Computational Visual Media
https://doi.org/10.1007/s41095-021-0209-9 Vol. 7, No. 2, June 2021, 169–185

Review Article

A survey on deep learning-based Monte Carlo denoising

Yuchi Huo[1], **Sung-eui Yoon**[1] (✉)

Abstract Monte Carlo (MC) integration is used ubiquitously in realistic image synthesis because of its flexibility and generality. However, the integration has to balance estimator bias and variance, which causes visually distracting noise with low sample counts. Existing solutions fall into two categories, in-process sampling schemes and post-processing reconstruction schemes. This report summarizes recent trends in the post-processing reconstruction scheme. Recent years have seen increasing attention and significant progress in denoising MC rendering with deep learning, by training neural networks to reconstruct denoised rendering results from sparse MC samples. Many of these techniques show promising results in real-world applications, and this report aims to provide an assessment of these approaches for practitioners and researchers.

Keywords rendering; Monte Carlo (MC) denoising; deep learning; ray tracing

1 Introduction

The synthesis of realistic images of virtual worlds is one of the primary driving forces for the development of computer graphics techniques [1, 2]. One of the firmly established bases for such a purpose is MC integration [3], which is renowned for its generality and heavy computational consumption. MC integration methods are attractive because of two distinct advantages. Firstly, they offer a unified framework for rendering almost every physically-based rendering effect. This significantly reduces the burden of exhaustive case-by-case customization of rendering pipelines. Secondly, most MC methods guarantee mathematical convergence to the ground truth, which is a critical virtue for high-quality rendering that requires temporal consistency and physical fidelity.

Classical MC integration methods, however, require a large number of samples to achieve faithful convergence. Despite continuously increasing computational power, the cost of realistic rendering remains a limiting, practical constraint, as it takes hours to render one high-quality image, or frame. When using few samples, MC integration results often suffer from estimator variance, which appears as visually distracting noise. The heavy computational consumption is one of the primary factors prohibiting a wider accessibility of MC integration. To address this problem, common approaches either devise more sophisticated sampling strategies to increase sampling efficiency, or develop local reconstruction functions to trade mathematical convergence for visually appealing denoising. Such a post-processing scheme is known as MC denoising, one of the most investigated areas in the rendering community.

Recently, deep learning techniques have earned unprecedented attention and exceeded many traditional algorithms in various domains [4, 5]. Derived from traditional MC reconstruction [6], MC denoising in combination with deep learning techniques has achieved notable progress and has become a topic of interest in recent years. Furthermore, industry has actively embraced the latest achievements. For example, in the movie industry, Pixar's RenderMan [7] adapted adaptive sampling and denoising filters in the production of Toy Story 4. Another example [8] in the gaming industry generates high-quality images with low sample counts for real-time use.

This report summarizes state-of-the-art techniques in deep learning-based MC denoising. We start with a direct introduction to the basic concepts and then

[1] KAIST, Daejeon, 31414, Republic of Korea. E-mail: Y. Huo, huo.yuchi.sc@gmail.com; S.-e. Yoon, sungeui@kaist.edu (✉).

Manuscript received: 2020-12-25; accepted: 2021-01-23

discuss the components of the area in detail (see Section 2). Afterwards, we provide a comprehensive overview which categorizes the related research into three topics:

- pixel denoising (Section 3),
- nontrivial-domain denoising (Section 4), and
- high-dimensional denoising (Section 5).

We conclude this report by summarizing and comparing the discussed techniques (see Section 6).

2 Deep learning-based Monte Carlo denoising concepts

Classical MC rendering estimates some target c, e.g., a pixel's color, through MC integration, as the sum of the contributions from M samples in some domain Ω, e.g., a pixel:

$$c = \int_{\Omega} f(s)\,\mathrm{d}s \approx \frac{1}{M}\sum_{m=1}^{M}\frac{f(s_m)}{p(s_m)} \tag{1}$$

where $f(s_m)$ and $p(s_m)$ denote the contribution and the sampling probability of the m-th sample, s_m, on the pixel, respectively. This kind of general MC integration produces estimation variance with low sample counts, leading to visually annoying noise. The problem inherently motivates the development of MC denoising techniques to filter the noisy input to achieve a plausible rendering quality with a reasonable time budget.

MC denoising can be formally described as a mapping g of an input x to the ground-truth r rendered by a high sample count (Fig. 1). In the most common case, x is a tuple correlated with a shading point p, such as a pixel, as $x_p = \{c_p, f_p\}$, where c_p represents noisy values achieved with low sample counts and f_p represents auxiliary features, e.g., surface normal or textures over multiple samples contributing to p. While using deep learning, the pursuit of optimal g can be formulated as the training of a neural network parameterized by a set of weights θ representing g. Through a supervised learning process that utilizes a dataset with N example pairs of $(x^1, r^1), \ldots, (x^N, r^N)$, the estimated parameters $\hat{\theta}$ are optimized via a loss function ℓ as

$$\hat{\theta} = \min \frac{1}{N}\sum_{n=1}^{N}\ell(r^n, g(X^n;\theta)) \tag{2}$$

where X^n is a block of per-pixel vectors in the neighborhood of x^n to produce the reconstructed output at pixel x^n [9]. In reference, the trained network takes seconds or minutes to generate $\hat{r}^n = g(X^n;\hat{\theta})$, a visually plausible approximation to the ground-truth that requires hours of rendering. Despite a lack of rigorous analysis of guarantees of mathematical convergence, this approximation reforms production pipelines by enabling rendering with a quality visually indistinguishable to the ground-truth, at a much faster speed, and will approaching an interactive rate in the near future.

3 Pixel denoising

3.1 Approaches

This section covers the approaches for a basic application scenario of MC denoising, the reconstruction of a single smooth image with the help of auxiliary features and noisy inputs. The neural networks take as input an image with noisy per-pixel colors, usually samples' average radiance estimated by path tracing [1], and predict the corresponding smoothed image. Because the results of most MC integration methods can be stacked in image space, directly denoising the per-pixel colors can work as a general post-processing add-on to

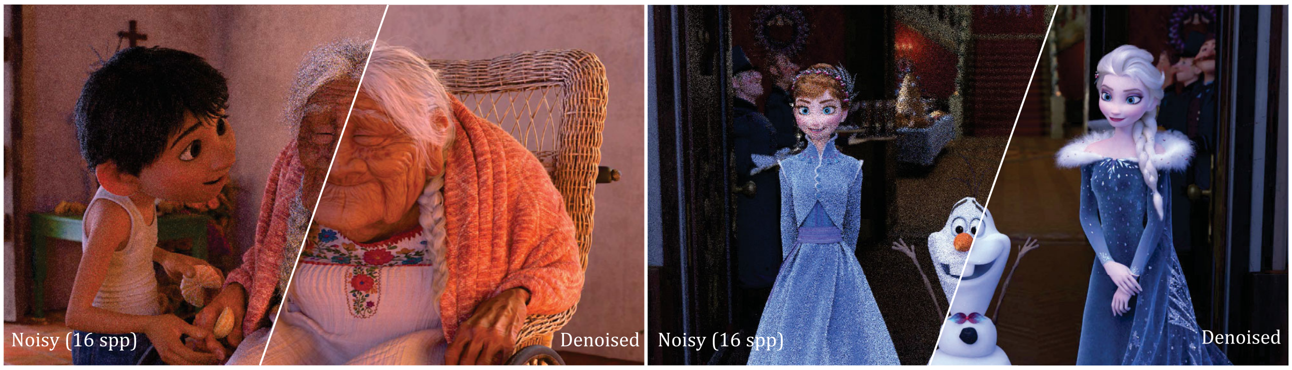


Fig. 1 Deep learning-based Monte Carlo denoising method trains a neural network to reduce Monte Carlo noise in input images. Reproduced with permission from Ref. [9], © Author 2018.

existing rendering pipelines without the need to reorganize data flows. Thus, pixel denoising rapidly became a popular solution both in academia and industry.

We categorize the research in pixel denoising according to prediction targets of neural networks, which imply the underlying problem formulation used by the denoising process. Overall, we categorize them as performing parameter prediction, radiance prediction, or kernel prediction. Table 1 summarizes the related papers.

3.2 Parameter prediction

An early attempt to utilize deep learning in MC denoising was motivated by a desire to predict optimal parameters of conventional MC filters [10]. Before this paper, the most successful MC denoising methods were based on handcrafted filters using additional scene features such as shading normals and texture albedo. The existing challenge was to optimize the parameters, i.e., filter bandwidths, of the filter models to reduce noise yet preserve scene details.

Kalantari et al. [10] observed that there is a complex relationship between noisy scene data and ideal filter parameters, and proposed to learn this relationship through deep learning. Their method uses cross-bilateral and cross non-local mean filters of various auxiliary features (world positions, shading normals, texture values, etc.) for the final reconstruction and a multilayer perceptron (MLP) neural network [11–13] to predict optimal weights for each feature in the filter. To use the framework, an MLP is first trained in an offline process on a set of noisy images of scenes with a variety of distributed effects to regress the optimal filter parameters that minimize the difference between the reconstructed output and the ground truth. At run-time, the trained network can then predict the filter parameters for new scenes to produce filtered images in just a few seconds. As shown in Fig. 2, the results were superior to previous approaches on a wide range of distributed effects such as depth of field, motion blur, area lighting, glossy reflections, and global illumination.

Xing and Chen [14] also adapted a parameter estimation network to address noise artifacts from path tracing. The method contains sampling and reconstruction stages. Stein's unbiased risk estimator (SURE) [15] is adopted to estimate the noise level per pixel that guides an adaptive sampling process. A modified MLP network is then used to predict the optimal reconstruction parameters. In the sampling stage, coarse samples are firstly generated, and then a noise level map is estimated with SURE to guide additional sampling. In the reconstruction stage, the modified MLP network is adopted to predict optimal reconstruction parameters of anisotropic filters for the final images, using the extracted features.

3.3 Kernel prediction

Based on the observation that predicting parameters of conventional handcrafted filters establishes local reconstruction kernels for pixels in an indirect way, another group of fruitful investigations aimed to directly predict local reconstruction kernels through kernel-predicting networks [9, 16, 17].

Explicit filters are useful for exploiting conventional MC denoising models, but may limit denoising capability even when using deep neural networks to

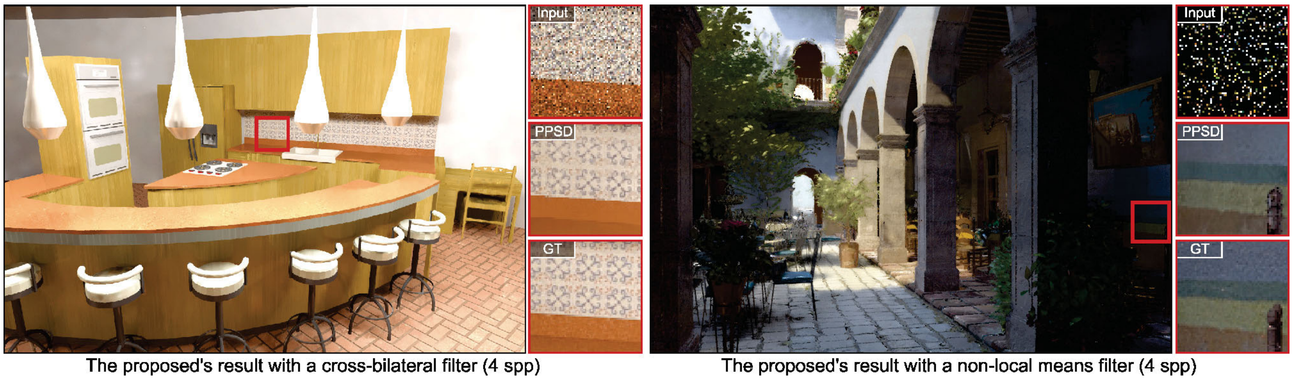

Input
PPSD
GT
The proposed's result with a cross-bilateral filter (4 spp)
Input
PPSD
GT
The proposed's result with a non-local means filter (4 spp)

Fig. 2 Result of using the trained network of Kalantari et al. [10] (PPSD) to drive a filter for denoising a new MC rendered image, with a cross-bilateral filter for the Kitchen scene (1200×800, left) and with a non-local means filter for the San Miguel Hallway scene (800×200, right). Both of these scenes were path-traced and contain severe noise at 4 samples per pixel. The trained network is able to estimate the appropriate filter parameters and effectively reduce the noise in only a few seconds. Reproduced with permission from Ref. [10], © ACM 2015.

predict the optimal parameters [16]. To address this problem, Bako et al. [16] proposed a novel, supervised learning approach that allows the filtering kernel to be more complex and general by leveraging a deep convolutional neural network (CNN) architecture [18, 19]. The approach introduced a novel, kernel-prediction network which employs the CNN to estimate the local weighting kernels used to compute each denoised pixel from its neighbors. The results demonstrate an improved accuracy compared to parameter-predicting MC denoisers and roughly 5–6 times faster convergence of the weighted kernel prediction than for direct radiance prediction. Other training techniques have been widely adopted, and some of them include decomposition of diffuse and specular components, separation of albedo from network prediction, and logarithmic transformation of specular color (Fig. 3).

Vogels et al. [9] expanded the capabilities of kernel-predicting networks using asymmetric loss functions that are designed to preserve details and provide the user with direct control over the variance-bias trade-off during inferencing. They also reconstituted the pipeline with some task-specific modules, including four distinct components. First, a source-aware encoder extracts low-level features and embeds them into a common feature space, enabling quick adaptation of a trained network to novel data. Second, spatial and temporal modules extract abstract, high-level features for kernel-based reconstruction. Third, a complete network is designed to preserve details and provide the user with direct control over the variance-bias trade-off during inferencing. Fourth, an error-prediction module infers reconstruction error maps for adaptive sampling. This modular design enables a production level MC denoising framework in terms of detail preservation, low-frequency noise removal, and temporal stability for processing various production and academic datasets. Finally, they shed light on the academic research by offering a theoretical analysis of convergence rates of kernel prediction architectures.

MC denoisers, also known as biased MC estimators, reduce MC noise by exploring the correlation between nearby pixels. As a result, they suffer from method-specific residual noise or systematic errors. Back et al. [20] aimed to mitigate such remaining errors by unifying an independent unbiased estimator and a correlated biased estimator with a kernel-predicting neural network. Their framework takes a pair of images, one with independent estimates, and the other with the corresponding correlated estimates generated by existing MC denoisers. A neural network is trained to exploit the correlation between these two pixel estimates and output a combination kernel for the weighted reconstruction of final images. The results of the unified framework outperform both single estimators both visually and numerically.

3.4 Radiance prediction

Parameter-predicting and kernel-predicting frameworks generally have achieved great success, but the kernel filtering scheme sometimes imposes restrictions on flexible fusion with state-of-the-art deep learning techniques. Therefore, another natural evolution of deep learning-based MC denoising trains neural networks to directly predict per-pixel color, i.e., the outgoing radiance towards the viewpoint at each footprint.

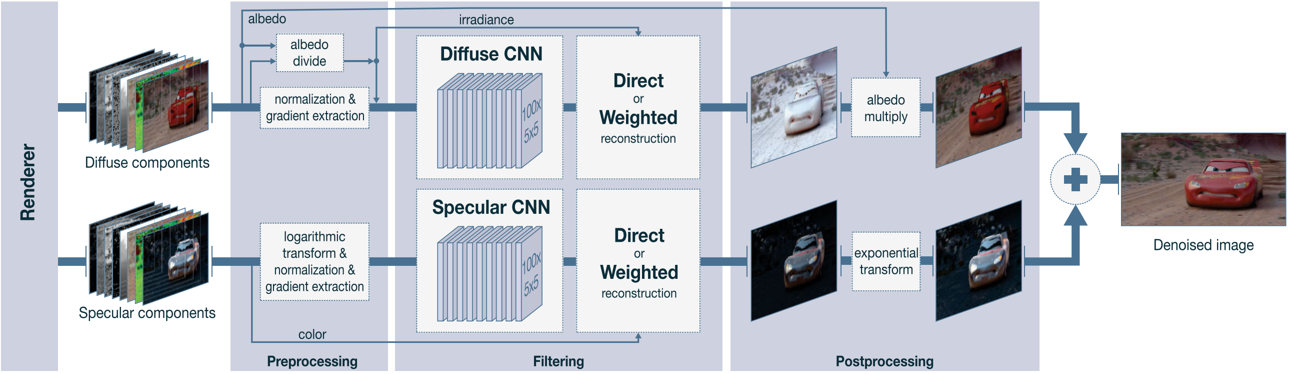


Fig. 3 Overview of the kernel-predicting framework [16]. It starts by independently preprocessing diffuse and specular data from the rendering system, and then feeds the information to two separate networks to denoise the diffuse and specular illumination, respectively. The output from each network undergoes reconstruction (direct reconstruction or weighted reconstruction through the predicted kernels) and postprocessing before being combined to obtain the final, denoised image. Reproduced with permission from Ref. [16], © Author 2017.

While most MC denoising methods rely on handcrafted optimization objectives like MSE or MAPE loss, which do not necessarily ensure perceptually plausible results, Xu et al. [21] presented an adversarial approach for MC denoising, following an insight that generative adversarial networks (GANs) [22, 23] can guide neural networks to produce more realistic high-frequency details. The adversarial approach to evaluate the reconstruction is based on the Wasserstein distance to measure perceptual similarity, which can be interpreted as the distance between the denoised and ground truth distributions. In addition, they adapt a feature modulation method to encode auxiliary features that allow features to better take effect at the pixel level, leading to fine-grained denoising results. Another GAN-based denoising method also considers denoising rendered images from a dataset containing 40 Pixar movie image frames with added Gaussian noise [24]. Because the network does not take auxiliary features as input, it can also denoise noisy photographs under natural light and CT scans.

A deep residual network (ResNet) [25] demonstrates significant improvement over a basic CNN. In order to take advantage of ResNet, a filter-free direct denoising method based on a standard-and-simple deep ResNet is trained to remove the noise of MC rendering [26]. The method directly maps the noisy input pixels to the smoothed output with only three common auxiliary features (depth, normal, and albedo), simplifying its integration into most production rendering pipelines. With the help of ResNet, the simple structure yields comparable accuracy to other state-of-the-art methods.

One distinguishing difference between MC denoising and natural image denoising is that auxiliary features, e.g., normals, can be extracted from the rendering pipeline, providing noise-free guidance for image reconstruction. However, the auxiliary features also contain redundant information, which reduces the efficiency of deep learning-based MC denoising. Yang et al. [27, 28] focused on how to extract useful information from auxiliary features. To address this problem, they first introduced an end-to-end CNN model to fuse feature buffers and predict a residual radiance map between noisy input and ground truth to reconstruct a final image. In addition, a high-dynamic range (HDR) image normalization method is proposed to train the model on HDR images in a more efficient and stable way [27]. In follow-up research, they proposed an autoencoder [29, 30] inspired network structure, a dual-encoder network with a feature fusion subnetwork, to first fuse auxiliary features. The fused features and a noisy image are then fed as inputs to a second encoder network to reconstruct a clean image by a decoder network [28]. Compared to conventional solutions using uncompressed auxiliary features, the method is able to generate satisfactory results in a significantly faster way.

While deep learning-based MC denoisers dramatically enhance rendering quality, the results are less reliable when there is insufficient information to calculate the features, such as variance and contrast. To address this issue, Kuznetsov et al. [31] proposed a deep learning approach for joint adaptive sampling and reconstruction of MC rendering results with extremely low sample counts. In addition to a conventional MC denoising network, they train a CNN to estimate sampling maps for guiding adaptive sample distribution over pixels. Finally, the denoising network produces denoised images from the adaptively sampled MC rendering results.

4 Nontrivial-domain denoising

4.1 Background

Conventional MC denoisers work in image space, where the basic geometric auxiliary features can be easily extracted from most rendering pipelines. This accessibility makes pixel-based MC denoisers a prevailing choice. However, the physical process of light transport occurs in a high-dimensional space where some important information is inevitably degraded when reducing everything into per-pixel radiance. To address this, a research stream aims to recover the lost information by utilizing various nontrivial domains, e.g., sample space and gradient domain, for high-quality rendering of illumination details or challenging effects. This section discusses the related approaches using nontrivial-domain features and their advantages in single-image denoising. The summary of these papers is in Table 1.

4.2 Sample space

In contrast to the traditional pixel-based MC denoisers, Gharbi et al. [17] proposed a sample-based kernel-splatting network. They observed that traditional MC denoisers exploit summary statistics of a pixel's sample distributions, which discards much of the samples' information and limits their denoising power. The proposed kernel-spatting network, learning the mapping between samples and images, embraces unfamiliar network architecture design to solve multiple challenges associated with the sample space: the order of the samples is arbitrary, and those samples must be treated in a permutation invariant manner. Instead of conventional gathering kernels, they suggested predicting spatting kernels that splat individual samples onto nearby pixels using a convolutional neural network. They claimed that, in addition, splatting is a natural solution to situations such as motion blur, depth-of-field, and many light transport problems, where it is easier to predict which pixels a sample contributes to, rather than to predict gathering kernels that need to determine informative relationships between relevant pixels. The new architecture yields higher-quality results both visually and numerically for low-sample count images and distributed-effect images.

Per-sample denoisers come with high computational costs because of the need to produce kernel weights and apply a large kernel for each sample in each pixel, which can limit its usability for higher sample counts. Based on this observation, Munkberg and Hasselgren [32] proposed to extract a compact representation of per-sample information by separating samples into a fixed number of partitions, called layers in their paper, via a data-driven method that learns unique kernel weights for each pixel in each layer and how to composite the filtered layers. This modification gives a practical denoiser the capability to strike a good trade-off between cost and quality. Furthermore, it provides an efficient way to control performance and memory characteristics, since the algorithm scales with the number of layers rather than the number of samples. Using two partitioned sample layers, the denoiser achieves interactive rates while producing image quality similar to larger networks.

Assuming that next event estimation (NEE) [33] is used in the rendering process, Lin et al. [34] decomposed the features of Gharbi et al. [17] into sample- and path-space features, where one-bounce paths are sample-space features and multi-bounce paths are path-space features. The key insight of the separation is to decompose the high-frequency illumination from short paths and low-frequency illumination from long paths. The three-scale features—pixel, sample, and path—are combined to preserve sharp details, using a feature attention mechanism and feature extractors.

4.3 Light field space

Most MC denoisers only use as features the outgoing radiance of samples in each pixel, while each sample is in fact a high-dimensional light path with information about the light field [35]. Lin et al. [36] observed that these methods show powerful denoising ability, but tend to lose geometric or lighting details and to blur sharp features during denoising. Based on the definition of the local light field (Fig. 5), the authors adapted a framework [37] for frequency analysis of light transport by calculating the frequency content of the local light field around a given ray. The local light field is defined as a 4D function, with two spatial dimensions and two angular dimensions. In the analysis, light transport operations, such as transport in free space or reflection, are transformed

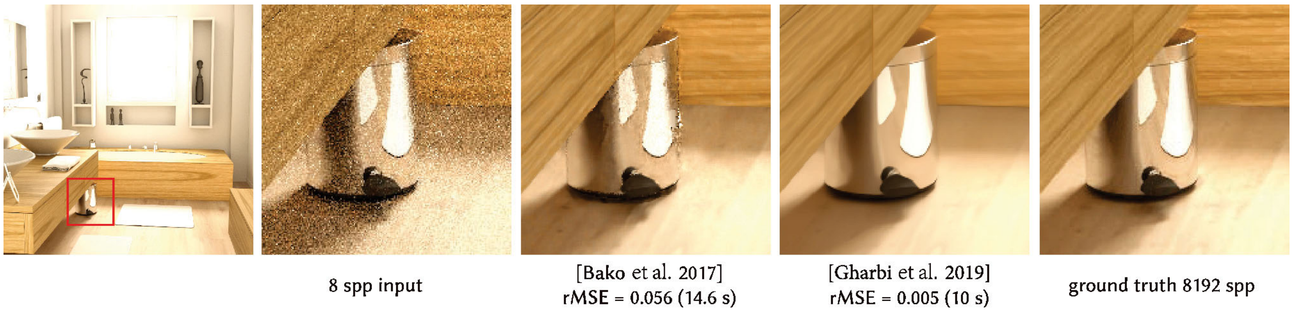


Fig. 4 Comparison between state-of-the-art pixel-based (Bako et al. [16]) and sample-space (Gharbi et al. [17]) MC denoising algorithms. The sample-space method works with the samples directly, using a splatting approach that makes it possible to appropriately handle various components of the illumination (indirect lighting, specular reflection, motion blur, depth of field, etc.) more effectively. Reproduced with permission from Ref. [17], © ACM 2019.

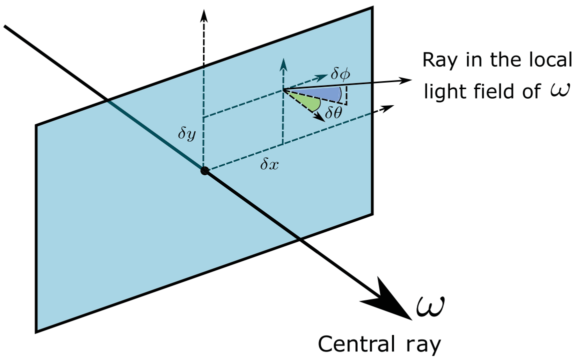


Fig. 5 The local light field is defined as a 4D function around the center ray (ω), parameterized by two spatial coordinates (δ_x and δ_y) and two angular coordinates (δ_θ and δ_ϕ). Reproduced with permission from Ref. [39], © IEEE 2020.

into operations on the Fourier spectrum, then approximately represented by the Fourier spectrum of the local light field, the covariance matrix [38]. A neural network makes use of this covariance matrix, a 4×4 matrix encoding the Fourier spectrum of the local light field at each pixel, to leverage the directional light transportation information. In addition, the author proposed a network extracting feature buffers separately from the color buffer and then integrated the two buffers into a shallow kernel predictor. Finally, they suggested an improved loss function considering perceptual loss. These modifications help to preserve illumination details.

Instead of using light-field-space features for image-space denoising, another category of research aims to directly reconstruct the denoised incident radiance field, i.e., the local light field at each pixel, for advanced goals such as unbiased path guiding [40–42]. We cover such works in Section 5.4.

4.4 Gradient domain

Gradient-domain rendering methods [43–45] develop a common denoising idea of estimating finite difference gradients of image colors to solve a screen-space Poisson problem. The gradient-domain information is believed to offer additional benefits because of the frequency content of the light transport integrand and the interplay with the gradient operator. Recent work combines this long-existing research direction with modern CNNs [46]. The new method replaces the conventional screened Poisson solver with a novel dense variant of the U-Net autoencoder, taking auxiliary feature buffers as inputs and using a perceptual image distance metric as loss function. The combination significantly improves the quality obtained from gradient-domain path tracing and yields notably improved image quality compared to simple image-space MC denoisers.

In other independent work, Guo et al. [47] proposed using a multi-branch autoencoder to replace the Poisson solver. The network end-to-end learns a mapping from a noisy input image and its corresponding image gradients to a high-quality image with low variance. One distinguishing feature of this work is that the authors train the network in a completely unsupervised manner by adjusting a non-trivial loss function between the noisy inputs and the outputs of the network. The loss function combines an energy function including a data fidelity term, a gradient fidelity term, and a regularizer constructed from selected rendering-specific features. In this way, the approach avoids the tedious and sometimes expensive rendering process to generate noise-free images for training, making it a technically unsupervised solution.

4.5 Photon denoising

While path tracing is a general MC integration approach for realistic rendering, it is not effective for simulating challenging light transport effects like caustics. Instead, photon mapping [48, 49] has been considered as the method of choice for rendering caustics, but it has not completely adopted progress in deep learning techniques. Some recent research bridges this gap by training a deep neural network to predict a kernel function aggregating photon contributions at each shading point [50]. Photon mapping traces a large number of photons from the light source, and then gathers the photon contributions at each shading point to achieve high-quality reconstructions of challenging light transportation results which are hard to trace from the camera. The authors mitigate the required number of photons with a network encoding individual photons into per-photon features, aggregating them in the neighborhood of a shading point to construct a photon local context vector, and inferring a kernel function from the per-photon and photon local context features. This work combines conventional deep learning-based denoisers for remaining light transport paths. The results show promising high-quality reconstructions of caustic effects with an order of magnitude fewer photons than previous photon mapping methods and significantly outperform path tracing-based MC rendering for caustic effects (Fig. 6).

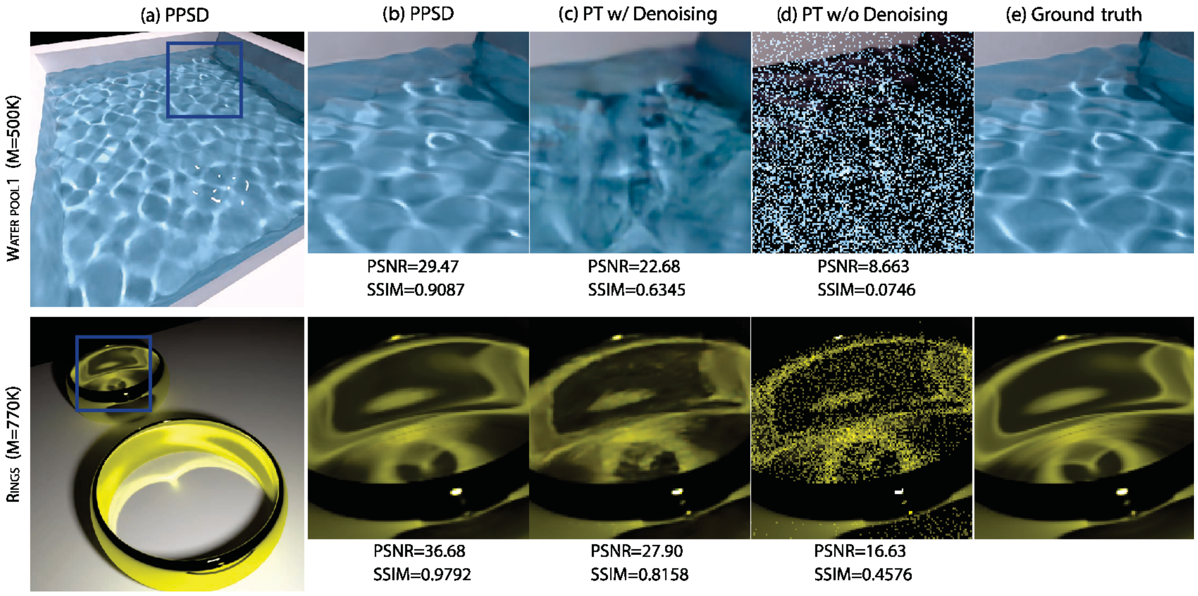


Fig. 6 Results of photon mapping denoising show high-quality reconstruction of caustic effects [50]. (a,b) Final results of the proposed method (PPSD). (c) Path-tracing results with an image-space denoiser [8]. (d) Results without denoiser. (e) Ground truth. Reproduced with permission from Ref. [50], © The Eurographics Association and John Wiley & Sons Ltd. 2020.

Stochastic progressive photon mapping [51] is an important global illumination method derived from photon mapping. It can simulate caustic effects in a progressive way, but suffers from both bias and variance with limited iterations, leading to visually annoying MC noise. Zeng et al. [52] recently proposed a deep learning-based method specially designed for denoising the biased renderings of stochastic progressive photon mapping. The method decomposes the light transport into two components, caustic and other, and denoises each part independently. It also employs additional photon-related auxiliary features and multi-residual blocks to enhance kernel predicting neural networks.

5 High-dimensional denoising

5.1 Overview

Single-image MC denoisers take as input one noisy image to produce one high-quality output without MC noise. However, such single-image output does not satisfy many applications that require higher-dimensional outputs. For example, producing computer animation requires a sequence of temporally consistent images, and path guiding to generate unbiased rendering results might need to denoise the whole incident light field at each shading point. In such scenarios, pixel-based MC denoisers are no longer adequate to generate the high-dimensional outputs without special designs for high-dimensional signal processing and consistency constraints. Here we categorize deep learning-based MC denoisers targeting high-dimensional applications into three types, temporal rendering, volume rendering, and radiance field reconstruction, and discuss each in detail. The related papers are summarized in Table 1.

5.2 Temporal rendering

One of the most important MC rendering applications is to generate a sequence of images for computer animation or interactive applications. Among many single-image denoisers, some focus on rendering quality, and others pay additional attention to the balance between quality and speed to achieve an interactive processing rate. Besides speed, an essential consideration is to enhance temporal stability between frames, to avoid low-frequency variances that may lead to flicking artifacts in animation. Pioneering research in this area was inspired by the good results achieved by recurrent neural networks (RNNs) [53, 54] in the context of video super-resolution and sub-pixel CNNs, and describes an RNN-based framework that dramatically improves temporal stability for sequences of sparsely sampled input images [8].

Compared to single-image denoisers, an RNN network takes sequential images as input to explore and impose constraints on temporal consistency. Its primary focus is on the reconstruction of global illumination with extremely low sampling budgets, at interactive rates. The primary novelty is the addition of recurrent connections to the network to improve temporal stability between frames. In addition, some modifications are suggested for processing MC noise, allowing larger pixel neighborhoods while improving the execution speed by an order of magnitude compared to a naive solution. The method shows impressive high-quality results at interactive rates and a promising future for high-quality real-time denoisers (Fig. 7).

Hasselgren et al. [58] combined temporal denoising with adaptive sampling to achieve high-quality rendering with high-frequency details. They proposed an adaptive rendering method that distributes samples via spatiotemporal joint training of neural network-based sample predictors and MC denoisers

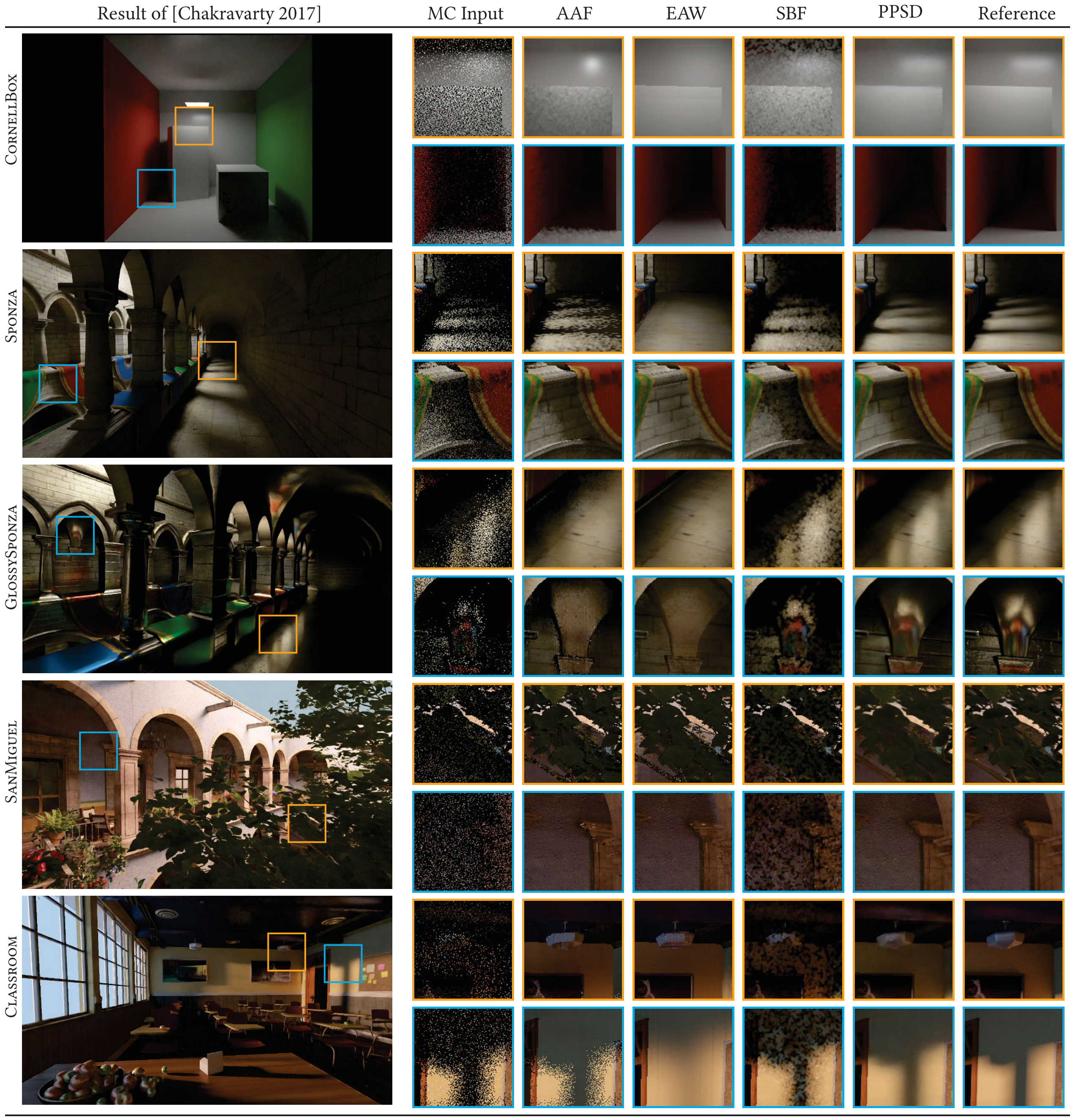


Fig. 7 Closeups of 1-bounce global illumination results for 1 spp input (MC), axis-aligned filter [55] (AAF), ÃĂ-Trous wavelet filter [56] (EAW), SURE-based filter [57] (SBF), and a deep learning-based denoiser [8] (PPSD). Compared to conventional methods, the deep learning-based MC denoiser yields higher rendering quality and temporal stability [70]. Reproduced with permission from Ref. [8], © ACM 2017.

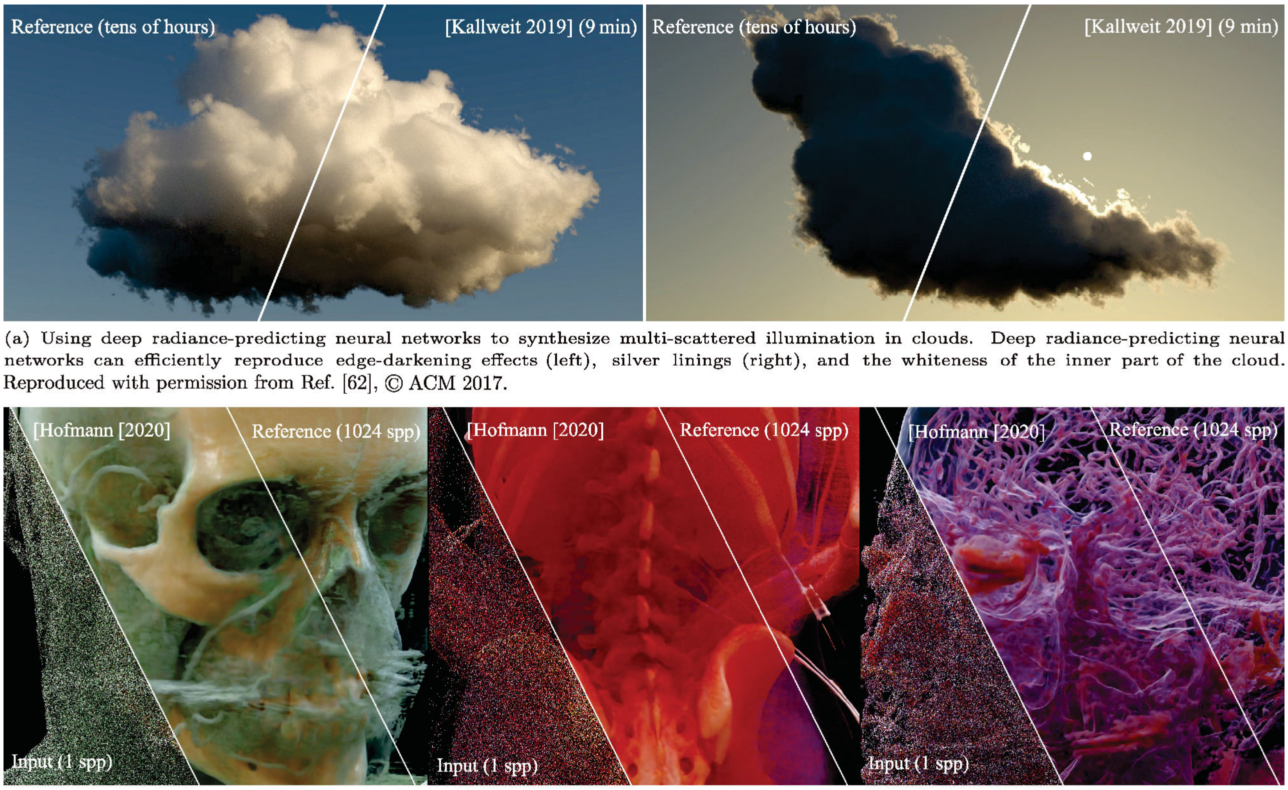
over multiple consecutive frames, increasing temporal stability and image fidelity. An optimized sample predictor enables the learning of spatio-temporal sampling strategies, which helps the rendering engine to adaptively place more samples in unoccluded regions or track specular highlights, where high-frequency details are hard to reconstruct. Such a framework allows a trade-off between quality and performance, while running at near real-time rates.

Meng et al. [59] focused on computation speed and proposed a novel and practical real-time approach to denoise noisy inputs in a data-dependent bilateral space, where the differentiable grid enables end-to-end training of denoising tasks. The proposed neural network learns to generate a guide image for first splatting noisy samples into the grid and then slicing it to read out the denoised data. In such a way, the approach avoids the explicit computation of per-pixel weights for large kernels. It achieves high-quality denoising with fast, spatially uniform filters, leading to significantly improved speed compared to basic kernel-prediction techniques.

While the aforementioned kernel-predicting neural network proposed by Vogel et al. [9] also contains a temporal denoiser module to boost temporal stability, the authors focus on animation rendering, which is slightly different from interactive rendering in terms of future frame visibility. In animation rendering, the temporal consistency constraints can be imposed on a temporal window, where the spatial features from previous and future individual frames can be extracted and warped, using motion vectors, to match the center frame. In this case, there is no need to insert recurrent connections to the module.

5.3 Volume rendering

As an important sub-category of realistic rendering, volume rendering [60, 61] significantly contributes to a wide variety of vivid visual effects for participating media, such as clouds, fogs, liquids, transparent solids, and medical data (Fig. 8). However, such rendering is usually conducted in 3D space, where a tremendous amount of light transport and scattering among particles can occur, causing difficulties or performance degradation for conventional image-space MC denoisers. Some recent research aims to adapt deep learning techniques to generate high-quality volume rendering images in



(a) Using deep radiance-predicting neural networks to synthesize multi-scattered illumination in clouds. Deep radiance-predicting neural networks can efficiently reproduce edge-darkening effects (left), silver linings (right), and the whiteness of the inner part of the cloud. Reproduced with permission from Ref. [62], © ACM 2017.

(b) Deep learning-based MC denoising of medical volumetric data from noisy input, compared to the converged ground truth. Reproduced with permission from Ref. [65], © ACM 2020.

Fig. 8 The combination of volume rendering and deep learning techniques can produce high-quality rendering results with low rendering cost.

3D space. Such methods share the same spirit as the previously discussed MC denoisers, using deep learning techniques to generate smooth images from a small number of MC samples.

Rendering clouds is considered to be a very challenging and time-consuming problem due to the intricacy of Lorenz-Mie scattering and the high albedo. In order to efficiently synthesize images of atmospheric clouds using a combination of MC integration and neural networks, Kallweit et al. [62] approached the problem in a data-driven way. They trained a neural network with residual connections, to predict the spatial and directional distribution of radiant flux from an offline dataset containing tens of cloud examples. In inferencing, the network takes as input visibility sample points of the cloud in a new scene to predict the radiance function for each shading configuration. The method contributes a key novelty that each visible sample contains a feature consisting of a hierarchical 3D descriptor of the cloud geometry with respect to the shading location and light source. While synthesizing images, the method stochastically samples the first scattering interaction with delta tracking, estimates direct in-scattering via MC integration, and predicts indirect in-scattering with the neural network.

The performance of the deep learning-based cloud rendering approach was later improved by decomposing the neural network architecture into some parts that can be precomputed and other parts that should be inferred at runtime. Panin and Nikolenko [63] introduced a latent space light probe approach that uses a separate neural network which accepts as input a descriptor of a grid cell in the cloud, and outputs the light probe for baking light probes offline. At runtime, the method uses a separate rendering network that takes as input a light probe and a much smaller 3D descriptor. Because collecting 3D descriptors takes about half of the total rendering time, using light probes to collect 3D descriptors and minimizing the size of 3D descriptors dramatically reduce the overall computation, yielding 2–3 times speedup over the previous approach.

Xu et al. [64] jointly leveraged gradient-domain information and photon mapping techniques for rendering homogenous participating media. They adopt an unsupervised gradient-domain deep learning framework [47] for image reconstruction of gradient-domain volumetric photon density estimation. The modified network contains encoded shift connections and takes as input a separated auxiliary feature branch which includes volume-based auxiliary features such as transmittance and photon density. The proposed method produces state-of-the-art rendering quality in volumetric photon mapping.

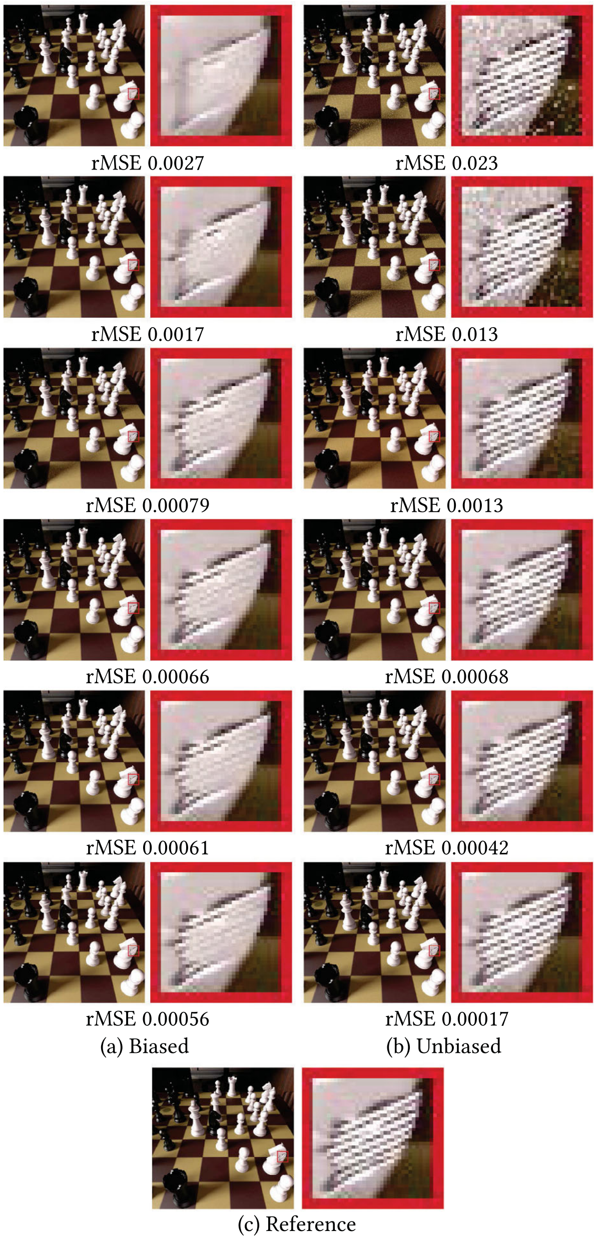
In the domain of medical imaging, MC rendering has turned out to be an efficient means to visualize and understand internal structures, especially for inexperienced users such as medical students, forensic staff, and patients. However, auxiliary features like depth and normal, vital for surface-based MC denoisers, are neither well-defined nor smooth for medical volumetric data. To address this, Hofmann et al. [65] modified surface-based MC denoisers for path-traced visualizations of medical volumetric data. Although noisy, special auxiliary features, such as model space position, world space normal, albedo and descriptors of first and second scattering events, are fed as guiding inputs to the neural network and contribute to generating high-quality rendering results from noisy images. Furthermore, the authors proposed a loss function specifically defined for a sharp reconstruction of specular highlights, and a GAN-inspired dual autoencoder architecture to enhance sharp edges and details like specular highlights, which are essential for interpretation. The overall architecture also considers temporal stability of video via feature reprojection between frames.

5.4 Radiance field reconstruction

Modern pixel-based MC denoisers have prevailed in a great range of rendering applications with satisfactory visual results. The denoised results, however, are mathematically biased estimates without a convergence guarantee, even if using hundreds or thousands of samples per pixel. In order to push the rendering quality to the edge for applications that are sensitive to numerical accuracy and visual fidelity, such as physical simulation, ground truth data generation, and high-quality rendering, some orthogonal approaches have pursued the ultimate in rendering quality via unbiased MC estimators. Recently, deep learning-based techniques have been used to reconstruct radiance fields to guide path tracing, under the name of path guiding [66, 67], for efficiently generating high-quality images with relatively large numbers of samples.

Bako et al. [40] noted that even modern deep learning-based MC denoisers do not produce acceptable final results for high-quality rendering, and turned to the recent path guiding techniques that aim to predict the incident radiance field at each pixel, which enables use of a guided probability distribution function (PDF) for first-bounce importance sampling. While existing path guiding approaches involve expensive online learning and offer benefits only at high sample counts, the authors proposed an offline, scene-independent deep learning-based approach that can importance-sample first-bounce light paths for general scenes. The predicted incident radiance field contains high-dimensional directional incident radiance information to directly modulate a per-pixel guiding PDF for unbiased MC integration. This increases the efficiency of sampling by putting more samples in informative directions, e.g., unoccluded regions. The primary advantage of offline learning is that it uses a data-driven scheme to learn a priori from a large set of training scenes, for reconstructing the full incident radiance by reusing nearby samples; thereby, the expensive online learning process that uses a large number of samples to fit a new scene can be abandoned. As reference, the trained network takes a small number of uniform initial samples as input to predict the full incident radiance field of each pixel, which is used to guide the remaining samples to generate the final results.

Instead of single-pass path guiding, another method takes a progressive adaptive sampling strategy that iteratively uses last-iteration samples to guide the sampling process for the next iteration [41]. In order to guide the progressive sampling process, the method considers the sampling as an action that can produce rewards, i.e., reducing reconstruction errors, and trains a quality-predicting neural network to predict the gain of different actions in a deep reinforcement learning (DRL) way [68, 69]. Via this action-based dynamic formulation, the quality-predicting neural network can learn an optimal sampling strategy from an offline dataset, using progressive sampling contexts in unseen scenes. The method decomposes the overall sampling process into two atomic sampling actions, doubling samples and refining directional resolution, and then uses the quality-predicting neural network to predict dynamic rewards of the two actions in different directions of pixels. In order to reconstruct the incident radiance field from the adaptive samples, the authors trained a CNN-based 4D neural network to generate a denoised radiance field for each pixel, which is used to guide path tracing in subsequent iterations. In general, the deep learning-guided unbiased sampling process guarantees mathematical convergence with sufficient samples, resulting in higher rendering quality compared to that from MC denoisers and other unbiased rendering approaches (Fig. 9).



(a) Biased (b) Unbiased

(c) Reference

Fig. 9 Equal-time comparisons between a biased MC denoiser (Bako et al. [16]) and unbiased path guiding using deep learning to generate guidance (Huo et al. [41]). Results are rendered within 1, 2, 8, 30, 60, and 120 minutes from top to bottom. While the MC denoiser converges faster with low sample counts, the deep learning-guided sampling method outperforms the MC denoiser with more samples and converges to the reference. Reproduced with permission from Ref. [41], © ACM 2020.

Denoised radiance fields can also be directly integrated into pixel colors for biased rendering [42]. The method uses an autoencoder neural network to denoise low-sample radiance caches for rendering indirect illumination, and then progressively increases samples to refine the radiance caches.

6 Conclusions

Thanks to the benefits from the unprecedented success of deep learning techniques, MC denoising has attracted strong attention in recent years. These techniques are naturally compatible with MC integration, one of the most general rendering frameworks used by many rendering pipelines. In general, only minor modification is required to extract auxiliary features, a wide range of applications is supported, they are scalable to both high-quality and performance-sensitive rendering, often GPU and TPU-friendly, and above all, they dramatically decrease the cost of MC rendering. Table 1 provides an overview of the methods discussed in this survey. For classifying different techniques, we use the following attributes in the summary table:

- *Rendering goals.* The exact goals of the neural networks or systems with respect to the entire rendering pipeline. Possible attributes of those specific targets include: PD, pixel denoising; RD, radiance denoising; VD, volumetric data denoising; AS, adaptive sampling; DE, rendering distributed effects; SD, sequential image denoising; and CA, rendering caustic effects.
- *Network inputs.* The type of features the neural networks take as input. Possible attributes include: P, noisy pixel colors generated by MC integration using a small number of samples per pixel; A, geometry- or scene-related auxiliary features such as surface normals, world positions, and texture albedo; S, sample colors defined on each MC sample rather than each pixel; R, radiance-field

Table 1 Summary of papers in Sections 3, 4, and 5. Attributes and abbreviations are given in Section 6

Method	Goal	Input	Predict	Domain	Speed	Remark
Kalantari et al. [10]	PD, DE	PA	parameter	image	O, 4	MLP, cross-bilateral and non-local means filters
Xing and Chen [14]	PD, AS	PA	parameter	image	O, 8	MLP, SURE, cross-bilateral filter
Bako et al. [16]	PD	PA	kernel	image	O, 16	CNN, kernel-predicting network
Vogels et al. [9]	PD, SD, AS	PA	kernel	image	O, 16	CNN, asymmetric loss functions
Back et al. [20]	PD	PA	kernel	image	O, 32	CNN, combine pixel estimates
Xu et al. [21]	PD	PA	radiance	image	O, 4	CNN, GAN, feature modulation, perceptual loss
Alsaiari et al. [24]	PD	P	radiance	image	O, 1	CNN, GAN
Yang et al. [27]	PD	PA	radiance	image	O, 4	CNN, HDR tonemapping
Yang et al. [28]	PD	PA	radiance	image	O, 4	CNN, feature encoder
Wong and Wong [26]	PD	PA	radiance	image	O, 8	CNN, ResNet
Kuzenetsov et al. [31]	PD, AS	PA	radiance	image	O, 5	CNN, autoencoder
Gharbi et al. [17]	PD, DE	SA	kernel	sample	O, 4	CNN, U-Net, kernel-splatting network
Munkberg and Hasselgren [32]	PD, DE	SA	kernel	sample	I, 8	CNN, layered embedding
Lin et al. [34]	PD	SA	kernel	sample	O, 1	CNN, three-scale features, attention mechanism
Lin et al. [36]	PD	PAR	kernel	radiance	O, 4	CNN, light transport covariance
Kettunen et al. [46]	PD	PAG	radiance	gradient	O, 4	CNN, U-Net, perceptual loss
Guo et al. [47]	PD	PAG	radiance	gradient	O, 4	CNN, unsupervised learning
Zhu et al. [50]	PD, CA	PAO	kernel	photon	O, 1	CNN, caustic decomposition
Zeng et al. [52]	PD, CA	PAO	kernel	photon	O, 1	CNN, caustic decomposition
Chaitanya et al. [8]	SD	PA	radiance	temporal	I, 1	RNN, autoencoder
Hasselgren et al. [58]	SD, AS	PA	radiance	temporal	I, 4	CNN, U-Net with recurrent feedback
Meng et al. [59]	SD	PA	kernel	temporal	I, 1	CNN, differentiable neural bilateral grid
Kallweit et al. [62]	VD	V	radiance	volume	O, 1	MLP, hierarchical 3D descriptors
Panin and Nikolenko [63]	VD	V	radiance	volume	O, 1	MLP, baking light probes
Hofman et al. [65]	VD	PA	radiance	image	O, 1	CNN, GAN, dual autoencoder
Xu et al. [64]	VD	PAVG	radiance	gradient	O, 1	CNN, photon density estimation
Bako et al. [40]	RD, AS	RA	radiance	radiance	O, 4	CNN, GAN
Huo et al. [41]	RD, AS	RA	radiance	radiance	O, 16	CNN, DRL, 4D convolution
Jiang and Kainz [42]	RD	RA	radiance	radiance	O, 1	CNN, autoencoder

sample colors defined on the incident radiance field with directional information; G, gradient-domain features, e.g., gradient maps; O, special descriptors of nearby photon information; V, descriptors of volumetric and lighting information in the 3D space.

- *Network prediction.* The underlying mathematical models or expected prediction outputs of the neural networks. The attributes can be classified into predicting filter parameters, predicting filtering kernels, and directly predicting radiance.
- *Rendering domain.* Traditionally, there exist variants of definitions of the rendering problem depending on the formulation and abstraction of the problem. Deep learning-based MC denoising techniques inherit such a taxonomy in terms of the relations between input, output, and features being explored. Common rendering domains include image domain, sample domain, radiance-field domain, gradient domain, photon domain, temporal domain, and volume domain.
- *Rendering speed.* Variants of MC denoisers make different tradeoffs between rendering quality and performance, thus satisfying different applications. Currently, deep learning-based MC denoisers pursue high-quality rendering with offline speed (O) or achieve interactive (I) frame rates at the cost of rendering details. The total time consumed depends on both neural network inference speed and the minimum samples per pixel (spp) for noisy network inputs. Here we report the minimum spp appearing in the original paper.
- *Technical remark.* Particular technical features and deep learning techniques used by each method.

In general, conventional MC integration approaches perform value estimation through stochastic schemes per footprint, e.g., pixel or shading point. On the other hand, deep learning-based MC denoising can be seen as a complementary postprocessing technique to explore the generality of spatial, temporal, and semantic correlations between rendering footprints and auxiliary features from offline datasets. It is not mandatory, in the conventional sense, but has achieved great success in practice and raised a lot of academic interest by revealing another dimension of the rendering problem, which is influencing in-depth studies and might lead to interesting next-generation rendering applications in the future. Some of the remaining open problems in this research area include the pursuit of efficient exploration of the high-dimensional path space, cooperation with sophisticated rendering frameworks such as Metropolis light transportation, the balance between mathematical convergence and regression efficiency, exploration of novel features and deep-learning models, and improved computation speed for robust real-time rendering. Hopefully, this survey will introduce deep learning-based MC denoising to a large audience and lead to follow-up research in different directions.

Acknowledgements

This work was supported by National Research Foundation of Korea (NRF) grant (MSIT) (No. 2019R1A2C3002833).

References

[1] Kajiya, J. T. The rendering equation. In: Proceedings of the 13th Annual Conference on Computer Graphics and Interactive Techniques, 143–150, 1986.

[2] Pharr, M.; Jakob, W.; Humphreys, G. *Physically based Rendering: From Theory to Implementation.* Morgan Kaufmann, 2016.

[3] Rubinstein, R. Y.; Kroese, D. P. *Simulation and the Monte Carlo Method.* Hoboken, NJ, USA: John Wiley & Sons, Inc., 2016.

[4] LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* Vol. 521, No. 7553, 436–444, 2015.

[5] Goodfellow, I.; Bengio, Y.; Courville, A.; Bengio, Y. *Deep Learning.* MIT Press, 2016.

[6] 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. *Computer Graphics Forum* Vol. 34, No. 2, 667–681, 2015.

[7] Dahlberg, H.; Adler, D.; Newlin, J. Machine-learning denoising in feature film production. In: Proceedings of the ACM SIGGRAPH 2019 Talks, Article No. 21, 2019.

[8] Chaitanya, C. R. A.; Kaplanyan, A. S.; Schied, C.; Salvi, M.; Lefohn, A.; Nowrouzezahrai, D.; Aila, T. Interactive reconstruction of Monte Carlo image sequences using a recurrent denoising autoencoder. *ACM Transactions on Graphics* Vol. 36, No. 4, Article No. 98, 2017.

[9] 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 Transactions on Graphics* Vol. 37, No. 4, Article No. 124, 2018.

[10] Kalantari, N. K.; Bako, S.; Sen, P. A machine learning approach for filtering Monte Carlo noise. *ACM Transactions on Graphics* Vol. 34, No. 4, Article No. 122, 2015.

[11] Hastie, T.; Tibshirani, R.; Friedman, J. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd edn. Springer Science & Business Media, 2009.

[12] Rosenblatt, F. Principles of neurodynamics. perceptrons and the theory of brain mechanisms. Technical Report. Cornell Aeronautical Lab Inc Buffalo NY, 1961.

[13] Rumelhart, D. E.; Hinton, G. E.; Williams, R. J. Learning internal representations by error propagation. Technical Report. California Univ San Diego La Jolla Inst for Cognitive Science, 1985.

[14] Xing, Q. W.; Chen, C. Y. Path tracing denoising based on SURE adaptive sampling and neural network. *IEEE Access* Vol. 8, 116336–116349, 2020.

[15] Stein, C. M. Estimation of the mean of a multivariate normal distribution. *The Annals of Statistics* Vol. 9, No. 6, 1135–1151, 1981.

[16] 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 Transactions on Graphics* Vol. 36, No. 4, Article No. 97, 2017.

[17] Gharbi, M.; Li, T.-M.; Aittala, M.; Lehtinen, J.; Durand, F. Sample-based Monte Carlo denoising using a kernel-splatting network. *ACM Transactions on Graphics* Vol. 38, No. 4, Article No. 125, 2019.

[18] LeCun, Y.; Boser, B.; Denker, J. S.; Henderson, D.; Howard, R. E.; Hubbard, W.; Jackel, L. D. Backpropagation applied to handwritten zip code recognition. *Neural Computation* Vol. 1, No. 4, 541–551, 1989.

[19] LeCun, Y.; Boser, B. E.; Denker, J. S.; Henderson, D.; Howard, R. E.; Hubbard, W. E.; Jackel, L. D. Handwritten digit recognition with a back-propagation network. In: Proceedings of the 2nd International Conference on Neural Information Processing Systems, 396–404, 1989.

[20] Back, J.; Hua, B.-S.; Hachisuka, T.; Moon, B. Deep combiner for independent and correlated pixel estimates. *ACM Transactions on Graphics* Vol. 39, No. 6, Article No. 242, 2020.

[21] Xu, B.; Zhang, J. F.; Wang, R.; Xu, K.; Yang, Y. L.; Li, C.; Tang, R. Adversarial Monte Carlo denoising with conditioned auxiliary feature modulation. *ACM Transactions on Graphics* Vol. 38, No. 6, Article No. 224, 2019.

[22] Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial nets. In: Proceedings of the 27th International Conference on Neural Information Processing Systems, Vol. 2, 2672–2680, 2014.

[23] Creswell, A.; White, T.; Dumoulin, V.; Arulkumaran, K.; Sengupta, B.; Bharath, A. A. Generative adversarial networks: An overview. *IEEE Signal Processing Magazine* Vol. 35, No. 1, 53–65, 2018.

[24] Alsaiari, A.; Rustagi, R.; Thomas, M. M.; Forbes, A. G. Image denoising using a generative adversarial network. In: Proceedings of the IEEE 2nd International Conference on Information and Computer Technologies, 126–132, 2019.

[25] He, K. M.; Zhang, X. Y.; Ren, S. Q.; Sun, J. Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 770–778, 2016.

[26] Wong, K. M.; Wong, T. T. Deep residual learning for denoising Monte Carlo renderings. *Computational Visual Media* Vol. 5, No. 3, 239–255, 2019.

[27] Yang, X.; Wang, D. W.; Hu, W. B.; Zhao, L. J.; Piao, X. L.; Zhou, D. S.; Zhang, Q.; Yin, B.; Cai, Q.; Wei, X. Fast reconstruction for Monte Carlo rendering using deep convolutional networks. *IEEE Access* Vol. 7, 21177–21187, 2019.

[28] Yang, X.; Wang, D. W.; Hu, W. B.; Zhao, L. J.; Yin, B. C.; Zhang, Q.; Wei, X.-P.; Fu, H. DEMC: A deep dual-encoder network for denoising Monte Carlo rendering. *Journal of Computer Science and Technology* Vol. 34, No. 5, 1123–1135, 2019.

[29] Ballard, D. H. Modular learning in neural networks. In: Proceedings of the 6th National Conference on Artificial Intelligence, Vol. 1, 279–284, 1987.

[30] Vincent, P.; Larochelle, H.; Bengio, Y.; Manzagol, P. A. Extracting and composing robust features with denoising autoencoders. In: Proceedings of the 25th International Conference on Machine Learning, 1096–1103, 2008.

[31] Kuznetsov, A.; Kalantari, N. K.; Ramamoorthi, R. Deep adaptive sampling for low sample count rendering. *Computer Graphics Forum* Vol. 37, No. 4, 35–44, 2018.

[32] Munkberg, J.; Hasselgren, J. Neural denoising with layer embeddings. *Computer Graphics Forum* Vol. 39, No. 4, 1–12, 2020.

[33] Hanika, J.; Droske, M.; Fascione, L. Manifold next event estimation. *Computer Graphics Forum* Vol. 34, No. 4, 87–97, 2015.

[34] Lin, W. H.; Wang, B. B.; Yang, J.; Wang, L.; Yan, L. Q. Path-based Monte Carlo denoising using a three-scale neural network. *Computer Graphics Forum* Vol. 40, 369–381, 2021.

[35] Levoy, M.; Hanrahan, P. Light field rendering. In: Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques, 31–42, 1996.

[36] Lin, W. H.; Wang, B. B.; Wang, L.; Holzschuch, N. A detail preserving neural network model for Monte Carlo denoising. *Computational Visual Media* Vol. 6, No. 2, 157–168, 2020.

[37] Durand, F.; Holzschuch, N.; Soler, C.; Chan, E.; Sillion, F. X. A frequency analysis of light transport. *ACM Transactions on Graphics* Vol. 24, No. 3, 1115–1126, 2005.

[38] Belcour, L.; Soler, C.; Subr, K.; Holzschuch, N.; Durand, F. 5D Covariance tracing for efficient defocus and motion blur. *ACM Transactions on Graphics* Vol. 32, No. 3, Article No. 31, 2013.

[39] Liang, Y. L.; Wang, B. B.; Wang, L.; Holzschuch, N. Fast computation of single scattering in participating media with refractive boundaries using frequency analysis. *IEEE Transactions on Visualization and Computer Graphics* Vol. 26, No. 10, 2961–2969, 2020.

[40] Bako, S.; Meyer, M.; DeRose, T.; Sen, P. Offline deep importance sampling for Monte Carlo path tracing. *Computer Graphics Forum* Vol. 38, No. 7, 527–542, 2019.

[41] Huo, Y.; Wang, R.; Zheng, R.; Xu, H.; Bao, H.; Yoon, S.-E. Adaptive incident radiance field sampling and reconstruction using deep reinforcement learning. *ACM Transactions on Graphics* Vol. 39, No. 1, Article No. 6, 2020.

[42] Jiang, G.; Kainz, B. Deep radiance caching: Convolutional autoencoders deeper in ray tracing. *Computers & Graphics* Vol. 94, 22–31, 2021.

[43] Lehtinen, J.; Karras, T.; Laine, S.; Aittala, M.; Durand, F.; Aila, T. Gradient-domain metropolis light transport. *ACM Transactions on Graphics* Vol. 32, No. 4, Article No. 95, 2013.

[44] Kettunen, M.; Manzi, M.; Aittala, M.; Lehtinen, J.; Durand, F.; Zwicker, M. Gradient-domain path tracing. *ACM Transactions on Graphics* Vol. 34, No. 4, Article No. 123, 2015.

[45] Hua, B. S.; Gruson, A.; Petitjean, V.; Zwicker, M.; Nowrouzezahrai, D.; Eisemann, E.; Hachisuka, T. A survey on gradient-domain rendering. *Computer Graphics Forum* Vol. 38, No. 2, 455–472, 2019.

[46] Kettunen, M.; Härkönen, E.; Lehtinen, J. Deep convolutional reconstruction for gradient-domain rendering. *ACM Transactions on Graphics* Vol. 38, No. 4, Article No. 126, 2019.

[47] 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 Transactions on Graphics* Vol. 38, No. 6, Article No. 223, 2019.

[48] Jensen, H. W. *Realistic Image Synthesis Using Photon Mapping*. AK Peters/CRC Press, 2001.

[49] Kang, C. M.; Wang, L.; Xu, Y. N.; Meng, X. X. A survey of photon mapping state-of-the-art research and future challenges. *Frontiers of Information Technology & Electronic Engineering* Vol. 17, No. 3, 185–199, 2016.

[50] Zhu, S.; Xu, Z.; Jensen, H. W.; Su, H.; Ramamoorthi, R. Deep kernel density estimation for photon mapping. *Computer Graphics Forum* Vol. 39, No. 4, 35–45, 2020.

[51] Hachisuka, T.; Ogaki, S.; Jensen, H. W. Progressive photon mapping. In: Proceedings of the ACM SIGGRAPH Asia 2008 papers, Article No. 130, 2008.

[52] Zeng, Z.; Wang, L.; Wang, B. B.; Kang, C. M.; Xu, Y. N. Denoising stochastic progressive photon mapping renderings using a multi-residual network. *Journal of Computer Science and Technology* Vol. 35, No. 3, 506–521, 2020.

[53] Rumelhart, D. E.; Hinton, G. E.; Williams, R. J. Learning representations by back-propagating errors. *Nature* Vol. 323, No. 6088, 533–536, 1986.

[54] Huang, Y.; Wang, W.; Wang, L. Bidirectional recurrent convolutional networks for multi-frame super-resolution. In: Proceedings of the 28th International Conference on Neural Information Processing Systems, Vol. 1, 235–243, 2015.

[55] Mehta, S. U.; Wang, B.; Ramamoorthi, R. Axis-aligned filtering for interactive sampled soft shadows. *ACM Transactions on Graphics* Vol. 31, No. 6, Article No. 163, 2012.

[56] Dammertz, H.; Sewtz, D.; Hanika, J.; Lensch, H. P. A. Edge-avoiding À-Trous wavelet transform for fast global illumination filtering. In: Proceedings of the Conference on High Performance Graphics, 67–75, 2010.

[57] Li, T. M.; Wu, Y. T.; Chuang, Y. Y. SURE-based optimization for adaptive sampling and reconstruction. *ACM Transactions on Graphics* Vol. 31, No. 6, Article No. 194, 2012.

[58] Hasselgren, J.; Munkberg, J.; Salvi, M.; Patney, A.; Lefohn, A. Neural temporal adaptive sampling and denoising. *Computer Graphics Forum* Vol. 39, No. 2, 147–155, 2020.

[59] Meng, X.; Zheng, Q.; Varshney, A.; Singh, G.; Zwicker, M. Real-time Monte Carlo denoising with the neural bilateral grid. In: Proceedings of the Eurographics Symposium on Rendering, 2020.

[60] Drebin, R. A.; Carpenter, L.; Hanrahan, P. Volume rendering. *ACM SIGGRAPH Computer Graphics* Vol. 22, No. 4, 65–74, 1988.

[61] Max, N. Optical models for direct volume rendering. *IEEE Transactions on Visualization and Computer Graphics* Vol. 1, No. 2, 99–108, 1995.

[62] Kallweit, S.; Müller, T.; McWilliams, B.; Gross, M.; Novák, J. Deep scattering: Rendering atmospheric clouds with radiance-predicting neural networks. *ACM Transactions on Graphics* Vol. 36, No. 6, Article No. 231, 2017.

[63] Panin, M.; Nikolenko, S. Faster RPNN: Rendering clouds with latent space light probes. In: Proceedings of the SIGGRAPH Asia 2019 Technical Briefs, 21–24, 2019.

[64] Xu, Z. L.; Sun, Q.; Wang, L.; Xu, Y. N.; Wang, B. B. Unsupervised image reconstruction for gradient-domain volumetric rendering. *Computer Graphics Forum* Vol. 39, No. 7, 193–203, 2020.

[65] Hofmann, N.; Martschinke, J.; Engel, K.; Stamminger, M. Neural denoising for path tracing of medical volumetric data. In: Proceedings of the ACM on Computer Graphics and Interactive Techniques, Article No. 13, 2020.

[66] Jensen, H. W. Importance driven path tracing using the photon map. In: *Rendering Techniques '95.* Hanrahan, P. M.; Purgathofer, W. Eds. Springer Vienna, 326–335, 1995.

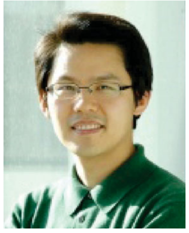
[67] Hey, H.; Purgathofer, W. Importance sampling with hemispherical particle footprints. In: Proceedings of the 18th Spring Conference on Computer Graphics, 107–114, 2002.

[68] Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A. A.; Veness, J.; Bellemare, M. G.; Graves, A.; Riedmiller, M.; Fidjeland, A. K.; Ostrovski, G. et al. Human-level control through deep reinforcement learning. *Nature* Vol. 518, No. 7540, 529–533, 2015.

[69] Silver, D.; Huang, A.; Maddison, C. J.; Guez, A.; Sifre, L.; van den Driessche, G.; Schrittwieser, J.; Antonoglou, I.; Panneershelvam, V.; Lanctot, M. et al. Mastering the game of Go with deep neural networks and tree search. *Nature* Vol. 529, No. 7587, 484–489, 2016.

[70] Nvidia. Interactive reconstruction of Monte Carlo image sequences using a recurrent denoising autoencoder. 2020. Available at https://research.nvidia.com/publication/interactive-reconstruction-monte-carlo-image-sequences-using-recurrent-denoising.

Yuchi Huo graduated from Zhejiang University and is working at the SGVR (Scalable Graphics, Vision, & Robotics) Lab. at KAIST (Korea Advanced Institute of Science and Technology). His research interests are in rendering, deep learning, image processing, and computational optics.



Sung-Eui Yoon is a professor at KAIST where he is currently leading the SGVR Lab. His research interests span scalable rendering, vision, and robotics problems including ray tracing, image search, and motion planning for robots.