



Denoising Monte Carlo renderings via a multi-scale featured dual-residual GAN

Yifan Lu¹ · Siyuan Fu¹ · Xiao Hua Zhang² · Ning Xie¹

Accepted: 7 June 2021 / Published online: 19 June 2021

© The Author(s), under exclusive licence to Springer-Verlag GmbH Germany, part of Springer Nature 2021

Abstract

Monte Carlo (MC) path tracing causes a lot of noise on the rendered image at a low samples per pixel. Recently, with the help of inexpensive auxiliary buffers and the generative adversarial network (GAN), deep learning-based denoising MC rendering methods have been able to generate noise-free images with high perceptual quality in seconds. In this paper, we propose a novel GAN structure for denoising Monte Carlo renderings, called dual residual connection GAN. Our key insight is that the dual residual connections can improve the chance of the optimal feature selection and implicitly increase the number of potential interactions between modules. We also propose a multi-scale auxiliary features extraction method, aiming to make full use of the rich geometry and texture information of auxiliary buffers. Moreover, we adopt a spatial-adaptive block with the deformable convolution to help the network adapt to the variance in spatial texture and edge features. Compared with the state-of-the-art methods, our network has fewer parameters and less inference time, and the results surpass the previous in terms of visual effects and quantitative metrics.

Keywords Denoising Monte Carlo renderings · Generative adversarial networks · Multi-scale auxiliary features · Dual residual connections

1 Introduction

Monte Carlo (MC) path tracing [18] is a general and powerful rendering technique for simulating light transport behavior and rendering photo-realistic images in computer graphics. Due to its generality and unbiased nature, the MC path tracing method has been widely used in animation production, visual effects, and video games [20]. However, it requires tracking a large number of ray paths within each pixel to render noise-free images, resulting in consuming a lot of ren-

dering time. This problem motivates researchers to develop many denoising approaches at a reduced sample rate (e.g., 1–64 samples per pixel (spp)) with the help of auxiliary buffers (e.g., albedo, normal, and depth buffers).

Recently, ACFM [35] and DMCR [26] apply the generative adversarial network (GAN) [9] for denoising Monte Carlo renderings at an offline rate to achieve more plausible results than traditional kernel filtering [4] and CNN-based approaches [2,31,34,36].

However, we find three main limitations of these methods. **First**, most of the MC denoising network structures apply several residual blocks to build a deeper network and thus improve denoising performance [26,31,35]. However, the residual connection [13] in the previous MC denoising methods is all embedded in the residual unit, which ignores the interaction of features between different residual units. **Second**, existing works ever since ACFM [35] modulate noisy feature maps based on encoded auxiliary features. The method can achieve better denoising performance than simply concatenating auxiliary buffers with the noisy image as network input (as previous works did [2,31]) as long as they are encoded properly. Existing works typically extract the features of auxiliary buffers in the form of full resolu-

✉ Ning Xie
seanxiening@gmail.com

Yifan Lu
luyifan0821@gmail.com

Siyuan Fu
AlexHuku@hotmail.com

Xiao Hua Zhang
zhxh@cc.it-hiroshima.ac.jp

¹ Center for Future Media, Department of Computer Science and Engineering, University of Electronic Science and Technology of China (UESTC), Chengdu 611731, China

² Hiroshima Institute of Technology, Hiroshima, Japan

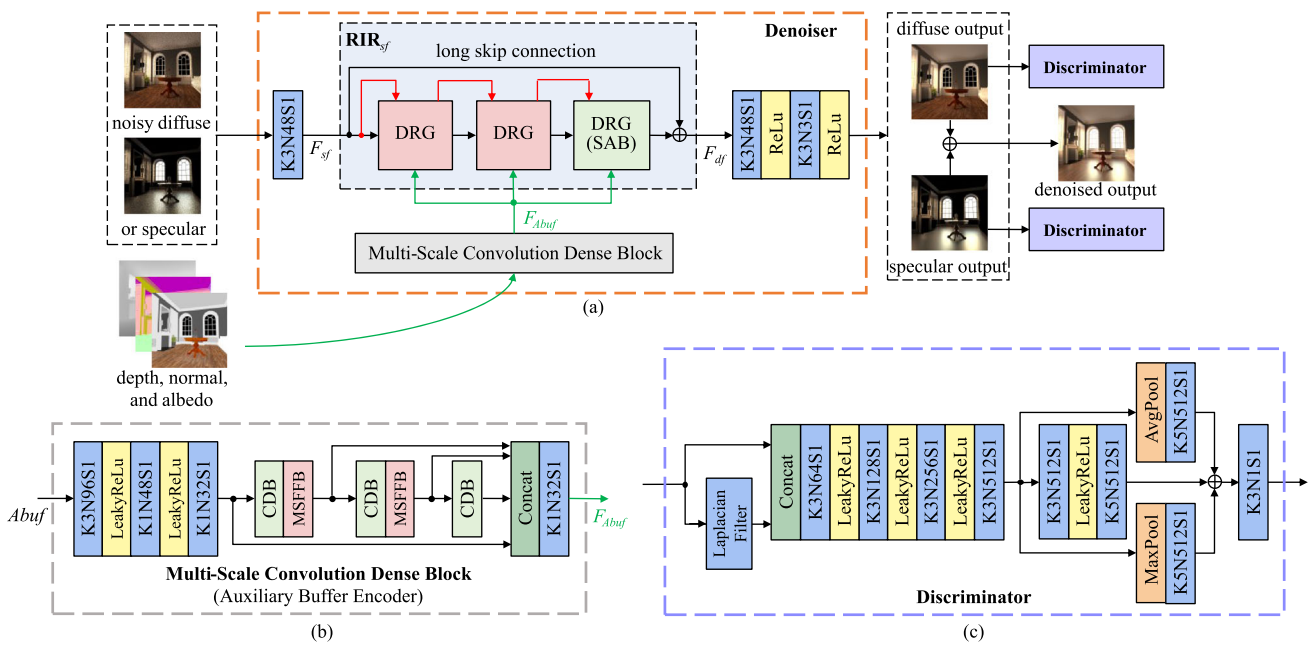


Fig. 1 The K3N48S1 represents a convolution operation where the kernel size is 3, the number of feature channels is 48, and stride is 1. **a** An overview of our network framework (DuRCGAN). Note that the denoisers and discriminators of diffuse and specular input are of different weights, respectively. The denoiser is based on a residual-in-residual (RIR) design, which stacks three dual residual groups (DRGs) and a long skip connection. The red line between the DRGs is *residual connection*

2 (see Fig. 2 and Sect. 3.1 in detail); **b** auxiliary buffer encoder network. We use a multi-scale convolution dense block (MSCDB) to extract spatially precise auxiliary features by convolution dense blocks (CDBs) and obtain a complementary set of contextual information across multiple spatial scales by multi-scale feature fusion blocks (MSFFBs). The encoded auxiliary features F_{Abuf} are used to modulate noisy features in DRGs; **c** discriminator network

tion (single scale) via several convolution layers [26,35] to achieve fine spatial details; however, operating on a single scale makes the receptive field fixed in each layer, and it is well known in the vision science that the size of the local receptive field in the same area is different [37]. **Third**, previous works often use traditional convolution operations to extract the local fixed-location features, which makes the network lack flexibility when facing low-frequency and high-frequency information simultaneously.

To address the above problems, we propose a novel adversarial approach for denoising Monte Carlo renderings, called dual residual connection GAN (**DuRCGAN**). Specifically, as illustrated in Fig. 1, we introduce the residual-in-residual (RIR) structure. The hierarchical connections inside the RIR allow the network to have more path options, which can increase the flow of information and the chance of the optimal feature selection. Moreover, we propose a multi-scale convolution dense block (MSCDB) as an auxiliary buffer encoder. It operates on full-resolution features to extract and maintain the fine spatial details of auxiliary features. During the encoding process, additional down-sampling and up-sampling layers are used to generate low-resolution features to obtain a complementary set of features across multiple spatial scales [37]. The encoded auxiliary features are used to modulate features from noisy input inside the proposed RIR

Table 1 The abbreviations we used in this section

RIR	Residual-in-residual
DRG	Dual residual group
DRB	Dual residual block
RU	Residual unit
MSCDB	Multi-scale convolution dense block
CDB	Convolution dense block
MSFFB	Multi-scale feature fusion block
CFM	Conditioned feature modulation
SKFF	Selective kernel feature fusion
SAB	Spatial-adaptive block

structure. Furthermore, we propose a spatial-adaptive block (SAB). It introduces deformable convolution [38] to help the network adapt to spatial variations between low-frequency and high-frequency features and thus recover more spatial details and textures. As shown in Fig. 6, DuRCGAN can achieve better visual results and quantitative metrics compared with previous state-of-the-art methods.

2 Related works

2.1 Learning-based Monte Carlo denoising

The key idea of denoising MC rendering is to reconstruct noise-free images from noisy input with the help of auxiliary features including albedo, normal, depth, and the corresponding variance buffers [34,39]. Recently, learning-based MC denoising approaches have leveraged deep neural networks to outperform traditional image-space methods [4].

Pixel-space reconstruction is the most common way of learning-based MC denoising. The pixel-based denoisers use the summary statistics of per-pixel sample distributions. As a pioneer, Kalantari et al. [19] used a multilayer perceptron neural network to estimate the parameters of denoising filters. Chaitanya et al. [5] proposed a recurrent neural network (RNN) to deal with image sequences at an interactive rate. Bako et al. [2] applied convolutional neural networks for predicting kernel filters. Vogels et al. [31] enriched Chaitanya et al. [5] and Bako et al.'s [2] works by considering multi-scale denoising and temporal coherence. Wong et al. [34] used several residual blocks to directly generate the noise-free images instead of predicting kernel filters [2]. Kuznetsov et al. [23] divided the denoising problem into two parts: adaptive sampling and reconstruction. Hasselgren [12] enriched Kuznetsov et al.'s [23] work by introducing multi-scale kernel prediction network and considering temporal denoising. Xu et al. [35] first applied the generative adversarial network to this mission. Moreover, they proposed the auxiliary feature conditioned modulation method to exert more additive and multiplicative interactions between the auxiliary features and noisy input. This is more effective than naively concatenating them with the noisy image as the network input. DMCR [26] enriched Xu et al.'s work [35] by introducing residual attention network and hierarchical features extraction method of auxiliary buffers. Meng et al. [27] introduced the neural bilateral grid [7] to build a light-weight network for real-time denoising.

Deep learning has also been utilized for sample-based MC denoising. This method worked on individual samples instead of pixel aggregates. Gharbi et al. [8] proposed a novel splatting method to predict per-sample splatting kernel. The kernel splats each sample onto nearby pixels to produce final results. Munkberg et al. [28] proposed a layering embedding denoising approach to speed up this operation. It separated samples into different layers and used a splatting kernel filter in each layer, respectively. However, memory requirements are substantial in the sample-based denoising method because each rendered sample has more scalar features. Meanwhile, the rendered samples still need to be averaged to pixels and processed by a pixel-based denoiser to generate features. Hence, in this paper, we focus on the design of the pixel-based denoising network structure.

In addition, Kettunen et al. [21] and Guo et al. [11] tried to reconstruct screened Poisson process for gradient-domain rendering, but it required additional input information. Vicini et al. [30] tried to consider deep Monte Carlo renderings denoising.

2.2 Generative adversarial networks

The generative adversarial network (GAN) [9] has been widely used in various image generation tasks, including image-to-image translation [32], image editing [17], and image super-resolution [33]. However, the training process of vanilla GAN is unstable because of gradient vanishing and mode collapse. Recently, several works focus on stabilizing the GAN's training and increase the sample diversity [1,10].

For MC denoising, Xu et al. [35] applied the VGG network [29] for the discriminator but failed to capture global information of the images. Moreover, DMCR [26] introduced a multi-scale PatchGAN discriminator [32] which means that no fully connected layer was used to discriminate images from coarse to fine scale.

3 Denoising network structure

Similar to previous work [35], our denoising network processes the diffuse and the specular noisy images separately and synthesizes the output of two networks to obtain the final denoised results. In this section, we elaborate on our proposed residual-in-residual (RIR) module and the multi-scale convolution dense block (Table 1).

3.1 Residual-in-residual (RIR) module

To achieve better results and make a deeper network, we introduce the residual-in-residual (RIR) module. It consists of a long skip connection and three dual residual groups (DRGs) (Fig. 1a). The long skip connection allows residual learning at a coarse level, which makes the network pay attention to learn high-frequency information. As illustrated in Fig. 2, the DRG consists of two dual residual blocks (DRBs) named as DRB^l and DRB^{l+1} and a middle skip connection to make a further step toward residual learning. In each DRB, it has two residual units (RUs) with dual residual connections (the blue and the red lines) and a short skip connection.

We now elaborate on the proposed dual residual connections. Recently, paired operations with dual residual connections have shown their effectiveness on image processing tasks [25]. In this paper, we introduce the dual residual connections into the MC denoising tasks and regard two RUs of the DRB as the paired operations. The dual residual connections consist of *residual connection-1* and *residual connection-2*. In practice, the *residual connection-1* (the blue

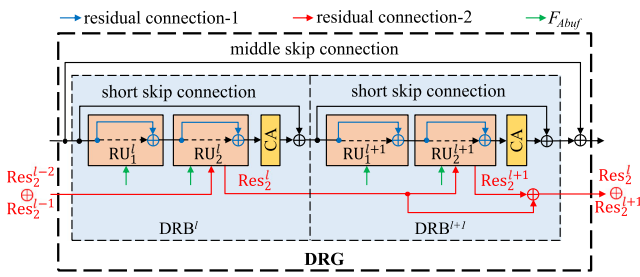


Fig. 2 Illustration of the dual residual group (DRG). DRG has two dual residual blocks (DRBs) named as DRB^l and DRB^{l+1} and a middle skip connection. Each DRB consists of two residual units (RUs), the dual residual connections (*residual connection-1* and *residual connection-2*), and a short skip connection. The F_{Abuf} produced by MSCDB is used to modulate noisy features in the RU

line in Fig. 2) is equivalent to the identity mappings in the standard residual unit [14], which can be viewed as the intra-RU residual connection. Besides *residual connection-1*, we introduce *residual connection-2* (the red line in Fig. 2) into the second RU of each DRB. As shown in Figs. 2 and 3a, let RU_2^l and RU_2^{l+1} be the second RU of DRB^l and DRB^{l+1} , respectively. Before its ReLU function, RU_2^{l+1} receives the intermediate residual (named as Res_2^l) from RU_2^l . After the ReLU function, it generates a new intermediate residual Res_2^{l+1} . To make full use of these intermediate residuals, we do element-wise addition operation on them ($Res_2^l \oplus Res_2^{l+1}$) and use its result as the input residual of the next DRG. The RU_2^l itself also benefits from this; it received the intermediate residual ($RU_2^{l-2} \oplus RU_2^{l-1}$) from the last DRG, as shown in Fig. 2. Therefore, the dual residual connections can implicitly increase the number of potential interactions between the intra-unit and inter-unit features, which can achieve better denoising results. The long, middle, and short skip connections and dual residual connections in the RIR allow the

network to have more path options, and more information can be bypassed through the multiple connections.

3.2 Multi-scale convolution dense block (MSCDB)

Auxiliary buffers are inexpensive rendering by-products, but they can provide rich geometry and texture information for noisy features, which can greatly improve denoising performance [39]. Hence, how to effectively extract these abundant features is a key problem in MC denoising.

The previous work [26] has shown the effectiveness of the convolution dense block (CDB) to extract auxiliary features. However, we find that the CDB only operates on the full-resolution (single-scale) auxiliary features, which can extract fine spatial details but fail to capture semantically reliable contextual information from multiple scales. Hence, in this paper, we propose the multi-scale convolution dense block (MSCDB) to extract and fuse diverse information from both full-resolution and low-resolution scales, as illustrated in Figs. 1b and 4. To decrease the number of network parameters, we abandoned the way of using CDB on multiple scales. Instead, we introduce a multi-scale feature fusion block (MSFFB, Fig. 3a) after CDB to obtain rich and semantically reliable contextual information while maintaining precise spatial features.

Specifically, the CDB operates on full resolution representations, and the MSFFB follows behind the CDB to fuse contextual information. MSFFB applies down-sampling operations to produce three resolution streams. In each resolution stream, we use one residual unit to extract features. Then, we apply up-sampling operations for two low-resolution feature maps to return to their full-resolution form. Motivated by Zamir et al.'s work [37], we introduce a selective kernel feature fusion (SKFF) module to aggregate features from three scales instead of simply concatenating

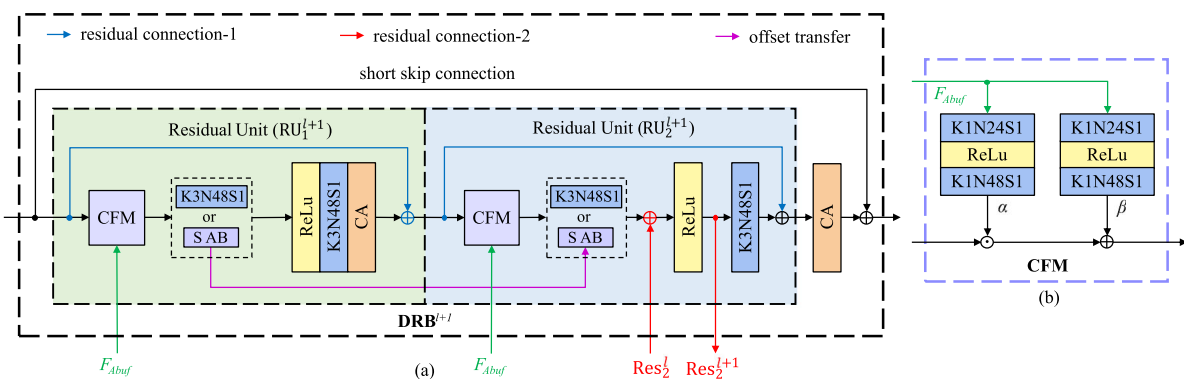


Fig. 3 Illustration of the $(l + 1)$ th dual residual block (DRB). It has two residual units (RU) with a short skip connection. In each RU, we apply the CFM [35] to modulate noisy features with auxiliary features F_{Abuf} and the channel attention (CA) [26] to exploit introduce the dual resid-

ual connections (the blue and red line) to make the network exploit the inter-channel relationship of features. We further use a spatial-adaptive block (SAB) to make the network adapt to spatial variations

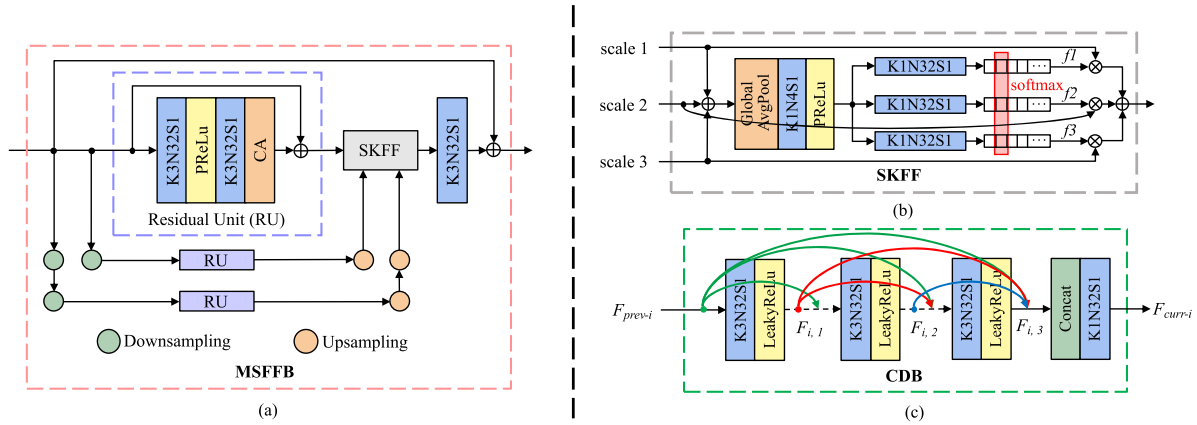


Fig. 4 The implementation of convolution dense block (CDB) and multi-scale feature fusion block (MSFFB) in the auxiliary buffer encoder network

them. The SKFF performs the element-wise addition operation on three scales features and then applies a global average pooling to squeeze the spatial dimension of the fusion features. This equals compute channel-wise statistics. Next, there is a channel-downscaling convolution layer to generate a latent vector, followed by three parallel channel-upscaling convolution layers to produce three feature descriptors. For select operation, we apply the softmax function to obtain three attention features f_1 , f_2 , and f_3 . Finally, we use f_1 , f_2 , and f_3 to recalibrate the input feature maps from three scales, respectively.

After passing through CDB and MSFFB, both full-resolution and progressive low-resolution features are extracted. We repeatedly stack them to extract deeper features. Finally, we concatenate the output of CDBs or MSFFBs and use a 1×1 convolution layer to fuse them into the final auxiliary features F_{Abuf} .

3.3 Spatial-adaptive block (SAB)

The standard convolution operation extracts the local fixed-location features, which may lead to calculating relevant and unrelated features simultaneously. As illustrated in Fig. 10b, the standard convolution operation makes the results blur at the junction of high-frequency and low-frequency information.

To address this problem, in this paper, we introduce a spatial-adaptive block (SAB) to help the network adapt to spatial changes. The core of SAB is the modulated deformable convolution (Fig. 5b) [6,38]. Compared with the standard convolution, the modulated deformable convolution can change the shapes of convolutional kernels and be formulated as:

$$y(p) = \sum_{p(i) \in N(p)} w_i \cdot x(p_i + \Delta p_i) * \Delta m_i$$

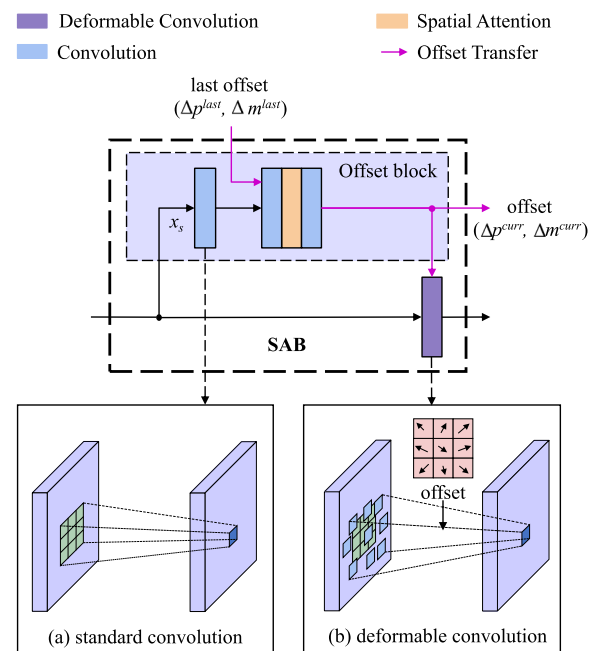


Fig. 5 The architecture of the spatial-adaptive block (SAB). The SAB consists of the offset block and deformable convolution layer. The deformable convolution uses the offset value obtained by the offset block to extract non-fixed location features

where $N(p)$ denotes the neighborhoods of location p with convolutional kernel size, and w_i and p_i denote the weight and the location in $N(p)$ (green squares shown in Fig. 5a). Δp_i and Δm_i are offset values and obtained via the offset block. Δp_i can change the location of p_i (blue squares shown in Fig. 5b), and Δm_i is the modulation scalar which lies in the range $[0, 1]$ to recalibrate features further. Hence, the modulated deformable convolution can adjust the spatial support regions, which helps the network deal with low-frequency and high-frequency features more effectively.





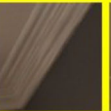

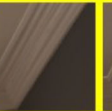
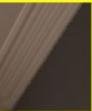


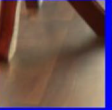
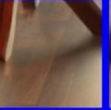
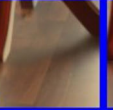












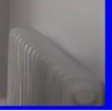

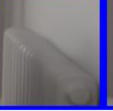
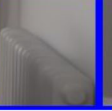

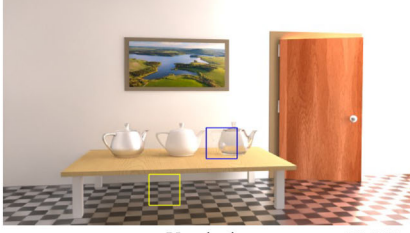

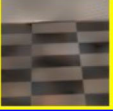






















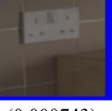
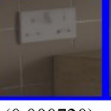

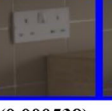

Ours		Noisy input	NFOR	KPCN	ACFM	DMCR	DuRCGAN (Ours)	Reference
								
								
The Grey & White Room	RMSE	(0.069899)	(0.004098)	(0.001349)	(0.001397)	(0.001088)	(0.000911)	
	SSIM	(0.2276)	(0.9209)	(0.9485)	(0.9457)	(0.9594)	(0.9621)	
	PSNR	(16.3160)	(30.1843)	(33.7373)	(34.2560)	(35.0955)	(35.4931)	
								
								
The White Room	RMSE	(0.014681)	(0.000879)	(0.000302)	(0.000291)	(0.000222)	(0.000199)	
	SSIM	(0.2781)	(0.9531)	(0.9640)	(0.9599)	(0.9713)	(0.9743)	
	PSNR	(21.1915)	(35.6054)	(37.6352)	(38.1307)	(39.1879)	(39.6325)	
								
								
Veach ajar	RMSE	(0.115230)	(0.003995)	(0.002845)	(0.001870)	(0.002126)	(0.001357)	
	SSIM	(0.1169)	(0.7971)	(0.8184)	(0.8332)	(0.8502)	(0.8655)	
	PSNR	(10.5093)	(27.0084)	(28.7843)	(30.6149)	(30.5849)	(32.0600)	
								
								
Country Kitchen	RMSE	(0.019842)	(0.001880)	(0.000743)	(0.000720)	(0.000606)	(0.000538)	
	SSIM	(0.3402)	(0.9526)	(0.9708)	(0.9662)	(0.9772)	(0.9784)	
	PSNR	(21.2128)	(33.3584)	(36.6166)	(37.0461)	(37.9129)	(38.1201)	

Fig. 6 We evaluate our network and compare it with the state-of-the-art methods, including NFOR [4], KPCN [2], ACFM [35], and DMCR [26], on test scene from [3] and rendered by the Tungsten renderer. For each scene, we also demonstrate two close-ups

Inspired by Chang et al. [6], in order to better estimate the current offset values, we transfer the offset values obtained in the last offset block $\{\Delta p^{\text{last}}, \Delta m^{\text{last}}\}$ to the current offset block (the purple line in Figs. 5 and 3a). Thus, we apply several standard convolution layers to extract features from the input features x_s and aggregate them with the last offset values $\{\Delta p^{\text{last}}, \Delta m^{\text{last}}\}$ to estimate current offset values $\{\Delta p^{\text{curr}}, \Delta m^{\text{curr}}\}$. Moreover, we introduce the spatial atten-

tion (SA) proposed by Zamir et al. [37] to help the offset block pay attention to spatial importance.

In order to reduce the number of network parameters and memory space occupation, we only replace the first standard convolution layer of each RU in the last DRG with the SAB (see Fig. 3a).

Table 2 The statistics of numerical performance show that our method can outperform the state-of-the-art approaches at any spp. Avg. indicates the average value over the entire test set calculated by SSIM, PSNR, or RMSE metrics, and B.P. indicates the percentage of all the best results of each method to the total test set

spp	Denoiser	SSIM \uparrow		PSNR \uparrow		RMSE \downarrow	
		Avg.	B.P. (%)	Avg.	B.P. (%)	Avg.	B.P. (%)
4	NFOR	0.8620	6.25	28.8931	6.25	0.008364	6.25
	KPCN	0.8865	0.00	30.8679	6.25	0.004411	0.00
	ACFM	0.8909	0.00	32.0561	12.50	0.003460	6.25
	DMCR	0.9134	6.25	31.4188	0.00	0.003434	6.25
	DuRCGAN (Ours)	0.9192	87.50	32.6834	75.00	0.002908	81.25
16	NFOR	0.9119	0.00	32.5877	0.00	0.003849	0.00
	KPCN	0.9206	0.00	34.9493	0.00	0.001984	0.00
	ACFM	0.9308	0.00	35.5749	6.25	0.001645	0.00
	DMCR	0.9416	6.25	35.6053	0.00	0.001689	6.25
	DuRCGAN (Ours)	0.9461	93.75	36.5493	93.75	0.001313	93.75
32	NFOR	0.9261	0.00	34.4881	0.00	0.002505	0.00
	KPCN	0.9374	0.00	36.9046	6.25	0.001339	6.25
	ACFM	0.9436	0.00	37.1799	0.00	0.001187	0.00
	DMCR	0.9518	6.25	37.4761	0.00	0.001200	6.25
	DuRCGAN (Ours)	0.9557	93.75	38.1509	93.75	0.000914	87.5

Best performance is highlighted in bold

Table 3 Average time cost and the number of parameters of each denoising approach

Method	Timing (s)	Parameter (M)	Device
NFOR	19.3	–	2.30 GHz Intel Xeon CPU
KPCN	4.6	2.25	Titan RTX
ACFM	1.1	1.53	Titan RTX
DMCR	1.1	2.05	Titan RTX
DuRCGAN (ours)	1.0	1.33	Titan RTX

Best performance is highlighted in bold

4 Experimental setup

4.1 Datasets

Training a robust generative adversarial network requires a large-scale and diverse dataset to avoid overfitting. We use the public denoising datasets [2,3] rendered by Tungsten renderer. The training datasets consist of 8 scenes, and each scene has about 200 pairs of noisy input, auxiliary buffers, and reference images rendered from different camera parameters, materials, textures, and illumination conditions. The reference images were rendered with 32,768 samples per pixel (spp), while the noisy images and corresponding auxiliary buffers were rendered with 32 spp. All training data are cropped into patches of size 128×128 by importance sampling [2]. We use albedo, normal, and depth maps as auxiliary buffers, and scale them to the same range [0.0–1.0]. We use diffuse data without albedo as the network input to preserve the texture information, and we then multiply the albedo map back to the denoised result. Before using specular or untextured diffuse RGB color buffers as the network input, we apply a logarithmic function to them to compress the high

dynamic range (HDR) of color values, i.e., $\log(1 + x)$, where x is the HDR color values.

4.2 Implementation details

The discriminator network is designed in a PatchGAN [16] style, which means that no fully connected layers are used to capture global features. We concatenate the original denoised/reference image and the image after applying a Laplacian filter, and as the input to the discriminator. This strategy lets the discriminator pay attention to the edge information of the image, while also prompting the generator to generate high-frequency details.

We implement our networks using PyTorch and train the networks on a single Titan RTX GPU. We use WassersteinGAN with a gradient penalty (WGAN-GP) [10] to stabilize the training process. We use Adam solver [22] with the default parameters and mini-batch size of 8 to train the network. The learning rate is set to $2e-4$ for both diffuse and specular branches and halved after training 5k, 10k, 15k, 20k iterations. The training time takes about 36 h for each branch. In the paper, we combine the symmetric mean abso-

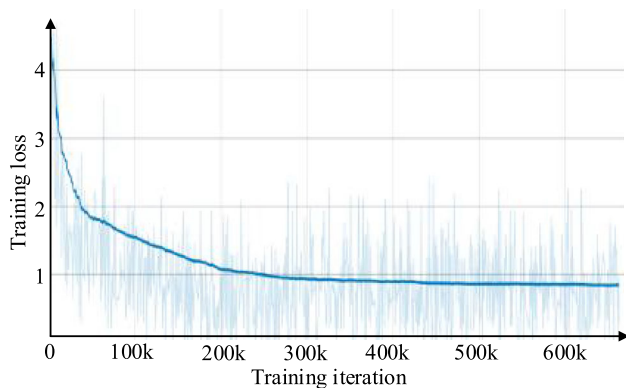


Fig. 7 Illustration of training loss

lute percentage error (SMAPE) loss and adversarial loss for training. The SMAPE loss enforces correctness at the low-frequency region, while adversarial loss focuses on high-frequency details of images. We use SMAPE instead of L1 and L2 loss since the SMAPE can stabilize to denoise HDR images [28,31]. We set the ratio between SMAPE and adversarial loss to 100:1 to make training more stable (Fig. 7).

5 Evaluation

5.1 Results

To evaluate our proposed network, we compare with four state-of-the-art offline denoising methods: NFOR [4], KPCN [2], ACFM [35], and DMCR [26]. The NFOR denoiser is the state-of-the-art regression-based method and embedded in the public Tungsten renderer, while KPCN, ACFM, and DMCR have the public model weights and training codes.

We measure the time cost and the number of parameters of each denoising approach. The time cost is averaged over all 1280×720 test images. The statistics of this information are presented in Table 3. The NFOR only has CPU implementation and takes 19.3 s on the 2.30 GHZ Intel Xeon CPU, and the KPCN takes 4.6 s since the kernel filter needs to calculate the final result pixel by pixel. Our method uses about 1.33 M parameters, which is fewer than the DMCR and ACFM but is still on a par with or even better than the previous method in terms of visual effects and quantitative metrics (see Table 2 and Fig. 6).

In addition, we choose three image quality metric methods, including relative MSE (RMSE), PSNR, and SSIM to compare quantitative results. To fairly compare with the above methods, we use their public codes to retrain the network on Tungsten training datasets. We design the experiments on noisy input images with different samples per pixel, including 4, 16, and 32 samples per pixel. Note that we only

train the network on 32 samples per pixel, instead of training a unique network for each spp.

Figure 6 shows the comparison of denoising results on four representative test scenes, including the Gray and White Room, the White Room, the Veach ajar, and the Country Kitchen. For each scene, we also show close-ups of the yellow and blue squares. NFOR leaves some splotchy artifacts on the low-frequency region of the image, which is caused by the lack of global information in the process of generating filter weights, and the edge details on the ceiling of the room (e.g., close-ups of the yellow squares in the Gray and White room) have also been softened. KPCN successfully denoises most low-frequency areas but fails to capture high-frequency ones since only stacking the standard convolution operations makes the network lack flexibility when facing different features. Both ACFM and DMCR try to make the network recover high-frequency information as much as possible, but they may produce smooth results in the junction of high-frequency and low-frequency areas (e.g., close-ups of the yellow squares in the Veach ajar). Our method performs on a par or even better than previous works in terms of visual effects and quantitative metrics. Similar to ACFM [35], we calculate the average denoising performance of these methods on the entire test set, as shown in Table 2. In addition, as illustrated in Fig. 8, we also apply the denoiser network trained at 32 spp to denoise 4 spp and 16 spp noisy images.

6 Analysis

6.1 Ablation on multi-scale convolution dense block

As presented in Sect. 3.2, we propose a multi-scale convolution dense block (MSCDB) to aggregate both high-resolution and progressive low-resolution hierarchical features of auxiliary buffers. This can preserve high-resolution and spatially precise auxiliary features as well as receive abundant contextual information from low-resolution representations.

To demonstrate its effectiveness, we only use five CDBs to operate on single-scale auxiliary features and ensure that it has similar parameters with MSCDB. Due to the lack of contextual complementary information, the black shadow on the pick wall and the edge of the ceiling are softened (Fig. 9b and d). Moreover, similar to Zamir et al. [37], we also analyzed the influence of the SKFF module. We concatenate the features from three scales and then mix them through two convolution operations with 1×1 kernel size. The SKFF module has only about 500 parameters, while the concatenating method will lead to more than 4000 parameters. However, the SKFF module used in MSCDB to fuse multi-scale features can achieve better visual effects and quantitative metrics than concatenating method (Fig. 9c and d).

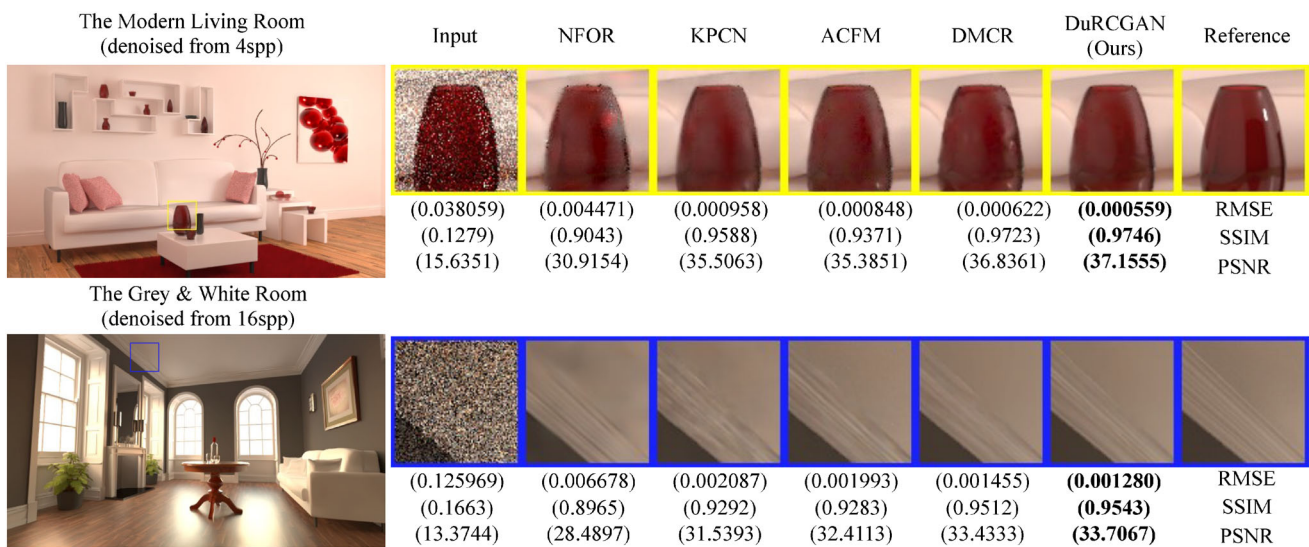


Fig. 8 We apply the denoiser network trained at 32 spp to denoise 4 spp and 16 spp noisy images. Our method can still recover more high-frequency information and achieve better denoising performance than previous methods

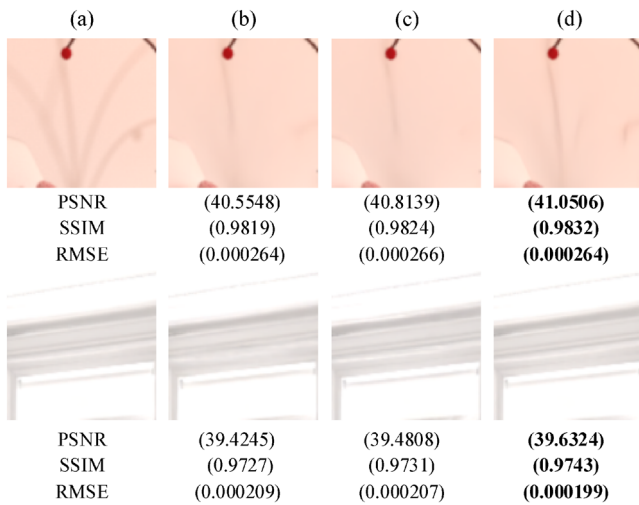


Fig. 9 Comparisons of different auxiliary features encoder. From left to right: a reference; b training with 5 CDBs; c simply concatenate multiple scales for training; d training with DuRCGAN (Ours)

6.2 Ablation on residual-in-residual (RIR)

To demonstrate the effect of our proposed RIR structure, we conduct ablation experiments on long, short, middle skip connections and residual connection-2, respectively (the residual connection-1 in the residual unit has been proven effective in various image processing tasks, so we only focus on the impact of residual connection-2). Table 4 shows the average metrics of the network without a specific connection on the entire test set. When residual connection-2 is removed, all the three metric values worsen no matter whether other skip connections are used or not. This indicates that the network without residual connection-2 cannot achieve better

denoising performance since the residual connection-2 can increase the number of potential interactions between residual units and the chance of the optimal feature selection. Moreover, residual connection-2 does not increase the number of network parameters.

In addition, using the long, middle, and short skip connections in the RIR structure can further improve the overall performance of the denoising networks. Among them, using short skip connections in the RIR has the most obvious improvement. A direct reason is that in our proposed RIR structure, the number of short skip connections is more than the other two connection types, and this also indirectly proves the superiority of multiple path options strategies of the network. These comparisons show that long, middle, short skip connections, and dual residual connections are essential for denoising networks. They also demonstrate the effectiveness of our proposed RIR structure.

6.3 Ablation on spatial-adaptive block

Spatial-adaptive block (SAB) introduces deformable convolution to make the network adapt to different spatial changes, which can improve its flexibility. In order to show the significance of deformable convolution, we conduct an ablation study on replacing deformable convolution in the last DRG with the standard convolution layers. As shown in Fig. 10b, it produces artifacts and blur at the junction of low-frequency and high-frequency information and failed to restore reflected illumination. In addition, to demonstrate the influence of reusing last offset values in SAB, we also remove the offset transfer between SAB (the purple line in Fig. 3) and only use the noisy features to estimate the offsets values. Fig-

Table 4 Ablation study of different RIR components

Long skip connection		✓	✓	✓	✓	✓	✓	✓	✓
Middle skip connection	✓		✓			✓		✓	✓
Short skip connection	✓			✓			✓	✓	✓
Residual connection-2	✓				✓	✓	✓		✓
SSIM	0.9531	0.8747	0.9484	0.9554	0.9528	0.9550	0.9533	0.9486	0.9557
PSNR	37.9483	32.1177	37.5297	37.4995	37.7982	37.6621	37.9892	37.6367	38.1509
RMSE	0.000990	0.003122	0.001096	0.001005	0.001048	0.001051	0.001051	0.001107	0.000914

Best performance is highlighted in bold

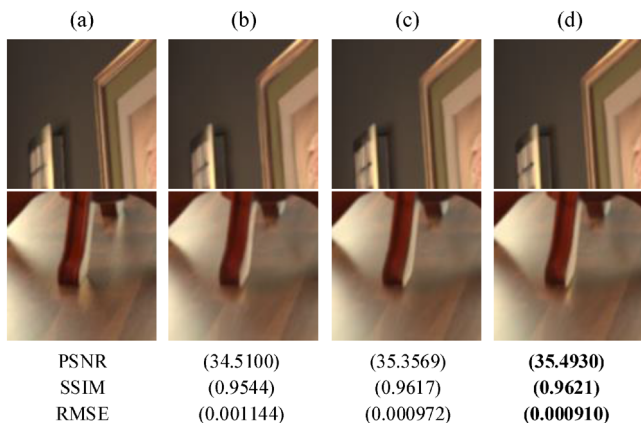


Fig. 10 Ablation on spatial-adaptive block. From left to right: **a** reference; **b** training without deformable convolution; **c** training without offset transfer; **d** training with DuRCGAN-Full (Ours)

ure 10c and d shows that training without offset transfer may produce unreal details in the edge of the frame and shadows. Therefore, the SAB can help the network to adapt to changes in spatial edges and textures.

6.4 Discussion

6.4.1 Limitations of training datasets

Large-scale and diverse datasets are essential to train a robust deep learning-based network. Our denoiser may produce poor performance and artifacts on some scenes due to lack of some special effects in the training datasets, including fog, motion blur, depth of field, smoke, etc. Figure 11 shows the limitations of our method on unknown effects, but our method can still recover some fine detail compared with previous approaches. Therefore, we would like to enlarge the training dataset to adapt to a wider range of rendering effects in the future. To make a further step, we will also extend Monte Carlo denoising to unsupervised learning like previous works [11,24], because rendering large-scale noise-free images (e.g., 16k, 32k) consumes a lot of time.

6.4.2 Adopt different denoiser structures for diffuse and specular noise colors

Considering that diffuse and specular have different noise characteristics, in this paper, we separate them but use the same network structure for training. However, specular buffer may contain extreme noise since the specular light paths are difficult to sample [27], and the albedo buffer may provide little information for specular noisy colors. It would be interesting to apply different network structures to diffuse and specular parts to adapt to this limitation (Fig. 12).

6.4.3 Multi-scale denoising structure

According to the improvement shown by the multi-scale convolution dense block (MSCDB), we believe that it would be beneficial to further introduce the multi-scale structure into the single-scale denoising network. Compared with a network that only uses a single-scale structure, the combination of single-scale and multi-scale structures can better help the network collect contextual information to generate semantically reliable denoising results, but it leads to an increase in network parameters and inference time. The OpenImageDenoise framework [15] has been widely used as a lightweight multi-scale structure (U-net), but it has not been operated on full-resolution features and is unable to capture spatial details. Hence, we will try to combine our single-scale denoiser with U-net in the future to obtain spatially accurate details and semantically reliable results.

6.4.4 Temporal denoising

In this paper, we focus on single-frame denoising at an offline rate. However, for 3D games, virtual reality, and other real-time applications, we would like to study how to denoise temporal sequences interactively at 1 spp with a generative adversarial network.

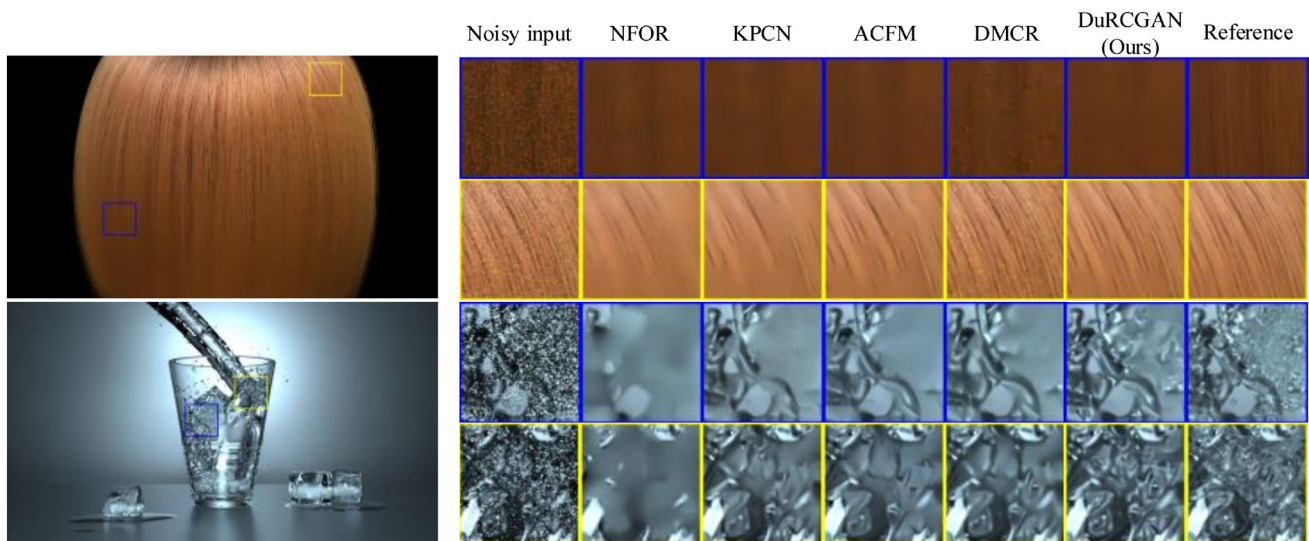


Fig. 11 The failure cases. Due to the inconsistency of the data distribution between the training and testing datasets, it is still difficult to recover high-frequency information for the hair and effects of water



Fig. 12 Comparisons of Intel® Open Image Denoise framework and our method

7 Conclusion

We have proposed a novel GAN structure (DuRCGAN) for denoising Monte Carlo renderings. It has fewer network parameters and better denoising performance than the state-of-the-art methods.

We proposed a multi-scale convolution dense block to exploit diverse features of auxiliary buffers. It not only maintains the spatial details at high resolution but also explores contextual information at low resolution. We also proposed the dual residual connections in the residual in residual structure to build a deeper network and increase the number of potential interactions between residual units, which increases the flexibility of the network and allows it to have more path options. Moreover, we further propose a spatial-adaptive block by introducing the deformable convolution to adapt to the spatial variations in textures and edges. Although our method has fewer network parameters and inference time than previous state-of-the-art methods, a comprehensive

experimental evaluation proves that our network structure is more robust and efficient.

Funding This work is part of the research supported by the National Nature Science Foundation of China under Grant No. 61602088, No. This work is part of the research supported by the Sichuan Provincial NSFC (No. 2018JY0528), the Fundamental Research Funds for the Central Universities No. Y03019023601008011, the interactive Technology Research Fund of the Research Center for Interactive Technology Industry, School of Economics and Management, Tsinghua University (No. RCITI2021T006) and sponsored by TiMi L1 Studio of Tencent corporation.

References

1. Arjovsky, M., Chintala, S., Bottou, L.: Wasserstein GAN. [arXiv:1701.07875](https://arxiv.org/abs/1701.07875) (2017)
2. 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. Graph.* **36**(4), 97–1 (2017)
3. Bitterli, B.: Rendering resources (2016). <https://benedikt-bitterli.me/resources/>
4. Bitterli, B., Rousselle, F., Moon, B., Iglesias-Guitián, J.A., Adler, D., Mitchell, K., Jarosz, W., Novák, J.: Nonlinearly weighted first-order regression for denoising Monte Carlo renderings. In: *Computer Graphics Forum*, vol. 35, pp. 107–117. Wiley Online Library (2016)
5. 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 auto-encoder. *ACM Trans. Graph. (TOG)* **36**(4), 1–12 (2017)
6. Chang, M., Li, Q., Feng, H., Xu, Z.: Spatial-adaptive network for single image denoising. [arXiv:2001.10291](https://arxiv.org/abs/2001.10291) (2020)
7. Gharbi, M., Chen, J., Barron, J.T., Hasinoff, S.W., Durand, F.: Deep bilateral learning for real-time image enhancement. *ACM Trans. Graph. (TOG)* **36**(4), 1–12 (2017)

8. Gharbi, M., Li, T.M., Aittala, M., Lehtinen, J., Durand, F.: Sample-based Monte Carlo denoising using a kernel-splatting network. *ACM Trans. Graph. (TOG)* **38**(4), 1–12 (2019)
9. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. *Adv. Neural Inf. Process. Syst.* **27**, 2672–2680 (2014)
10. Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., Courville, A.C.: Improved training of Wasserstein GANs. In: *Advances in Neural Information Processing Systems*, pp. 5767–5777 (2017)
11. 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. Graph. (TOG)* **38**(6), 1–13 (2019)
12. Hasselgren, J., Munkberg, J., Salvi, M., Patney, A., Lefohn, A.: Neural temporal adaptive sampling and denoising. In: *Computer Graphics Forum*, vol. 39, pp. 147–155. Wiley Online Library (2020)
13. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778 (2016)
14. He, K., Zhang, X., Ren, S., Sun, J.: Identity mappings in deep residual networks. In: *European Conference on Computer Vision*, pp. 630–645. Springer (2016)
15. Intel open image denoise. <https://www.openimagedenoise.org/documentation.html>
16. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1125–1134 (2017)
17. Jo, Y., Park, J.: Sc-fegan: Face editing generative adversarial network with user's sketch and color. In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1745–1753 (2019)
18. Kajiya, J.T.: The rendering equation. In: *Proceedings of the 13th Annual Conference on Computer Graphics and Interactive Techniques*, pp. 143–150 (1986)
19. Kalantari, N.K., Bako, S., Sen, P.: A machine learning approach for filtering Monte Carlo noise. *ACM Trans. Graph.* **34**(4), 122–1 (2015)
20. Keller, A., Fascione, L., Fajardo, M., Georgiev, I., Christensen, P., Hanika, J., Eisenacher, C., Nichols, G.: The path tracing revolution in the movie industry. In: *ACM SIGGRAPH 2015 Courses*, pp. 1–7 (2015)
21. Kettunen, M., Härkönen, E., Lehtinen, J.: Deep convolutional reconstruction for gradient-domain rendering. *ACM Trans. Graph. (TOG)* **38**(4), 1–12 (2019)
22. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. [arXiv:1412.6980](https://arxiv.org/abs/1412.6980) (2014)
23. Kuznetsov, A., Kalantari, N.K., Ramamoorthi, R.: Deep adaptive sampling for low sample count rendering. In: *Computer Graphics Forum*, vol. 37, pp. 35–44. Wiley Online Library (2018)
24. Lehtinen, J., Munkberg, J., Hasselgren, J., Laine, S., Karras, T., Aittala, M., Aila, T.: Noise2noise: learning image restoration without clean data. [arXiv:1803.04189](https://arxiv.org/abs/1803.04189) (2018)
25. Liu, X., Suganuma, M., Sun, Z., Okatani, T.: Dual residual networks leveraging the potential of paired operations for image restoration. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7007–7016 (2019)
26. Lu, Y., Xie, N., Shen, H.T.: DMCR-GAN: Adversarial denoising for Monte Carlo renderings with residual attention networks and hierarchical features modulation of auxiliary buffers. In: *SIGGRAPH Asia 2020 Technical Communications*, pp. 1–4 (2020)
27. Meng, X., Zheng, Q., Varshney, A., Singh, G., Zwicker, M.: Real-time Monte Carlo denoising with the neural bilateral grid (2020)
28. Munkberg, J., Hasselgren, J.: Neural denoising with layer embeddings. In: *Computer Graphics Forum*, vol. 39, pp. 1–12. Wiley Online Library (2020)
29. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. [arXiv:1409.1556](https://arxiv.org/abs/1409.1556) (2014)
30. Vicini, D., Adler, D., Novák, J., Rousselle, F., Burley, B.: Denoising deep Monte Carlo renderings. In: *Computer Graphics Forum*, vol. 38, pp. 316–327. Wiley Online Library (2019)
31. Vogels, T., Rousselle, F., McWilliams, B., Röthlin, G., Harvill, A., Adler, D., Meyer, M., Novák, J.: Denoising with kernel prediction and asymmetric loss functions. *ACM Trans. Graph. (TOG)* **37**(4), 1–15 (2018)
32. Wang, T.C., Liu, M.Y., Zhu, J.Y., Tao, A., Kautz, J., Catanzaro, B.: High-resolution image synthesis and semantic manipulation with conditional GANs. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8798–8807 (2018)
33. Wang, X., Yu, K., Wu, S., Gu, J., Liu, Y., Dong, C., Qiao, Y., Change Loy, C.: Esrgan: enhanced super-resolution generative adversarial networks. In: *Proceedings of the European Conference on Computer Vision (ECCV)* (2018)
34. Wong, K.M., Wong, T.T.: Deep residual learning for denoising Monte Carlo renderings. *Comput. Vis. Med.* **5**(3), 239–255 (2019)
35. Xu, B., Zhang, J., Wang, R., Xu, K., Yang, Y.L., Li, C., Tang, R.: Adversarial Monte Carlo denoising with conditioned auxiliary feature modulation. *ACM Trans. Graph.* **38**(6), 224–1 (2019)
36. Yang, X., Wang, D., Hu, W., Zhao, L.J., Yin, B.C., Zhang, Q., Wei, X.P., Fu, H.: Demc: A deep dual-encoder network for denoising Monte Carlo rendering. *J. Comput. Sci. Technol.* **34**(5), 1123–1135 (2019)
37. Zamir, S.W., Arora, A., Khan, S., Hayat, M., Khan, F.S., Yang, M.H., Shao, L.: Learning enriched features for real image restoration and enhancement. [arXiv:2003.06792](https://arxiv.org/abs/2003.06792) (2020)
38. Zhu, X., Hu, H., Lin, S., Dai, J.: Deformable convnets v2: More deformable, better results. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 9308–9316 (2019)
39. Zwicker, M., Jarosz, W., Lehtinen, J., Moon, B., Ramamoorthi, R., Rousselle, F., Sen, P., Soler, C., Yoon, S.E.: Recent advances in adaptive sampling and reconstruction for Monte Carlo rendering. In: *Computer Graphics Forum*, vol. 34, pp. 667–681. Wiley Online Library (2015)

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



Yifan Lu received the B.Ec. degree in instructional technology from the Nanchang Hangkong University, NanChang, China, in 2019. Currently, he is studying for M. Eng. at the University of Electronic Science and Technology of China. His current research interests include computer graphics, game engine and deep learning.



Siyuan Fu is currently studying for his B.S. degree at University of Electronic Science and Technology of China. His research interest falls in image processing, rendering technics, and deep learning.



Ning Xie received the ME and Ph.D. degrees from the Department of Computer Science, Tokyo Institute of Technology, Tokyo, Japan, in 2009 and 2012, respectively. In 2012, he was appointed as a research associate in the same institute. From 2017, he is an associate professor in the School of Computer Science and Engineering, UESTC. His research interests include computer graphics, game engine, and the theory and application of artificial intelligence and machine learning. His research

is supported by research grants including NSFC (China), MOE (China), CREST(Japan) and The Ministry of Education, Culture, Sports, Science and Technology(Japan).



Xiaohua Zhang received a Ph.D. degree in information science and engineering from the Tokyo Institute of Technology, Japan, in 2000. After working at NHK Engineering System Inc. from 2000 to 2003, he joined the faculty of the Hiroshima Institute of Technology as a professor. His research interests include computer graphics, image processing, computer vision, pattern recognition, and machine learning. He is a member of the IEEE, IEICE, ITE, IIEEJ, and IIAE.