91). Association for Computing Machinery, p 157–164
22. Moon B, Carr NA, Yoon S (2014) Adaptive rendering based on weighted local regression. In: ACM SIGGRAPH 2014 Talks (SIGGRAPH '14). Association for Computing Machinery, Article 67, pp 14
23. Moon B, Medonagh S, Mitchell K, Gross M (2016) Adaptive polynomial rendering. ACM Transactions on Graphics(TOG) 35(4):10. Article 40
24. Overbeck RS, Donner C, Ramamoorthi R (2009) Adaptive wavelet rendering. In: ACM SIGGRAPH Asia 2009 papers (SIGGRAPH Asia '09). Association for Computing Machinery, Article 140, pp 12
25. Pharr M, Humphreys G (2016) Physically based rendering: From theory to implementation 3rd, 1–1167. Morgan Kaufmann:USA
26. Paris S, Durand F (2009) A fast approximation of the bilateral filter using a signal processing approach. Int J Comput Vis 81(1):24–52
27. Rousselle F, Knous C, Zwicker M (2011) Adaptive sampling and reconstruction using greedy error minimization. In Proceedings of the 2011 SIGGRAPH Asia Conference (SA '11). Association for Computing Machinery, Article 159, pp 12
28. Rousselle F, Knous C, Zwicker M (2012) Adaptive rendering with non-local means filtering. ACM Transactions on Graphics(TOG) 31(6):11. Article 195
29. Rousselle F, Manzi M, Zwicker M (2013) Robust denoising using feature and color information. Computer Graphics Forum 32(7):121–130
30. Santos JD, Sen P, Oliveira MM (2018) A framework for developing and benchmarking sampling and denoising algorithms for Monte Carlo rendering. The Visual Computer 34(6):765–778
31. Schied C, Kaplanyan A, Wyman C, Patney A, Chaitanya CRA, Burgess J, Liu S, Dachsbacher C, Lefohn A, Salvi M (2017) Spatiotemporal variance-guided filtering: real-time reconstruction for path-traced global illumination. In: Proceedings of High Performance Graphics (HPG '17), Association for Computing Machinery, Article 2, pp 1–12
32. Schied C, Peters C, Dachsbaecher C (2018) Gradient estimation for real-time adaptive temporal filtering. In: Proceedings of the ACM on Computer Graphics and Interactive Techniques, Article 24 pp 16
33. Sen P, Darabi S (2012) On filtering the noise from the random parameters in Monte Carlo rendering. ACM Transactions on Graphics(TOG) Article 18:15
34. Sen P, Zwicker M, Rousselle F, Yoon S, Kalantari NK (2015) Denoising your Monte Carlo renders: recent advances in image-space adaptive sampling and reconstruction. In: ACM SIGGRAPH 2015 Courses (SIGGRAPH '15). Association for Computing Machinery, Article 11 pp 255
35. Soler C, Subr K, Durand F, Holzschuch N, Sillion FX (2009) Fourier depth of field. ACM Transactions on Graphics(TOG) 28(2):12. Article 18
36. Vogels T, Rousselle F, Mcwilliams B, Rothlin G, Harvill A, Adler D, Meyer M, Novak J (2018) Denoising with Kernel prediction and asymmetric loss functions. ACM Transactions on Graphics(TOG) 37(4):15. Article 124
37. Wang Z, Bovik AC, Sheikh HR, Simoncelli EP (2004) Image quality assessment: From error visibility to structural similarity. IEEE Transactions on Image Processing 13(4):600–612
38. Zwicker M, Jarosz W, Lehtinen J, Moon B, Ramamoorthi R, Rousselle F, Sen P, Soler C, Yoon S (2015) Recent advances in adaptive sampling and reconstruction for Monte Carlo rendering. Computer Graphics Forum 34(2):667–681

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Terms and Conditions

Springer Nature journal content, brought to you courtesy of Springer Nature Customer Service Center GmbH (“Springer Nature”).

Springer Nature supports a reasonable amount of sharing of research papers by authors, subscribers and authorised users (“Users”), for small-scale personal, non-commercial use provided that all copyright, trade and service marks and other proprietary notices are maintained. By accessing, sharing, receiving or otherwise using the Springer Nature journal content you agree to these terms of use (“Terms”). For these purposes, Springer Nature considers academic use (by researchers and students) to be non-commercial.

These Terms are supplementary and will apply in addition to any applicable website terms and conditions, a relevant site licence or a personal subscription. These Terms will prevail over any conflict or ambiguity with regards to the relevant terms, a site licence or a personal subscription (to the extent of the conflict or ambiguity only). For Creative Commons-licensed articles, the terms of the Creative Commons license used will apply.

We collect and use personal data to provide access to the Springer Nature journal content. We may also use these personal data internally within ResearchGate and Springer Nature and as agreed share it, in an anonymised way, for purposes of tracking, analysis and reporting. We will not otherwise disclose your personal data outside the ResearchGate or the Springer Nature group of companies unless we have your permission as detailed in the Privacy Policy.

While Users may use the Springer Nature journal content for small scale, personal non-commercial use, it is important to note that Users may not:

1. use such content for the purpose of providing other users with access on a regular or large scale basis or as a means to circumvent access control;
2. use such content where to do so would be considered a criminal or statutory offence in any jurisdiction, or gives rise to civil liability, or is otherwise unlawful;
3. falsely or misleadingly imply or suggest endorsement, approval , sponsorship, or association unless explicitly agreed to by Springer Nature in writing;
4. use bots or other automated methods to access the content or redirect messages
5. override any security feature or exclusionary protocol; or
6. share the content in order to create substitute for Springer Nature products or services or a systematic database of Springer Nature journal content.

In line with the restriction against commercial use, Springer Nature does not permit the creation of a product or service that creates revenue, royalties, rent or income from our content or its inclusion as part of a paid for service or for other commercial gain. Springer Nature journal content cannot be used for inter-library loans and librarians may not upload Springer Nature journal content on a large scale into their, or any other, institutional repository.

These terms of use are reviewed regularly and may be amended at any time. Springer Nature is not obligated to publish any information or content on this website and may remove it or features or functionality at our sole discretion, at any time with or without notice. Springer Nature may revoke this licence to you at any time and remove access to any copies of the Springer Nature journal content which have been saved.

To the fullest extent permitted by law, Springer Nature makes no warranties, representations or guarantees to Users, either express or implied with respect to the Springer nature journal content and all parties disclaim and waive any implied warranties or warranties imposed by law, including merchantability or fitness for any particular purpose.

Please note that these rights do not automatically extend to content, data or other material published by Springer Nature that may be licensed from third parties.

If you would like to use or distribute our Springer Nature journal content to a wider audience or on a regular basis or in any other manner not expressly permitted by these Terms, please contact Springer Nature at

onlineservice@springernature.com