Adversarial Monte Carlo Denoising with Conditioned Auxiliary Feature Modulation

BING XU, KooLab, Kujiale, China
JUNFEI ZHANG, KooLab, Kujiale, China
RUI WANG*, State Key Laboratory of CAD & CG, Zhejiang University, China
KUN XU, BNRist, Department of Computer Science and Technology, Tsinghua University, China
YONG-LIANG YANG, University of Bath, UK
CHUAN LI, Lambda Labs Inc, USA
RUI TANG*, KooLab, Kujiale, China

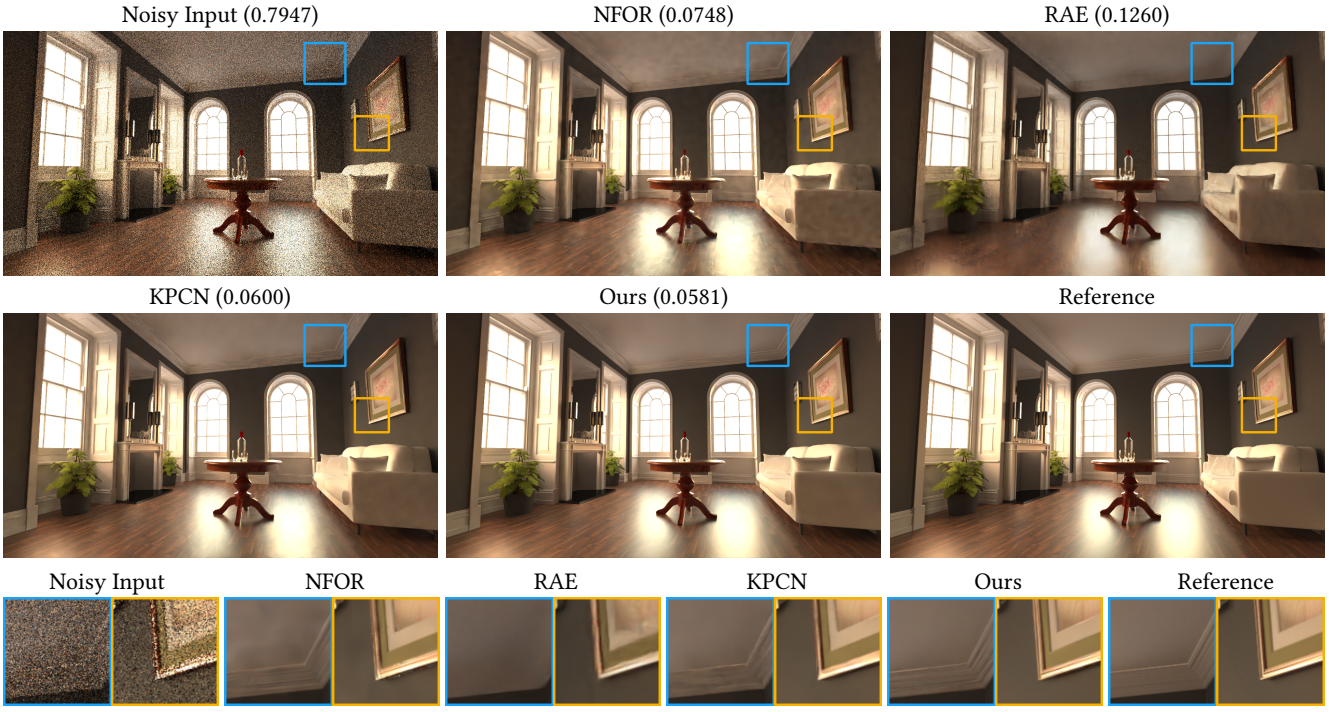

Fig. 1. Left to right; top to bottom: (a) 32 spp noisy image generated using a path tracer; (b) NFOR [Bitterli et al. 2016]; (c) Recurrent autoencoder [Chaitanya et al. 2017]; (d) KPCN [Bako et al. 2017] (e) our adversarial MC denoiser; (f) reference path-traced image with 32k spp. Numbers above indicate metric (1-SSIM).

*The co-corresponding authors: Rui Wang, rwang@cad.zju.edu.cn; Rui Tang, ati@qunhemail.com.

Authors' addresses: BING XU, KooLab, Kujiale, China, xubinggl@gmail.com; JUNFEI ZHANG, KooLab, Kujiale, China, ahui@qunhemail.com; RUI WANG, State Key Laboratory of CAD & CG, Zhejiang University, China, rwang@cad.zju.edu.cn; KUN XU, BNRist, Department of Computer Science and Technology, Tsinghua University, China, xukun@tsinghua.edu.cn; YONG-LIANG YANG, University of Bath, UK, yy753@bath.ac.uk; CHUAN LI, Lambda Labs Inc, USA, c@lambdalabs.com; RUI TANG, KooLab, Kujiale, China, ati@qunhemail.com.

Denoising Monte Carlo rendering with a very low sample rate remains a major challenge in the photo-realistic rendering research. Many previous works, including regression-based and learning-based methods, have been explored to achieve better rendering quality with less computational cost. However, most of these methods rely on handcrafted optimization objectives, which lead to artifacts such as blurs and unfaithful details. In this paper, we present an adversarial approach for denoising Monte Carlo rendering. Our key insight is that generative adversarial networks can help denoiser networks to produce more realistic high-frequency details and global illumination by learning the distribution from a set of high-quality Monte Carlo path tracing images. We also adapt a novel feature modulation method to utilize auxiliary features better, including normal, albedo and depth. Compared to previous state-of-the-art methods, our approach produces a better reconstruction of

0730-0301/2019/11-ART224 $15.00
https://doi.org/10.1145/3355089.3356547

the Monte Carlo integral from a few samples, performs more robustly at different sample rates, and takes only a second for megapixel images.

CCS Concepts: • **Computing methodologies → Computer graphics**; *Rendering*; Ray tracing.

Additional Key Words and Phrases: path tracing, Monte Carlo denoising, adversarial learning, feature modulation

ACM Reference Format:
BING XU, JUNFEI ZHANG, RUI WANG, KUN XU, YONG-LIANG YANG, CHUAN LI, and RUI TANG. 2019. Adversarial Monte Carlo Denoising with Conditioned Auxiliary Feature Modulation. *ACM Trans. Graph.* 38, 6, Article 224 (November 2019), 12 pages. https://doi.org/10.1145/3355089.3356547

1 INTRODUCTION

Along with the rapid improvements in hardware and gradually increasing perceptual demands of users, Monte Carlo path tracing is becoming more popular in movie production and video games due to its generality and unbiased nature [Keller et al. 2015; Zwicker et al. 2015]. However, its high estimator variance and low convergence rate motivate researchers to investigate efficient denoising approaches at reduced sample rates with the help of inexpensive by-products (e.g., feature buffers). In the past few years, regression-based kernel filtering approaches [Bitterli et al. 2016; Moon et al. 2014] and learning-based methods [Bako et al. 2017; Chaitanya et al. 2017; Kalantari et al. 2015; Vogels et al. 2018] have achieved great success. In particular, the deep learning based methods have achieved more plausible denoising results, since they effectively leverage convolutional neural networks to break the limitation of only utilizing information from pixel sets in specific images. However, based on our practice of employing the state-of-the-art methods, we found that nearly all of them rely on handcrafted optimization objectives like MSE or MAPE loss which do not necessarily ensure perceptually plausible results. Fig. 1 shows some typical cases where recent works [Bako et al. 2017; Chaitanya et al. 2017] have struggled to handle extremely noisy regions as in high-frequency area thus led to over-smoothed output with approximately correct colors. Our primary focus is to reconstruct the visually convincing global illumination as previous approaches while recovering high-frequency details as much as possible.

Generative adversarial networks (GANs) [Goodfellow et al. 2014] have recently demonstrated remarkable progress in various areas, including image generation [Radford et al. 2015], image translation [Huang et al. 2018; Isola et al. 2017; Wang et al. 2018a] and unsupervised domain adaptation [Zhu et al. 2017]. GANs exhibit a strong ability to model specific data distribution and enable computers to sample novel realistic images from it. Likewise, we contend that physically-based renderers share similar targets to image synthesis, yet are generally subject to more restrictions, or in other words, enjoy more favorable conditions. This is due to the fact that renderers already have a complete description of the 3D scene geometry, the materials and textures, the camera, and the lighting conditions. Each ray can be seen as a sample on the light transport distribution in Monte Carlo rendering. Hence, from another point of view, the auxiliary buffer guided image-space denoising problem can be interpreted as a conditional generative problem by model-learning on a large-scale training set.

Motivated by the above, we propose an adversarial approach to evaluate the reconstruction by leveraging Wasserstein distance [Arjovsky et al. 2017] to measure perceptual similarity, which can be interpreted as the distance between the denoised and ground truth distributions. Wasserstein distance generally performs better than *KL*/*JS* divergence, as it provides smoother measurement. The proposed method effectively learns the statistics of the ground truth images rendered with a considerably high sample rate, based on deep feature representation. We also propose a method to utilize better the inexpensive rendering by-products, such as per-pixel normal, albedo and depth [Zwicker et al. 2015], which have been proven to contain rich information and effectively guide the image filtering process. We demonstrate the advantage of our framework by comprehensively comparing it with the state-of-the-arts on publicly available datasets, and show the effectiveness of our design choices via a thorough analysis.

Overall our work makes two major contributions: 1) the first adversarial learning framework for Monte Carlo denoising that outperforms the state-of-the-art methods, and 2) a novel conditioned auxiliary feature modulation method that better utilizes feature information at the pixel level.

2 RELATED WORK

A comprehensive review on Monte Carlo (MC) denoising and generative adversarial network (GAN) is beyond the scope of this paper, with both topics having been extensively studied in the field. Hence, we focus on the most relevant to our work. Regarding the former, we discuss image-space auxiliary feature guided denoising approaches, while for the latter, we focus on GANs for image reconstruction. We also review network conditioning and scattering effects decomposition as they are germane to the design choices of our framework.

Image space Monte Carlo denoising. As explained in a recent survey [Zwicker et al. 2015], the key idea of Monte Carlo denoising is to minimize the reconstruction error by selecting and tuning appropriate filters that model the relationship between the noisy input and denoised output. Different types of filters have been proposed such as zero-order [Moon et al. 2014], first-order [Bitterli et al. 2016], and even high-order ones [Moon et al. 2016]. Within this process, further guidance can be provided by the inexpensive rendering by-products to improve the results, such as edge-stopping functions including albedo, normal, depth information [Dammertz et al. 2010; Kalantari et al. 2015; Li et al. 2012; McCool 1999; Sen and Darabi 2012], and shading-related information including virtual flash image [Moon et al. 2013], visibility and an ambient occlusion map [Kalantari et al. 2015; Rousselle et al. 2013].

While traditional regression-based approaches have achieved impressive results by adopting models in different forms, they are restricted by the available neighborhood pixels relevant to the specific form of the filter. Recent state-of-the-art approaches have leveraged supervised learning to outperform traditional ones based on empirical rules. As a pioneer, Kalantari et al. [2015] used machine learning to estimate the weights of the filter automatically. Bako et al. [2017] used convolutional neural networks to infer not just the filter weights but also the form of a more complex filter kernel itself. Chaitanya et al. [2017] utilized a recurrent neural network to force

the temporal coherence, enabling interactive denoising for real-time applications. Vogels et al. [2018] extended the kernel prediction strategy in the work [Bako et al. 2017] by also considering temporal coherence at multiple scales. In concurrent works, Kettunen et al. [2019] attempted reconstruction for gradient-domain rendering, whilst Gharbi et al. [2019] utilized raw Monte Carlo samples as high-order statistics and a novel splatting approach to achieve better results with larger computational cost and storage space though.

Scattering effects decomposition. Instead of denoising the final noisy color image, there are works having obtained better performance by considering different scattering effects during light transport simulation. Zimmer et al. [2015] first proposed a general decomposition framework to reduce artifacts caused by conflicting light transport phenomena. Bako et al. [2017] built on this framework to separate and denoise diffuse and specular components respectively before reconstructing the final output image. This leads to considerable improvements in MC denoising. In addition, Bauszat et al. [2011] achieved better global illumination filtering by splitting the integral in the render equation into direct and indirect parts due to the different characteristics therein.

Generative adversarial network for image reconstruction. Goodfellow et al. [2014] introduced the generative adversarial network (GAN) as a competitive game between a generator and a discriminator. While the goal of the generator is to fool the discriminator by generating perceptually convincing images, the discriminator is trying to distinguish the generated outputs from the real targets. GANs are known for high-quality image generation in spite of the gradient vanishing and mode collapse problem of the vanilla version. There has been extensive work researching on the various ways to stabilize the adversarial training [Arjovsky et al. 2017; Gulrajani et al. 2017; Radford et al. 2015]. GANs are also widely used to help improve perceptual quality for image restoration problems such as image super-resolution [Ledig et al. 2017], Gaussian denoising [Divakar and Venkatesh Babu 2017; Galteri et al. 2017] or blind denoising [Chen et al. 2018], deblurring [Kupyn et al. 2018], etc.

Network conditioning. Many works rely on prior information as a condition to help address various ill-posed problems like image generation and image restoration. Such priors include depths for image dehazing [He et al. 2010] , semantic masks for image translation [Isola et al. 2017], edge features for image upsampling [Fattal 2007], etc. Our auxiliary feature guided MC denoising falls into the same category. We are inspired by previous studies on conditional network normalization. These techniques have shown effectiveness on image style transfer [Dumoulin et al. 2017; Ghiasi et al. 2017; Huang and Belongie 2017] and visual reasoning [Perez et al. 2017]. The key idea is applying a conditioned function to generate parameters for feature-wise affine transformation in batch normalization. Perez et al. [2018] proved that the feature modulation layer does not have to be applied after a normalization one. This technique has been further extended to spatial feature modulation to condition network on semantic information [Park et al. 2019; Wang et al. 2018b]. The ability to preserve spatial information is crucial for low-level tasks, so as for MC denoising.

3 PROBLEM BACKGROUND

In classic Monte Carlo path tracing, the outgoing radiance c of a pixel is approximated by the sum of the contributions from N path samples in the path space Ω:

$$c = \int_{\Omega} f(x)dx \approx \frac{1}{N}\sum_{i=1}^{N}\frac{f(x_i)}{p(x_i)} \tag{1}$$

where $f(x_i)$ and $p(x_i)$ denote the radiance contribution and the probability of the i-th sampled path on the pixel, respectively.

The above general Monte Carlo integration produces highly noisy images at a small sample rate; and quadratically more samples are needed to linearly reduce the variance even for trivial scenes, making the rendering process rather costly [Bauszat et al. 2011]. This inherently motivates MC denoising to achieve a plausible rendering quality within a limited time budget. Previous joint-filtering based algorithms indicated that auxiliary feature buffers can be obtained directly as by-products of conventional rendering process and provide helpful guidance for post-processing image-space denosing. The key is how to define the mathematical model of the filter and how to optimize its parameters [Zwicker et al. 2015].

With the advances of convolutional neural networks, recent works focus on how to learn a general relationship between the noisy input and the denoised output with the help of the low-cost feature buffers, breaking the limitations of conventional approaches with a fixed form of the filtering model. Although such learning-based approaches have gained better performance due to the generality of the CNN-based model, there are still two major issues to be considered in practice:

- **i)** The loss function should be able to better reflect the perceptual quality of the denoised image with respect to the ground truth. Recent works empirically define loss function based on typical image-space metrics such as L1 and L2, which tend to produce blurry results. This motivates us to employ an effective adversarial mechanism with the general loss function to enhance the denoising further(see Section 4.3).
- **ii)** The auxiliary features should be efficiently exploited to recover both the high-frequency and low-frequency shading effects. Previous methods simply concatenate all the feature channels with noisy color as input, providing only point-wise biasing. We further incorporate a more advanced conditioned auxiliary feature modulation strategy that allows features to take effect at the pixel level, leading to fine-grained denoising result (see Section 4.2).

4 ADVERSARIAL MONTE CARLO DENOISING

In this section, we elaborate on how to denoise Monte Carlo renderings in an adversarial manner with a denoising network as well as a critic network. The former network allows expressing complex nonlinear relationship from input to output; and the latter ensures the indistinguishable quality between the output and the ground truth. Besides, the auxiliary feature conditioned modulation is incorporated in the former, providing more guidance potential for filtering the noisy input through additive and multiplicative interactions.

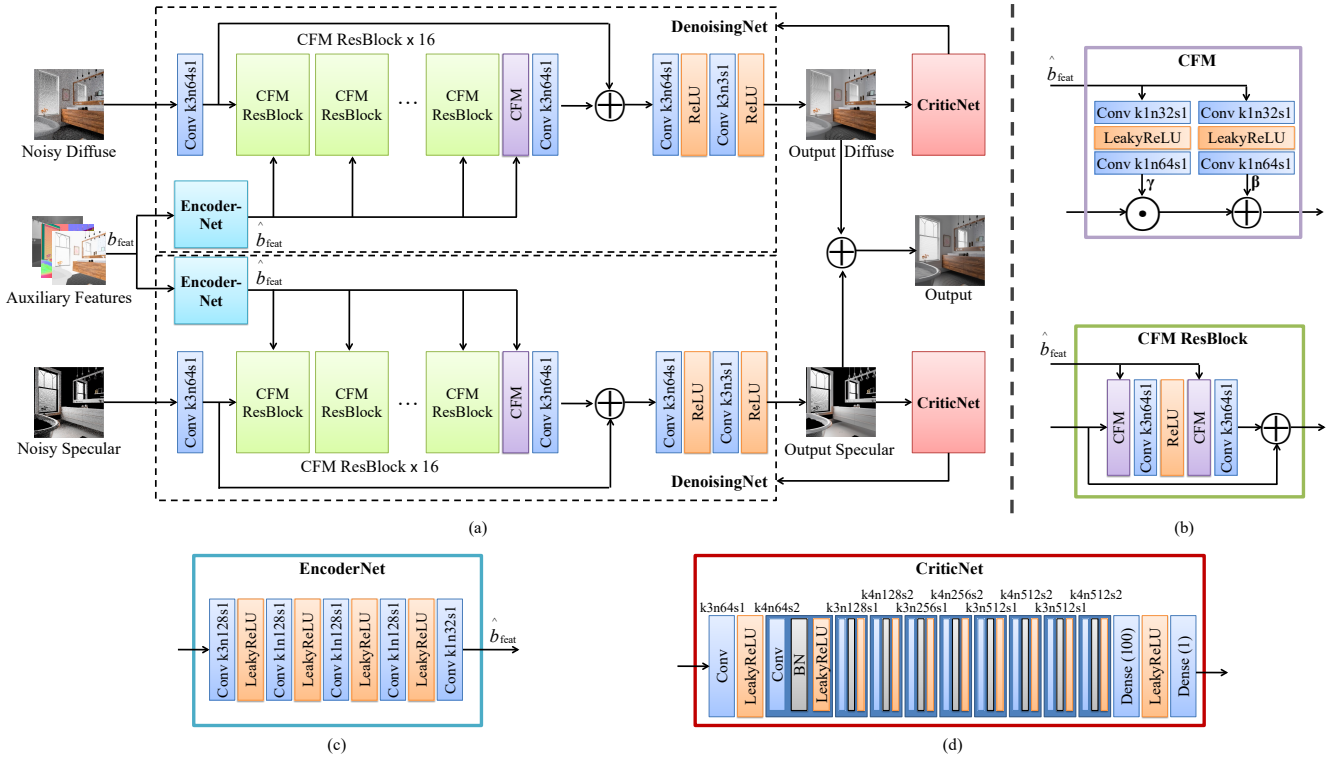

Fig. 2. (a) Overview of our adversarial framework; (b) Illustration of the residual block (ResBlock) for conditioned feature modulation; (c) EncoderNet; (d) CriticNet. Interpretation of network layer annotations: e.g, k3n128s1 indicates that kernel size is 3, number of feature channels is 128 and stride is 1.

4.1 ADVERSARIAL MC DENOISING FRAMEWORK

We illustrate our denoising framework in Fig. 2(a). Similar to KPCN [Bako et al. 2017], we process diffuse and specular components via separate paths with the same architecture but different parameters. This is theoretically valid since the BRDF factor in integrals of render equation [Kajiya 1986] can be split into two parts according to different scattering effects. The denoising network G is defined as (symbol c below stands for either diffuse or specular color):

$$c_{out} = G(c_{in}, b_{feat}), \tag{2}$$

where c_{out} is the denoised output of the noisy input c_{in}, and b_{feat} denotes auxiliary feature buffers, including normal (3 channels), depth (1 channel), albedo (3 channels) and others if applicable. We optimize G's parameters θ_G using generative adversarial training:

$$\min_{\theta_G} \max_{\theta_D} D(G(c_{in}, b_{feat}), c_{gt}), \tag{3}$$

where D is the critic network with parameters θ_D. G and D are jointly trained to minimize the difference between the sampling distribution of ground truth renderings c_{gt} and the sampling distribution of images generated by G.

4.2 AUXILIARY BUFFER CONDITIONED MODULATION

For denoising Monte Carlo rendering, traditional feature-guided filtering approaches are generally based on joint filtering or cross bilateral filtering [Bauszat et al. 2011]. The basic idea is to utilize auxiliary geometry and texture information to guide the estimation of the filtering weights, where strong assumptions are made on the correlation between the low-cost auxiliary features and the noisy input image [De Vries et al. 2017]. Due to the lack of samples in the neighborhood (e.g., only the spatial-similar or intensity-similar pixel samples), these approaches often result in visual artifacts [Bauszat et al. 2011].

Recent learning-based approaches use deep features to gather information from a larger neighborhood and to reduce the need of handcrafted assumptions. However, most of the those limit the influence of auxiliary features to early layers by just concatenating them with the noisy input image [Bako et al. 2017; Chaitanya et al. 2017]. Moreover, concatenation-based conditioning approach amounts to conditional biasing, namely, by adding a bias to the hidden layers based on conditioning representation (detailed explanation can be found on [Dumoulin et al. 2018]).

Inspired by the success of the conditional normalization methods in image style transfer literature [Dumoulin et al. 2017; Huang and Belongie 2017], we propose to integrate auxiliary feature information into our network by a conditioned feature modulation (**CFM**) technique similar to [Park et al. 2019; Wang et al. 2018b]. Apart from the conditional biasing mentioned above, it also involves conditional scaling, namely, by scaling the hidden layers based on conditioning representation. We define one operation of **CFM** as follows (see Fig. 2(b) for detailed illustration):

$$\mathbf{CFM}(L_{in}) = \boldsymbol{\gamma}(\hat{b}_{feat}) \odot L_{in} \oplus \boldsymbol{\beta}(\hat{b}_{feat}). \tag{4}$$

In the above equation, **CFM** modulates the feature maps L_{in} at multiple layers of G conditioning on $\hat{b}_{feat}$, which denotes transformed auxiliary features by the shared **EncoderNet** (see Fig. 2(c)). $\odot$ and $\oplus$ denote element-wise multiplication and addition. $\boldsymbol{\gamma}$ and $\boldsymbol{\beta}$ represent $\hat{b}_{feat}$-dependent scaling and shifting operation matrices: γ is composed of $\gamma^i_{c,y,x}$ and β is composed of $\beta^i_{c,y,x}$ which are the learned transformations. Letting C^i be the number of channels in the i-th layer, H^i and W^i respectively be the height and width of the feature maps in i-th layer, here we use symbols $\gamma^i_{c,y,x}$ and $\beta^i_{c,y,x}$ to denote the conversion operation from $\hat{b}_{feat}$ to the scaling and biasing values at the site (c, y, x) in i-th hidden feature layer ($c \in C^i, y \in H^i, x \in W^i$). The operation matrix ($\boldsymbol{\gamma}$ or $\boldsymbol{\beta}$) is modeled by a two-layer convolutional network (Fig. 2(b)). Besides, b_{feat}, the input of **EncoderNet**, is a concatenation of all auxiliary buffer channels.

By combining multiplicative and additive interactions at multiple layers, our framework allows stronger influence from auxiliary features. As discussed in a recent breakthrough [Perez et al. 2018], such modulation can be viewed as using one network to generate parameters of another network, making a new form of hypernetwork [Ha et al. 2016]. This also coincides with the joint filtering reconstruction approach where the auxiliary features guide the image filtering by modifying the weights of the filter [Bitterli et al. 2016].

4.3 DENOISING AND CRITICISING IN AN ADVERSARIAL APPROACH

As stated before, learning-based denoising approaches fail to cope with some complex situations like high-frequency noisy regions. We have implemented and experimented on various general convolutional denoising networks, for instance, the AutoEncoder network architecture in [Chaitanya et al. 2017], and found that it is the loss function rather than the network structure plays a more vital role in reconstruction quality. One phenomenon is that increasing the contribution of structural loss such as gradient loss [Chaitanya et al. 2017] effectively improves the results to a certain extent. This motivates us to automatically select loss function in order to wisely drive the training process.

Existing works [Goodfellow 2016; Lotter et al. 2015] indicate that using pixel-wise content loss like L1 or L2 loss tends to produce blurry results, since denoising and most other image reconstruction tasks are essentially ill-posed problems. There exist several possible solutions, and the pixel-wise content loss will end up in the average of these possible solutions [Ledig et al. 2017; Lotter et al. 2015]. Selecting L2 as loss function will maximize PSNR (peak signal-to-noise ratio) value, but this is not enough to guarantee perceptual quality [Wang et al. 2003]. And in general, pixel-level loss functions do not ensure a mechanism conforming to human visual perception. From this perspective, the ideal solution is to have a differentiable metric which naturally reflects human visual system. It is not a trivial task and inspires us to use generative adversarial networks, as such implicit models can be used for efficient learning even there is no direct definition or knowledge of the data distribution [Miyato et al. 2018]. This is because training a discriminator in GAN is equivalent to estimating the density ratio between the model and target distribution [Mohamed and Lakshminarayanan 2016].

In our work, we combine L1 content loss and adversarial loss given by the diffuse/specular critic network. We use L1 instead of L2 to reduce splotchy artifacts from reconstructed images as in [Bako et al. 2017; Chaitanya et al. 2017; Zhao et al. 2016]. While the existence of the content loss ensures the quality of the low frequency part, the adversarial loss (by trying to fool the critic) encourages our denoiser to generate results residing on the manifold of the noise-free images with high frequency details. The ablation study on the effectiveness of the adversarial loss can be found in Section 6 (see Fig. 9).

Our critic is based on Wasserstein-GAN [Arjovsky et al. 2017] with a gradient penalty, which enables stable training of a wide variety of GAN architectures with almost no hyper-parameter tuning. Furthermore, Wasserstein distance has less restriction on balancing the training process of the generator and the critic, making it possible to pre-train the latter on large-scale datasets first, and then, fine tuning it on one small render- or domain-specific dataset.

5 EVALUATION

5.1 TRAINING SETUP

Dataset. Large-scale datasets are necessary to avoid over-fitting for deep neural networks. In order to train our denoising critic network, we collected 1000 different indoor scene frames obtained from our commercial renderer, including 900 for training and 100 for validation. These scene frames were selected from diverse room designs with abundant illumination conditions as well as various materials and geometries, which span different denoising circumstances (see example scenes in Fig. 3). The reference images for training were rendered with 16k samples per pixel. Since state-of-the-art methods [Bako et al. 2017; Bitterli et al. 2016] use public scenes rendered by Tungsten [Bitterli et al. 2016] as testing or part of training data, we also downloaded datasets released by KPCN (noted by a Tungsten dataset) for evaluation. This dataset consists of sampled frames from 8 simple scenes (see Fig. 4).

All dataset images are divided into patches of size 128x128, and the auxiliary feature buffers are normalized to the same range of [0.0, 1.0]. Similar to Bako et al. [2017] and Chaitanya et al. [2017], the color values are stored as 16-bit half precision floating point (FP16) values to maintain a high dynamic range (HDR) of illumination. The specular component is the remaining radiance with diffuse

Fig. 3. Example reference images rendered from 1000 indoor scenes by our commercial renderer. The complete set can be found in supplemental material.

Fig. 4. Example reference images rendered by Tungsten. The whole set includes approximately 200 images with modified environment maps, camera parameters, and/or materials from each of 8 scenes [Bitterli 2016]

component excluded as in Bako et al. [2017]. For demonstration and comparison of the evaluation metrics, we apply gamma tone mapping to convert it into a low dynamic range (LDR). Moreover, we use the same input pre-processing strategies as in KPCN including using an untextured diffuse color buffer and applying the log transform on the specular one. We use only three auxiliary features (depth, normal and albedo), far fewer than KPCN which calculates the variance and gradient value for each buffer respectively. It can be seen in Section 5.3 that our method outperforms KPCN in most situations with fewer auxiliary buffers and this helps to save storage and IO cost.

Implementation details. As both diffuse and specular branches share the same network architecture design, we elaborate one branch as an example for the sake of simplicity. In the following, noisy color shall stand for either diffuse or specular color. We use PyTorch for implementation and train the networks on NVIDIA GTX 2080Ti. We use 16 Resblocks modified with our auxiliary buffer conditioned modulation as shown in Fig. 2 for the denoiser network. Residual blocks help to enhance training of deep networks [He et al. 2016]. Batch normalization layers from the original Resblock are removed to save computational cost without affecting network performance as in Wang et al. [2018c]. The auxiliary feature encoder network contains five convolutional layers each followed by a Leaky ReLU layer (parameter=0.1); and the network architecture design of the critic network is similar to the widely used VGG [Simonyan and Zisserman 2014], which takes an input size of 128*128 (network details are shown in Fig. 2(d)).

Training GANs is still a challenging task and we use auxiliary features and WGAN-GP to stabilize the training. In particular, the Wasserstein metric replaces the *JS* divergence of vanilla GANs to have a much smoother value space. This relaxes the requirement of skill level for generator and discriminator. In addition to this method, there are many other works dedicated to solving this problem [Roth et al. 2017; Salimans et al. 2016]. Adversarial training here follows the original WGAN [Arjovsky et al. 2017] for tuning hyperparameters and involves adopting a gradient penalty to enforce the Lipschitz constraint as in WGAN-GP [Gulrajani et al. 2017]. We train the network using an Adam optimizer (0.9, 0.999 for two betas and 1e-8 for epsilon) with the initialization given by He et al. [2015]. The learning rate is set to 1e-4 for the diffuse branch and a smaller value of 1e-6 for the specular branch. Both branches halve the learning rate at most four times after every 50k iterations. Lastly, the weight ratio between L1 loss and adversarial loss is set to 200:1 to make training more stable. The training takes about 2 days on a single 2080Ti GPU for each branch.

5.2 EVALUATION METRICS AND STATE-OF-THE-ART METHODS

To compare various MC denoising algorithms, we chose three different evaluation metrics, including relative MSE, SSIM and PSNR. The RMSE and SSIM heat maps for all the results are provided in the supplemental viewer. We only present subsets of metrics in this paper for concise demonstration purposes and some parts of the image are zoomed-in using bilinear interpolation to facilitate detailed comparison.

Based on the discussions in Section 2, we have selected three state-of-the-art MC denoising methods to compare the outcomes with, including NFOR [Bitterli et al. 2016], KPCN [Bako et al. 2017], and RAE [Chaitanya et al. 2017]. Denoiser by NFOR is shipped with the public Tungsten renderer, whilst KPCN has codes and training weights available. RAE has not released training and inference codes, so we ran the open executable file which only accepts LDR images as input instead. It should be noted that RAE is trained on its own large-scale dataset that is not in public. We trained and validated our adversarial MC denoiser on our dataset (see Fig. 3). Its representative coverage benefits our denoiser and critic network for generalization ability. As different rendering systems can have inconsistent sampling strategy and noise levels, to fairly compare with the above baseline methods on Tungsten scenes, we also trained on these scenes to adapt our denoiser to the Tungsten renderer.

5.3 RESULTS

To compare our work with state-of-the-art methods, we conduct experiments on noisy input images with different sample per pixel (spp). The complete testing results with 4, 16, 32, 64 and 128 spp can be found in the supplemental. Fig. 5 shows some typical results and closeups from four public Tungsten scenes, including **Bathroom** with mirrors, **PinkRoom** with colorful glasses under reflection and refraction situations, **HorseRoom** and **WhiteRoom** that are relatively dim and make global illumination more difficult to denoise. These demonstrate the ability of our network to preserve features (e.g., object edges, shadows in both flat and complex regions, etc.) and hence, the advantage of our work over previous approaches especially in relation to high-frequency details (e.g., room corners, ceilings, etc.) while adapting to different lighting setups. Full resolution images can be found in the supplemental interactive viewer[1] for detailed inspection.

Overall our work performs consistently on a par or better than the state-of-the-art methods in terms of both perceptual quality and quantitative metrics. NFOR, one of the best traditional offline filtering methods, suffers from splotchy-looking results and residual noise due to limited filter kernel size and insufficient statistics from only neighborhood pixels for filter weights estimation. Learning-based methods (KPCN and RAE) generally obtain better results in low-frequency areas, but may produce over-smoothed ones with approximate colors for shading details. Our approach is satisfactory

[1] http://adversarial.mcdenoising.org

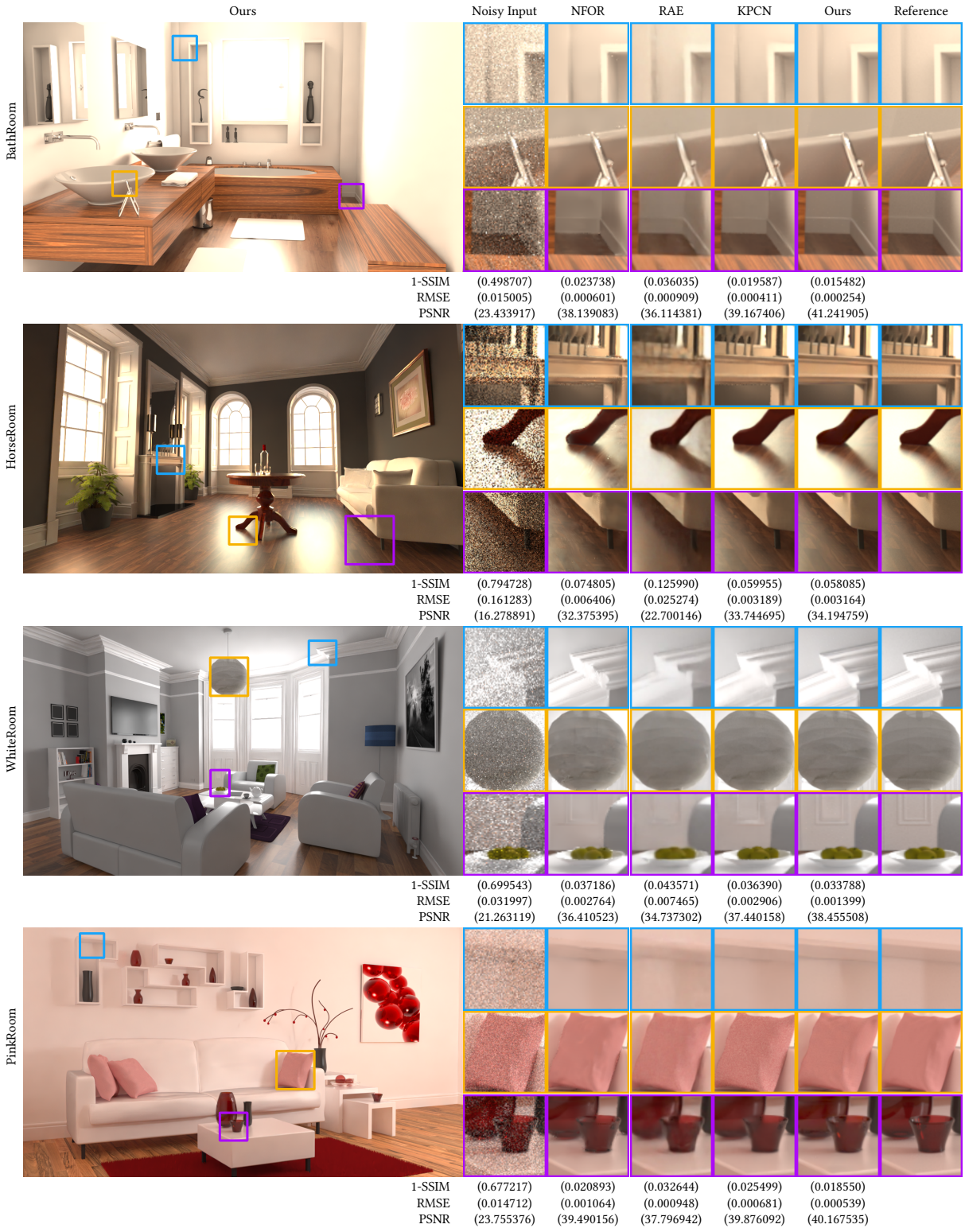

Fig. 5. We compare our results with baseline methods: NFOR [Bitterli et al. 2016], RAE [Chaitanya et al. 2017] and KPCN [Bako et al. 2017] on a test set rendered by Tungsten. Scenes rendered in 32 spp are from Bitterli [2016] with publicly available lighting and camera settings. Full resolution images with SSIM and RMSE heat maps can be found in the supplemental viewer.

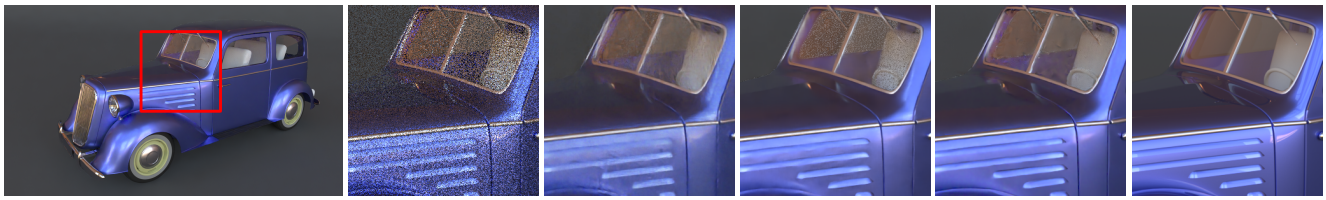

Fig. 6. We apply the adversarial denoiser trained on 32 spp to denoise 4 spp noisy input to demonstrate its successful extrapolation to low sample rates.

Table 1. Aggregate numerical performance of all methods over the entire Tungsten testset. Avg. denotes the average of specific metrics and B.P. denotes the percentage of the very method to be the best one among all. Perceptually and quantitatively, our method outperforms the state-of-the-arts especially for very noisy inputs with low spp. When spp increases and input becomes less challenging, the advantage over others slightly decrease in metrics but the perceptual quality is still comparable or even better. The discussion of convergence behavior can be seen in Section 6.5.

spp	Denoiser	1-SSIM ↓		PSNR ↑		RMSE ↓	
		AVG.	B.P.	AVG.	B.P.	AVG.	B.P.
4	NFOR	0.1614	0.00%	28.3028	0.00%	0.0374	0.00%
	RAE	**0.0751**	37.93%	29.5359	0.00%	0.0080	6.90%
	KPCN	0.0891	13.79%	32.1290	0.00%	0.0059	3.45%
	Ours	0.0773	**48.28%**	**34.3188**	**100.00%**	**0.0038**	**89.66%**
16	NFOR	0.0707	10.34%	32.6832	10.34%	0.0180	3.45%
	RAE	0.0549	3.45%	34.3337	0.00%	0.0033	3.45%
	KPCN	0.0531	20.69%	36.4538	0.00%	0.0024	6.90%
	Ours	**0.0463**	**65.52%**	**37.8608**	**89.66%**	**0.0019**	**86.21%**
32	NFOR	0.0493	10.34%	34.8495	6.90%	0.0118	3.45%
	RAE	0.0482	0.00%	36.0105	0.00%	0.0028	0.00%
	KPCN	0.0426	20.69%	38.4051	3.45%	0.0017	10.34%
	Ours	**0.0366**	**68.97%**	**39.6197**	**89.66%**	**0.0013**	**86.21%**
64	NFOR	0.0389	6.90%	37.3478	6.90%	0.0067	3.45%
	RAE	0.0395	0.00%	38.0982	0.00%	0.0016	0.00%
	KPCN	0.0349	20.69%	40.2623	27.59%	0.0012	34.48%
	Ours	**0.0296**	**72.41%**	**40.9673**	**65.52%**	**0.0009**	**62.07%**
128	NFOR	0.0305	10.34%	39.5127	6.90%	0.0036	3.45%
	RAE	0.0338	0.00%	39.8093	0.00%	0.0011	0.00%
	KPCN	0.0288	27.59%	42.0056	**48.28%**	0.0008	**62.07%**
	Ours	**0.0248**	**62.07%**	**42.0803**	44.83%	**0.0007**	34.48%

in both low-frequency and high-frequency areas due to a smoother global illumination effect and better detail preservation, thus resulting in more visually pleasing and perceptually natural denoised output. Table 1 shows aggregate numerical performance over the entire test set comparing to the state-of-the-arts. More comparisons and details of each result can be found in the supplemental viewer. We also employed the adversarial denoiser trained at 32 spp to denoise 4 spp noisy images to demonstrate the good generality for low sample counts relative to other approaches (see Fig. 6).

Since we separate diffuse and specular buffers, we also provide denoised results of each buffer to compare with KPCN that relies on the same strategy. As shown in Fig. 7 and Fig. 8, our work consistently leads to better results for both diffuse and specular components.

5.4 RECONSTRUCTION PERFORMANCE

Our approach also strikes a good balance between denoising quality and computational cost. Inference (denoising) takes an average of 1.1s (550ms for diffuse or specular, respectively) on a single 2080Ti GPU for a 1280x720 image, while KPCN takes 3.9s in the same settings as the kernel filtering process requires more time. And our approach takes far less pre-processing time and memory due to

Fig. 7. Denoised diffuse components on 16 spp (the first row) and 64 spp (the second row) scenes: KPCN vs. ours.

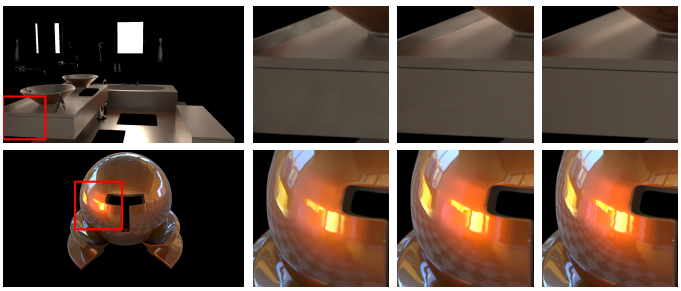

Fig. 8. Denoised specular component on 16 spp scenes: KPCN vs. ours.

fewer feature buffers used. NFOR only provides CPU implementation which takes more than 10 seconds on a 3.4GHz Intel Xeon processor. Besides this, the number of our inference network parameters is 2M compared to 3M of RAE which is with highly interactive rates, leading to competitive running speed.

6 ANALYSIS

In order to study the effects of various design choices in our work, we modified the different components of our framework and compared the performance with the full model.

6.1 EFFECTIVENESS OF THE ADVERSARIAL LOSS AND CRITIC NETWORK

By adjusting the loss function, we show that the critic network help guide the training of the denoiser network and produce perceptually more pleasing results than using handcrafted loss (L1 in our experiment; other objectives like L2 and SSIM losses have been proven to perform worse than the L1 loss [Bako et al. 2017]).

For example, the edges of green leaves and reflected textures in Fig. 9 are better preserved in our result with the adversarial loss than

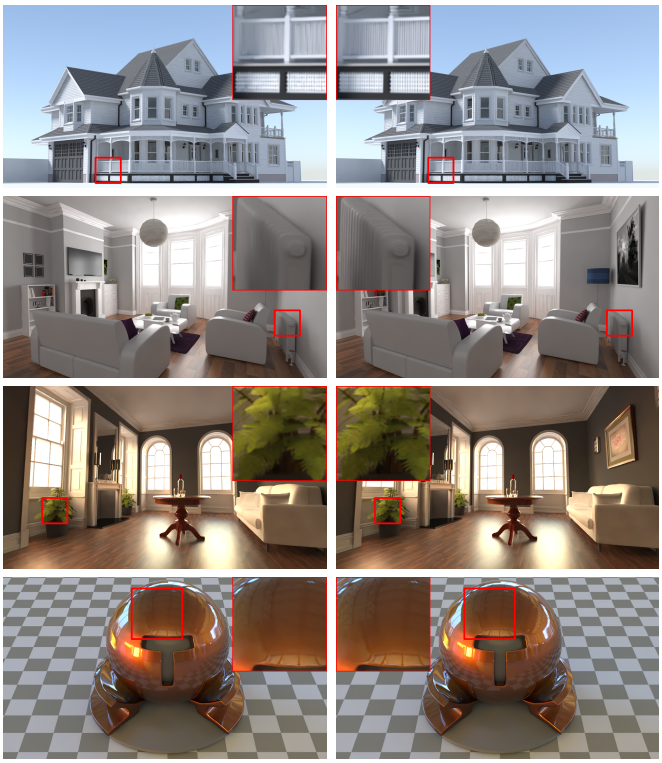

Fig. 9. Comparisons of high frequency details without (left) and with (right) adversarial loss. Scenes credit to Bitterli [2016]

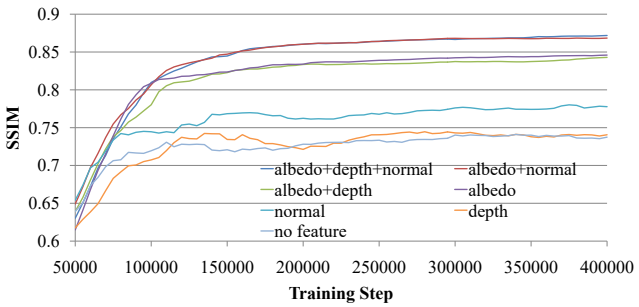

Fig. 10. Convergence graph of SSIM on the validation set for different combination of the auxiliary feature buffers.

Fig. 11. Comparisons of different feature utilizing methods. From left to right: (a) Training with no auxiliary features; (b) Concatenate the auxiliary features and noisy color as fused input; (c) Full model of CFM with a combination of shifting and scaling; (d) Reference.

using only L1 loss, thus conforming with the hypothesis that perceptual similarity is difficult to be captured by handcrafted objectives. The classic per-pixel metrics such as the L1 and L2 losses assume no inter-connection between the pixels. In contrast, the adversarial loss can capture visual patterns that are both high-dimensional and structurally correlated [Zhang et al. 2018]. See the supplemental viewer for the full display of the superior results of our method.

Despite the L2 distance between pre-trained VGG features being a popular choice to balance the adversarial loss, [Chen and Koltun 2017; Gatys et al. 2016; Ledig et al. 2017], our experiments observed no significant improvement from it. This is largely due to the use of auxiliary features, which act as a smooth context constraint that regularizes GANs.

6.2 EFFECTIVENESS OF AUXILIARY FEATURE BUFFER

In order to demonstrate the denoising enhancement due to auxiliary feature buffers, we conduct an ablation study on activating only subsets of buffers. In Fig. 10, we plot the convergence curve of SSIM metric on the validation set. We drew a similar experimental conclusion as Chaitanya et al. [2017]. With the 7 control groups, it is easy to tell that albedo is the most significant influence factor, normal is less influential while depth, shown by the experiment, has almost no effect (see the bottom two curves in Fig. 10). That is to say, when the feature provides more texture and contour information, the network gets more guidance to restore edges. Additionally, all these features are geometry-relevant and research on shading related features remains as future work.

6.3 EFFECTIVENESS OF FEATURE CONDITIONED MODULATION

As presented in Section 4.2, we adopt conditioned modulation to apply additive and multiplicative interactions. This helps to automatically model the relationship between the auxiliary features and the noisy input image. To demonstrate the effectiveness, we compare with simply concatenating them as a fused input into the modified network as in previous works [Bako et al. 2017; Chaitanya et al. 2017] (see Fig. 11). In our implementation, we intentionally increased the number of layers, as well as the channels per layer, for the feature concatenation approach. Hence the compared networks had a similar number of parameters to ensure fairness.

Conventional input-concatenation approaches limit the effectiveness of auxiliary features to early layers. In contrast, the proposed conditioned modulation layers perform scaling and shifting at different scales: point-wise shifting modulates the feature activation; scaling selectively suppresses or highlights feature activation. Whilst all "a posterior" MC denoising methods inherently leave an unexplored gap between 2D image space and the high-dimensional path space, the combination of scaling and shifting is still more powerful than feature concatenation, which is equivalent to biasing. It should be noted that our point-wise multiplicative interaction also resembles

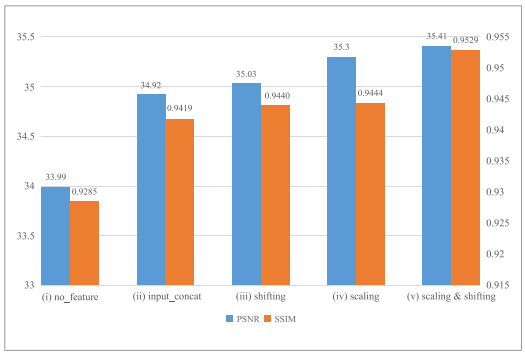

Fig. 12. Ablation study on the variants of the full model to evaluate our conditioned feature modulation (CFM) method. Average performance of PSNR and SSIM over test scenes on the diffuse branch. From left to right: (i) Training with no auxiliary features; (ii) Concatenate the auxiliary features and noisy color as fused input; (iii) Variant of CFM using only a shifting operation; (iv) Variant of CFM using only a scaling operation; (v) Full model of CFM with a combination of shifting and scaling.

the attention mechanism which has been widely used in machine learning applications.

Also, to test the influence of $\boldsymbol{\gamma}$ mapping (for multiplicative interaction) and $\boldsymbol{\beta}$ mapping (for additive operation), we trained two model variants by taking out multiplication or addition respectively. One is to deactivate the $\boldsymbol{\gamma}$ operation by setting all elements of $\boldsymbol{\gamma}(\hat{b}_{feat})$ to 1.0; the other is to deactivate the $\boldsymbol{\beta}$ operation by setting all elements of $\boldsymbol{\beta}(\hat{b}_{feat})$ to 0.0 (see details in Fig. 2(b) and Section 4.2). We show the performance on different evaluation metrics of these variants in Fig. 12, where we take the diffuse component as an experimental example. The results show that the conditioned modulation is more effective than the commonly used concatenation with the input. Besides this, modulation block can learn to condition the auxiliary feature buffers through either additive or multiplicative interaction alone. However, neither of them performs as well as the combination of the two. It also suggests that the multiplicative interaction plays a slightly more important role than the additive one.

Another design choice is whether different auxiliary features should be fused into the input of **EncoderNet** (Fig. 2(c)). We experimented on one variant by consecutively chaining the buffer-specific modulation layers. Specifically, in each **CFM** Resblock (see Fig. 2(b) and Equation 4), the **CFM** function becomes auxiliary buffer-specific (**CFM**_normal, **CFM**_depth and **CFM**_albedo are applied to modulate network layers sequentially instead of being one **CFM** conditioned on fused auxiliary features). The results do not show much difference, indicating that the auxiliary features do not need multiplicative operations among themselves.

6.4 DIFFUSE AND SPECULAR DECOMPOSITION

To verify the necessity of such strategy, we conducted training and denoising without decomposition. This led to unsatisfactory results as shown in Fig. 13, where the illumination on the floor is disturbing.

The texture appears to be erroneously enhanced and the reflected illumination is weakened. Adding the adversarial loss is helpful to a certain extent but cannot fundamentally eliminate this problem. The explanation would be that the diffuse and specular components have different noise patterns with different characteristics. Thus

Fig. 13. Left: Reflection is not reconstructed well without separating diffuse and specular components. Right: Reflection is well reconstructed by separating diffuse and specular components.

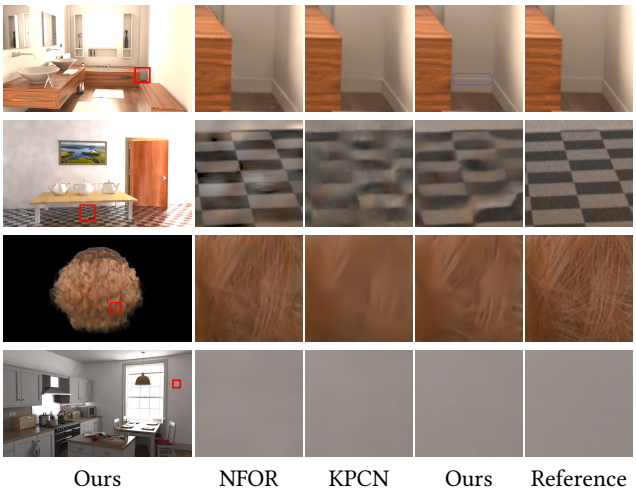

Fig. 14. Zoomed-in failure cases due to limitations.

they may require different convolutional kernels or utilize auxiliary features in different ways. Decoupling the two buffers leads to better reconstruction results, which coincides with the approach of KPCN [Bako et al. 2017]. Additionally, as in render equation [Kajiya 1986], we need to separate the diffuse and specular integrals theoretically if we want to calculate an untextured color as we do in our implementation, so too for KPCN. Yet RAE [Chaitanya et al. 2017] uses untextured color without separating these paths, which tends to produce similar artifacts as in Fig. 13.

To understand better the mutual relation between the two components, we tried either sharing the feature pre-processing layers in the **EncoderNet** module (Fig. 2(c)) between diffuse and specular branches, or jointly training them to obtain a single adversarial loss on the final reconstructed image. However, both variants by sharing the encoder net or critic net parameters result in worse performance. Thus we choose to decompose these two buffers as KPCN does. How to combine the two components efficiently to achieve a compact end-to-end architecture remains as future work .

6.5 DISCUSSION and LIMITATIONS

Limitations. While our method demonstrates an overall better perceptual quality than the state-of-the-arts, it also has various limitations. First, some failure cases are difficult to interpret due to the black-box nature of deep neural networks. For instance, some low contrast features were magnified with respect to the reference (the blue-circled skirting line on the wall; top row in Fig. 14). These behaviors could be potentially improved by enlarging the training

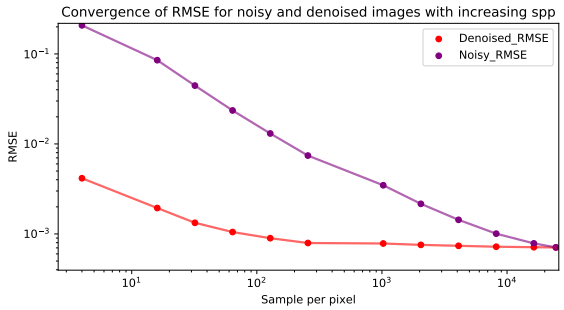

Fig. 15. Convergence of RMSE for both noisy input and denoised images with increasing spp. The vertical axis is the average RMSE of the test scenes. The horizontal axis is spp. The original spps are [4,16,32,64,128,256,1024,2048,4096,8192,16384,24576]. We can see that at some point two curves intersect. The reference is rendered with 32k spp.

dataset. Inconsistent training and test data distributions also lead to failure cases, the specific effects of which are not covered by the training set, such as fog or smoke.

Moreover, our method may perform poorly in cases where the input feature buffers do not capture the details. For example, the denoiser cannot restore fine details of hair due to lack of depth information (third row in Fig. 14). Another case is the blurry specular texture of the floor (second row in Fig. 14), as Tungsten dataset does not include scenes with special specular albedo thus no distinction between albedo buffers used by the diffuse/specular branch is ensured. Simply adding specular albedo buffer and increasing the diversity of the training scenes will resolve this problem.

Rendering reference images of training data is computationally intensive as it demands large sample counts. Limited samples (e.g., 8k, 32k) inevitably cause remaining noise for some complex effects and hence the learnt networks have the risk of over-fit noise. Some residual noise can be seen when the images are zoomed in on a very large scale (bottom row in Fig. 14).

Convergence Discussion. Monte Carlo path tracing is known to be inherently unbiased but converges with a high computational cost. The "a posterior" denoising methods including ours effectively accelerate the convergence by deriving a posterior from the statistics of a set of samples. Smaller errors can be obtained by increasing the samples but converging to zero is unreachable by our approach. This is also the general case for all learning-based methods, and the reason lies in the gap between training data and testing data. The trained model is not exactly restricted by the empirical variance of rendered pixel means, thus error vanishing with increasing sample counts of specific rendering image is not guaranteed (see Fig. 15).

7 CONCLUSION AND FURTHER WORK

We have presented the first adversarial learning framework for the Monte Carlo denoising problem and have achieved state-of-the-art results with improved perceptual quality. Moreover, our framework sheds light on exploring the relationship between auxiliary features and noisy images by neural networks. Comprehensive evaluations have demonstrated the effectiveness and efficiency of our framework over previous works.

Future work. Our work can be extended in several ways in the future. First, the current network architecture is far from optimal. Network designs can be fine-tuned to simplify the model, and various strategies including custom-precision and model pruning can be explored to accelerate the inference process. We would also like to study how to encode temporal coherence into our framework for interactive applications. Second, in addition to additive and multiplicative operations, more complex relationships based on Attention mechanism [Xu et al. 2015] or hypernetworks [Ha et al. 2016] can be exploited for better feature guidance. Variant network structures other than GANs (e.g., variational autoencoder) can be investigated to further improve the quality of denoised images in terms of human perception. Third, it would be beneficial to extend our approach to handle a greater range of rendering effects, like depth of field and motion blur. More data from different domains are needed here to test its potential. Finally, how to achieve comparable quality for MC denoising with 'light-weight' learning is worth exploring, as generating noise-free ground truth on a large scale is rather expensive [Lehtinen et al. 2018].

ACKNOWLEDGMENTS

We gratefully thank the anonymous reviewers for their constructive suggestions, and Qing Ye, Qi Wu, Junrong Huang for helpful discussions and cluster rendering support. This work is partially funded by National Key R&D Program of China (No. 2017YFB1002605), NSFC (No. 61872319, 61822204, 61521002), Zhejiang Provincial NSFC (No. LR18F020002), CAMERA - the RCUK Centre for the Analysis of Motion, Entertainment Research and Applications (EP/M023281/1), and a gift from Adobe.

REFERENCES

Martin Arjovsky, Soumith Chintala, and Léon Bottou. 2017. Wasserstein gan. *arXiv preprint arXiv:1701.07875* (2017).

Steve Bako, Thijs Vogels, Brian McWilliams, Mark Meyer, Jan Novák, Alex Harvill, Pradeep Sen, Tony Derose, and Fabrice Rousselle. 2017. Kernel-predicting convolutional networks for denoising Monte Carlo renderings. *ACM Trans. Graph.* 36, 4 (2017), 97–1.

Pablo Bauszat, Martin Eisemann, and Marcus Magnor. 2011. Guided image filtering for interactive high-quality global illumination. In *Computer Graphics Forum*, Vol. 30. 1361–1368.

Benedikt Bitterli. 2016. Rendering resources. https://benedikt-bitterli.me/resources/.

Benedikt Bitterli, Fabrice Rousselle, Bochang Moon, José A Iglesias-Guitián, David Adler, Kenny Mitchell, Wojciech Jarosz, and Jan Novák. 2016. Nonlinearly Weighted First-order Regression for Denoising Monte Carlo Renderings. In *Computer Graphics Forum*, Vol. 35. 107–117.

Chakravarty R Alla Chaitanya, Anton S Kaplanyan, Christoph Schied, Marco Salvi, Aaron Lefohn, Derek Nowrouzezahrai, and Timo Aila. 2017. Interactive reconstruction of Monte Carlo image sequences using a recurrent denoising autoencoder. *ACM Transactions on Graphics (TOG)* 36, 4 (2017), 98.

Jingwen Chen, Jiawei Chen, Hongyang Chao, and Ming Yang. 2018. Image blind denoising with generative adversarial network based noise modeling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3155–3164.

Qifeng Chen and Vladlen Koltun. 2017. Photographic image synthesis with cascaded refinement networks. In *Proceedings of the IEEE International Conference on Computer Vision*. 1511–1520.

Holger Dammertz, Daniel Sewtz, Johannes Hanika, and Hendrik Lensch. 2010. Edge-avoiding À-Trous wavelet transform for fast global illumination filtering. In *Proceedings of the Conference on High Performance Graphics*. Eurographics Association, 67–75.

Harm De Vries, Florian Strub, Jérémie Mary, Hugo Larochelle, Olivier Pietquin, and Aaron C Courville. 2017. Modulating early visual processing by language. In *Advances in Neural Information Processing Systems*. 6594–6604.

Nithish Divakar and R Venkatesh Babu. 2017. Image denoising via CNNs: an adversarial approach. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 80–87.

Vincent Dumoulin, Ethan Perez, Nathan Schucher, Florian Strub, Harm de Vries, Aaron Courville, and Yoshua Bengio. 2018. Feature-wise transformations. *Distill* (2018). https://doi.org/10.23915/distill.00011 https://distill.pub/2018/feature-wise-transformations.

Vincent Dumoulin, Jonathon Shlens, and Manjunath Kudlur. 2017. A learned representation for artistic style. *Proc. of ICLR* 2 (2017).

Raanan Fattal. 2007. Image upsampling via imposed edge statistics. *ACM transactions on graphics (TOG)* 26, 3 (2007), 95.

Leonardo Galteri, Lorenzo Seidenari, Marco Bertini, and Alberto Del Bimbo. 2017. Deep generative adversarial compression artifact removal. In *Proceedings of the IEEE International Conference on Computer Vision*. 4826–4835.

Leon A Gatys, Alexander S Ecker, and Matthias Bethge. 2016. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2414–2423.

Michaël Gharbi, Tzu-Mao Li, Miika Aittala, Jaakko Lehtinen, and Frédo Durand. 2019. Sample-based Monte Carlo denoising using a kernel-splatting network. *ACM Transactions on Graphics (TOG)* 38, 4 (2019), 125.

Golnaz Ghiasi, Honglak Lee, Manjunath Kudlur, Vincent Dumoulin, and Jonathon Shlens. 2017. Exploring the structure of a real-time, arbitrary neural artistic stylization network. *arXiv preprint arXiv:1705.06830* (2017).

Ian Goodfellow. 2016. NIPS 2016 tutorial: Generative adversarial networks. *arXiv preprint arXiv:1701.00160* (2016).

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in neural information processing systems*. 2672–2680.

Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. 2017. Improved training of wasserstein gans. In *Advances in Neural Information Processing Systems*. 5767–5777.

David Ha, Andrew Dai, and Quoc V Le. 2016. Hypernetworks. *arXiv preprint arXiv:1609.09106* (2016).

Kaiming He, Jian Sun, and Xiaoou Tang. 2010. Single image haze removal using dark channel prior. *IEEE transactions on pattern analysis and machine intelligence* 33, 12 (2010), 2341–2353.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*. 1026–1034.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

Xun Huang and Serge Belongie. 2017. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE International Conference on Computer Vision*. 1501–1510.

Xun Huang, Ming-Yu Liu, Serge Belongie, and Jan Kautz. 2018. Multimodal unsupervised image-to-image translation. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 172–189.

Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. 2017. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1125–1134.

James T Kajiya. 1986. The rendering equation. In *ACM SIGGRAPH computer graphics*, Vol. 20. ACM, 143–150.

Nima Khademi Kalantari, Steve Bako, and Pradeep Sen. 2015. A machine learning approach for filtering Monte Carlo noise. *ACM Trans. Graph.* 34, 4 (2015), 122–1.

Alexander Keller, Luca Fascione, Marcos Fajardo, Iliyan Georgiev, P Christensen, Johannes Hanika, Christian Eisenacher, and Gregory Nichols. 2015. The path tracing revolution in the movie industry. In *ACM SIGGRAPH 2015 Courses*. ACM, 24.

Markus Kettunen, Erik Härkönen, and Jaakko Lehtinen. 2019. Deep convolutional reconstruction for gradient-domain rendering. *ACM Transactions on Graphics (TOG)* 38, 4 (2019), 126.

Orest Kupyn, Volodymyr Budzan, Mykola Mykhailych, Dmytro Mishkin, and Jiří Matas. 2018. Deblurgan: Blind motion deblurring using conditional adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 8183–8192.

Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, 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*. 4681–4690.

Jaakko Lehtinen, Jacob Munkberg, Jon Hasselgren, Samuli Laine, Tero Karras, Miika Aittala, and Timo Aila. 2018. Noise2noise: Learning image restoration without clean data. *arXiv preprint arXiv:1803.04189* (2018).

Tzu-Mao Li, Yu-Ting Wu, and Yung-Yu Chuang. 2012. SURE-based optimization for adaptive sampling and reconstruction. *ACM Transactions on Graphics (TOG)* 31, 6 (2012), 194.

William Lotter, Gabriel Kreiman, and David Cox. 2015. Unsupervised learning of visual structure using predictive generative networks. *arXiv preprint arXiv:1511.06380* (2015).

Michael D McCool. 1999. Anisotropic diffusion for Monte Carlo noise reduction. *ACM Transactions on Graphics (TOG)* 18, 2 (1999), 171–194.

Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. 2018. Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957* (2018).

Shakir Mohamed and Balaji Lakshminarayanan. 2016. Learning in implicit generative models. *arXiv preprint arXiv:1610.03483* (2016).

Bochang Moon, Nathan Carr, and Sung-Eui Yoon. 2014. Adaptive rendering based on weighted local regression. *ACM Transactions on Graphics (TOG)* 33, 5 (2014), 170.

Bochang Moon, Jong Yun Jun, JongHyeob Lee, Kunho Kim, Toshiya Hachisuka, and Sung-Eui Yoon. 2013. Robust image denoising using a virtual flash image for Monte Carlo ray tracing. In *Computer Graphics Forum*, Vol. 32. 139–151.

Bochang Moon, Steven McDonagh, Kenny Mitchell, and Markus Gross. 2016. Adaptive polynomial rendering. *ACM Transactions on Graphics (TOG)* 35, 4 (2016), 40.

Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. 2019. Semantic Image Synthesis with Spatially-Adaptive Normalization. *arXiv preprint arXiv:1903.07291* (2019).

Ethan Perez, Harm De Vries, Florian Strub, Vincent Dumoulin, and Aaron Courville. 2017. Learning visual reasoning without strong priors. *arXiv preprint arXiv:1707.03017* (2017).

Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. 2018. Film: Visual reasoning with a general conditioning layer. In *Thirtyfilm-Second AAAI Conference on Artificial Intelligence*.

Alec Radford, Luke Metz, and Soumith Chintala. 2015. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434* (2015).

Kevin Roth, Aurelien Lucchi, Sebastian Nowozin, and Thomas Hofmann. 2017. Stabilizing training of generative adversarial networks through regularization. In *Advances in neural information processing systems*. 2018–2028.

Fabrice Rousselle, Marco Manzi, and Matthias Zwicker. 2013. Robust denoising using feature and color information. In *Computer Graphics Forum*, Vol. 32. 121–130.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. 2016. Improved techniques for training gans. In *Advances in neural information processing systems*. 2234–2242.

Pradeep Sen and Soheil Darabi. 2012. On filtering the noise from the random parameters in Monte Carlo rendering. *ACM Trans. Graph.* 31, 3 (2012), 18–1.

Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014).

Thijs Vogels, Fabrice Rousselle, Brian McWilliams, Gerhard Röthlin, Alex Harvill, David Adler, Mark Meyer, and Jan Novák. 2018. Denoising with kernel prediction and asymmetric loss functions. *ACM Transactions on Graphics (TOG)* 37, 4 (2018), 124.

Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. 2018a. High-resolution image synthesis and semantic manipulation with conditional gans. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 8798–8807.

Xintao Wang, Ke Yu, Chao Dong, and Chen Change Loy. 2018b. Recovering realistic texture in image super-resolution by deep spatial feature transform. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 606–615.

Xintao Wang, Ke Yu, Shixiang Wu, Jinjin Gu, Yihao Liu, Chao Dong, Yu Qiao, and Chen Change Loy. 2018c. Esrgan: Enhanced super-resolution generative adversarial networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 0–0.

Zhou Wang, Eero P Simoncelli, and Alan C Bovik. 2003. Multiscale structural similarity for image quality assessment. In *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*, Vol. 2. Ieee, 1398–1402.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhutdinov, Richard Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. *arXiv preprint arXiv:1502.03044* (2015).

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. 2018. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 586–595.

Hang Zhao, Orazio Gallo, Iuri Frosio, and Jan Kautz. 2016. Loss functions for image restoration with neural networks. *IEEE Transactions on computational imaging* 3, 1 (2016), 47–57.

Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*. 2223–2232.

Henning Zimmer, Fabrice Rousselle, Wenzel Jakob, Oliver Wang, David Adler, Wojciech Jarosz, Olga Sorkine-Hornung, and Alexander Sorkine-Hornung. 2015. Path-space Motion Estimation and Decomposition for Robust Animation Filtering. In *Computer Graphics Forum*, Vol. 34. 131–142.

Matthias Zwicker, Wojciech Jarosz, Jaakko Lehtinen, Bochang Moon, Ravi Ramamoorthi, Fabrice Rousselle, Pradeep Sen, Cyril Soler, and S-E Yoon. 2015. Recent advances in adaptive sampling and reconstruction for Monte Carlo rendering. In *Computer Graphics Forum*, Vol. 34. 667–681.