Novel accelerated Stochastic Progressive Photon Mapping rendering with neural network

Qiwei Xing¹, Chunyi Chen ¹*, Zhihua Li¹

¹Changchun University of Science and Technology, Changchun City 130022, China
E-mail: 673659600@qq.com; chenchunyi@cust.edu.cn; 814155716@qq.com

Abstract: Recently, deep learning-based approaches have led to dramatic improvements for Monte Carlo rendering at the low sampling rate. Most of these approaches are aimed at path tracing. However, they are not suitable for photon mapping. In this paper, we develop a novel accelerate stochastic progressive photon mapping approaches with neural network. First, our framework utilizes the particle-based rendering and focuses on photon density estimation. We train a neural network to predict a kernel function to aggregate photon contributions at shading point. Then we construct a estimation images with the prediction network. During experiments, we could find that there are spike pixels and noises in estimation images sometimes. So we present the improved denoising network to post-process the estimation images. Finally, we can obtain the high-quality reconstructions of complex global illumination effects like caustics with an order of magnitude fewer photons compared with previous photon mapping methods. Besides, our denoising network can reduce most multi-scale noises on both low-frequency and high-frequency areas while preserving more illumination details, especially caustics, compared with other state-of-the-art learning-based denoising methods.

1. Introduction
Stochastic progressive photon mapping (SPPM) has been widely used in rendering images with global illumination effect in computer graphics. Compared with the common photo mapping (PM) and progressive photon mapping(PPM), this method could generate images with caustics and specular-diffuse-specular lighting effects with less memories and time. Unfortunately, this method is a biased but consistent rendering approach. In other words, SPPM always suffers from both variance and bias with necessary iterations or inappropriate scenario configurations. And it always takes a lot of time to obtain a high-quality and noise-free images. How to reduce the variance and bias of the SPPM is the main problem for this research. According to the Ref[6], the bias of renderings generated by SPPM usually has large low-frequency noises on diffuse surfaces and the variance has small high-frequency noises on glossy surfaces. So, this found can help us produce the noise-free images.

In recent years, deep learning methods have been utilized in denoising unbiased path tracing, but they are not efficient for biased renderings because of the randomly of the photon collection. In this paper, we present a novel accelerate rendering method designed for biased SPPM renderings.

Firstly, we propose the deep learning-based algorithm for particle-based rendering that could generate efficient, high-quality images with a small number of photons. We mainly focus on the photon density estimation problem, which is the key component of all particle-based methods and present a novel convolutional neural network to estimate accurate photon density at any surface points of the scene, just given sparsely distributed photons.

Secondly, according to Ref[6], to avoid conflicting rendering constraints with sparse distributed...
photons and keep more sharp caustics, we separate radiance of final results into two components: caustic part and global part. Then we utilize a neural network to reconstruct these images separately.

Finally, we combine these reconstruction results and get the high-quality images at the end.

Our framework is shown in Fig.1.

Figure 1. Overview of our accelerated SPPM method. Given a set of photons of any shading point with the standard SPPM algorithm, we input these photon’s properties to the Feature Extractor network that compute per-photon features. The original per-photon features and the deep context are concatenated and processed by a kernel prediction network to predict a kernel weight. Next, these predicted kernel weights are used to compute the photon contributions and then obtain the reflected radiance. Sometimes, our results obtained within above process are noisy or with spike pixels. So, we present a denoising framework to post-process the estimation images. Given the noisy estimation images, we separate the caustic and global parts and then feed them to the denoising network respectively. After denoising, we can get the two separate reconstructed outputs and post-processed to get the final denoised image.

Utilizing the above methods, we could generate high-quality images using less photons. On the contrary, various of path tracing and photon mapping improved approaches fail to do so; even when combined with advanced progressive and adaptive techniques, SPPM and APPM are limited by the necessary samples to achieve comparable results.

2. Related Work

Monte Carlo path integration. Kajiya[11] firstly introduced the rendering equation and Monte Carlo(MC) path tracing. Then various approaches for MC path integration have been presented, including light tracing[12], bidirectional path tracing(BDPT)[13-14], and Metropolis light transport(MLT)[15-17]. These approaches could simulate complex light transport with accurate global illumination effect in an unbiased way. Unfortunately, MC methods requires too many samples, especially for low probability paths, such as the classical caustic or specular-diffuse-specular (SDS) paths. Our method is to improve SPPM technique, which is efficient for caustics and SDS, and our goal is to achieve sparse reconstruction.

Photon density estimation. Previous work investigated progressive methods to overcome the shortcomings of memory bottleneck and enable arbitrarily large photon numbers[13,23,18-19], bidirectional methods to improve rendering glossy objects[20], adaptive methods to optimize photon tracing[21], and the combination of unbiased MC methods and photon mapping[22-24]. A lot of related works have been introduced to improve the kernel density estimation with standard statistics for adaptive kernel bandwidth[25-27] or anisotropic kernel shapes[28]. Other methods utilized ray differentials[20], blue noise distribution[30-32], and Gaussian mixture filtering[33] to improve the reconstruction. In contrast, we make use of the common kernel density estimation with a novel module based on neural network and keep the rest unchanged in the standard SPPM framework.

Monte Carlo denoising. There is little progress in sparse reconstruction with low sample counts in photon mapping while many methods have been proposed to achieve MC rendering with low sample counts. A recent survey of sparse sampling and reconstruction is developed by Zwicker et al.[34]. MC denoising methods can be categorized into two classes: methods with pre-process which rely on prior theoretical knowledge[35-38] and methods based on post-process which reduce the noise in rendered images with few assumptions about the image signal[11,39-40].

In recent years, techniques with deep learning framework have been introduced to denoising the
MC images\cite{7,41}. Kalantari et al.\cite{11} first presented the MC denoising methods with neural network. They utilized a multilayer perceptron neural network to learn the relationship between the noisy scene data and the ideal filter parameters and made use of learned model for new scenes for a wide range of distributed effects. Bako et al.\cite{7} introduced a novel denoising network which is to estimate the local weighting kernels with CNN(convolutional neural network) and with predicted results, they computed each pixel to be denoised from its neighbors. Here we set the Bako’s methods as KPCN. From their results, it could see that KPCN has shown great improvement over the prior MC denoising algorithms. Later, Vogels et al.\cite{8} improved the KPCN, who combined KPCN and several task-specific modules and optimized the assembly with an asymmetric loss. Although they built upon a multi-scale architecture to address residual low-frequency noise, their method could not directly solve the single-frame denoiser problem. Wong and Wong\cite{9,10} used residual learning to generate high-quality path tracing renderings. Their model, which is called RDP by ours, directly predicted the corresponding noise-free color instead of per-pixel kernel weights. Xu et al.\cite{42} also proposed the denoising network to predict the final color based on the GAN(generative adversarial network). They also adapted a well-designed conditioned auxiliary feature buffers. However, their approaches are not suitable to denoise SPPM just because SPPM is a pixel-based method rather than sample-based. In contrast, our network separately considers individual scene-space photon samples around every shading point and predicts a kernel to gather per-photon contributions.

3. Accelerated SPPM Framework

3.1. Learning to estimate photon density

First, we present our photon density estimation approach with neural network. Our method focuses on density estimation only. Based on the standard SPPM, we upgrade the traditional, distance-based with a novel, local-context-aware kernel function, which is based on the learned neural network. Particularly, when given a visible shading point, we consider its $K$ nearest neighbor photons with the bandwidth $r$ which is adaptively selected. The properties of these individual photons are utilized as the input of our network, including the photon directions $\{p_i\}_{i=1}^{K}$, photon positions $\{d_i\}_{i=1}^{K}$, and photon contributions $\{\phi_i\}_{i=1}^{K}$. Based on the origin equation, we make use of our network to regress per-photon kernel weights and compute radiance estimates with the follow modified one\cite{1}:

$$L(p_i) = \frac{\rho}{N_\pi r^2} \sum_{j=1}^{K} \Theta_{ij} \{p_i,\{p\},\{d\},\{\phi\}\}$$

(1)

where $\Theta_{ij}$ is our predicted kernel weight for photon $i$. Here we utilize all photons in a local neighborhood for per-photon kernel prediction and we obtain the photon statistics and associates per-photon information to compute kernels for photon aggregation.

3.1.1. Pre-processing for input

According to Ref[43], photon distributions are highly different between the visible shading points and scenes. So, it is a challenge to design a general network which could generalize different inputs. Besides, the effect of deep neural network is closely related to input data. The more normalized the input data, the more accurate the results of neural network. So, we pre-process the input photon properties for better performance of our network.

Because light intensity could have high dynamic range(HDR), the photon contribution can vary widely in range. We use a mapping function to pre-process the photon contributions\cite{43},

$$M_u(a) = \frac{\log(u+a) - \log(a)}{\log(u+a) - \log(a) + 1}$$

(2)

where $a=0.001$ is an additional parameter. Here $M_u(u)$ maps HDR values $u$ from $[0, \infty]$ to $[0,1]$. Further, we linearly map the values from $[0,1]$ to $[-1,1]$ and use them as the input of our network. As for photon positions $x$ and directions $d$, we transform them into the local coordinate of the visible shading point, which is constructed with the information (shading normal, depth, texture, world
position) of the shading point and two orthogonal directions that are randomly selected in the tangent plane.

Our learned kernel bandwidth $r$ is determined by the distance of the $K$th nearest photon. This leads to a big range of bandwidth values given various photon distributions $\tau_i$, which is highly challenging for our neural network to process. Motivated by the bandwidth normalization used in traditional kernels\cite{44-45}, we divide the photon positions in the local coordinates by the bandwidth $r$, and scale the final density estimates by $1/r^2$, which is shown in Equation(1).

After all above processing, different terms of our network input are all normalized into the range of [-1,1], which makes our network could correlate and leverage different photon properties from various domains in the efficient way.

3.1.2. Our network architecture

As described above, our inputs are essentially a set of multi-feature 3D points in a unit sphere. In the points set, there is no meaningful inherent point ordering, and the number of points(K) is not fixed. Inspired by Ref\cite{46}, we leverage the network shown in Fig.2, which accept an arbitrary number of inputs and are invariant to permutations of inputs.

As shown in Fig. 2, our network contains two major sub-networks: the feature extractor and the kernel predictor.

They process every photon individually. The feature extractor processes each photon which utilizes the photon properties as input and extracts meaningful features with MLPs (multi-layer perceptron). In particular, when designing the feature extractor part, we use three FNs (full-connected layers). For each layer, there follows the activation layers, whose activation function is the $ReLU$\cite{46}. The feature extractor is mainly utilized to transform the input into a learned 32-channel feature vector with linear and non-linear operations. Then we aggregate these per-photon feature by the max pooling and average pooling operators whose output is the deep photon context vector. The vector represents the local photon statistics in a non-linearly transformed space. Next, the kernel predictor makes use of the across-photon context and the per-photon features to predict a single scalar, which is, in other words, the kernel weight. Finally, these kernel weights are used to linearly combine the original photon contributions (shown in Equation (1)). When we designed the kernel predictor network, we utilized three-layer FNs with $ReLU$ activations functions.
3.2. Post-denoising with deep residual network

3.2.1. Pre-processing for input and the final reconstruction

After our photon density estimation, we could reconstruct accurate photon density from sparse photons. However, there exists a small amount of noise in the reconstruction images, especially when input photon is fewer. Hence, inspired by Ref[6] and Ref[10], we make use of a denoising framework. Fig. 3 shows the overview of our denoising framework.

![Denoising Network Diagram](image)

Figure 3. Our network is based on the ResNet architecture[9]. We use 16 ResBlocks (basic ResNet building blocks). Experiment results showed that our choice (shown in Fig. 4) performs best in terms of both training efficiency and quality of output. There are a total of 35 convolutional layers in our network.

Denoising the images rendered with PM is different from the methods for MC. MC denoising methods always use a weighted reconstruction, which leads to conflicting denoising constraints. That is to say, when a group of photons contribute to a surface, we could hardly decide whether to average out and blur out the contributions or stay sharp[6]. Therefore, we separate the rendering result into two parts: caustic and global, which represent our radiance estimation result of two types of photons: caustic photons and global photons. When photons go through a delta BSDF and reach a diffuse surface or reach a glossy surface at the maximum depth without any diffuse bounces, we called these photons as caustic photons, and the other photons are called global photons.

As said above, the bias and the variance bring us multi-scale noises into one rendering of SPPM. To solve it, we proposed an image-space denoising method by modeling and estimating a filtering function \( F \). It is used to compute the denoised result \( \hat{c}_i \) of the pixel \( i \). \( \hat{c}_i \) is evaluated as the weighted sum of the pixel result \( c_j \) in its neighborhood \( N(i) \) centered at the pixel \( i \)[6]:

\[
\hat{c}_i = \sum_{j \in N(i)} F(X_j, \theta_j)c_j
\]

where \( X = \{ x_j : j \in N(i) \} \) is the inputs based on the selected auxiliary features and \( \theta_j \) represents the filter specific parameters of the filtering function \( F(\cdot) \) defined by the method. Thus, we can express our denoising framework as follows[6]:

\[
\hat{c}_g = \sum_{j \in N(i)} F(X_j, \theta_g)c_j, \quad \hat{c}_c = \sum_{j \in N(i)} F(X_j, \theta_c)c_j
\]

where \( \hat{c}_g \) and \( \hat{c}_c \) represents the denoised results of global part and caustic part respectively. As both caustic and global parts have HDR values, whose optimization process is highly unstable, we apply a logarithmic transform as the pre-processing step to each part[6]:

\[
\hat{c} = \log(1 + c).
\]

After denoised separately, we utilize exponential transformation to each output and then recombine them[6]:

\[
\hat{c} = \exp(\hat{c}_g) + \exp(\hat{c}_c) - 2
\]
3.2.2. **Our network architecture**

We proposed an improved deep residual learning (ResNet)(shown in Fig.4) based on the network presented in Ref[10]. This type of network has shown good performance on several inverse problems[47-49]. The depth of a ResNet can reach over one hundred convolutional layers with continued improvement in performance. The residual learning capability makes ResNet an ideal candidate for the denoising task in a supervised learning setting as it can focus on learning and mapping the differences between the noisy input and the corresponding ground truth at different scales. That is to say, it is suitable for denoising the SPPM.

![ResNet Diagram](image)

Figure 4. Inside our ResBlock, there are 2 convolutional layers, each having 128 filters of size 3×3. Each convolutional layer is followed by a batch normalization layer to form a sub-unit, and a parametric rectifier[10] is a sandwiched between these two sub-units.

As shown in Fig.4, we modify the network architecture which is different from the original ResNET proposed in Ref[50] by He et al. We add one parametric rectifier[47] as an activation unit to improve the performance of our network. We also include a network wide skip connection[47,51-52] for added mapping flexibility.

3.2.3. **Auxiliary Features**

Besides the features from previous work (e.g., shading normal, depth, albedo), we utilize other auxiliary feature to better reduce noises while preserving more illumination details:

- **Distance**: the distance from the visible intersection point to the camera;
- **Photon Density**: the local gathered photon count;
- **Photon Flux**: the accumulated flu;
- **Photon Gradient**: the energy-direction distribution of the photons over a region.

4. **Experiments Setup**

In this section, we would describe our network implementation and training details.

4.1. **Dataset**

A major difficulty of training network is that it always requires a sufficiently large and representative dataset to learn the complex relationship between its inputs and outputs while avoiding overfitting[60].

There is no public dataset which is used for our network so far. Hence, we designed our own reasonably large dataset by taking 30 scenes curated by Bitterli et al[53], covering simple scenes to complex scenes with indirect illumination effect, and caustic effect. A few examples of these scenes are shown in Fig. 5.
4.2. Implementation and Training

We implemented our networks in TensorFlow\[54-55\] with a learning rate of $2 \times 10^{-4}$, and setting the gradient clip threshold as 1.0.

When implementing the photon density estimator network, we apply the standard SPPM method to generate photon densities for every point with a total number of 1 billion photon paths. For each scene, we store 10 million photon paths and a 256×256 multi-channel image that contains the ground truth radiances and other information (positions, shading normal, albedo, tracing depth and BRDFs). Besides, we randomly selected K from 100 to 800 and use from 0.3 million to 4 million photons to train our network. We utilized a batch size of 2000 random shading points.

As for our denoising network, we trained the caustic and global parts respectively with a batch size of 128. Inspired by Bako et al.\[7\], we kept the number of parameters reasonably low for speeding up both training and inference. Each network only has about 250000 trainable parameters. We initialized them with Xavier initialization\[56\].

5. Results and Analysis

5.1. Comparison with PT algorithm

| Scene   | PT(64spp) PSNR | SSIM | PT(1024spp) PSNR | SSIM |
|---------|----------------|------|------------------|------|
| Red Wine | 36.37 | 0.9606 | 24.82 | 0.7455 |
| Water Pool | 31.47 | 0.9623 | 17.86 | 0.5512 |
| Dragon | 33.80 | 0.9026 | 31.80 | 0.8167 |

Figure 6. We show our final results (at the left of above images). We compare against pure path tracing with 64spp and 1024spp on different three scenes. Obviously, our results are with best quality and we could draw the same conclusion from the statistics under the images.
We compare our method with pure PT algorithm (with the same spp(sampling rate per pixel) and with the large sampling rate per pixel), on three challenging scenes(Red wine, Dragon, Water pool) that involve complex caustics, diffuse-specular interactions. For each scene, we shoot photons in 0.1 second, which generates about 0.8M photons with maximum five photons per path; we only keep those photons involving the light-specular paths in the scenes. Because of various compositions of scenes, there are 85K(Red wine), 50K(Water pool) and 77K(Dragon) valid photons that used in above three scenes respectively. As shown in Fig. 6, the results rendered by PT with the same spp could denoise the images, but it could not model the true caustic effect and wave effect. Besides, from the statistical indicators below the images, our method outperforms in the quality of images than the PT algorithm.

5.2. Comparison with PM algorithm

![Figure 7](image)

Figure 7. We show our final results (at the left of above images). We compare against PM\cite{57} with the same input photons and pure PPM\cite{3} with the same and SPPM\cite{1} with the same photon counts on two different scenes. Obviously, our results are with best quality and we could draw the same conclusion from the statistics under the images.

We compare our method with the classical PM with the same k-NN photons as inputs and compare ours with various progressive methods designed to progressively reduce the bandwidth with large photon counts on two challenging scenes(Water pool 2, Glass egg). Note that, across all different scenes with different photon counts, our method with 500 photons as our input performs better than all the comparison PM methods, including standard PM\cite{57}, PPM\cite{3}, and SPPM\cite{1}. As shown in Fig.7, the Water pool2 scene was rendered with the same number of photons and the Glass egg scene was rendered with different number of photons to ensure the quality of images. Whether with the same number of photons or not, our results could generate more true caustic and wave effect than PM’s and PPM’s results with ten times the photon counts as ours. SPPM utilizes common statistical information of local photons to improve the density estimation of PPM. So, it could obtain better results. However, it requires as large as the number of photons to ensure the quality of images. In contrast, our method could achieve significantly better results than SPPM with the same number of photons. From the statistics indicators below the images, our method outperform the best of these variants in quality of images.
5.3. Comparison between our method (with denoising part) and that (without denoising part)

![Comparison of denoising and non-denoising](image)

Figure 8. We compare against our results without denoising process. Obviously, the image rendered with denoising process (b) is with smoother texture and less aliasing than that without denoising process (a).

To demonstrate the necessary of denoising process, we make the comparison experiments between the improved SPPM algorithms with denoising parts and the one without denoising part on two scenes. As shown in Fig. 8, our method with denoising process achieves better than the one without denoising part. With the denoising part, more illumination details are better preserved, and both the high-frequency and low-frequency noises are better removed. The lower error (shown in the quality statistics below the images) confirm that our method with denoising process produces the higher quality.

![Quality statistics for different methods](image)

Figure 9. We show our final results (at the left of above images). We compare against KPCN\cite{57} and RDP\cite{3} with the same input images on three different scenes. Our denoising network removes more multi-scale noises while better preserving illumination details. Besides, from the statistics under the images, our rendered images are with best quality.

To fully test the potential performance of our denoising network, we further compare our denoising network with the state-of-the-art denoising network including the KPCN\cite{7} and RDP\cite{9}, on the same scenes. The result can be seen in Fig. 9. From the figure, our denoising network gains a huge performance boost.

The advantage of our denoising network are two parts. One is the architecture of our network. The unique architecture of ResNet makes itself with sophisticated mapping possibilities through the
identity mapping. This is the main consideration why we choose ResNet as our denoising network solution. The other advantage of our denoising network is that we introduce the photon features based on the general features, which could improve the quality of denoising results. Fig. 10 shows the impact of photon features on our network performance. With photon-related features, our denoising network converges faster.

![Figure 10. Impact of auxiliary features on a smaller ResNet similar to our network. Previous work always utilized the texture, depth and shading normal as the feature to optimize the denoising network[7-11].](image)

Here, we presented the albedo of the materials, the distance between the visible intersection point to the camera, the photon density, the photon flux and the photon gradient. From the images, our L1-loss is less than the previous work (Here we choose the KPCN for comparison). That is to say, our auxiliary features seem to accelerate the denoising network convergence. So our denoising approaches outperforms than other denoising methods when denoising the images rendered by photon mapping.

6. Conclusions
We presented a novel accelerated stochastic progressive photon mapping rendering method with neural network. Firstly, in order to aggregate the photon density of every visible intersect point in the scenes, we introduce a deep neural network to learn the kernel function. Next we could obtain more accurate images with various effects including the caustics effects, wave effects and global illumination effects. Finally, to improve the generated results, we utilize a denoising process with the improved ResNet. We proposed a series of auxiliary photon-related features to better handle noises while preserving correct illumination details, especially caustics. Compared with the state-of-the-art rendering algorithms including the approaches based on PT and the ones based on PM, our method performs better.

Acknowledgements
This work was supported in part by the National Natural Science Foundation of China under Grant U19A2063, in part by the Jilin Provincial Science & Technology Development Program of China under Grant 20190302113GX, in part by the 13th Five-year Science & Technology Research Program of Jilin Provincial Department of Education under Grant JJKH20200792KJ.

References
[1] Hachisuka T, Jensen H W. (2009) Stochastic progressive photon mapping. ACM Transactions on Graphics (TOG), 28: No.5
[2] Jensen H W. (2001) Realistic image synthesis using photon mapping. A K Peters/CRC Press, New York.
[3] Hachisuka T, Ogaki S, Jensen H W. (2008) Progressive Photon Mapping. ACM Transactions on Graphics (TOG), 27: 1-8.
[4] Kang C, Wang L, Xu Y, et al. (2016) A survey of photon mapping state-of-the-art research and future challenges. Frontiers of Information Technology & Electronic Engineering., 17:185-199.
[5] Ritschel T, Dachsacher C, Grosch T, et al. (2012) The state of the art in interactive global illumination. In: Computer Graphics Forum. Oxford, UK., pp: 160-188.
[6] Zeng Z, Wang L, Wang B B, et al. (2020) Denoising Stochastic Progressive Photon Mapping Renderings Using a Multi-Residual Network. Journal of Computer Science and Technology. 35: 506-521.
[7] Bako S, Vogels T, McWilliams B, et al. (2017) Kernel-predicting convolutional networks for denoising Monte Carlo renderings. ACM Trans. Graph. 36: 97:1-97:14.
[8] Vogels T, Rousselle F, McWilliams B, et al. (2018) Denoising with kernel prediction and asymmetric loss functions. ACM Transactions on Graphics (TOG). 37: 1-15.
[9] Wong K M, Wong T T. (2018) Robust deep residual denoising for Monte Carlo rendering. In: SIGGRAPH Asia 2018 Technical Briefs. Tokyo, Japan. 14: 1-4.
[10] Wong K M, Wong T T. (2019) Deep residual learning for denoising Monte Carlo renderings. Computational Visual Media. 5: 239-255.
[11] Kalantari N K, Bako S, Sen P. (2015) A machine learning approach for filtering Monte Carlo noise. ACM Trans. Graph. 34: 122:1-122:12.
[12] Dutré P, Lafortune E P, Willems Y D. (1993) Monte Carlo light tracing with direct computation of pixel intensities. In: 3rd International Conference on Computational Graphics and Visualisation Techniques. Alvor, Portugal. pp: 128 – 137.
[13] Willems E P L Y D. (1993) Bi-Directional Path Tracing. In: International Conference on Computational Graphics & Visualization Techniques. pp: 145-153.
[14] Veach E, Guibas L J. (1955) Optimal combining sampling techniques for Monte Carlo rendering. In: Proceedings of the 22nd annual conference on Computer graphics and interactive techniques. pp: 419-428.
[15] Veach E. (1997) Robust Monte Carlo methods for light transport simulation. Stanford University PhD thesis, 1610.
[16] Pauly M., Kollig T. (2000), Keller A. Metropolis light transport for participating media. In: Rendering Techniques 2000. Vienna, Austria. pp:11-22.
[17] Cline D, Talbot J, Egbert P. (2005) Energy redistribution path tracing. ACM Transactions on Graphics (TOG)., 24: 1186-1195.
[18] Hachisuka T, Jarosz W, Jensen H W. (2010) A progressive error estimation framework for photon density estimation. ACM Transactions on Graphics (TOG)., 29: 1-12.
[19] Knaus C, Zwicker M. (2011) Progressive photon mapping: A probabilistic approach. ACM Transactions on Graphics (TOG)., 30: 1-13.
[20] Vorba J. (2011) Bidirectional photon mapping. In: Proc. of the Central European Seminar on Computer Graphics (CESCG’11).
[21] Hachisuka T, Jensen H W. (2011) Robust adaptive photon tracing using photon path visibility. ACM Transactions on Graphics (TOG)., 30: 1-11.
[22] Georgiev I, Krivánek J, Slusallek P. (2011) Bidirectional light transport with vertex merging. In: SIGGRAPH Asia 2011 Sketches., Hong Kong, China, 27: 1-2.
[23] Hachisuka T, Pantaleoni J, Jensen H W. (2012) A path space extension for robust light transport simulation. ACM Transactions on Graphics (TOG)., 31: 1-10.
[24] Georgiev I, Krivánek J, Davidovic T, et al. (2012) Light transport simulation with vertex connection and merging. ACM Trans. Graph., 31: 192:1-192:10.
[25] Jensen H W, Christensen N J. (1995) Photon maps in bidirectional Monte Carlo ray tracing of
complex objects. Computers & Graphics., 19: 215-224.
[26] Kaplanyan A S, Dachsbacher C. (2013) Adaptive progressive photon mapping. ACM Transactions on Graphics (TOG), 32: 1-13.
[27] Kang C M, Wang L, Xu Y N, et al. (2016) Adaptive Photon Mapping Based on Gradient. Journal of Computer Science and Technology., 31: 217-224.
[28] Schjøth L, Sporring J, Fogh Olsen O. (2008) Diffusion based photon mapping. In: Computer Graphics Forum. Oxford, UK. pp: 2114-2127.
[29] Schjøth L, Frisvad J R, Erleben K, et al. (2007) Photon differentials. In: Proceedings of the 5th international conference on Computer graphics and interactive techniques. Australia and Southeast Asia. Pp:179-186.
[30] Spencer B, Jones M W. (2009) Into the blue: Better caustics through photon relaxation. In: Computer Graphics Forum. Oxford, UK. pp: 28: 319-328.
[31] Spencer B, Jones M W. (2013) Photon parameterisation for robust relaxation constraints. In: Computer Graphics Forum. Oxford, UK., pp: 32: 83-92.
[32] Spencer B, Jones M W. (2013) Progressive photon relaxation. ACM Transactions on Graphics (TOG), 32: 1-11.
[33] Jakob W, Regg C, Jarosz W. (2011) Progressive expectation-maximization for hierarchical volumetric photon mapping. In: Computer Graphics Forum. Oxford, UK. pp: 1287-1297.
[34] Zwicker M, Jarosz W, Lehtinen J, et al. (2015) Recent advances in adaptive sampling and reconstruction for Monte Carlo rendering. In: Computer graphics forum. Oxford, UK. pp: 667-681.
[35] Durand F, Holzschuch N, Soler C, et al. (2005) A frequency analysis of light transport. ACM Transactions on Graphics (TOG), 24: 1115-1126.
[36] Egan K, Tseng Y., Holzschuch N, et al. (2009) Frequency Analysis and Sheared Reconstruction for Rendering Motion Blur. ACM Transactions on Graphics (TOG), 28: No. 93.
[37] Yan L Q, Mehta S U, Ramamoorthi R, et al. (2015) Fast 4D sheared filtering for interactive rendering of distribution effects. ACM Transactions on Graphics (TOG), 35: 1-13.
[38] Wu L, Yan L Q, Kuznetsov A, et al. (2017) Multiple Axis-Aligned Filters for Rendering of Combined Distribution Effects. In: Computer Graphics Forum. Oxford, UK. pp: 155-166.
[39] Overbeck R S, Donner C, Ramamoorthi R. (2009) Adaptive wavelet rendering. ACM Trans. Graph., 28: 140.
[40] Rousselle F, Manzi M, Zwicker M. (2013) Robust denoising using feature and color information. In: Computer Graphics Forum., pp: 121-130.
[41] Chaitanya C R A, Kaplanyan A S, Schied C, et al. (2017) Interactive reconstruction of Monte Carlo image sequences using a recurrent denoising autoencoder. ACM Transactions on Graphics (TOG), 36: 1-12.
[42] Xu B, Zhang J, Wang R, et al. (2019) Adversarial Monte Carlo denoising with conditioned auxiliary feature modulation. ACM Trans. Graph., 38: 224:1-224:12.
[43] Zhu S, Xu Z, Jensen H W, et al. (2020) Deep Kernel Density Estimation for Photon Mapping. In: Computer Graphics Forum., pp: 35-45.
[44] Wand M P, Jones M C. (1994) Kernel smoothing. Chapman and Hall/CRC, New York.
[45] Shirley P, Wade B, Hubbard P M, et al. (1995) Global illumination via density-estimation. In: Rendering Techniques' 95. Vienna, Austria, 219-230.
[46] Qi C R, Su H, Mo K, et al. (2017) Pointnet: Deep learning on point sets for 3d classification and segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. Honolulu, USA. pp: 652-660.
[47] Ledig C, Theis L, Huszár F, et al. (2017) Photo-realistic single image super-resolution using a generative adversarial network. In: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition. Honolulu, USA. pp: 4681-4690.
[48] Nah S, Hyun Kim T, Mu Lee K. (2017) Deep multi-scale convolutional neural network for dynamic scene deblurring. In: Proceedings of the IEEE Conference on Computer Vision and
Pattern Recognition. Honolulu, USA. pp: 3883-3891.

[49] Lim B, Son S, Kim H, et al. (2017) Enhanced deep residual networks for single image super-resolution. In: Proceedings of the IEEE conference on Computer vision and Pattern Recognition. Honolulu, USA. pp: 136-144.

[50] He K, Zhang X, Ren S, et al. (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. Las Vegas, USA. pp: 770-778.

[51] He K, Zhang X, Ren S, et al. (2016) Identity mappings in deep residual networks. In: European conference on computer vision. Amsterdam, The Netherlands, pp: 630-645.

[52] Mao X J, Shen C, Yang Y B. (2016) Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections. arXiv preprint arXiv:1603.09056.

[53] Bitterli B. Rendering resources, 2016. https://benedikt-bitterli.me/resources/, March 2020.

[54] Abadi M, Barham P, Chen J, et al. (2016) Tensorflow: A system for large-scale machine learning. In: 12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16). pp: 265-283.

[55] Da K. (2014) A method for stochastic optimization. arXiv preprint arXiv:1412.6980, 2014.

[56] Glorot X, Bengio Y. (2010) Understanding the difficulty of training deep feedforward neural networks. In: Proceedings of the thirteenth international conference on artificial intelligence and statistics. pp: 249-256.

[57] Jensen H W. (1996) Global illumination using photon maps. In: Eurographics workshop on Rendering techniques. Springer, Vienna, pp: 21-30.