Ma MC, Wang L, Xu YN *et al.* Unsupervised reconstruction for gradient-domain rendering with illumination separation. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 39(6): 1281-1291 Nov. 2024. DOI: 10.1007/s11390-024-3142-4

Unsupervised Reconstruction for Gradient-Domain Rendering with Illumination Separation

Ming-Cong Ma (马鸣聪), Lu Wang* (王　璐), *Member, CCF*, Yan-Ning Xu (徐延宁), *Member, CCF* and Xiang-Xu Meng (孟祥旭), *Member, CCF*

School of Software, Shandong University, Jinan 250101, China

E-mail: 201734818@mail.sdu.edu.cn; luwang_hcivr@sdu.edu.cn; xyn@sdu.edu.cn; mxx@sdu.edu.cn

Received February 2, 2023; accepted August 7, 2024.

Abstract Gradient-domain rendering methods can render higher-quality images at the same time cost compared with traditional ray tracing rendering methods, and, combined with the neural network, achieve better rendering quality than conventional screened Poisson reconstruction. However, it is still challenging for these methods to keep detailed information, especially in areas with complex indirect illumination and shadows. We propose an unsupervised reconstruction method that separates the direct rendering from the indirect, and feeds them into our unsupervised network with some corresponding auxiliary channels as two separated tasks. In addition, we introduce attention modules into our network which can further improve details. We finally combine the results of the direct and indirect illumination tasks to form the rendering results. Experiments show that our method significantly improves image quality details, especially in scenes with complex conditions.

Keywords gradient-domain rendering, unsupervised network, deep learning

1 Introduction

Monte Carlo ray tracing methods are widely used in many fields due to their flexibility and versatility. However, the methods also suffer from slow convergence, significant image noise at a sampling rate of low spp (samples per pixel), and expensive expenditure at a sampling rate of high spp.

Gradient-domain rendering[1] uses spatial correlation to alleviate the problem of excessive image noise at low spp. This method additionally outputs the images' horizontal and vertical gradients, and these high-frequency details provide constraints on image reconstruction, thus allowing the screened Poisson solvers to produce better results. Since image reconstruction is essentially an optimization iteration problem, adding more constraints benefits the problem's solution. As a result, more and more auxiliary features are beginning to be introduced into the reconstruction process. These new constraints undoubtedly optimize the quality of the reconstructed image, but they also significantly increase the time and resource consumption of image rendering and reconstruction.

The emergence of deep neural network methods has led to new directions of exploration for image reconstruction. Well-trained network models[2–4] have shown good enough performance to replace the screened Poisson solvers regarding reconstruction quality and processing efficiency. GradNet[2] has even reduced the expense of training by applying unsupervised networks to image reconstruction, thus making deep neural networks more competitive. However, these currently existing methods make maintaining some indirect illumination and shadow details difficult. Some of the denoising methods of Schie *et al.*[5] keep details by distinguishing between direct and indirect illumination. This direct and indirect illumination separation has succeeded in traditional ray trac-

Regular Paper

This work has been partially supported by the National Key Research and Development Program of China under Grant No. 2020YFB1708900 and the National Natural Science Foundation of China under Grant No. 62272275.

*Corresponding Author

ing methods. However, there is currently no gradient-domain algorithm separating direct and indirect illumination. Separated direct and indirect illumination have different lighting properties, which helps the subsequent network process and optimize illumination independently better to reconstruct lighting and shadow details in different dimensions. Naturally, the gradient information can be easily separated for direct and indirect illumination, followed by a reconstruction stage on direct and indirect illumination independently.

In this paper, we decompose the rendering task into two separate rendering tasks of direct and indirect illumination, supplemented with auxiliary features and gradient channels. Then, the two separated tasks pass through the rendering pipeline relatively independently. Our method combines the direct and indirect illumination reconstruction results only in the final stage. We have experimentally demonstrated that our method does not result in additional energy loss or quality degradation. Moreover, the final rendering results are of higher quality regarding detail reconstruction than those of traditional methods, especially in indirect illumination and shadows, because the high-quality direct illumination channels are processed separately. Through ablation experiments, we have also demonstrated that both the illumination separation process and the network model's improvement positively impact the quality of reconstruction results.

In summary, our main contributions are:

- a gradient-domain rendering process for network reconstruction that separates the direct and indirect illumination to improve the final reconstruction quality;
- an unsupervised network with an attention module that can better preserve illumination details while removing noise.

2 Related Work

2.1 Gradient-Domain Rendering and Image Reconstruction

In addition to using color information, the gradient-domain rendering methods introduce spatial information to assist Monte Carlo rendering in convergence, also known as image gradients. Since the seminal work of gradient-domain Metropolis transport by Lehtinen *et al.*[6], the gradient-domain methods have been also extended to path tracing by Kettunen *et al.*[7]. The combination of other related techniques with gradient-domain methods also brings different advantages. For example, Manzi *et al.*[1] applied bidirectional path tracing to improve the ray path.

In gradient-domain rendering, image reconstruction usually implies using a screened Poisson solver by Bhat *et al.*[8] or iteration-based optimization methods by Rousselle *et al.*[9]. For better results, subsequent work[10–12] usually used additional auxiliary features as new constraints, increasing the time cost and memory overhead. He *et al.*[13] proposed a non-uniform sampling method based on a gain control strategy, using different spp for high-frequency and low-frequency regions to improve the rendering quality in a targeted manner.

2.2 Deep Learning for Denoising and Gradient-Domain Rendering

In recent years, deep learning networks have been applied positively in traditional Monte Carlo path-tracking image denoising[14–17]. Han *et al.*[18] built network extraction features from G-buffers and P-buffers, respectively, and weighted the results of the two to obtain the final denoising result. Alpay and Akyuz[19] conducted higher spp at lower resolutions and finally passed the network, which performed denoising and upsampling to restore resolutions. Zhang *et al.*[20] used an automatically selected feature set to denoise volume rendering results. Firmino *et al.*[21] used the confidence interval method to use the result only when the denoising network result is beneficial and reduces the impact of the network at high sampling numbers. Currius *et al.*[22] proposed a real-time denoising network for hair rendering. In contrast, the application of deep networks to gradient-domain rendering is somewhat later. To the best of our knowledge, Kettunen *et al.*[3] firstly combined deep learning with gradient-domain rendering and achieved results beyond traditional Monte Carlo path-tracing image-denoising. However, Kettunen *et al.*[3] used a supervised network to perform denoising.

In contrast, the training datasets for supervised networks require many high-quality images as ground truth, which is a considerable computational overhead. Unsupervised learning, on the other hand, due to its nature, can avoid the computation for ground truth, and thus focuses the vast majority of its effort on the training dataset, thus allowing the dataset size to be increased. Unsupervised networks, therefore, have certain unique advantages. Guo *et al.*[2] presented unsupervised deep learning solution to screened

Poisson reconstruction in gradient-domain rendering. The unsupervised deep learning solution achieved better results than traditional screened Poisson reconstruction and was comparable to the supervised network. Xu *et al.*[4] proposed an unsupervised convolutional neural network for image reconstruction in gradient-domain volumetric photon density estimation. However, all these methods still need to be improved to keep illumination details while removing noise, especially in areas with detailed shadows and indirect illumination. Our method focuses on unsupervised networks to keep illumination details by separating direct and indirect illumination.

3 Background

The key idea of gradient-domain rendering is using highly correlated shift paths of the base path to estimate finite differences between adjacent pixels. These finite differences are generally referred to as gradients.

The color channel (base image) and the gradient channels output from the gradient-domain renderer are shown in Fig.1. For the final result, the gradient-domain rendering reconstructs the final image from the typically more noisy color channel and less noisy gradient channels by solving a screened Poisson equation[23].

$$\hat{I} = \arg\min_{I} \|\alpha(I - I_{\mathrm{b}})\|^{n} + \left\| \begin{pmatrix} H_{\mathrm{dx}} I \\ H_{\mathrm{dy}} I \end{pmatrix} - \begin{pmatrix} I_{\mathrm{dx}} \\ I_{\mathrm{dy}} \end{pmatrix} \right\|^{n}, \tag{1}$$

where α is used to control the weight between the gradient and the color, I_{b} is the color image channel generated by the gradient-domain renderer, I_{dx} and I_{dy} are the horizontal pixel gradient and the vertical pixel gradient of I_{b}, respectively. In contrast, H_{dx} and H_{dy} are the finite difference operators in the vertical and horizontal directions, respectively. The final output image $\hat{I}$ minimizes both its pixel difference and gradient difference from the input image.

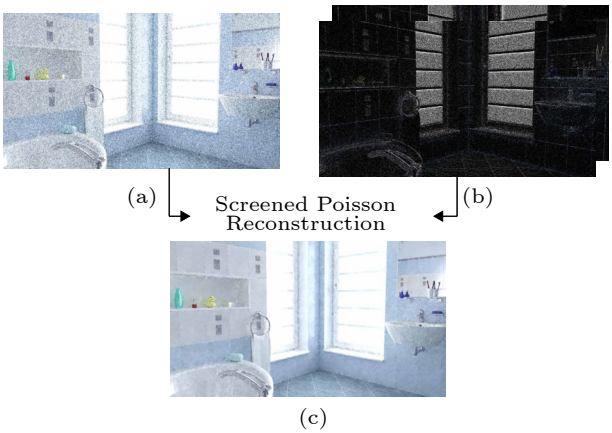

Fig.1. Basic process of gradient-domain rendering. (a) Color channel. (b) Gradient channels. (c) Reconstruction image.

In the reconstruction process, the result is unbiased when using the L_2 norm, but it usually results in a poor visual experience with anomalous data. The result of L_1 reconstruction is much better visually, but unfortunately, it is biased[6, 7, 24].

4 Our Method

Fig.2 presents an overview of our method. Our reconstructed models have the same structure and do

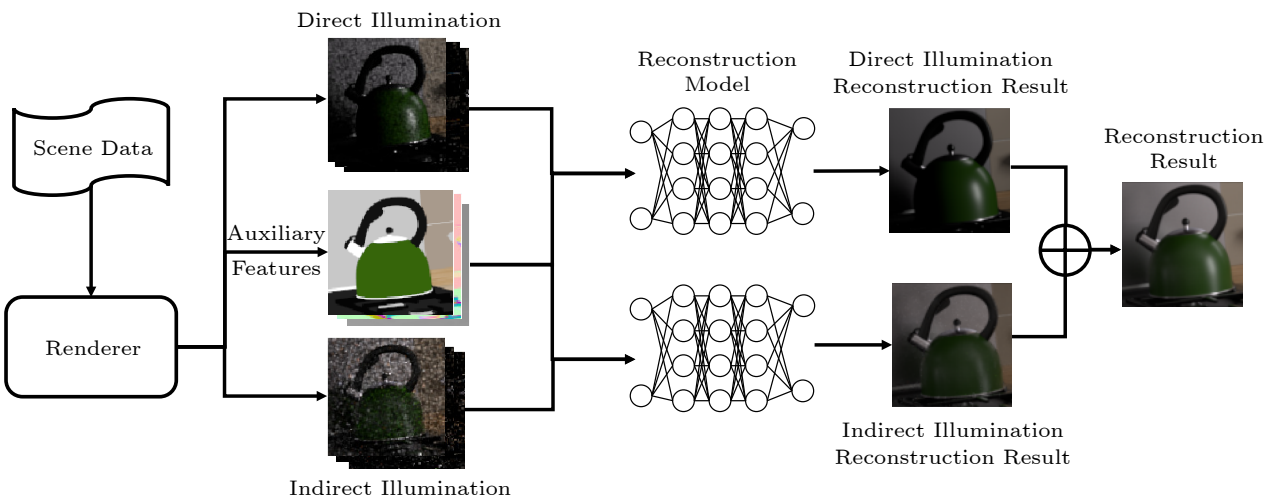

Fig.2. Overview of our method. The direct and indirect illumination datasets are fed into the same reconstruction models for independent image reconstruction, and the same auxiliary features are shared between them. The individual reconstruction results are summed to obtain the final reconstructed image.

not significantly differentiate between direct and indirect illumination. A well-trained network has general reconstruction capabilities for direct and indirect illumination and does not require a different design. Although our reconstruction models use the same network structure, the two reconstruction models have different parameters after the final training because they use different data during the training phase.

4.1 Illumination Separation

In our method, the direct and indirect illumination are separated, and the gradient data is applied separately. Therefore, compared with (1), our method can be expressed by the following formula:

$$\begin{aligned}\hat{I} &= \psi^d(x^d) + \psi^i(x^i)\\ &= \psi^d(I^d_{\rm b}, I^d_{\rm dx}, I^d_{\rm dy}, F) + \psi^i(I^i_{\rm b}, I^i_{\rm dx}, I^i_{\rm dy}, F),\end{aligned}$$

where ψ^d and ψ^i represent the reconstruction network model of direct illumination and the reconstruction network model of indirect illumination, respectively. x^d and x^i represent the training sample set of direct and indirect illumination, respectively. $I^d_{\rm b}, I^d_{\rm dx}, I^d_{\rm dy}$ represent the base color, horizontal gradient, and vertical gradient channels of direct illumination, respectively. Similarly, $I^i_{\rm b}, I^i_{\rm dx}, I^i_{\rm dy}$ represent the base color, horizontal gradient, and vertical gradient channels of indirect illumination, respectively. Note that, both the direct illumination samples and the indirect illumination samples use the same feature channels F. After the radiance calculations of direct and indirect illumination are completed, radiance data are added together to obtain the final denoising result.

Fig.3 shows the illumination separation results of a kitchen scene. We can notice the fact that the noise scales of direct illumination are different from those of indirect illumination. It is reasonable to treat the direct and indirect illumination separately. After the two separate reconstruction tasks are completed, the images are superimposed to form the final reconstruction result.

4.2 Network Architecture

As shown in Fig.4 and Fig.5, our network architecture extends from GradNet[2], whose reliability of the multi-branch structure was proven. Therefore our method follows this structure.

The individual structure of each branch is similar to U-net[25], with branches connected to each other in the upsampling phase by a residual structure. The data-branch is designed to extract low-frequency contents from noisy images, and the gradient-branch is designed to extract high-frequency details so that only the gradient images are involved in the gradient-branch.

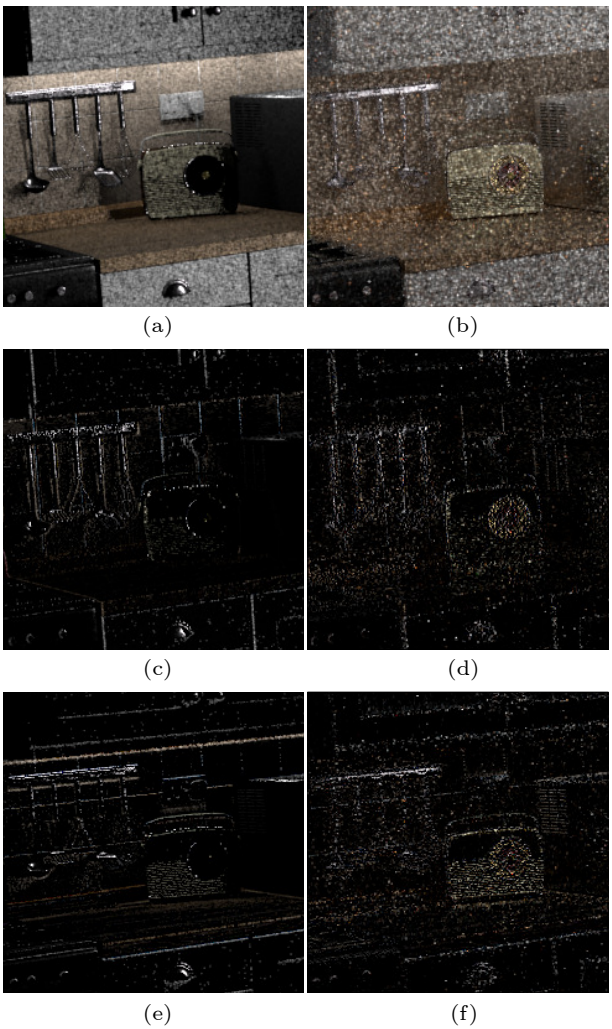

Fig.3. Illumination separation results of a kitchen scene, including base color and two gradient channels for direct and indirect illumination, respectively. (a) Color channel of direct illumination. (b) Color channel of indirect illumination. (c) Vertical gradient channel of direct illumination. (d) Vertical gradient channel of indirect illumination. (e) Horizontal gradient channel of direct illumination. (f) Horizontal gradient channel of indirect illumination.

In our network, $I_{\rm b}$ and F are the input image and related auxiliary channels, respectively, as the input of the data-branch, which are used to extract low-frequency data; $I_{\rm dx}$ and $I_{\rm dy}$ are the horizontal gradient and the vertical gradient of the input image, respectively, as the gradient-branch input for extracting high-frequency details.

The edge information and subtle structural fea-

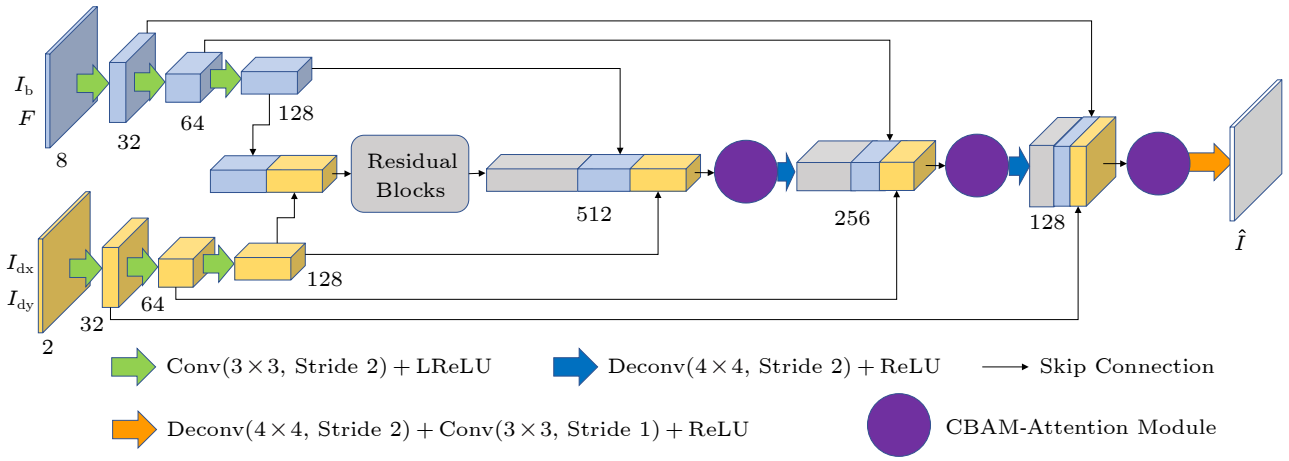

Fig.4. Overview of our network architecture. Our network adopts the results of dual branches, where different branches correspond to channels with distinct features. Attention modules are only added at the upsampling stage to prevent too many attention modules from mistaking noise for features for retention.

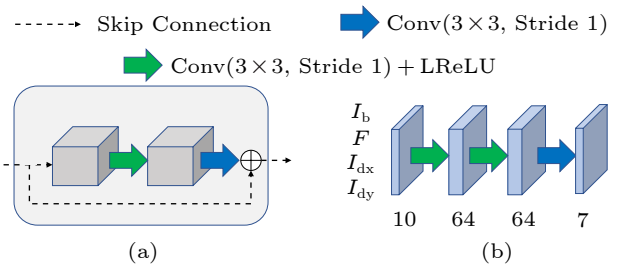

Fig.5. Auxiliary sub-models of the main network. The residual block is of classical design, while the G-branch only provides the loss function for the main network. (a) Residual block. (b) G-branch.

tures of objects are what we want the network to pay special attention to and retain as much as possible. At the same time, the rendering noise caused by low spp is ignored by the network as useless information as much as possible. Based on the above motivations, we introduce the attention module. It allows the network to distinguish between details of different importances as much as possible. The attentional module uses the classical CBAM[26] (convolutional block attention module) structure, which is only used in the upsampling stage. The introduction of the attention module effectively eliminates the noise in the reconstruction result, making the final result much cleaner. At the same time, the judgment of the partial edge of the object is more accurate, and the partial edge details are effectively distinguished.

We also try to introduce other network models, such as the classic U-net-Block[25], to replace part of the structure in the network or apply the attention module to other steps, such as skip connections. However, the final result proves that these more complex structures cannot bring positive feedback for the final reconstruction. In our experiments, we find interesting rules. Macroscopically speaking, the more complex the convolutional structure of the network and the more convolutional layers, the more difficult it is to remove the noise. The network mistakenly retains this noise that should have been eliminated as a feature. The introduction of the attention module can eliminate this noise within a specific range, but the excessive use of the attention module will cause the details that should be preserved to be blurred or even destroyed. Therefore, our final network structure reaches the balance between denoising and detail preserving.

4.3 Loss Function

For unsupervised networks, the most important thing is the design of the loss function.

The loss function applied to the final network training consists of the following three components: a data loss $\mathcal{L}_{\text{Data}}$, a gradient loss $\mathcal{L}_{\text{Grad}}$, and a first-order loss $\mathcal{L}_{\text{1st}}$. They can be expressed in the following form:

$$\mathcal{L}_{\text{Total}} = \alpha \times \mathcal{L}_{\text{Data}} + \mathcal{L}_{\text{Grad}} + \beta \times \mathcal{L}_{\text{1st}}.$$

Based on the existing loss function[2], which performs well, we adjust the weights of some of its loss terms to fit our improved network architecture. The data loss, gradient loss, and first-order loss take a similar strategy as in GradNet[2]. Parameter α is used to adjust the effect of the color channel on the final loss function. In supervised networks, this term is usually adjusted to be close to 1.0 to be visually close to the ground truth. However, due to the lack of reliable ground truth in unsupervised networks, it is difficult to trust the loss results of color channel as in su-

pervised networks. We adjust the value of the parameter α to try out its effect on the final result and finally settle on 1.0 with the same weight as the gradient loss. The parameter β will increase slowly with the epoch, and eventually, stabilize at 2.

4.4 Dataset Preparation

Since unsupervised networks are trained without the ground truth, it is much easier to generate only relatively low-quality training samples. Although GradNet[2] open-sources its associated dataset, we cannot use it directly as its data samples do not separate direct and indirect illumination. To ensure experimental fairness and maintain consistency in the data size, we choose seven scenes from Benedikt Bitterli①, and generate 100 frames for each scene at 64 spp, finally acquiring 700 frames of images with high-resolution (1 280×720). We randomly intercept 20 patches of 256×256 size from each frame image, which ultimately constitute a training dataset containing 14 000 samples. In comparison, the GradNet[2] dataset contains 12 654 samples, which has roughly the same number of samples as ours.

For each frame, the renderer generates the base images and the corresponding gradients with our gradient-domain path tracer extended from Manzi *et al.*[23] at a sampling rate of 64 spp, along with three additional buffers (i.e., albedo, normal, and depth). The albedo buffer is recorded for the first non-specular hit point on the path, which has been normalised to the range [0, 1]. The normal buffer records the same point, but it needs to be normalised to the range [0, 1]. The depth buffer records the location of the hit point and uses the scene location information to complete the normalisation to the range [0, 1]. Therefore, the albedo, normal, and depth buffers are all normalised before entering the network. Occasionally, there will be additional buffers with non-numeric values, and those are set to zero.

For the construction of the validation set, we choose two scenes (BATHROOM, CLASSROOM) used in the training set and render the results under different spp with a new perspective and parameters. Also, three entirely new scenes (KITCHEN-1, KITCHEN-2, and DININGROOM) are selected, and their results under different spp are rendered. Five scenes are combined to form the final validation set while generating their ground truth at 81 960 spp as a reference for final comparison.

4.5 Training

Our network is implemented based on the PyTorch framework[2]. We use some of the same base parameters as in Guo *et al.*[2] to train our network. We use mini-batch SGD (stochastic gradient descent) and apply the Adam solver[27] for training with the moment parameters $\beta_1 = 0.5$ and $\beta_2 = 0.999$. We tune the learning rate with the basic learning rate of 0.000 1 and the power of 0.95 every other epoch. The initial weights of the network are initialized using the method mentioned in He *et al.*[28]. We use an NVIDIA GTX 3090 GPU for training and execute 151 epochs. The training samples are fed into the network in batches of 32, and four processes are trained in parallel simultaneously. Ninety percent of the samples are used as the training set, and the remaining ten percent as the validation set.

5 Results and Discussion

To evaluate our method, we compare it with existing unsupervised network reconstruction techniques. Afterwards, we analyze in detail the various design choices in our models. Finally, we discuss the performance of the method again under different spp inputs. The reference images are generated by path tracing at an extremely high sampling rate.

5.1 Runtime Performance

We compare the average time taken by the network to infer the test data samples. The average time taken by GradNet[2] to infer a frame of 720×1 080 resolution is 265.54 ms, while the average time taken by our network to infer the same frame is 283.86 ms. Although our network adds additional structures compared with GradNet[2], the increased time overhead of our network is not readily noticeable but with significant quality improvement. On the other hand, our method does not aim at real-time rendering, and the slight increase in time overhead is perfectly acceptable for offline rendering.

①Bitterli B. Rendering resources, 2016. https://benedikt-bitterli.me/resources/, Sept. 2024.

5.2 Quality Evaluation

We compare our method with classical screened Poisson image reconstruction method (using L_1)[1] and the existing state-of-the-art unsupervised network GradNet[2]. As shown in Fig.6, our method performs better in terms of the RelMSE (relative mean square error) score and the details of image preserva-

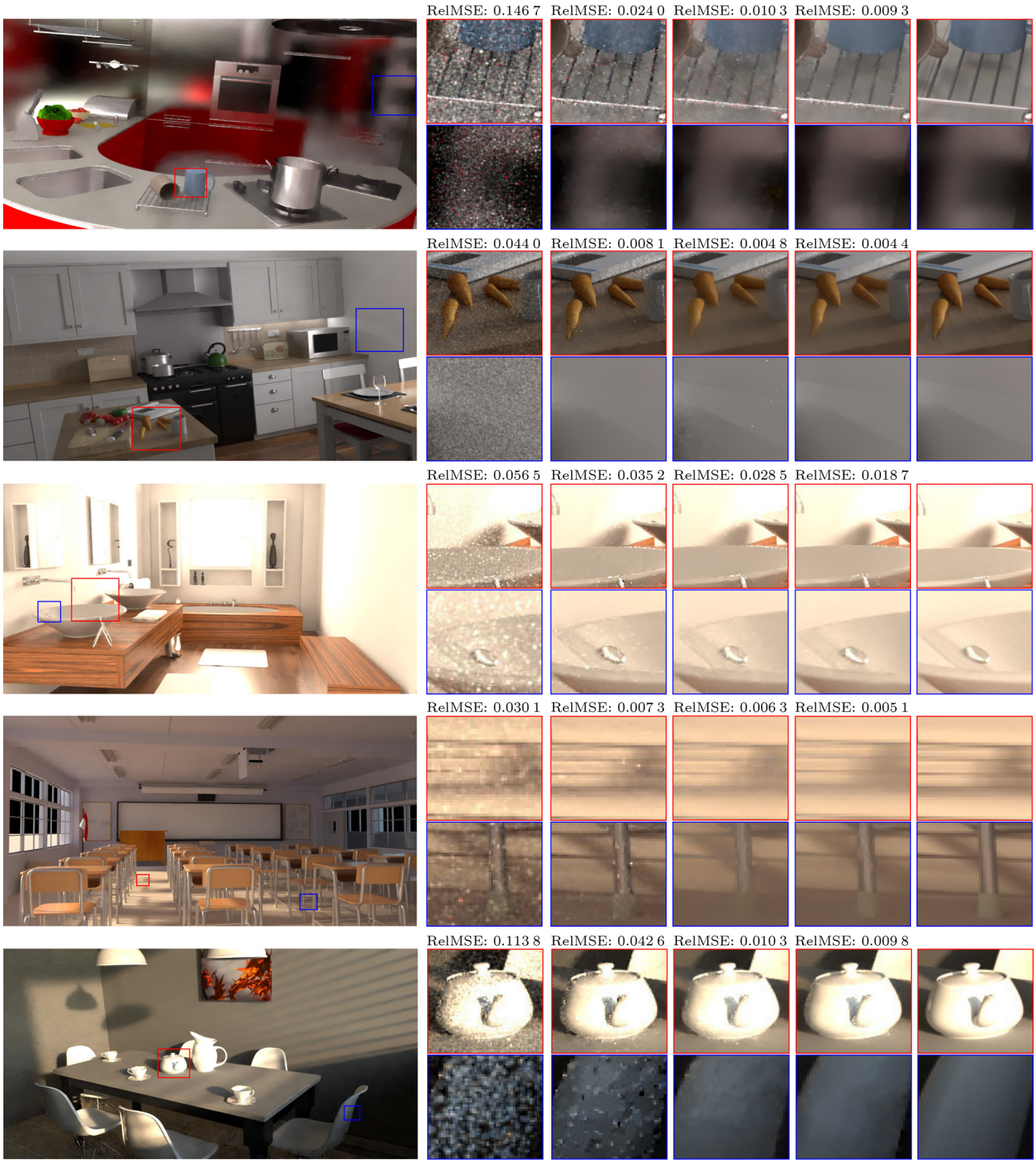

Fig.6. Image reconstruction results using our method, compared with existing reconstruction methods: L_1 Possion reconstruction[1] and GradNet[2] in five scenes: KITCHEN-1 (64 spp), KITCHEN-2 (32 spp), BATHROOM (32 spp), CLASSROOM (64 spp), and DININGROOM (16 spp) (from line 1 to line 5, respectively). (a) Reconstruction results with our method. (b) Details of the input image. (c) Details of the L_1 reconstruction[1]. (d) Details of GradNet[2]. (e) Details of our method. (f) Ground truth.

tion, especially in the conservation of indirect illumination and shadow details. The reconstruction results of different methods for different scenes under other evaluation metrics are shown in Table 1. It can be seen that our method has the best results in almost all scenes under different evaluation criteria, including two different LPIPS (learned perceptual image patch similarity) methods and SSIM (structural similarity index measure), and only in the KITCHEN-2 scene, its SSIM score is slightly smaller than that of GradNet[2] by a very small margin.

Table 1. Reconstruction Quality Comparisons with Existing Methods

Scene	Method	SSIM	LPIPS (ALEX)	LPIPS (VGG)
KITCHEN-1	Input	0.859 8	0.622 7	0.580 1
	L_1[1]	0.961 3	0.194 8	0.235 3
	GradNet[2]	0.980 7	0.104 2	0.170 3
	Ours	**0.982 4**	**0.096 7**	**0.168 2**
KITCHEN-2	Input	0.923 8	0.510 3	0.497 4
	L_1[1]	0.986 3	0.152 6	0.177 5
	GradNet[2]	**0.990 2**	0.052 2	0.090 5
	Ours	0.990 1	**0.045 5**	**0.083 3**
BATHROOM	Input	0.826 1	0.416 2	0.418 6
	L_1[1]	0.842 2	0.095 1	0.195 3
	GradNet[2]	0.810 2	0.064 7	0.180 8
	Ours	**0.924 0**	**0.053 6**	**0.175 4**
CLASSROOM	Input	0.946 2	0.376 7	0.399 9
	L_1[1]	0.989 8	0.070 2	0.136 3
	GradNet[2]	0.988 9	0.032 0	0.080 3
	Ours	**0.991 2**	**0.029 7**	**0.077 2**
DININGROOM	Input	0.835 8	0.530 8	0.550 8
	L_1[1]	0.892 9	0.191 8	0.334 7
	GradNet[2]	0.958 3	0.073 1	0.146 9
	Ours	**0.975 5**	**0.067 4**	**0.132 8**

Note: The lowest error is marked in bold.

It is clear that the unsupervised method has a significant improvement over the classical method, and the results of our method without illumination separation will be evaluated in Subsection 5.3 to determine the network performance improvement further.

5.3 Ablation Experiments

In the ablation experiments, we focus on comparing the impact of the illumination separation process and the network structure with the attention module.

Impact of Illumination Separation. Figs.7(a) and 7(b) both use the GradNet[2] network, while introducing the illumination separation process in Fig.7(b). It can be seen that the shaded boundaries are more explicit in the latter, while the RelMSE decreases. This illustrates the positive impact of the illumination separation process on the final results. The positive effects of illumination separation are also illustrated in Figs.7(c) and 7(d), while using our network structure. Our network with illumination separation also outperforms the one without. We can conclude that both for GradNet[2] and our network, the illumination separation process has resulted in a more confident results.

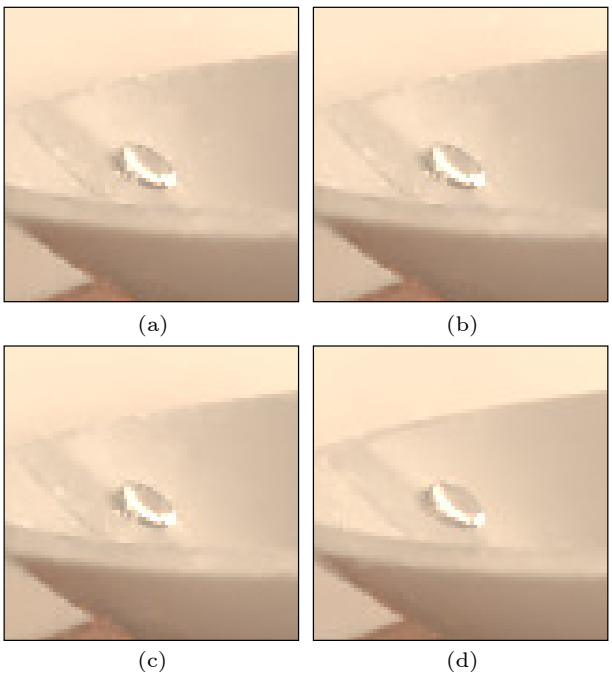

Fig.7. Ablation experiment results. (a) Method without the attention module and without illumination separation, RelMSE: 0.026 2. (b) Method without the attention module and with illumination separation, RelMSE: 0.020 8. (c) Method with the attention module and without illumination separation, RelMSE: 0.023 5. (d) Method with the attention module and with illumination separation, RelMSE: 0.017 7.

Impact of the Network Structure with the Attention Module. Figs.7(a) and 7(c) show results of different network structures without illumination separation. Fig.7(a) use the original network of GradNet[2], while Fig.7(c) adds the attention module. It can be seen that the noise is better removed after introducing the attention module, also the objective evaluation metric RelMSE is reduced. The results of different networks with illumination separation (Figs.7(b) and 7(d)) also show the positive effect of the attention module. Fig.7(d) shows results of our method, where both the attention module and the illumination separation process are introduced, the shadow shape and the noise problems are considerably im-

proved. The objective evaluation metric RelMSE is also reduced to a greater extent, suggesting that the use of both the attention module and the illumination separation process can further enhance the noise removal capability of the network.

5.4 Performance Under Different Spp

To test the effect of our network under different spp inputs, we have generated a total of 11 sets of test data with different spp, with the corresponding spp ranging from 2^0 to 2^{10}. However, limited by the error size, we only show some of the results. Especially in higher spp, the difference between different methods is minimal, and it is not easy to show specific differences.

The RelMSE evaluations of our method and existing methods under different spp are shown in Fig.8. Although our method uses only 64 spp of input during training, it shows superior reconstruction ability for different spp of the test set. Although at a higher spp, the final denoising results of different methods are similar because the quality of the initial image generated by the renderer is sufficient. However, our method has more evident advantages in low spp images with more noise and poorer quality. While the RelMSE evaluation index is lower, the visual effect is also cleaner.

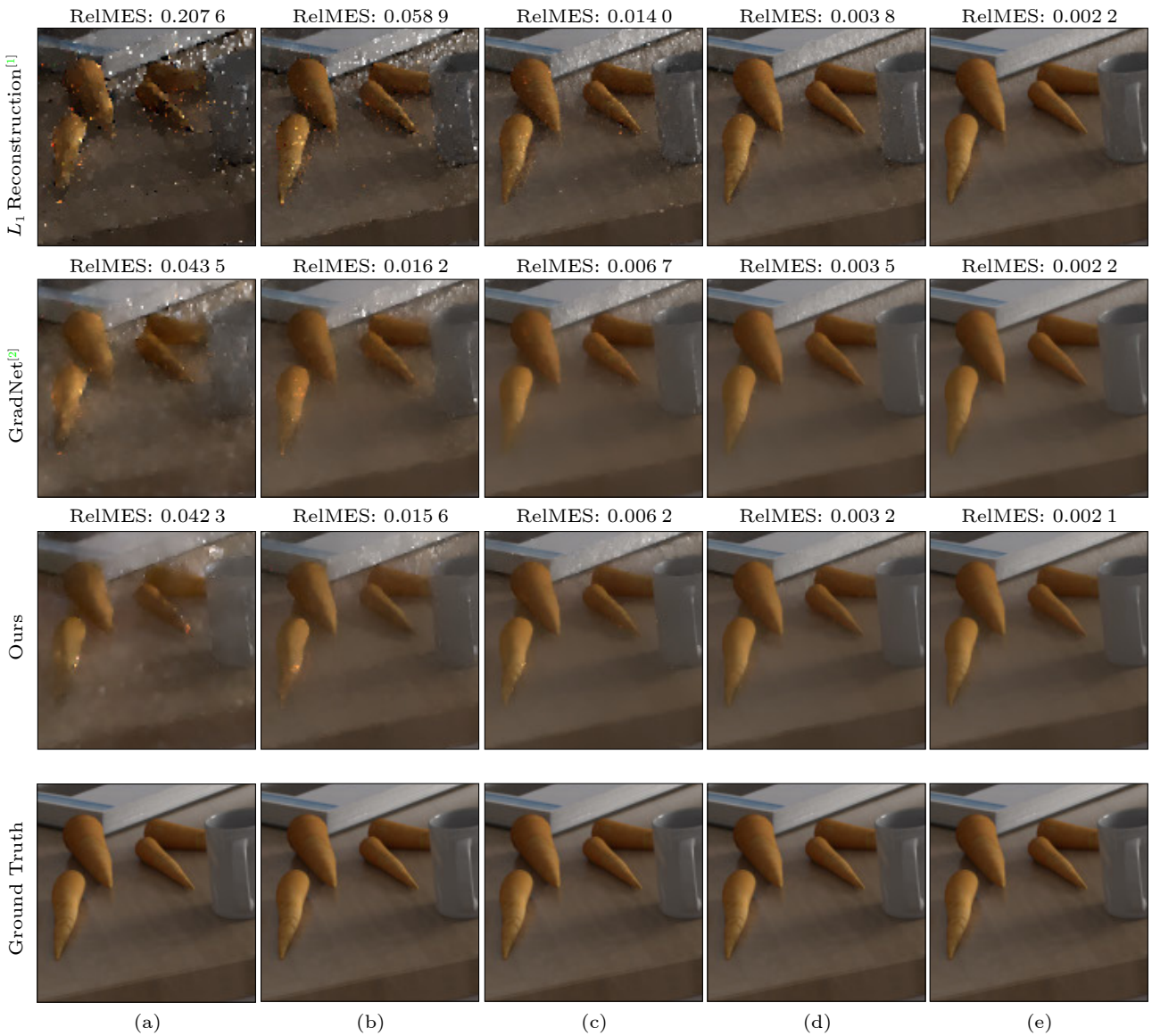

Fig.8. Results of our method compared with other methods in different spp. Evaluations are performed at different spp in the KITCHEN-2 scene. The classical screened Poisson reconstruction method[1] is used in L_1 reconstruction, and the existing unsupervised network reconstruction[2] is used in GradNet reconstruction. (a) 1 spp. (b) 4 spp. (c) 16 spp. (d) 64 spp. (e) 256 spp.

6 Conclusions

We proposed a gradient-domain rendering process for network reconstruction that separates direct and indirect illumination to improve the final reconstruction quality. The illumination separation process effectively preserves extra detail in the final result, especially indirect illumination and shadow. Meanwhile, we presented an unsupervised network with an attention module, which can better preserve features while removing noise. The introduction of the attention module improves the network's ability to deal with noise to a certain extent, making the final results more clean and friendly in terms of visual effects.

Several interesting future work can be done in the following areas. The processing flow of illumination separation should have a certain generality to the network reconstruction method. It may be possible to use this flow in other methods for further improvement and optimization. It may also be helpful to distinguish between material types, such as distinguishing between diffuse and specular results for separate processing. Combining with the temporal domain is also an interesting idea, making full use of the continuity between frames to improve the quality of single-frame generation further.

Conflict of Interest The authors declare that they have no conflict of interest.

References

[1] Manzi M, Kettunen M, Aittala M, Lehtinen J, Durand F, Zwicker M. Gradient-domain bidirectional path tracing. In *Proc. the Eurographics Symposium on Rendering 2015*, Jun. 2015, DOI: 10.2312/sre.20151168.

[2] Guo J, Li M, Li Q, Qiang Y, Hu B, Guo Y, Yan L Q. Gradnet: Unsupervised deep screened Poisson reconstruction for gradient-domain rendering. *ACM Trans. Graphics*, 2019, 38(6): 223. DOI: 10.1145/3355089.3356538.

[3] Kettunen M, Härkönen E, Lehtinen J. Deep convolutional reconstruction for gradient-domain rendering. *ACM Trans. Graphics*, 2019, 38(4): Article No. 126. DOI: 10.1145/3306346.3323038.

[4] Xu Z, Sun Q, Wang L, Xu Y, Wang B. Unsupervised image reconstruction for gradient-domain volumetric rendering. *Computer Graphics Forum*, 2020, 39(7): 193–203. DOI: 10.1111/cgf.14137.

[5] Schied C, Kaplanyan A, Wyman C, Patney A, Chaitanya C R A, Burgess J, Liu S, Dachsbacher C, Lefohn A, Salvi M. Spatiotemporal variance-guided filtering: Real-time reconstruction for path-traced global illumination. In *Proc. the 2017 High Performance Graphics*, Jul. 2017, Article No. 2. DOI: 10.1145/3105762.3105770.

[6] Lehtinen J, Karras T, Laine S, Aittala M, Durand F, Aila T. Gradient-domain metropolis light transport. *ACM Trans. Graphics*, 2013, 32(4): Article No. 95. DOI: 10.1145/2461912.2461943.

[7] Kettunen M, Manzi M, Aittala M, Lehtinen J, Durand F, Zwicker M. Gradient-domain path tracing. *ACM Trans. Graphics*, 2015, 34(4): Article No. 123. DOI: 10.1145/2766997.

[8] Bhat P, Curless B, Cohen M, Zitnick C L. Fourier analysis of the 2D screened Poisson equation for gradient domain problems. In *Computer Vision–ECCV 2008*, Forsyth D, Torr P, Zisserman A (eds.), Springer, 2008, pp.114–128. DOI: 10.1007/978-3-540-88688-4_9.

[9] Rousselle F, Jarosz W, Novák J. Image-space control variates for rendering. *ACM Trans. Graphics*, 2016, 35(6): Article No. 169. DOI: 10.1145/2980179.2982443.

[10] Manzi M, Vicini D, Zwicker M. Regularizing image reconstruction for gradient-domain rendering with feature patches. *Computer Graphics Forum*, 2016, 35(2): 263–273. DOI: 10.1111/cgf.12829.

[11] Back J, Yoon S E, Moon B. Feature generation for adaptive gradient-domain path tracing. *Computer Graphics Forum*, 2018, 37(7): 65–74. DOI: 10.1111/cgf.13548.

[12] Ha S, Oh S, Back J, Yoon S E, Moon B. Gradient outlier removal for gradient-domain path tracing. *Computer Graphics Forum*, 2019, 38(2): 245–253. DOI: 10.1111/cgf.13634.

[13] He H, Zhao Y, Liu H. Gradient-domain path tracing with gain control strategy. *Journal of Computer-Aided Design & Computer Graphics*, 2022, 34(2): 294–304. DOI: 10.3724/SP.J.1089.2022.18837. (in Chinese)

[14] Bako S, Vogels T, Mcwilliams B, Meyer M, NováK J, Harvill A, Sen P, Derose T, Rousselle F. Kernel-predicting convolutional networks for denoising Monte Carlo renderings. *ACM Trans. Graphics*, 2017, 36(4): Article No. 97. DOI: 10.1145/3072959.3073708.

[15] Chaitanya C R A, Kaplanyan A S, Schied C, Salvi M, Lefohn A, Nowrouzezahrai D, Aila T. Interactive reconstruction of Monte Carlo image sequences using a recurrent denoising autoencoder. *ACM Trans. Graphics*, 2017, 36(4): Article No. 98. DOI: 10.1145/3072959.3073601.

[16] Vogels T, Rousselle F, Mcwilliams B, Röthlin G, Harvill A, Adler D, Meyer M, Novák J. Denoising with kernel predictionand asymmetric loss functions. *ACM Trans. Graph.*, 2018, 37(4): Article No. 124. DOI: 10.1145/3197517.3201388.

[17] Suppan C, Chalmers A, Zhao J, Doronin A, Rhee T. Neural screen space rendering of direct illumination. In *Proc. the 29th Pacific Conference on Computer Graphics and Applications*, Oct. 2021, pp.37–42. DOI: 10.2312/pg.20211385.

[18] Han K B, Odenthal O G, Kim W J, Yoon S E. Pixel-wise guidance for utilizing auxiliary features in Monte Carlo Denoising. *Proceedings of the ACM on Computer Graphics and Interactive Techniques*, 2023, 6(1): Article No. 11. DOI: 10.1145/3585505.

[19] Alpay K C, Akyuz A O. Denoising and guided upsampling of Monte Carlo path traced low resolution renderings. In *Proc. the 2022 ACM SIGGRAPH Posters*, Aug. 2022, Article No. 37. DOI: 10.1145/3532719.3543250.

[20] Zhang X, Ott M, Manzi M, Gross M, Papas M. Automatic feature selection for denoising volumetric renderings. *Computer Graphics Forum*, 2022, 41(4): 63–77. DOI: 10.1111/cgf.14587.

[21] Firmino A, Frisvad J R, Jensen H W. Progressive denoising of Monte Carlo rendered images. *Computer Graphics Forum*, 2022, 41(2): 1–11. DOI: 10.1111/cgf.14454.

[22] Currius R R, Assarsson U, Sintorn E. Real-time hair filtering with convolutional neural networks. *Proceedings of the ACM on Computer Graphics and Interactive Techniques*, 2022, 5(1): Article No. 15. DOI: 10.1145/3522606.

[23] Manzi M, Kettunen M, Durand F, Zwicker M, Lehtinen J. Temporal gradient-domain path tracing. *ACM Trans. Graphics*, 2016, 35(6): Article No. 246. DOI: 10.1145/2980179.2980256.

[24] Hua B S, Gruson A, Nowrouzezahrai D, Hachisuka T. Gradient-domain photon density estimation. *Computer Graphics Forum*, 2017, 36(2): 31–38. DOI: 10.1111/cgf.13104.

[25] Ronneberger O, Fischer P, Brox T. U-Net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention—MICCAI 2015*, Navab N, Hornegger J, Wells W M, Frangi A F (eds.), Springer, 2015, pp.234–241. DOI: 10.1007/978-3-319-24574-4_28.

[26] Woo S, Park J, Lee J Y, Kweon I S. CBAM: Convolutional block attention module. In *Computer Vision—ECCV 2018*, Ferrari V, Hebert M, Sminchisescu C, Weiss Y (eds.), Springer, 2018, pp.3–19. DOI: 10.1007/978-3-030-01234-2_1.

[27] Kingma D P, Ba J. Adam: A method for stochastic optimization. In *Proc. the 3rd International Conference on Learning Representations*, May 2017.

[28] He K, Zhang X, Ren S, Sun J. Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification. In *Proc. the 2015 IEEE International Conference on Computer Vision* (*ICCV*), Dec. 2015, pp.1026–1034. DOI: 10.1109/ICCV.2015.123.

Ming-Cong Ma received his M.S. degree in computer science and technology from Shandong University, Jinan, in 2020. He currently is a Ph.D. candidate in the School of Software, Shandong University, Jinan. His research interests include machine learning and image denoising.

Lu Wang received her Ph.D. degree in computer science from Shandong University, Jinan, in 2009. She is a professor in the School of Software, Shandon University, Jinan. Her research interests include photorealistic rendering and high performance rendering.

Yan-Ning Xu received his Ph.D. degree in computer science from Shandong University, Jinan, in 2006. He is an associate professor in the School of Software, Shandong University, Jinan. His research interests include photorealistic rendering and computer aided design.

Xiang-Xu Meng received his Ph.D. degree in computer science from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, in 1998. He is a professor in the School of Software, Shandong University, Jinan. His current research interests include human-computer interaction, virtual reality, computer graphics, and visualization.