Computational Visual Media
DOI 10.1007/s41095-xxx-xxxx-x Vol. x, No. x, month year, xx–xx

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 state-of-the-art report summarizes the recent trends of the post-processing reconstruction scheme. Recent years have seen increasing attention and significant progress in denoising MC renderings 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 overview for practitioners and researchers to assess these approaches.

Keywords Rendering, Monte Carlo 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 [30, 53]. One of the firmly established bases for such a purpose is MC integration [55], which is renowned for its generality and heavy computational consumption. MC integration methods are attractive because of two distinct advantages. First, 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. Second, 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.

Classic MC integration methods, however, expect a large number of samples to achieve faithful convergence. Despite continuously increasing computational power, the cost of realistic rendering remains a limiting, practical constraint that takes hours to render one frame of high-quality images. When using a small number of samples, the 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 focused areas in the rendering community.

Recently, deep learning techniques have earned unprecedented focus and exceeded many traditional algorithms in various domains [14, 37]. Derived from traditional MC reconstruction [70], MC denoising through the combination with deep learning techniques achieves notable progress and becomes a hot topic in recent years. Furthermore, the industry actively embraces the latest achievements. For example, in the movie industry, Pixar's RenderMan [9] adapts adaptive sampling and denoising filters for the production of Toy Story 4. Another example [7] in the gaming industry is to generate high-quality images with low sample counts for real-time use.

This report summarizes the state-of-the-art techniques in deep learning-based MC denoising. We start with a hands-on introduction of the basic concept and then discuss in details the architecture

1 KAIST, Daejeon, 31414, South Korea. E-mail: huo.yuchi.sc@gmail.com, sungeui@kaist.edu

Manuscript received: 2014-12-31; accepted: 2015-01-30.

Fig. 1 Deep learning-based Monte Carlo denoising method trains a neural network to reduce Monte Carlo noises in input images. This image is excerpted from Refs. [61].

of the area (Sec. 2). Afterward, we provide a comprehensive overview by categorizing the related research into three topics:

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

We then conclude this report by providing the summary and comparisons of discussed techniques (Sec. 6).

2 Concept of Deep Learning-Based Monte Carlo Denoising

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

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

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 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 to a shading point p, such as a pixel, as $x_p = \{c_p, f_p\}$, where c_p is noisy values achieved with low sample counts and f_p is auxiliary features, e.g. surface normal or textures over multiple samples contributing to p. While using deep learning, the pursuing of optimal g can be formulated as the training of a neural network parameterized by a set of weights θ to represent g. Through a supervised learning process that utilizes a dataset with N example pairs of $(x^1, r^1), ..., (x^N, r^N)$, 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 around the neighborhood of x^n to produce the reconstructed output at pixel x^n [61]. 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 on the guarantee of mathematical convergence, this approximation reforms production pipelines by enabling rendering quality that is visually indistinguishable to the ground-truth in a much fast running time, approaching to an interactive rate in a near future.

3 Pixel Denoising

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 [30], and predict the corresponding smoothed image. Because the results of most MC integration methods can be stacked into the image space, directly denoising the per-pixel colors can work as a general post-processing add-on to existing rendering pipelines without the necessity of reforming data flows. As thus, pixel denoising soon becomes a popular solution in the academic society and industry.

We categorize the related researches in pixel denoising according to prediction targets of neural

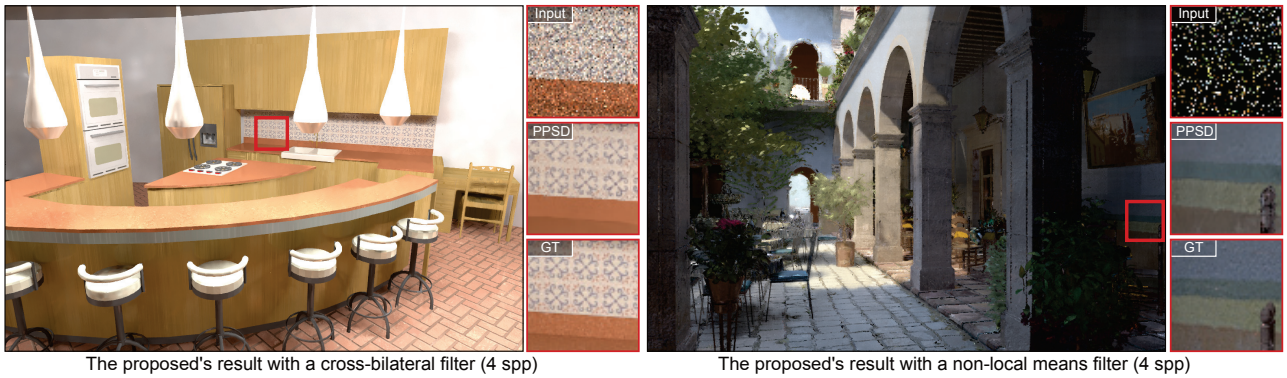


Fig. 2 Result of using the trained network of Kalantari et al. [31] (PPSD) to drive a filter for denoising a new MC rendered image, with a cross-bilateral filter for the KITCHEN scene (1200 × 800) on the left and with a non-local means filter for the SAN MIGUEL HALLWAY scene (800 × 1200) on the right. Both of these scenes are path-traced and contain severe noise at 4 samples per pixel (spp). The trained network is able to estimate the appropriate filter parameters and effectively reduce the noise in only a few seconds. The image is excerpted from Refs. [31].

networks, which imply the underlying problem formulation of the denoising process. Overall, we categorize them into parameter prediction, radiance prediction, and kernel prediction (Tab. 1).

3.1 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 [31]. 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. [31] observed there is a complex relationship between the noisy scene data and the ideal filter parameters, and proposed to learn this relationship through deep learning. The method uses cross-bilateral and cross non-local means filters of various auxiliary features (i.e., world positions, shading normals and texture values, etc.) for the final reconstruction and a multilayer perceptron (MLP) neural network [20, 54, 56] to predict optimal weights for each feature in the filter. To use the framework, a 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 only 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 et al. [63] also adapted a parameter estimation network to address the noise artifacts of path tracing. The method contains sampling and reconstruction stages. Stein's unbiased risk estimator (SURE) [59] 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, 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.

3.2 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 researches aims to directly predict local reconstruction kernels through the kernel-predicting networks [4, 13, 61].

The explicit filters are handy to exploit conventional MC denoising models, but might limit denoising capability even using deep neural networks to predict the optimal parameters [4]. To address this problem, Bako et al. [4] 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 [38, 39]. 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. Through the results, there has been observed improved accuracy compared to parameter-predicting MC denoisers and roughly 5 to 6 times faster convergence speed of the weighted kernel prediction than that of the direct radiance prediction. Some other training skills 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. [61] 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 inference. 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 inference. Forth, an error-predicting module to infer 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 various production and academic data sets purpose. Finally, they shed light on the academic research by offering a theoretical analysis of convergence rates of kernel predicting architectures.

MC denoisrs, also known as biased MC estimators, reduce MC noises by exploring the correlation among nearby pixels. As a result, they suffer from method-specific residual noise or systematic errors. Back et al. [2] aimed to mitigate such remaining errors by unifying independent unbiased estimator and 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 among 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 visually and numerically.

3.3 Radiance Prediction

Parameter-predicting and kernel-predicting frameworks generally have achieved great success, but the kernel filtering scheme sometimes imposes restrictions on a flexible fusing 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 toward viewpoint at each footprint.

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. [64] present an adversarial approach for MC denoising, following an insight that generative adversarial networks (GANs) [8, 15] 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 adapted 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 [1]. Because the network does not take auxiliary features as input, it can also denoise noisy photographs under natural light and CT scans.

Deep residual network (ResNet) [21] demonstrates significant improvement over vanilla CNN. In order to take advantage of the ResNet, a filter-free direct denoising method based on a standard-and-simple deep ResNet is trained to remove the noise of MC rendering [62]. 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 to most production rendering pipelines. With the help of ResNet, the simple structure yields comparable accuracy compared to the other state-of-the-arts.

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

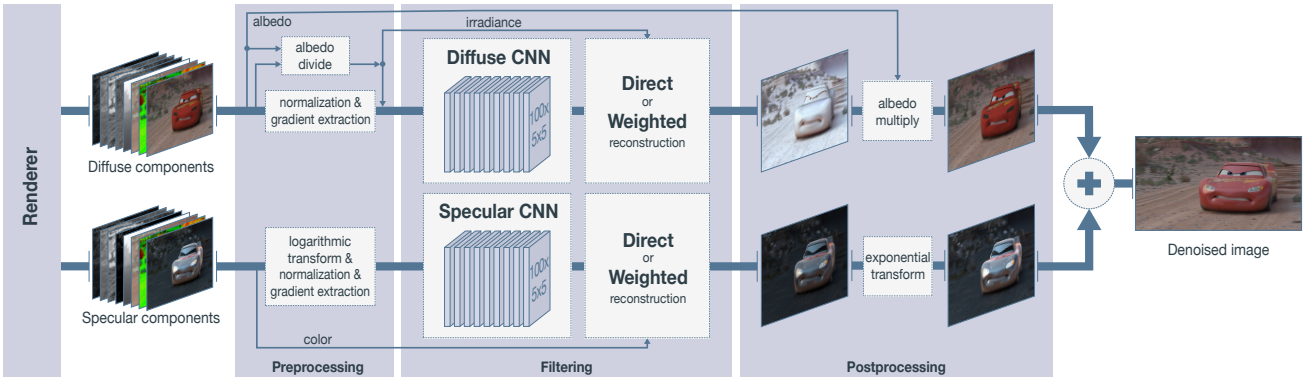


Fig. 3 An overview of the kernel-predicting framework [4]. It starts by preprocessing diffuse and specular data coming from the rendering system independently, and then feeds the information to two separate networks, which denoise the diffuse and specular illumination, respectively. The output from each network undergoes reconstruction (direct reconstruction or weighted reconstruction through the predicted kernels) and postprocessing before being combined to obtain the final, denoised image. This image is excerpted from Refs. [4].

et al. [66, 67] focused on the topic of 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 GT 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 [66]. In a follow-up research, they proposed an Autoencoder [5, 60] inspired network structure, Dual-Encoder network with a feature fusion subnetwork, to fuse auxiliary features firstly. The fused features and a noisy image are then fed as inputs of a second encoder network to reconstruct a clean image by the decoder network [67]. Compared with 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 no sufficient information to calculate the features, such as variance and contrast. To address this issue, Kuznetsov et al. [36] 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

Conventional MC denoisers work on the image space, where the basic geometry 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 transportation is on a high-dimensional space where some important information is inevitably degraded when reducing the everything into per-pixel radiance. To address this, a stream of researches aims to discover 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 (Tab. 1).

4.1 Sample Space

In contrast to the traditional pixel-based MC denoisers, Gharbi et al. [13] proposed a sample-based kernel-splatting network. The authors 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 should 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

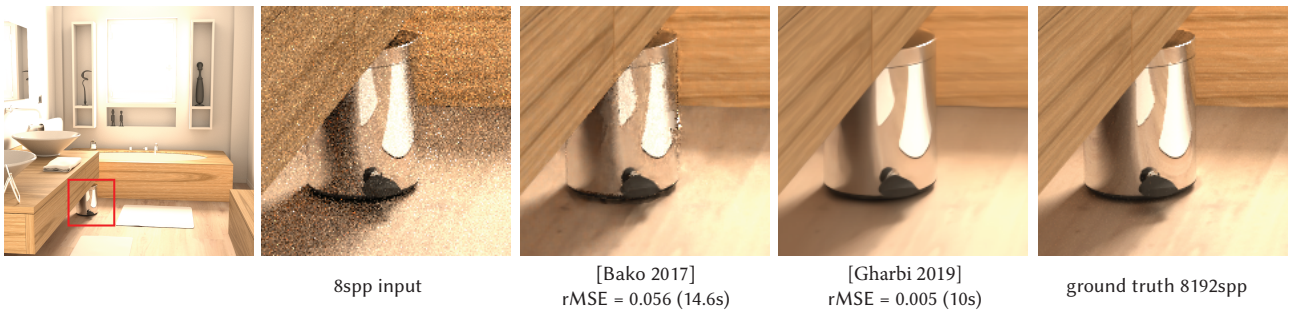


Fig. 4 Comparisons between state-of-the-art pixel-based (Bako et al. [4]) and sample-space (Gharbi et al. [13]) MC denoising algorithms. Sample-space method works with the samples directly, it uses a splatting approach that makes it possible to appropriately handle various components of the illumination (indirect lighting, specular reflection, motion blur, depth of field, etc) more effectively. The image is excerpted from Refs. [13].

network. They claimed that, in addition, splatting is a natural solution to situations such as motion blur, depth-of-field, and many light transport paths, where it is easier to predict which pixels a sample contributes to, rather than to predict gathering kernels that need to figure out informative relationship between relevant pixels. The new architecture yields higher-quality results both visually and numerically in 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 might limit its usability for higher sample counts. Based on this observation, Munkberg et al. [50] proposed to extract a compact representation of per-sample information by separating samples into a fixed number of partitions, denoted as layers in their paper, via a data-driven way that learns unique kernels 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) [18] is used in the rendering process, Lin et al. [45] decomposed the features of Gharbi et al. [13] 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 together to preserve sharp details, using a feature attention mechanism and feature extractors.

4.2 Light Field Space

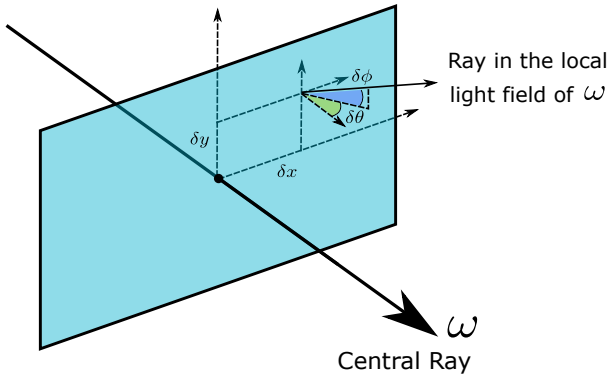


Fig. 5 The Local Light Field is defined as a 4D function around the center ray (ω), parameterized by two spatial coordinates (δ_x and δ_y) and two angular coordinates (δ_θ and δ_ϕ). This image is excerpted from Refs. [43].

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 of the light field [41]. Lin et al. [44] 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 [12] 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 into operations on the Fourier spectrum, then approximately represented by the Fourier spectrum of the local light field, the covariance matrix [6]. A neural network is proposed to make use of this covariance matrix, a 4×4 matrix encoding 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 two buffers into a shallow kernel predictor. Finally, they suggested a 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 researches aims to directly reconstruct denoised incident radiance field, i.e., the local light field at each pixel, for advanced goals such as unbiased path guiding [3, 26, 29]. We cover such kind of works in Sec. 5.3.

4.3 Gradient Domain

The gradient-domain rendering methods [24, 35, 40] 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. A recent work combines this long-existing research direction with modern CNNs [34]. 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 functions. The combination significantly improves the quality obtained from gradient-domain path tracing and yields notably improved image quality compared to vanilla image-space MC denoisers.

In another independent work, Guo et al. [16] propose to use a multi-branch autoencoder to replace the Poisson solver. The proposed 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 tweaking 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.

4.4 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 [28, 33] has been considered as the method of choice for rendering caustics, but not properly adapted with the trend led by the progress of deep learning techniques. A recent research bridges this gap by training a deep neural network to predict a kernel function aggregates photon contributions at each shading point [69]. Photon mapping traces a large number of photons from the light source, then gathers the photon contributions at each shading point to achieve high-quality reconstructions of challenging light transportation hard to be traced from the camera. The authors proposed to 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 compared to previous photon mapping methods and significantly outperform those of path tracing-based MC rendering in rendering caustic effects (Fig. 6).

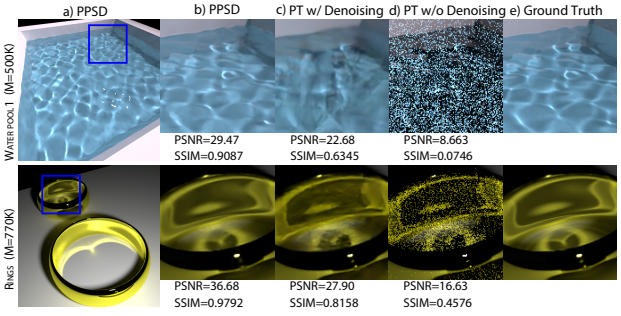


Fig. 6 The results of photon mapping denoising show high-quality reconstruction of caustic effects [69]. (a) and (b) The final results of the proposed method (PPSD). (c) Path-tracing (PT) results with an image-space denoiser [7]. (d) PT results without denoiser. (e) Ground truth. This image is excerpted from Refs. [69].

Stochastic progressive photon mapping [17] is one of the important global illumination methods 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. [68] recently proposed a deep learning-based method specially designed for denoising the biased renderings of stochastic progressive photon mapping. The method decomposes the light transportation into two components, caustic and others, 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

Single-image MC denoisers take as input one noisy image to produce one high-quality output without MC noise. However, their single-image output does not satisfy many applications that require higher-dimensional outputs. For example, producing computer animations requires a sequence of temporally consistent images, and path guiding to generate unbiased rendering results might need to denoise the whole incident light field on each shading point. In these scenarios, pixel-based MC denoisers are no longer adequate to generate the high-dimensional outputs without special designs for high-dimensional signal processing and consistent 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 details (Tab. 1).

5.1 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 are focusing on rendering quality, and others put additional attention to the balance between quality and speed to achieve an interactive processing speed. Besides the processing speed, an essential consideration is to enhance temporal stability between frames to avoid low-frequency variances that might lead to serious flicking artifacts in animation. The pioneering research is inspired by the good results of using recurrent neural networks (RNNs) [25, 57] in the context of video super-resolution and sub-pixel CNNs, and describes an RNN-based framework that drastically improves temporal stability for sequences of sparsely sampled input images [7].

Compared to single-image denoisers, the RNN network takes sequential images as input to explore temporal coherency 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 of high-quality real-time denoisers (Fig. 7).

Hasselgren et al. [19] 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 spatio-temporal joint training of neural network-based sample predictors and MC denoisers 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 disoccluded regions.20201133 or track specular highlights, where high-frequency details are hard to reconstruct. Such a framework allows trade-off between quality and performance, while running at near real-time rates.

Meng et al. [48] focused on the computation speed and proposed a novel and practical real-time approach denoises 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 the vanilla kernel-prediction techniques.

It is worth noting that the aforementioned kernel-predicting neural network proposed by Vogel et al. [61] also contains a temporal denoiser module to boost temporal stability. However, the authors focus on animation rendering, which is slightly different from interactive rendering in terms of future frame visibility.

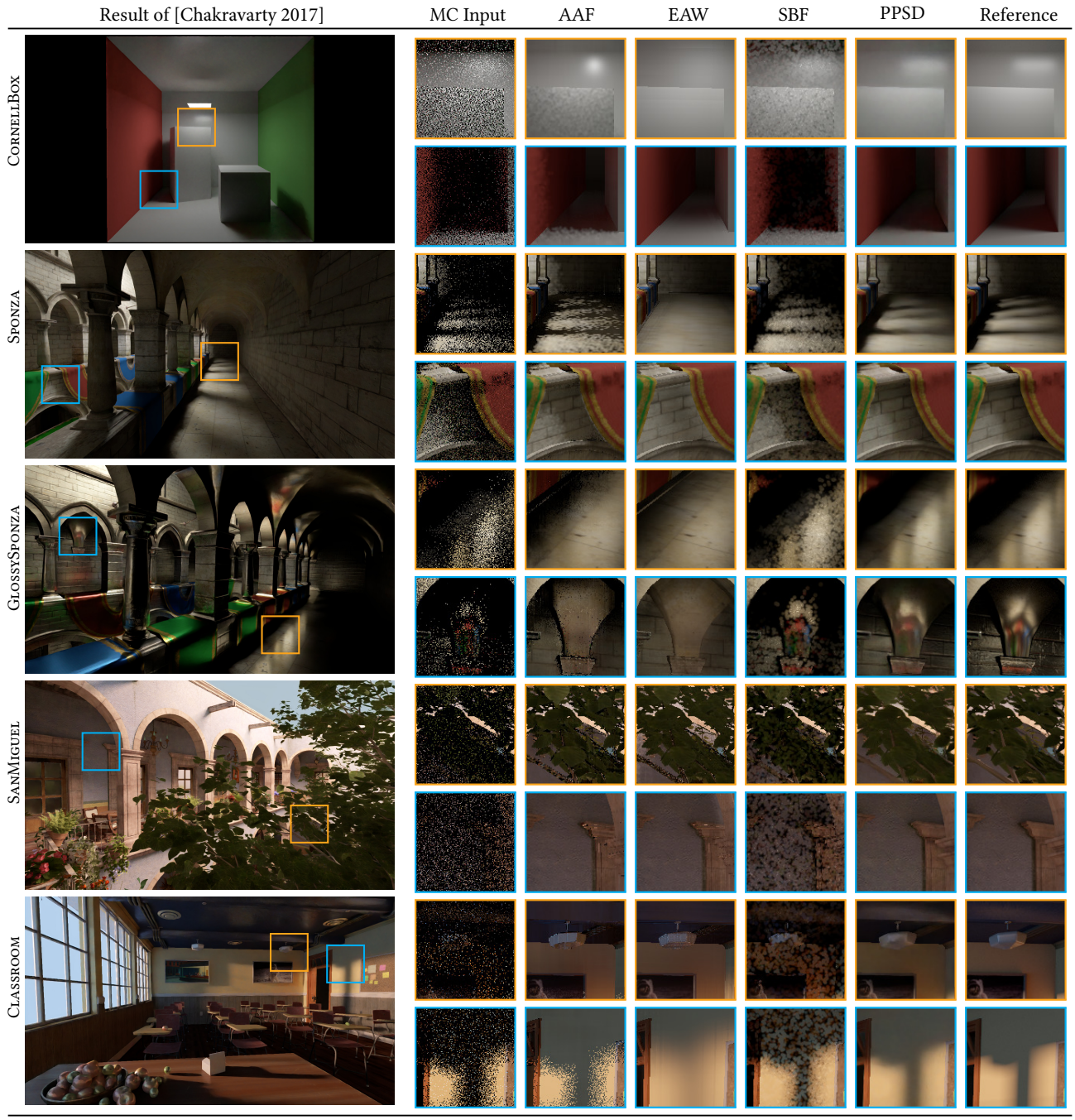


Fig. 7 Closeups of 1-bounce global illumination results for 1 spp input (MC), axis-aligned filter [47] (AAF), À-Trous wavelet filter [10] (EAW), SURE-based filter [42] (SBF), and result of the proposed deep learning-based denoiser [7] (PPSD). Compared to the conventional method, the deep learning-based MC denoiser yields higher rendering quality and temporal stability. Full resolution images and video sequences are provided in the project's main page [51]. The image is excerpted from Refs. [7].

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 way, there is no need to insert recurrent connections to the module.

5.2 Volume Rendering

As an important section of realistic rendering, volume rendering [11, 46] significantly contributes to a wide variety of vivid visual effects for participating media,

such as clouds, fogs, liquid, transparent solid, and medical data (Fig. 8). However, such rendering is usually conducted in the 3D space, where exists a tremendous amount of possible light transport and scattering among particles, causing difficulties or performance declines for conventional image-space MC denoisers. Some recent researches aim to adapt deep learning techniques to generate high-quality volume rendering images in the 3D space. This kind of method shares the same spirit of the discussed MC denoisers so far, using deep learning techniques to generate smooth images with 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. [32] approached the problem in a data-driven way. They trained a neural network, featured with residual connections, to predict the spatial and directional distribution of radiant flux from an offline dataset containing tens of cloud examples. In inference, the network takes visibility sample points of the cloud in a new scene as input to predicts the radiance function for each shading configuration. The method contributes a key novelty that each visible sample contains a feature of a hierarchical 3D descriptor of the cloud geometry with respect to the shading location and the 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.

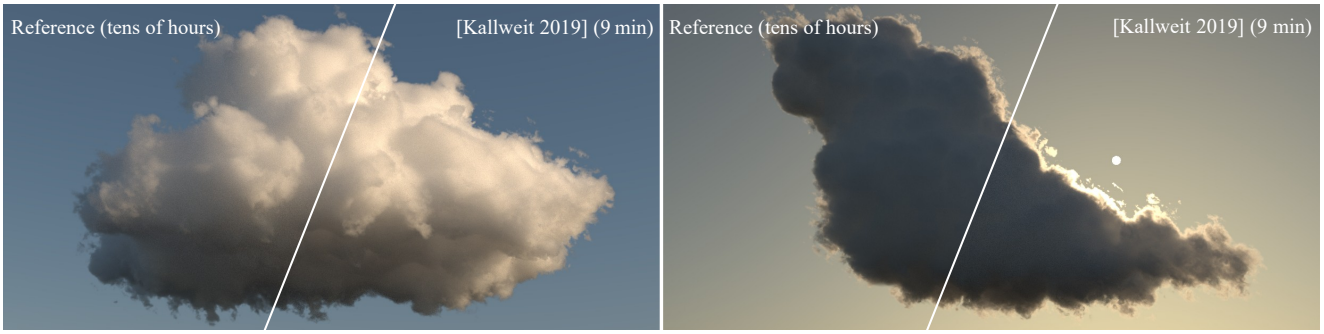
The performance of the deep learning-based clound rendering approach was later improved by decomposing the neural network architecture into some parts that can be precomputed and other parts that should be inferred at runtime. Panin et al. [52] introduced a latent space light probes approach that uses 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 computational overhead, yielding 2 to 3 times speedup over the previous approach.

Xu et al. [65] jointly leveraged gradient-domain information and photon mapping techniques for rendering homogenous participating media. The authors adopt the unsupervised gradient-domain deep learning framework [16] 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.

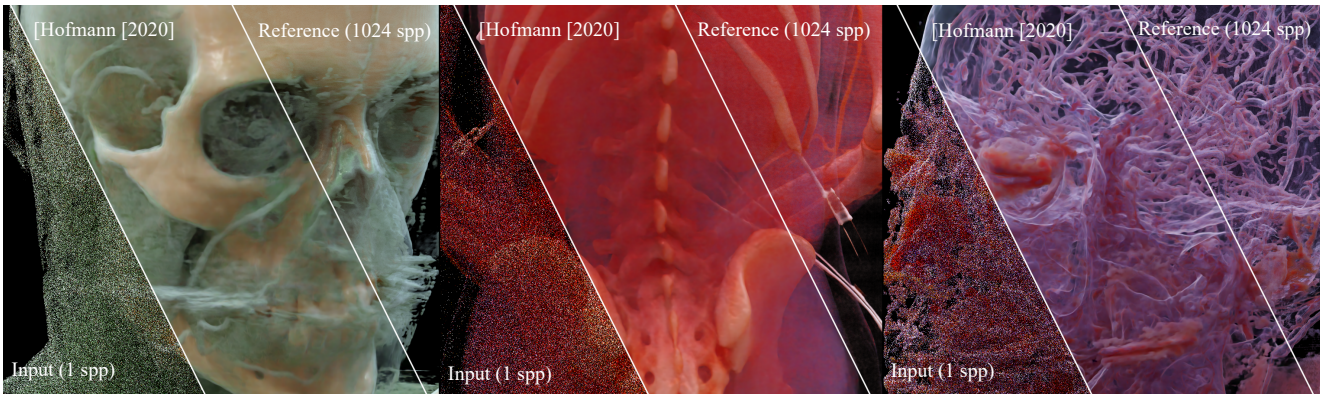
In the domain of medical imaging, MC rendering turned out to be an efficient means to visualize and understand internal structures, especially for inexperienced users such as medical students, forensic staff, and patients. However, the auxiliary features like depth and normal vital for surface-based MC denoisers are neither well-defined nor smooth for medical volumetric data. To address this, Hofmann et al. [23] transferred surface-based MC denoisers for the 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 scatter events, are fed as guiding inputs of 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 the sharp edges and details like specular highlights, which are essential for interpretation. The overall architecture also considers temporal stability of videos via feature reprojection between frames.

5.3 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 estimation without convergence guarantee, even 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 keep pursuing the ultimate of rendering quality via unbiased MC estimators. Recently, deep learning-based techniques are used to reconstruct radiance fields



(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 lining (right), and the whiteness of the inner part of the cloud. This image is excerpted from Refs. [32].



(b) Deep learning-based MC denoising of medical volumetric data from noisy input, compared to the converged ground truth. This image is excerpted from Refs. [23].

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

to guide path tracing, known as path guiding [22, 27], for efficiently generating high-quality images with relatively high samples.

Bako et al. [3] noticed that even the 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 incident radiance filed on each pixel, which enables 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 per-pixel guiding PDF for unbiased MC integration, which increases the efficiency of sampling by putting more samples in informative directions, e.g., disoccluded regions. The primary advantage of offline learning is that it uses a data-driven scheme to learn from a large set of training scenes a priori, for reconstructing the full incident radiance by reusing nearby samples. Therefore the expensive online learning process that uses a large amount of samples to fit a new scene can be abandoned. In reference, the trained network takes a small amount 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 sampling process in the next iteration [26]. In order to guide the progressive sampling process, the method considers the sampling as taking 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 [49, 58]. Via this action-based dynamic formulation, the quality-predicting neural network can learn from an offline dataset an optimal sampling strategy under progressive sampling contexts in unseen

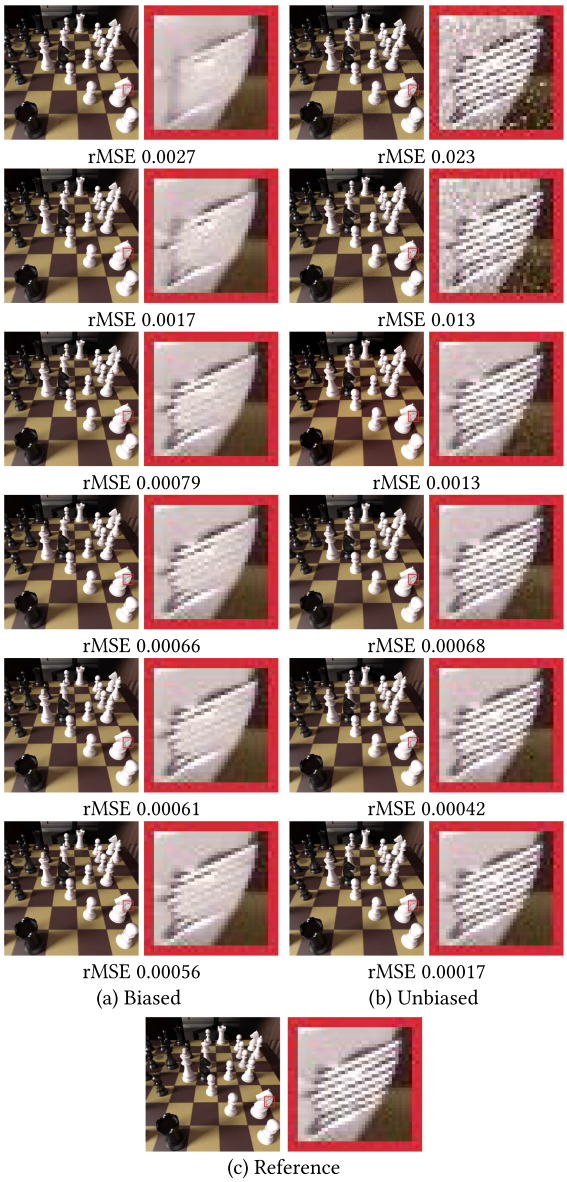


Fig. 9 Equal-time comparisons between biased MC denoiser (Bako et al. [4]) and unbiased path guiding using deep learning to generate guidance (Huo et al. [26]). The results are rendered within 1, 2, 8, 30, 60, and 120 minutes from top to bottom. While the MC denoiser achieves faster convergence with low sample counts, the deep learning-guided sampling method outperforms the MC denoiser in high samples and converges to the reference. This image is excerpted from Refs. [26].

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 for guiding path tracing in the subsequent iterations. In general, the deep learning-guided unbiased sampling process guarantees mathematical convergence with high samples, resulting in higher rendering quality compared to that of MC denoisers and other unbiased rendering approaches (Fig. 9).

Denoised radiance fields can also be directly integrated into pixel colors for biased rendering [29]. The method uses an autoencoder neural network to denoise low-sample radiance caches for rendering indirect illumination, then progressively increases samples to refine the radiance caches.

6 Conclusion

Thanks to benefits from the unprecedented success of deep learning techniques, MC denoising has been attracting strong attention in the past years. These techniques are naturally compatible with MC integration, one of the most general rendering framework used by many rendering pipelines. In general, they require only minor modification to extract auxiliary features, support a wide range of applications, scalable to both high-quality and performance-sensitive rendering, often GPU and TPU-friendly, and above all, dramatically decrease the cost of MC rendering. Tab. 1 provides an overview of the discussed methods 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 have been contributing to a specific target 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 images 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 sample 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 samples rather than each pixel; R, Radiance-field sample colors defined on the incident radiance field with directional information; G, Gradient-domain features, e.g., gradient maps; O, specific descriptors of nearby

photon information; V, descriptors of volumetric and lighting information in the 3D space.

- **Network Prediction.** The underlying mathematical models or the 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 ways to formulate and abstract the problem. Deep learning-based MC denoising techniques inherit such a taxonomy in terms of the intricacies 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 consumption depends on both neural network inference speed and the minimum spp of noisy network inputs. Here we report the minimum spp appears in the original paper.
- **Technical Remark.** Distinguished 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 pursuits of efficient exploration of the high-dimensional path space, cooperation with sophisiticated rendering framework 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 introduces deep learning-based MC denoising to a large audience and leads to follow-up researches in different directions.

Acknowledgements

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

References

[1] A. Alsaiari, R. Rustagi, M. M. Thomas, A. G. Forbes, et al. Image denoising using a generative adversarial network. In *2019 IEEE 2nd International Conference on Information and Computer Technologies (ICICT)*, pages 126–132. IEEE, 2019.

[2] J. Back, B.-S. Hua, T. Hachisuka, and B. Moon. Deep combiner for independent and correlated pixel estimates. *ACM Transactions on Graphics (TOG)*, 39(6):1–12, 2020.

[3] S. Bako, M. Meyer, T. DeRose, and P. Sen. Offline deep importance sampling for monte carlo path tracing. In *Computer Graphics Forum*, volume 38, pages 527–542. Wiley Online Library, 2019.

[4] S. Bako, T. Vogels, B. McWilliams, M. Meyer, J. Novák, A. Harvill, P. Sen, T. Derose, and F. Rousselle. Kernel-predicting convolutional networks for denoising monte carlo renderings. *ACM Trans. Graph.*, 36(4):97–1, 2017.

[5] D. H. Ballard. Modular learning in neural networks. In *AAAI*, pages 279–284, 1987.

[6] L. Belcour, C. Soler, K. Subr, N. Holzschuch, and F. Durand. 5d covariance tracing for efficient defocus and motion blur. *ACM Transactions on Graphics (TOG)*, 32(3):1–18, 2013.

[7] C. R. A. Chaitanya, A. S. Kaplanyan, C. Schied, M. Salvi, A. Lefohn, D. Nowrouzezahrai, and T. Aila. Interactive reconstruction of monte carlo image sequences using a recurrent denoising autoencoder. *ACM Transactions on Graphics (TOG)*, 36(4):1–12, 2017.

[8] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. A. Bharath. Generative adversarial networks: An overview. *IEEE Signal Processing Magazine*, 35(1):53–65, 2018.

Tab. 1 Summary of papers in Sec. 3, 4 and 5. See various attributes and their symbols in Sec. 6.

Methods	Goal	Input	Predict	Domain	Speed	Remark
Kalantari et al. [31]	PD, DE	PA	parameter	image	O, 4	MLP, cross-bilateral and non-local means filters
Xing et al. [63]	PD, AS	PA	parameter	image	O, 8	MLP, SURE, cross-bilateral filter
Bako et al. [4]	PD	PA	kernel	image	O, 16	CNN, kernel-predicting network
Vogels et al [61]	PD, SD, AS	PA	kernel	image	O, 16	CNN, asymmetric loss functions
Back et al. [2]	PD	PA	kernel	image	O, 32	CNN, combine pixel estimates
Xu et al. [64]	PD	PA	radiance	image	O, 4	CNN, GAN, feature modulation, perceptual loss
Alsaiari et al. [1]	PD	P	radiance	image	O, 1	CNN, GAN
Yang et al. [66]	PD	PA	radiance	image	O, 4	CNN, HDR tonemapping
Yang et al. [67]	PD	PA	radiance	image	O, 4	CNN, feature encoder
Wong et al. [62]	PD	PA	radiance	image	O, 8	CNN, ResNet
Kuzenetsov et al. [36]	PD, AS	PA	radiance	image	O, 5	CNN, autoencoder
Gharbi et al. [13]	PD, DE	SA	kernel	sample	O, 4	CNN, U-Net, kernel-splatting network
Munkberg et al. [50]	PD, DE	SA	kernel	sample	I, 8	CNN, layered embedding
Lin et al. [45]	PD	SA	kernel	sample	O, 1	CNN, three-scale features, attention mechanism
Lin et al. [44]	PD	PAR	kernel	radiance	O, 4	CNN, light transport covariance
Kettunen et al. [34]	PD	PAG	radiance	gradient	O, 4	CNN, U-Net, perceptual loss
Guo et al. [16]	PD	PAG	radiance	gradient	O, 4	CNN, unsupervised learning
Zhu et al. [69]	PD, CA	PAO	kernel	photon	O, 1	CNN, caustic decomposition
Zeng et al. [68]	PD, CA	PAO	kernel	photon	O, 1	CNN, caustic decomposition
Chaitanya et al. [7]	SD	PA	radiance	temporal	I, 1	RNN, autoencoder
Hasselgren et al. [19]	SD, AS	PA	radiance	temporal	I, 4	CNN, U-Net with recurrent feedback
Meng et al. [48]	SD	PA	kernel	temporal	I, 1	CNN, differentiable neural bilateral grid
Kallweit et al. [32]	VD	V	radiance	volume	O, 1	MLP, hierarchical 3D descriptors
Panin et al. [52]	VD	V	radiance	volume	O, 1	MLP, baking light probes
Hofman et al. [23]	VD	PA	radiance	image	O, 1	CNN, GAN, dual autoencoder
Xu et al. [65]	VD	PAVG	radiance	gradient	O, 1	CNN, photon density estimation
Bako et al. [3]	RD, AS	RA	radiance	radiance	O, 4	CNN, GAN
Huo et al. [26]	RD, AS	RA	radiance	radiance	O, 16	CNN, DRL, 4D convolution
Jiang et al. [29]	RD	RA	radiance	radiance	O, 1	CNN, autoencoder

[9] H. Dahlberg, D. Adler, and J. Newlin. Machine-learning denoising in feature film production. In *ACM SIGGRAPH 2019 Talks*, pages 1–2. 2019.

[10] H. Dammertz, D. Sewtz, J. Hanika, and H. P. Lensch. Edge-avoiding à-trous wavelet transform for fast global illumination filtering. In *Proceedings of the Conference on High Performance Graphics*, pages 67–75. Citeseer, 2010.

[11] R. A. Drebin, L. Carpenter, and P. Hanrahan. Volume rendering. *ACM Siggraph Computer Graphics*, 22(4):65–74, 1988.

[12] F. Durand, N. Holzschuch, C. Soler, E. Chan, and F. X. Sillion. A frequency analysis of light transport. *ACM Transactions on Graphics (TOG)*, 24(3):1115–1126, 2005.

[13] M. Gharbi, T.-M. Li, M. Aittala, J. Lehtinen, and F. Durand. Sample-based monte carlo denoising using a kernel-splatting network. *ACM Transactions on Graphics (TOG)*, 38(4):1–12, 2019.

[14] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio. *Deep learning*, volume 1. MIT press Cambridge, 2016.

[15] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27:2672–2680, 2014.

[16] J. Guo, M. Li, Q. Li, Y. Qiang, B. Hu, Y. Guo, and L.-Q. Yan. Gradnet: unsupervised deep screened poisson reconstruction for gradient-domain rendering. *ACM Transactions on Graphics (TOG)*, 38(6):1–13, 2019.

[17] T. Hachisuka, S. Ogaki, and H. W. Jensen. Progressive photon mapping. In *ACM SIGGRAPH Asia 2008 papers*, pages 1–8. 2008.

[18] J. Hanika, M. Droske, and L. Fascione. Manifold next event estimation. In *Computer graphics forum*, volume 34, pages 87–97. Wiley Online Library, 2015.

[19] J. Hasselgren, J. Munkberg, M. Salvi, A. Patney, and A. Lefohn. Neural temporal adaptive sampling and denoising. In *Computer Graphics Forum*, volume 39, pages 147–155. Wiley Online Library, 2020.

[20] T. Hastie, R. Tibshirani, and J. Friedman. *The elements of statistical learning: data mining, inference, and prediction*. Springer Science & Business Media, 2009.

[21] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[22] H. Hey and W. Purgathofer. Importance sampling with hemispherical particle footprints. In *Proceedings of the 18th spring conference on Computer graphics*, pages 107–114, 2002.

[23] N. Hofmann, J. Martschinke, K. Engel, and M. Stamminger. Neural denoising for path tracing of medical volumetric data. *Proceedings of the ACM on*

Computer Graphics and Interactive Techniques, 3(2):1–18, 2020.

[24] B.-S. Hua, A. Gruson, V. Petitjean, M. Zwicker, D. Nowrouzezahrai, E. Eisemann, and T. Hachisuka. A survey on gradient-domain rendering. In *Computer Graphics Forum*, volume 38, pages 455–472. Wiley Online Library, 2019.

[25] Y. Huang, W. Wang, and L. Wang. Bidirectional recurrent convolutional networks for multi-frame super-resolution. *Advances in neural information processing systems*, 28:235–243, 2015.

[26] Y. Huo, R. Wang, R. Zheng, H. Xu, H. Bao, and S.-E. Yoon. Adaptive incident radiance field sampling and reconstruction using deep reinforcement learning. *ACM Transactions on Graphics (TOG)*, 39(1):1–17, 2020.

[27] H. W. Jensen. Importance driven path tracing using the photon map. In *Eurographics Workshop on Rendering Techniques*, pages 326–335. Springer, 1995.

[28] H. W. Jensen. *Realistic image synthesis using photon mapping*. AK Peters/CRC Press, 2001.

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

[30] J. T. Kajiya. The rendering equation. In *Proceedings of the 13th annual conference on Computer graphics and interactive techniques*, pages 143–150, 1986.

[31] N. K. Kalantari, S. Bako, and P. Sen. A machine learning approach for filtering monte carlo noise. *ACM Trans. Graph.*, 34(4):122–1, 2015.

[32] S. Kallweit, T. Müller, B. Mcwilliams, M. Gross, and J. Novák. Deep scattering: Rendering atmospheric clouds with radiance-predicting neural networks. *ACM Transactions on Graphics (TOG)*, 36(6):1–11, 2017.

[33] C.-m. Kang, L. Wang, Y.-n. Xu, and X.-x. Meng. A survey of photon mapping state-of-the-art research and future challenges. *Frontiers of Information Technology & Electronic Engineering*, 17(3):185–199, 2016.

[34] M. Kettunen, E. Härkönen, and J. Lehtinen. Deep convolutional reconstruction for gradient-domain rendering. *ACM Transactions on Graphics (TOG)*, 38(4):1–12, 2019.

[35] M. Kettunen, M. Manzi, M. Aittala, J. Lehtinen, F. Durand, and M. Zwicker. Gradient-domain path tracing. *ACM Transactions on Graphics (TOG)*, 34(4):1–13, 2015.

[36] A. Kuznetsov, N. K. Kalantari, and R. Ramamoorthi. Deep adaptive sampling for low sample count rendering. In *Computer Graphics Forum*, volume 37, pages 35–44. Wiley Online Library, 2018.

[37] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

[38] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.

[39] Y. LeCun, B. E. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. E. Hubbard, and L. D. Jackel. Handwritten digit recognition with a back-propagation network. In *Advances in neural information processing systems*, pages 396–404, 1990.

[40] J. Lehtinen, T. Karras, S. Laine, M. Aittala, F. Durand, and T. Aila. Gradient-domain metropolis light transport. *ACM Transactions on Graphics (TOG)*, 32(4):1–12, 2013.

[41] M. Levoy and P. Hanrahan. Light field rendering. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 31–42, 1996.

[42] T.-M. Li, Y.-T. Wu, and Y.-Y. Chuang. Sure-based optimization for adaptive sampling and reconstruction. *ACM Transactions on Graphics (TOG)*, 31(6):1–9, 2012.

[43] Y. Liang, B. Wang, L. Wang, and N. Holzschuch. Fast computation of single scattering in participating media with refractive boundaries using frequency analysis. 2019.

[44] W. Lin, B. Wang, L. Wang, and N. Holzschuch. A detail preserving neural network model for monte carlo denoising. *Computational Visual Media*, pages 1–12, 2020.

[45] W. Lin, B. Wang, J. Yang, L. Wang, and L.-Q. Yan. Path-based monte carlo denoising using a three-scale neural network. *Computer Graphics Forum*, n/a(n/a).

[46] N. Max. Optical models for direct volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 1(2):99–108, 1995.

[47] S. U. Mehta, B. Wang, and R. Ramamoorthi. Axis-aligned filtering for interactive sampled soft shadows. *ACM Transactions on Graphics (TOG)*, 31(6):1–10, 2012.

[48] X. Meng, Q. Zheng, A. Varshney, G. Singh, and M. Zwicker. Real-time monte carlo denoising with the neural bilateral grid. In C. Dachsbacher and M. Pharr, editors, *Eurographics Symposium on Rendering - DL-only Track*. The Eurographics Association, 2020.

[49] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.

[50] J. Munkberg and J. Hasselgren. Neural denoising with layer embeddings. In *Computer Graphics Forum*, volume 39, pages 1–12. Wiley Online Library, 2020.

[51] Nvidia. Interactive reconstruction of monte carlo image sequences using a recurrent denoising autoencoder. https://research.nvidia.com/publication/interactive-reconstruction-monte-carlo-image-sequences-using-recurrent-denoising, 2020.

[52] M. Panin and S. Nikolenko. Faster rpnn: Rendering clouds with latent space light probes. In *SIGGRAPH Asia 2019 Technical Briefs*, pages 21–24. 2019.

[53] M. Pharr, W. Jakob, and G. Humphreys. *Physically based rendering: From theory to implementation*. Morgan Kaufmann, 2016.

[54] F. Rosenblatt. Principles of neurodynamics. perceptrons and the theory of brain mechanisms.

Technical report, Cornell Aeronautical Lab Inc Buffalo NY, 1961.
[55] R. Y. Rubinstein and D. P. Kroese. *Simulation and the Monte Carlo method*, volume 10. John Wiley & Sons, 2016.
[56] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning internal representations by error propagation. Technical report, California Univ San Diego La Jolla Inst for Cognitive Science, 1985.
[57] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986.
[58] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489, 2016.
[59] C. M. Stein. Estimation of the mean of a multivariate normal distribution. *The annals of Statistics*, pages 1135–1151, 1981.
[60] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103, 2008.
[61] T. Vogels, F. Rousselle, B. McWilliams, G. Röthlin, A. Harvill, D. Adler, M. Meyer, and J. Novák. Denoising with kernel prediction and asymmetric loss functions. *ACM Transactions on Graphics (TOG)*, 37(4):1–15, 2018.
[62] K.-M. Wong and T.-T. Wong. Deep residual learning for denoising monte carlo renderings. *Computational Visual Media*, 5(3):239–255, 2019.
[63] Q. Xing and C. Chen. Path tracing denoising based on sure adaptive sampling and neural network. *IEEE Access*, 8:116336–116349, 2020.
[64] B. Xu, J. Zhang, R. Wang, K. Xu, Y.-L. Yang, C. Li, and R. Tang. Adversarial monte carlo denoising with conditioned auxiliary feature modulation. *ACM Trans. Graph.*, 38(6):224–1, 2019.
[65] Z. Xu, Q. Sun, L. Wang, Y. Xu, and B. Wang. Unsupervised image reconstruction for gradient-domain volumetric rendering. In *Computer Graphics Forum*, volume 39, pages 193–203. Wiley Online Library, 2020.
[66] X. Yang, D. Wang, W. Hu, L. Zhao, X. Piao, D. Zhou, Q. Zhang, B. Yin, Q. Cai, and X. Wei. Fast reconstruction for monte carlo rendering using deep convolutional networks. *IEEE Access*, 7:21177–21187, 2018.
[67] X. Yang, D. Wang, W. Hu, L.-J. Zhao, B.-C. Yin, Q. Zhang, X.-P. Wei, and H. Fu. Demc: A deep dual-encoder network for denoising monte carlo rendering. *Journal of Computer Science and Technology*, 34(5):1123–1135, 2019.
[68] Z. Zeng, L. Wang, B.-B. Wang, C.-M. Kang, and Y.-N. Xu. Denoising stochastic progressive photon mapping renderings using a multi-residual network. *Journal of Computer Science and Technology*, 35:506–521, 2020.
[69] S. Zhu, Z. Xu, H. W. Jensen, H. Su, and R. Ramamoorthi. Deep kernel density estimation for photon mapping. In *Computer Graphics Forum*, volume 39, pages 35–45. Wiley Online Library, 2020.
[70] M. Zwicker, W. Jarosz, J. Lehtinen, B. Moon, R. Ramamoorthi, F. Rousselle, P. Sen, C. Soler, and S.-E. Yoon. Recent advances in adaptive sampling and reconstruction for monte carlo rendering. In *Computer graphics forum*, volume 34, pages 667–681. Wiley Online Library, 2015.

Yuchi Huo is graduated from Zhejiang University and working at SGVR Lab. at KAIST. His research interests are in rendering, deep learning, image processing, and computational optics.

Sung-Eui Yoon is a professor at KAIST (Korea Advanced Institute of Sci. and Tech.). Currently, he is leading SGVR Lab. (Scalable Graphics, Vision, & Robotics) at KAIST. His research interests span scalable rendering, vision, and robotics problems including ray tracing, image search, and motion planning for robots.