DOI: 10.1111/cgf.14052
Eurographics Symposium on Rendering 2020
C. Dachsbacher and M. Pharr
(Guest Editors)

Volume 39 (2020), Number 4

Deep Kernel Density Estimation for Photon Mapping

Shilin Zhu[1*] Zexiang Xu[1*] Henrik Wann Jensen[1, 2] Hao Su[1] Ravi Ramamoorthi[1]

[1]University of California, San Diego [2]Luxion

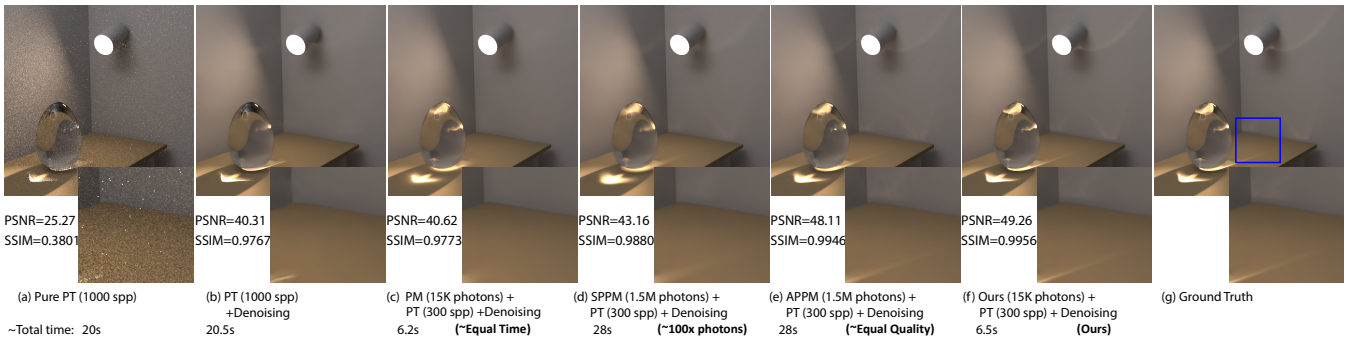

Figure 1: *We present a novel learning-based photon mapping (PM) method that can be used to synthesize photorealistic images (f) with detailed caustics (shown and compared in the insets) from very sparse photons for scenes with complex diffuse-specular interactions. In particular, we use our method with only 15k photons (∼0.06 photons per pixel) to compute accurate global illumination for light-specular paths. We use path tracing (PT) with a moderate number (300) of samples per pixel (spp) to compute the other paths and apply the Optix learning-based denoiser (based on [CKS*17]) to remove the Monte Carlo (MC) noise. In contrast, pure PT leads to noisy results lacking focused caustics (a) even with 1000 spp that is significantly more than our photon and path samples. While this noise can be mitigated using a learning-based denoiser, this introduces artifacts and cannot recover the caustics (b). Combining PT and standard PM [Jen96] with 15k photons, and then denoising (c), avoids these artifacts but still does not reconstruct caustics accurately from such low photon counts. While providing 1.5M photons (this is 100 times the number of photons our method uses) and applying the advanced stochastic progressive PM (SPPM) [HJJ10] enables a more accurate result (d), it is still slightly worse than ours. In contrast, our result (f) accurately reproduces the caustic effects in the global illumination, as compared to the ground truth (g), with significantly fewer samples. Ours is comparable with (if not better than) the result from adaptive progressive PM (APPM) [KD13] with 100 times the number of photons (e).*

Abstract

Recently, deep learning-based denoising approaches have led to dramatic improvements in low sample-count Monte Carlo rendering. These approaches are aimed at path tracing, which is not ideal for simulating challenging light transport effects like caustics, where photon mapping is the method of choice. However, photon mapping requires very large numbers of traced photons to achieve high-quality reconstructions. In this paper, we develop the first deep learning-based method for particle-based rendering, and specifically focus on photon density estimation, the core of all particle-based methods. We train a novel deep neural network to predict a kernel function to aggregate photon contributions at shading points. Our network encodes individual photons into per-photon features, aggregates them in the neighborhood of a shading point to construct a photon local context vector, and infers a kernel function from the per-photon and photon local context features. This network is easy to incorporate in many previous photon mapping methods (by simply swapping the kernel density estimator) and can produce high-quality reconstructions of complex global illumination effects like caustics with an order of magnitude fewer photons compared to previous photon mapping methods. Our approach largely reduces the required number of photons, significantly advancing the computational efficiency in photon mapping.

CCS Concepts

• ***Computing methodologies*** → ***Rendering;***

1. Introduction

Computing global illumination is crucial for photorealistic image synthesis. Ray tracing-based methods have been widely used to simulate complex light transport effects with global illumination in film, animation, video game and other industrial fields. The most successful approaches are based on either Monte Carlo (MC) integration, like path tracing [Kaj86, Vea97], or particle density estimation, like photon mapping [Jen96]. Photon mapping techniques are able to efficiently simulate caustics and other challenging light transport effects, which are very hard and even impossible for pure Monte Carlo-based methods to simulate.

In general, both MC-based and particle-based methods require numerous samples to render noise-free images, and are thus computationally expensive. Recently, significant progress has been made in denoising MC images rendered with low sample counts using deep learning techniques [CKS*17, BVM*17]. However, there is relatively little work in particle-based methods for low-sample reconstruction and current photon mapping techniques still require a very large number of traced photons to achieve accurate, artifact-free radiance estimation.

We present the first deep learning-based approach for particle-based rendering that enables efficient, high-quality global illumination with a small number of photons. Our approach is particularly good at reconstructing diffuse-specular interactions like caustics, for which previous photon mapping methods require large photon sample counts (and path-tracing at reasonable sample counts can miss altogether). We focus on photon density estimation—a key component of all particle-based methods—and introduce a novel deep neural network that can estimate accurate photon density at any surface points in a scene given only sparsely distributed photons.

Previously, the most successful density estimation methods for photon mapping are kernel-based methods that use traditional kernel functions (like a uniform or cone kernel) to compute output radiance at a surface point as a weighted sum of nearby photons. While previous methods have improved the kernels by controlling the kernel bandwidths or shapes [KD13, SSFO08, KWX*16], traditional kernel functions still require a large enough count of photons located in a small enough bandwidth around every surface shading point, for which a very large number of photons need to be traced, to compute accurate photon density. In contrast, *we propose to learn to predict a kernel function at each shading point to effectively aggregate nearby photon contributions*. Our predicted kernels leverage data priors and are able to compute accurate photon density estimation for complex global illumination from photon counts that are an order of magnitude fewer than traditional methods.

Our network considers local photons around a queried surface point within a predefined bandwidth as input. Unlike traditional methods that often treat photons individually or leverage standard statistics to aggregate photons, *we leverage learned local photon statistics—encoded as a deep photon context vector inferred by the network—around a surface point for per photon kernel weight estimates*. Specifically, the network first processes individual photons to extract per-photon features and aggregates them across photons using pooling operations to obtain a deep photon context feature that represents the local photon statistics. The network processes the individual per-photon features concatenated with the local context to compute per-photon kernel weights, which are used to perform density estimation by a weighted sum. We demonstrate that this approach of learning kernel prediction is more efficient than a baseline that directly estimates photon density from the aggregated deep context vector.

To train our network, we create diverse photon distributions by tracing photons in 500 procedurally generated scenes with complex shapes and materials. We sample surface points on diffuse surfaces, which form a 512×512 image (one pixel per point) in each scene, and we compute the ground truth photon density of each point using progressive photon mapping [HOJ08] with billions of photons. Note that, our network focuses on local photon distribution properties of surface points. Hence, every surface point in a scene is a training datum, allowing us to train a generalizable network without a lot of images.

In Fig. 1, we demonstrate that, using only 15k photons, our method can synthesize high-quality images. Conversely, variations of path tracing and photon mapping fail to do so; even when combined with advanced progressive and adaptive techniques, SPPM and APPM require significantly more samples (1.5M photons) to achieve comparable results. This makes our approach an important step towards making photon mapping computationally efficient. Moreover, our experiments leverages an effective practical hybrid approach: using our method for reconstructing light-specular (LS) paths – the light transport paths that interact with specular surfaces before arriving at light sources—and low sample-count path tracing with learning-based denoising for all other light transport paths. This leverages the advantages of both MC denoising and our efficient photon density estimation technique.

2. Related Work

Monte Carlo path integration. Kajiya [Kaj86] introduced the rendering equation and Monte Carlo (MC) path tracing. Since then, various methods for MC path integration have been developed, including light tracing [DLW93], bidirectional path tracing (BDPT) [LW93, VG95], and Metroplis light transport (MLT) [Vea97, PKK00, CTE05]. These methods are able to simulate complex light transport with accurate global illumination in an unbiased way. However, pure MC based methods typically require a very large number of samples (traced paths), especially for very low probability paths like the classical caustic or specular-diffuse-specular (SDS) paths. We base our method on the photon mapping technique, which is efficient for caustics and SDS, and we aim to achieve sparse reconstruction.

Monte Carlo denoising. While there is little progress in sparse reconstruction with low sample counts in photon mapping, many approaches have been proposed to achieve MC rendering with low sample counts. A recent survey of sparse sampling and reconstruction is presented by Zwicker et al. [ZJL*15]. MC denoising meth-

* Equal contribution.

ods can be categorized into a-priori methods that rely on prior theoretical knowledge [DHS*05,ETH*09,YMRD15,WYKR17], and a-posteriori methods that filter out the noise in rendered images with few assumptions about the image signal [ODR09,RMZ13,KBS15].

Recently, deep learning techniques have been introduced to achieve MC denoising [CKS*17, BVM*17], and many methods utilize kernel prediction [BVM*17, VRM*18, XZW*19]. Kalantari et al. [KBS15] propose to predict the parameters of fixed filtering functions using fully-connected neural networks. Bako et al. [BVM*17] leverage deep convolution neural networks to predict kernels to linearly combine the original noisy radiances of neighboring pixels. Gharbi et al. [GLA*19] make use of individual screen-space path samples and predict a kernel for each sample that splats the radiance contributions to its neighboring pixels. Deep learning techniques have also been extended to gradient domain rendering [KHL19]. In contrast, we apply deep learning in photon density estimation and leverage local photon statistics for density estimation from sparse photons. Our network considers individual scene-space photon samples around each shading point and predicts a kernel to gather per-photon contributions. Our approach is the first that introduces deep learning in photon mapping and demonstrates learning-based kernel prediction in this context.

Photon density estimation. The rendering equation [Kaj86, ICG86] can be approximated by particle density estimation [SWH*95, Jen96, WHSG97]. Most particle-based methods are based on the original photon mapping framework [Jen96]; it first traces rays from light sources to distribute photons in a scene, and then gathers neighboring photons at individual shading points to approximate radiance estimates. Photon mapping achieves low variance in the rendered images and leads to blurred, less noticeable artifacts at the cost of introducing bias in the estimates. Photon mapping is able to consistently converge to the correct solution by increasing the number of photons towards infinity and reducing the bandwidth towards zero.

Previous work has investigated progressive methods to overcome the memory bottleneck and enable arbitrarily large photon numbers [HOJ08, HJ09, HJJ10, KZ11], bidirectional methods to improve rendering glossy objects [Vor11], adaptive methods to optimize photon tracing [HJ11], and the combination of unbiased MC methods and photon mapping [GKS11, HPJ12, GKDS12]. Many relevant works have been presented to improve the kernel density estimation by utilizing standard statistics for adaptive kernel bandwidth [JC95, KD13, KWX*16] or anisotropic kernel shapes [SSFO08]. Other works leverage ray differentials [SFES07], blue noise distribution [SJ09, SJ13a, SJ13b], traditional linear regression [HSR*15] and Gaussian mixtures fitting [JRJ11] to improve the reconstruction. In contrast, we focus on accurately computing photon density with sparse photons, which hasn't been explored in previous work. Essentially, we replace the traditional kernel density estimation with a novel deep learning based module, and keep the rest unchanged in the standard photon mapping framework. This potentially enables the combination of our technique and previous photon mapping techniques that focus on other components in the framework.

3. Background: Density estimation

Photon mapping techniques compute reflected radiance via density estimation. Kernel density estimation [WJ94] is the most widely used density estimation method in statistics, and has been widely applied in photon mapping. Early works use the uniform kernel that treats nearby photons equally [Jen96, HOJ08]; subsequent works extend photon density estimation to support arbitrary smooth kernels [HJJ10, KZ11]. In general, the reflected radiance at a shading location $\boldsymbol{x}$ is computed by:

$$L(\boldsymbol{x},\boldsymbol{\omega}) \approx \frac{1}{N}\sum_{i=1}^{N} k_r(\boldsymbol{x},\boldsymbol{x}_i)\tau_i, \tag{1}$$

where N is the total number of photon paths that are emitted in a scene, $\boldsymbol{\omega}$ is the reflected direction, $\boldsymbol{x}_i$ is the location of a photon, τ_i is the photon contribution and k_r represents the kernel function with bandwidth r. In general, the photon contribution τ_i is the product of the BRDF and the photon energy. In this work, we only compute photon density on diffuse surfaces, as is done in many classical photon mapping methods. In this case, the BRDF at a shading point is ρ/π, where ρ is the albedo. Correspondingly, $\tau_i = \phi\rho/\pi$, where ϕ represents the accumulated path contribution divided by the sampling probability, which can be also interpreted as the energy flux carried by the photon. Therefore, $\boldsymbol{\omega}$ can be removed and ρ can be taken out of the summation in Eqn. 1. We therefore consider the photon energy ϕ as the photon contribution in this work.

The kernel k_r assigns linear weights to photons, which are used to linearly combine the contributions of photons in a local window with radius r. Traditionally, k_r is a uniform function $(1/(\pi r^2))$ or a function of the distance from the shading point to a photon $(\|\boldsymbol{x} - \boldsymbol{x}_i\|)$. Instead, we propose to leverage data priors to predict kernels to aggregate photon contributions.

4. Learning to compute photon density

In this section, we present our learning-based approach for photon density estimation. Our approach is light-weight and focuses on density estimation only; we keep the main framework of standard photon mapping and upgrade the traditional, distance-based and photon-independent kernel functions (k_r in Eqn. 1) to novel, learned and local-context-aware kernel functions represented by a deep neural network (see Fig. 2).

In particular, given a shading point, our network considers its K nearest neighbor photons, which adaptively selects the bandwidth r. Multiple properties of individual photons are used as input for the network, including photon positions $\{\boldsymbol{x}_i\}_{i=1}^{K}$, photon directions $\{\boldsymbol{d}_i\}_{i=1}^{K}$ and photon contributions $\{\phi_i\}_{i=1}^{K}$. We also supply the number of nearest photons K to the network to let it better understand the local photon distribution. Our network (denoted as Φ) regresses per-photon kernel weights to compute radiance estimates via a weighted sum similar to Eqn. 1:

$$L(\boldsymbol{x}) \approx \frac{\rho}{N\pi r^2}\sum_{i=1}^{K} \Phi_{r,i}(\boldsymbol{x},\{\boldsymbol{x}_i\},\{\boldsymbol{d}_i\},\{\phi_i\})\phi_i, \tag{2}$$

where $\Phi_{r,i}$ represents the predicted kernel weight for photon i. Note that, our network uses information about *all* photons in a local

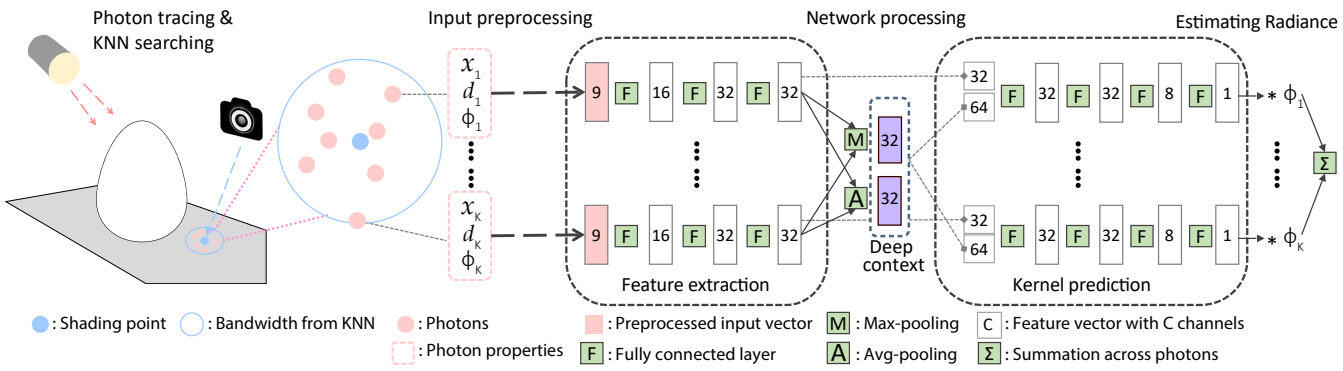

Figure 2: *Overview of our deep photon density estimation network. Given a set of photons within the bandwidth of a shading point, we pre-process these photons' properties and input them to feature extractor MLPs that compute per-photon features. These are aggregated using max- and average-pooling to construct a deep context feature. The original per-photon features and the deep context are concatenated and processed by a kernel prediction MLP that predicts a kernel weight. Finally, these kernel weights are used to sum the photon contributions and produce the reflected radiance.*

neighborhood for per-photon kernel prediction; it obtains deep photon statistics and associates per-photon information with statistical context to compute kernels for photon aggregation.

4.1. Input pre-processing

Photon distributions are highly diverse across shading points and across scenes, making it challenging to design a network that generalizes across different inputs. Besides, deep neural networks are known to benefit from normalized input data to correlate values from different domains. Therefore, we pre-process the input photon properties to allow for better generalizability and performance.

Since light intensities can have very high dynamic range (HDR), the photon contributions τ_i can vary widely in range, which is highly challenging for a network to process. We introduce a mapping function to pre-process the photon contributions,

$$t_a(u) = \frac{\log(u+a) - \log(a)}{\log(u+a) - \log(a) + 1}, \tag{3}$$

where $a = 0.01$ is an additional parameter. Essentially, $t_a(u)$ maps HDR values u from $[0,\infty]$ to $[0,1]$. We further linearly map these values to $[-1,1]$ and provide them as network input. We observe that such a mapping process facilitates the network learning.

For photon positions $\boldsymbol{x}_i$ and directions $\boldsymbol{d}_i$, we first transform them into the local coordinate frame of the shading point; the coordinate frame is constructed using the position and normal of the shading point and two orthogonal directions that are randomly selected in the tangent plane. This transforms the network inputs into a consistent coordinate system and improves generalizability.

The bandwidth r of our learned kernel is determined by the distance of the K^{th} nearest photon. This leads to a large range of bandwidth values given various photon distributions, which is highly challenging for a deep neural network to process. Motivated by the bandwidth normalization used in traditional kernels [WJ94, SWH*95], we divide the photon positions in the local coordinates by the bandwidth r, and scale the final density estimates by $1/r^2$, which is shown in Eqn. 2. This normalizes all input photon positions into a unit sphere and post-scales the computed photon density by the actual window area. As a result, our network is invariant to the actual bandwidths, and effectively generalizes to different photon distributions and supports different numbers of *total emitted photons* that will introduce different bandwidths for the same K.

Note that, different terms of our network input are all normalized into the range of $[-1,1]$, which enables our network to correlate and leverage different photon properties from various domains in an efficient way. Our input pre-processing also makes our network translation-, rotation-, and scale- invariant to diverse photon distributions, leading to good generalization across different scenes and different numbers of emitted photons.

4.2. Network architecture

The inputs to our network are essentially a set of multi-feature 3D points in a unit sphere. In the set, there is no meaningful inherent point ordering and the number of points (K) is not fixed. We thus leverage PointNet [QSMG17] style neural networks with multi-layer perceptrons, which accept an arbitrary number of inputs and are invariant to permutations of inputs. As shown in Fig. 2, our network consists of two sub-networks, a feature extractor and a kernel predictor; they are both fully connected neural networks and process each photon individually.

The feature extractor first processes each individual photon; it considers the pre-processed photon properties (9 channels including positions, directions and contributions) as input, and extracts meaningful features using multilayer perceptrons. Specifically, we use three fully connected layers in the feature extractor, and each layer is followed by a ReLu activation layer. The feature extractor leverages linear and non-linear operations to transform the original input into a learned 32-channel feature vector. These per-photon features are then aggregated across photons by max-pooling and average-pooling operations which output the deep photon context

vector. This vector represents the local photon statistics in a learned non-linearly transformed space. The kernel predictor then leverages the across-photon context and the per-photon features to predict a single scalar that represents the kernel weight for each photon. These per-photon kernel weights are the final output of our network and will be used to linearly combine the original photon contributions as expressed in Eqn. 2. The kernel predictor is also a three-layer fully connected neural network with ReLU as activation layers, which is similar to the feature extractor but with different channels at each layer.

Note that, unlike previous work that treats each photon independently, we propose to correlate per-photon information with local context information across photons. Our feature extractor transforms photon properties into learned feature vectors, which allows for collecting photon statistics in the learned neural feature space to obtain the photon context for the following kernel prediction. Our whole network is very light-weight, and involves only six fully connected layers; this ensures a highly efficient inference process. We show that such a light-weight network is able to effectively reconstruct accurate photon density from sparse photons.

4.3. Training details

Data generation. Monte Carlo denoising usually requires a large number of images to train and is hard to generalize across different type of scenes. Our method focuses on local photon distributions; in other words, to learn proper data priors, we desire the diversity of photon distributions in terms of individual shading points and not necessarily of the entire scenes. This allows for good generalizability of our network with a relatively small number of training scenes, which can even be very different from our final testing scenes. Inspired by [XSHR18, XBS*19], we procedurally create shapes from primitives with random sizes and random bump maps; a set (randomly from 1 to 16) of such shapes are then placed in a box and distributed roughly as a grid. We also place multiple area lights with random locations and rotations in the scene, and randomly assign specular materials and diffuse materials to the scene objects.

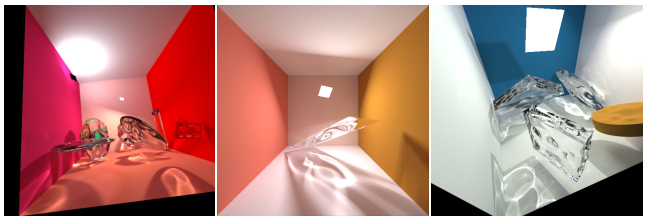

Figure 3: *Examples of our procedurally generated training scenes.*

A few examples of these scenes are shown in Fig 3; complex light transport effects with diverse photon distributions are simulated. To sample shading points in each scene, we shoot rays from a camera through an image plane with 512×512 pixels and select the first diffuse intersections as target shading points. We trace photons from light sources and keep the ones that contribute to the indirect lighting. Progressive photon mapping [HOJ08] is then applied to compute ground truth photon densities for each point with a total number of about 1 billion photon paths. For each scene, we store 10 million photon paths and a 512×512 multi-channel image that contains the ground truth radiances and other necessary information (positions, normals and BRDFs) of shading points. We create 500 scenes for training our neural networks and test our network on scenes that are significantly different from our training data (see Fig. 1 and Fig. 8).

Loss function. We supervise our network with the ground truth radiance estimates. The final radiances are in high dynamic range, which can easily make the training dominated by high-intensity values; we therefore tone-map the radiance estimates using the μ-law as in [KR17]. The mapping function $p_\mu(v)$ is given by:

$$p_\mu(v) = \frac{\log(1+\mu v)}{\log(1+\mu)}, \tag{4}$$

and we set $\mu = 5000$ following [KR17]. We tone-map both our estimated radiance and the ground truth radiance, and we apply L_2 loss on the mapped values.

Training parameters. We randomly select K from 100 to 800 and use from 0.3 million to 4 million photons to train our network, which makes it generalize well to various bandwidths and photon counts. We use Adam to train our network for 6000 epochs with an initial learning rate of 10^{-4} and a batch size of 2000 random shading points.

5. Experiments

We now present a comprehensive evaluation of our method.

Ablation study. We first justify the choices of our network design. In particular, we compare our network with a baseline network that estimates the final radiance without predicting kernels; this comparison network has a similar network architecure but directly outputs the final irradiance from the across-photon deep context vector. Figure 4 shows the training processes of these networks; our network converges significantly faster than the baseline method. This demonstrates the effectiveness of combining kernel density estimation and deep learning and is consistent with previous results on denoising for path tracing [BVM*17, VRM*18, GLA*19].

Evaluation scenes and photon generation. We evaluate our method on six challenging scenes (GLASS EGG, RED WINE, RINGS, WATER POOL1, WATER POOL2, DRAGON) that involve complex caustics and other diffuse-specular interactions with LS paths. In theory, LS paths can never be reconstructed by path tracing if we use a point light source; we therefore use area lights in the scenes to allow for reasonable comparisons with PT. For each scene, we shoot photons for 0.1 second, which generates about 0.8M photon paths with at maximum five photons per path; we only keep those photons that involve light-specular paths in the scenes. We denote the number of valid photons we consider as M, which is a number that is different from the total emitted photon paths N in Eqn. 1. Because of various compositions of scenes, there are 15k (GLASS EGG), 85k (RED WINE), 77k (RINGS), 50k (WATER POOL1), 100k (WATER POOL1) and 125k (WATER POOL1) valid photons that are used in the six scenes respectively. We also evaluate with the number of photons that are traced in one second—corresponding to ten times the number of photons traced in 0.1

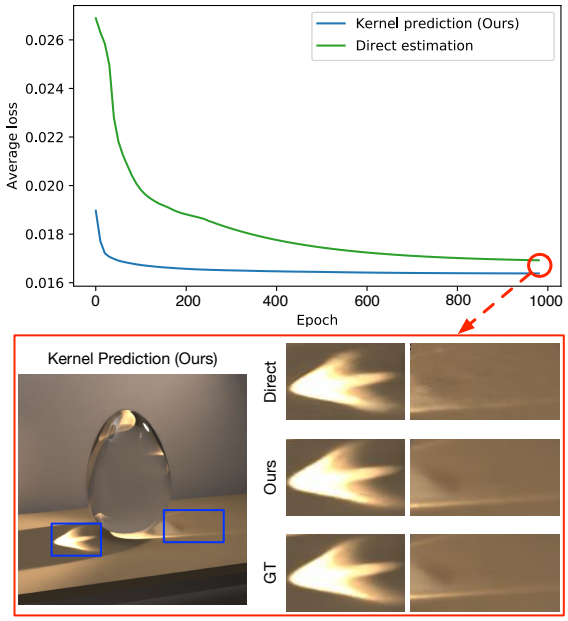

Figure 4: *Kernel prediction. We compare the optimization speed of our kernel-prediction network, and a baseline direct-estimation network. Our network converges faster to the lower loss value. On the bottom, we show the rendered images from the two networks trained with 1000 epochs. Our results are closer to the ground truth with more details in the caustics.*

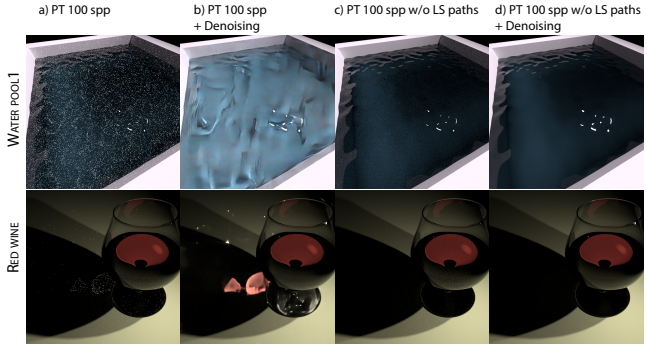

Figure 5: *Path tracing and (PT) without light-specular paths (LS). We show PT and denoising results using 100 spp with and without light-specular paths. The noise can be seen more clearly when zooming into the electronic PDF.*

seconds—to justify the generalization of our network to different numbers of emitted photons, and compare with the other methods with photons that are traced in ten seconds to justify the quality of our sparse reconstruction.

Combining MC denoising and deep photon mapping. We evaluate our deep photon density estimation by combining our method with MC denoising. Specifically, we apply our learning-based density estimation to only compute the challenging light transport effects which involve LS paths that are extremely hard to trace in PT and likely to introduce caustics. In addition, we use path tracing with relatively low sample counts to compute the remaining light transport paths, and use modern learning-based denoising—the Optix built-in denoiser based on [CKS*17]—to remove the MC noise.

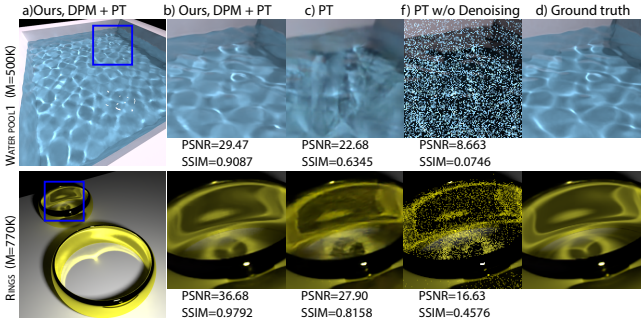

Figure 6: *We show our final results in full images (a). Our final results are computed by combining our deep photon mapping results and path tracing with denoising. We compare against pure path tracing using 1000 spp with (c) and without(d) denoising on insets. Obviously, path tracing alone even with 1000 spp cannot handle the LS paths.*

By removing LS paths in PT, we also make PT and MC denoising much easier. As shown in Fig. 5, PT without LS paths can be effectively denoised using modern learning-based denoising techniques with 100 spp, whereas full PT with LS paths introduces extensive noise with the same 100 spp, causing denoising to fail completely. In fact, the standard PT plus denoising pipeline is not able to recover the complex light transport effects with even 1000 spp (see Figs. 1,6). In contrast, we demonstrate a practical way of combining our efficient deep photon mapping with MC denoising for photorealistic image synthesis, in which we leverage the benefits of low-sample reconstruction in both scene-space particle density estimation and screen-space MC integration.

Parameters of our network and comparison methods. We observe that it is very hard for a single network to generalize across different numbers of input photons (K). We thus use a fixed K when training per network, and specifically we train two networks with $K = 50$ and $K = 500$ for the evaluation. We also compare with a variant of our network that has four times the channels at each layer in our network architecture to evaluate if larger network capacity leads to higher performance. This large network generally leads to better performance (see Tab. 1), but it requires about three times longer inference time (see Tab. 2); please see the following parts in the section for more discussion about quality and performance. In the experiments, we use DPM (deep photon mapping) to denote the network with regular capacity and DPM-L (or Ours-L) to denote the one with larger capacity.

In all experiments, we compare with the classical photon mapping (PM) with the same k-NN photons as inputs. We also compare with various progressive methods that are designed to progressively reduce the bandwidth with large photon counts. In particular, for

Scene	(*M*)	Ours-50	Ours-L-50	PM-50	Ours-500	Ours-L-500	PM-500	PPM	APPM
GLASS EGG	(15k)	0.013	0.006	0.085	0.013	0.006	0.165	0.085	0.080
	(150k)	0.012	0.006	0.036	0.008	0.004	0.079	0.065	0.043
	(1.5M)	0.013	0.007	0.031	0.006	0.003	0.027	0.030	0.030
RED WINE	(85k)	0.052	0.028	0.116	0.044	0.021	0.222	0.134	0.111
	(850k)	0.035	0.027	0.053	0.023	0.014	0.102	0.064	0.047
	(8.5M)	0.032	0.030	0.045	0.014	0.011	0.037	0.031	0.026
RINGS	(77k)	0.042	0.023	0.069	0.023	0.008	0.153	0.137	0.143
	(770k)	0.041	0.024	0.046	0.011	0.006	0.042	0.050	0.049
	(7.7M)	0.045	0.020	0.066	0.012	0.009	0.023	0.017	0.014
WATER POOL1	(50k)	0.244	0.174	0.281	0.214	0.146	0.323	0.327	0.277
	(500k)	0.214	0.173	0.221	0.135	0.115	0.244	0.249	0.193
	(5.0M)	0.237	0.186	0.259	0.107	0.105	0.124	0.206	0.125
WATER POOL2	(102k)	0.178	0.125	0.226	0.167	0.095	0.260	0.262	0.224
	(1.0M)	0.132	0.121	0.147	0.115	0.080	0.221	0.211	0.155
	(10.2M)	0.134	0.128	0.159	0.066	0.061	0.102	0.163	0.088
DRAGON	(125k)	0.066	0.054	0.073	0.052	0.043	0.089	0.126	0.102
	(1.2M)	0.056	0.054	0.061	0.034	0.033	0.044	0.083	0.054
	(12.5M)	0.059	0.059	0.078	0.028	0.027	0.031	0.059	0.035

Table 1: *Quantitative RMSE evaluation. We test our networks trained with different K ($K = 50$ and 500, denoted with Ours-K) on six novel scenes with different numbers of valid photons (M). We also test a variant of our network architecture with enlarged four times capacity (Ours-Large) using the same K. We compare RMSE against standard photon mapping (PM) [Jen96] under the same conditions, and also progressive PM (PPM) [HOJ08] and adaptive PPM (APPM) [KD13]. We highlight the best and the second best results in red and blue for each row; note that, all of them are our results. We also highlight the best result of the comparison methods in yellow, which is often worse than any of our network settings.*

Photon tracing	Photon gathering	Number of photons	DPM-50	DPM-L-50	DPM-500	DPM-L-500
0.1s	0.12s∼ 0.5s	15k∼ 125k	0.3s	1.0s	3.0s	10.0s
1.0s	1.2s∼ 5.0s	150k∼ 1.2M	0.3s	1.0s	3.0s	10.0s
10.0s	12s∼ 50s	1.5M∼ 12M	0.3s	1.0s	3.0s	10.0s

Table 2: *Timing. We show the corresponding running time in seconds for each photon mapping component. Our experiments are run with photons that are traced within 0.1s, 1.0s and 10.0s in each scene. We list the corresponding gathering time to find the neighboring photons for about 512×512 surface shading points. The numbers of total photons are also shown, corresponding to the M in Tab. 1. We list the network inference time for 512×512 surface shading points for our regular network (DPM) and a large network (DPM-L) with $K = 50$ and 500. Note that, the network inference time is determined by its capacity and K, and is independent of the number of total photons in the scene.*

density estimation at fixed surface points, we compare with progressive photon mapping (PPM) [HOJ08]. Given a certain number of input photons, the quality of PPM is influenced by the initial radius and the number of photons per iteration. To make a fair comparison, we compare 30 different variants (10 radii and 3 photon counts per iteration) of the two parameters and choose the best settings (with lowest RMSEs) for each scene. We also compare with adaptive progressive photon mapping [KD13] similarly using the best radius and number of photons per iteration from 30 different variants of parameters. For visual comparisons, we compare with stochastic PPM (SPPM) [HJ09], when there are transparent surfaces in a scene which require sampling multiple surface points per pixel.

Quantitative and qualitative evaluation. We now evaluate our method quantitatively and qualitatively with different numbers of photons counts (*M*) and different variations of training parameters (input photon number *K* and capacity). Table 1 shows quantitative RMSE evaluation of photon density estimation on the six testing scenes; the numbers are averaged across about 260k surface shading points sampled by tracing rays from a camera and selecting the first diffuse hit points in the scenes. Note that, across all these different scenes with different photon counts, our method with $K = 500$ performs consistently better than all the comparison PM methods, including standard PM [Jen96], PPM [HOJ08] and APPM [KD13], with the same number of total photons. Most of our results are better than PM's and PPM's results with ten times the photon counts as ours. APPM leverages traditional statistical information of local photons to improve the density estimation of PPM, which is able to achieve fairly good results; however, it requires the number of photons to be large enough to obtain good statistics. In contrast, our method leverages learned statistics in the network, which achieves significantly better results than APPM with the same number of photons; ours is actually comparable to

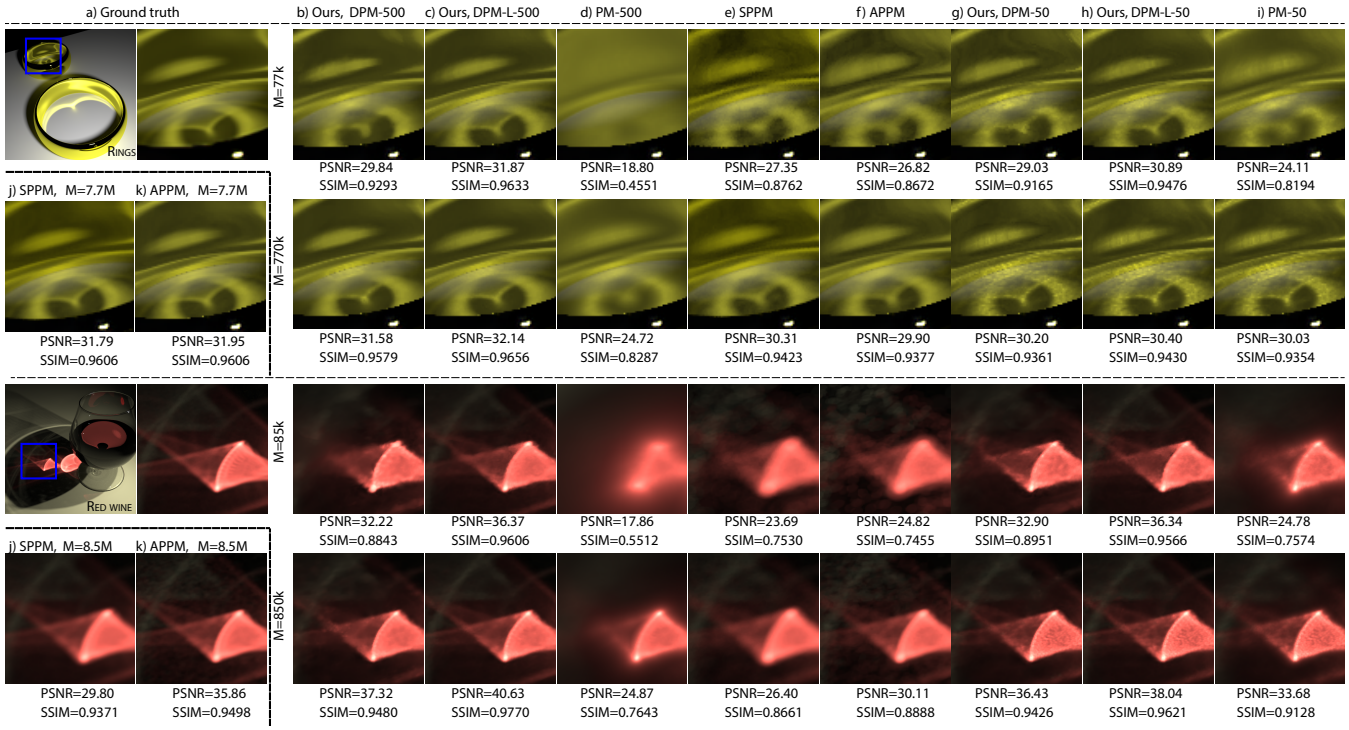

Figure 7: *We show results of our method with different numbers of input photons (K). We compare against PM, SPPM and APPM with the same number of total photons (M) on insets marked in the left-top ground truth image. We also show the results of APPM and PPM with ten times the largest number of photons our method uses (j, k). The PSNRs and SSIMs of the insets are shown correspondingly.*

Mean DSSIM	Ours-50	Ours-Large-50	PM-50	Ours-500	Ours-Large-500	PM-500
	0.0346	0.0342	0.0337	0.0281	0.0277	0.0260

Table 3: *Temporal stability. We show the mean DSSIM between pairs of adjacent frames over a sequence of 30 rendered frames. Results have been averaged over all the different scenes and amount of photons.*

the APPM that uses ten times the total number of photons. Note that, the APPM and PPM results are selected from the results of tens of APPM and PPM variants with different hyper-parameters for their best performance; yet, our method still outperforms the best of these variants.

To visually illustrate the numbers in Tab. 1, we demonstrate all the rendering results of RINGS and RED WINE with the first two rows (first two M) in Fig. 7; we also show the visual results of APPM and PPM with larger M in Fig. 7.j, k. Additionally, we show results of three testing scenes in Fig. 8, where we compare our our DPM-500 with PM and SPPM. In Fig. 1, we show the result of our DPM-50 and compare with PT, PM, SPPM and APPM. In general, our method with $K = 500$ outperforms the comparison methods with the same number of photons qualitatively and quantitatively. And our results are comparable to (if not better than) the comparison methods that use ten times the number of photons in the scene. While the larger network with $K = 500$ (Ours-L-500) performs better than the regular network, the larger one also requires longer inference time (see Tab. 2). Therefore, our regular network with $K = 500$ is generally the best choice for most cases, which stably achieves high-quality results. However, when timing is not a critical issue, the large network will be a better choice for higher accuracy.

In most cases, the Ours-500 ($K = 500$) results are better than the Ours-50 ($K = 50$) ones, indicating that our network is usually in favor of more nearest neighbor photons (K) as input. Essentially, a larger K allows for better local deep statistics in the deep context feature, which enables better kernel predictions. Note that, this is not the case for standard PM using the same nearest neighbor strategy for bandwidth selection. Photon mapping either introduces obvious non-smooth artifacts with a small bandwidth (Fig. 7.i) or outputs over-smooth results without details with a large bandwidth (Fig. 7.d). APPM tends to resolve this issue by wisely reducing the bandwidth according to the photon statistics. In contrast, our method achieves significantly better results than APPM when there are only sparsely distributed photons. Our method is able to leverage a relatively large bandwidth without introducing any obvious over-smoothing issues. This is thanks to our learning based context-aware kernel prediction approach. In particular, our approach allows for every single photon to leverage across-photon information

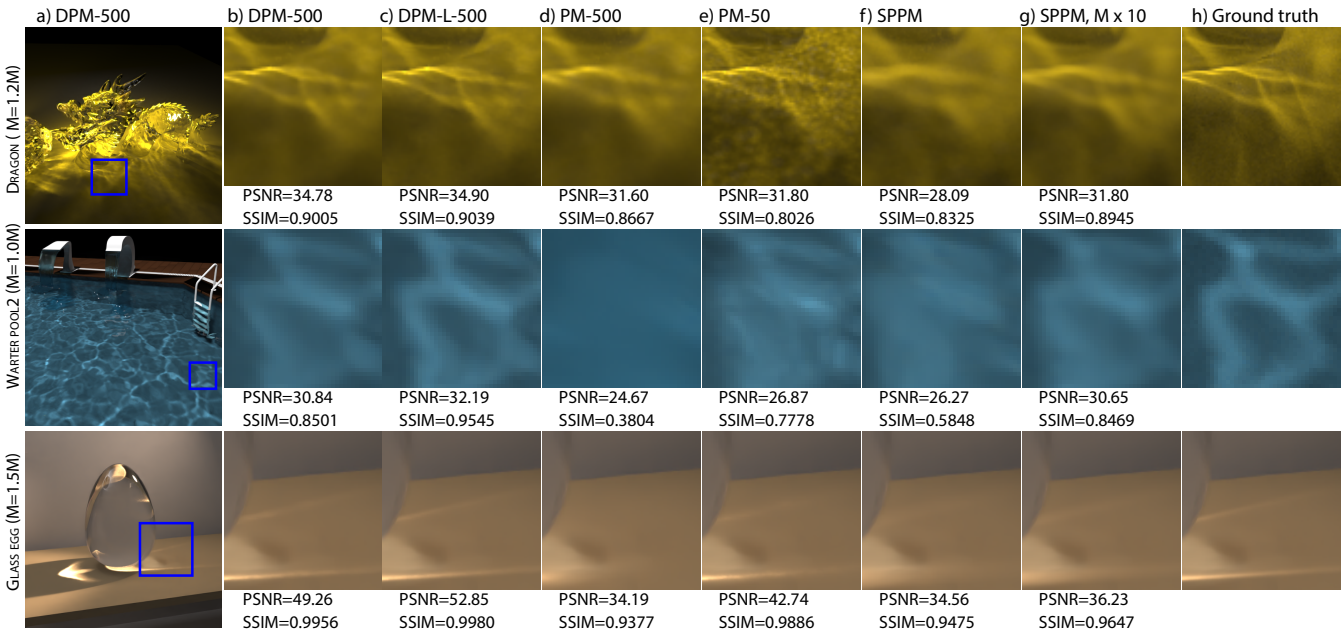

Figure 8: *We show our results on full images (a). We compare against PM with the same input photons (d) and SPPM with the same (f) and ten times (g) the total photon counts on insets. PSNRs and SSIMs are also calculated for all insets and listed below.*

in the learned deep context feature to tell if it is an outlier or an important contributing element to the shading point's reflected radiance; a corresponding kernel weight is assigned to each photon based on the decision made by data priors in the network. Therefore, our method is able to effectively utilize the sparse photons in a large area to generate photorealistic images that are of high smoothness and have many details.

Timing. We use Optix to trace photons and do path tracing for all the results. All experiments are run on one NVIDIA 1080 Ti GPU. Path tracing runs at about 50 spp per second in all six scenes with an image resolution of 512×512. It takes about 0.1, 1.0 and 10 seconds to emit photons. We show the corresponding photon gathering time and network inference time for 512×512 surface shading points in Tab. 2. In particular, we build Kd-Trees to do the neighboring search at each shading point and all methods take similar time to gather neighboring photons. Note that the running time of our network is linear with the number of input photons K; it is also determined by the number of shading points that are required to be computed, and the listed timing corresponds to 512×512 shading points. The total running time for our method is the summation of the photon tracing, gathering and the network inference time; the total running time for the other methods is the summation of tracing and gathering. Note that, across all the experiments (Tab. 1, Fig. 7, Fig. 8), our results of DPM-500 with photons traced in 1 second are comparable to the best results of comparison methods with photons traced in 10 seconds; however, to achieve the comparable results, our DPM-500 takes about 5.2s$\sim$ 9s total time, whereas the comparison methods require 22s$\sim$ 60.0s total time to compute the same number of shading points. Our method takes significantly shorter time to achieve the comparable quality.

Network capacity and K. While our network is mainly trained with relatively sparse photons (small M), our network with K=500 overall generalizes well across different numbers of total photons (M) and, in most cases, achieves better performance when M increases. However, for K=50, there is too little information for the network to leverage and higher performance is often not ensured with a larger M. Nonetheless, our network with K=50 still works well and performs better than the comparison methods when there are tens of thousands of photons. We also observe that a larger network (Ours-L) with larger capacity leads to clearly better results than our regular network. Of course, a larger network requires higher computational cost or longer inference time as shown in Tab. 2. Yet, the larger network with $K = 50$ can already often achieve reasonably good results, which takes shorter running time than $K = 500$. We leave the exploration of more variants of the network capacity and K as future work.

Temporal consistency. Since our method deals with shading points in 3D space and is independent of view directions, we have observed that it has good across-frame consistency when changing the view in a scene with a fixed set of photons. We follow [VRM*18] and use the mean DSSIM between consecutive frames to evaluate the temporal consistency when moving the camera. Results in Table 3 show comparable temporal stability between our results and standard PM outputs. We leave the extensions of our network to recurrent architectures and general temporal consistency with other dynamic components in a scene as future work.

Progressive density estimation. Our current framework requires a fixed number of input photons for each trained network. Progressive photon mapping accepts different numbers of photons per iter-

ation with reduced bandwidth. Nonetheless, we have demonstrated that our network architecture supports accurate photon density estimation with various fixed photon numbers. In other words, a progressive method can potentially be achieved by training a sequence of networks with different numbers of inputs. A universal network for any given number of input photons may require introducing recurrent networks in the framework, which is an interesting direction of future work.

6. Conclusions and Future Work

In this paper, we have presented the first deep learning-based method for density estimation in particle-based rendering. We introduce a deep neural network that learns a kernel function to aggregate photons at each shading point and renders accurate caustics with significantly fewer photons than previous approaches, with minimal overhead. Learning-based MC denoising has significantly improved path tracing results and our work extends these benefits to the popular photon mapping method.

Our method could be improved in the future with more advanced machine learning approaches, perhaps based on generative adversarial networks (GANs), just as has been done with path tracing [XZW*19]. More broadly, we believe this paper points towards denoisers specialized to many other approaches for realistic image synthesis such as Metropolis Light Transport and Vertex Connection and Merging.

Acknowledgements

We thank Kalyan Sunkavalli for insightful discussions. This work was supported in part by NSF grants 1617234, 1703957 and 1764078, ONR grant N000141712687, the Ronald L. Graham Chair, a Google Fellowship, an Adobe Fellowship and the UC San Diego Center for Visual Computing.

References

[BVM*17] 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 (TOG) 36*, 4 (2017), 97. 2, 3, 5

[CKS*17] 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 (TOG) 36*, 4 (2017), 98. 1, 2, 3, 6

[CTE05] CLINE D., TALBOT J., EGBERT P.: Energy redistribution path tracing. In *ACM Transactions on Graphics (TOG)* (2005), vol. 24, ACM, pp. 1186–1195. 2

[DHS*05] DURAND F., HOLZSCHUCH N., SOLER C., CHAN E., SILLION F. X.: A frequency analysis of light transport. *ACM Transactions on Graphics (TOG) 24*, 3 (2005), 1115–1126. 3

[DLW93] DUTRÉ P., LAFORTUNE E. P., WILLEMS Y. D.: Monte carlo light tracing with direct computation of pixel intensities. 2

[ETH*09] EGAN K., TSENG Y.-T., HOLZSCHUCH N., DURAND F., RAMAMOORTHI R.: Frequency analysis and sheared reconstruction for rendering motion blur. In *ACM Transactions on Graphics (TOG)* (2009), vol. 28, ACM, p. 93. 3

[GKDS12] GEORGIEV I., KRIVÁNEK J., DAVIDOVIC T., SLUSALLEK P.: Light transport simulation with vertex connection and merging. *ACM Trans. Graph. 31*, 6 (2012), 192–1. 3

[GKS11] GEORGIEV I., KŘIVÁNEK J., SLUSALLEK P.: Bidirectional light transport with vertex merging. In *SIGGRAPH Asia 2011 Sketches* (2011), ACM, p. 27. 3

[GLA*19] 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 (TOG) 38*, 4 (2019), 1–12. 3, 5

[HJ09] HACHISUKA T., JENSEN H. W.: Stochastic progressive photon mapping. In *ACM Transactions on Graphics (TOG)* (2009), vol. 28, ACM, p. 141. 3, 7

[HJ11] HACHISUKA T., JENSEN H. W.: Robust adaptive photon tracing using photon path visibility. *ACM Transactions on Graphics (TOG) 30*, 5 (2011), 114. 3

[HJJ10] HACHISUKA T., JAROSZ W., JENSEN H. W.: A progressive error estimation framework for photon density estimation. In *ACM Transactions on Graphics (TOG)* (2010), vol. 29, ACM, p. 144. 1, 3

[HOJ08] HACHISUKA T., OGAKI S., JENSEN H. W.: Progressive photon mapping. In *ACM Transactions on Graphics (TOG)* (2008), vol. 27, ACM, p. 130. 2, 3, 5, 7

[HPJ12] HACHISUKA T., PANTALEONI J., JENSEN H. W.: A path space extension for robust light transport simulation. *ACM Transactions on Graphics (TOG) 31*, 6 (2012), 191. 3

[HSR*15] HUANG X., SUN X., REN Z., TONG X., GUO B., ZHOU K.: Irradiance regression for efficient final gathering in global illumination. *Frontiers of Computer Science 9*, 3 (2015), 456–465. 3

[ICG86] IMMEL D. S., COHEN M. F., GREENBERG D. P.: A radiosity method for non-diffuse environments. *Acm Siggraph Computer Graphics 20*, 4 (1986), 133–142. 3

[JC95] JENSEN H. W., CHRISTENSEN N.: Photon maps in bidirectional monte carlo ray tracing of complex objects. *Computers & Graphics 19*, 2 (1995), 215–224. 3

[Jen96] JENSEN H. W.: Global illumination using photon maps. In *Rendering Techniques' 96*. Springer, 1996, pp. 21–30. 1, 2, 3, 7

[JRJ11] JAKOB W., REGG C., JAROSZ W.: Progressive expectation-maximization for hierarchical volumetric photon mapping. In *Computer Graphics Forum* (2011), vol. 30, Wiley Online Library, pp. 1287–1297. 3

[Kaj86] KAJIYA J. T.: The rendering equation. In *ACM SIGGRAPH computer graphics* (1986), vol. 20, ACM, pp. 143–150. 2, 3

[KBS15] KALANTARI N. K., BAKO S., SEN P.: A machine learning approach for filtering monte carlo noise. *ACM Trans. Graph. 34*, 4 (2015), 122–1. 3

[KD13] KAPLANYAN A. S., DACHSBACHER C.: Adaptive progressive photon mapping. *ACM Transactions on Graphics (TOG) 32*, 2 (2013), 16. 1, 2, 3, 7

[KHL19] KETTUNEN M., HÄRKÖNEN E., LEHTINEN J.: Deep convolutional reconstruction for gradient-domain rendering. *ACM Transactions on Graphics (TOG) 38*, 4 (2019), 1–12. 3

[KR17] KALANTARI N. K., RAMAMOORTHI R.: Deep high dynamic range imaging of dynamic scenes. *ACM Trans. Graph. 36*, 4 (2017), 144–1. 5

[KWX*16] KANG C.-M., WANG L., XU Y.-N., MENG X.-X., SONG Y.-J.: Adaptive photon mapping based on gradient. *Journal of Computer Science and Technology 31*, 1 (2016), 217–224. 2, 3

[KZ11] KNAUS C., ZWICKER M.: Progressive photon mapping: A probabilistic approach. *ACM Transactions on Graphics (TOG) 30*, 3 (2011), 25. 3

[LW93] LAFORTUNE E. P., WILLEMS Y.: Bi-directional path tracing. In *Compugraphics' 93* (1993), pp. 145–153. 2

[ODR09] OVERBECK R. S., DONNER C., RAMAMOORTHI R.: Adaptive wavelet rendering. *ACM Trans. Graph. 28*, 5 (2009), 140. 3

[PKK00] PAULY M., KOLLIG T., KELLER A.: Metropolis light transport for participating media. In *Rendering Techniques 2000*. Springer, 2000, pp. 11–22. 2

[QSMG17] QI C. R., SU H., MO K., GUIBAS L. J.: Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2017), pp. 652–660. 4

[RMZ13] ROUSSELLE F., MANZI M., ZWICKER M.: Robust denoising using feature and color information. In *Computer Graphics Forum* (2013), vol. 32, Wiley Online Library, pp. 121–130. 3

[SFES07] SCHJØTH L., FRISVAD J. R., ERLEBEN K., SPORRING J.: Photon differentials. In *Proceedings of the 5th international conference on Computer graphics and interactive techniques in Australia and Southeast Asia* (2007), pp. 179–186. 3

[SJ09] SPENCER B., JONES M. W.: Into the blue: Better caustics through photon relaxation. In *Computer Graphics Forum* (2009), vol. 28, Wiley Online Library, pp. 319–328. 3

[SJ13a] SPENCER B., JONES M. W.: Photon parameterisation for robust relaxation constraints. In *Computer Graphics Forum* (2013), vol. 32, Wiley Online Library, pp. 83–92. 3

[SJ13b] SPENCER B., JONES M. W.: Progressive photon relaxation. *ACM Transactions on Graphics (TOG) 32*, 1 (2013), 1–11. 3

[SSFO08] SCHJØTH L., SPORRING J., FOGH OLSEN O.: Diffusion based photon mapping. In *Computer Graphics Forum* (2008), vol. 27, Wiley Online Library, pp. 2114–2127. 2, 3

[SWH*95] SHIRLEY P., WADE B., HUBBARD P. M., ZARESKI D., WALTER B., GREENBERG D. P.: Global illumination via density-estimation. In *Rendering Techniques' 95*. Springer, 1995, pp. 219–230. 3, 4

[Vea97] VEACH E.: *Robust Monte Carlo methods for light transport simulation*, vol. 1610. Stanford University PhD thesis, 1997. 2

[VG95] VEACH E., GUIBAS L. J.: Optimally combining sampling techniques for monte carlo rendering. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques* (1995), ACM, pp. 419–428. 2

[Vor11] VORBA J.: Bidirectional photon mapping. In *Proc. of the Central European Seminar on Computer Graphics (CESCG'11)* (2011). 3

[VRM*18] 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 (TOG) 37*, 4 (2018), 124. 3, 5, 9

[WHSG97] WALTER B., HUBBARD P. M., SHIRLEY P., GREENBERG D. P.: Global illumination using local linear density estimation. *ACM Transactions on Graphics (TOG) 16*, 3 (1997), 217–259. 3

[WJ94] WAND M. P., JONES M. C.: *Kernel smoothing*. Chapman and Hall/CRC, 1994. 3, 4

[WYKR17] WU L., YAN L.-Q., KUZNETSOV A., RAMAMOORTHI R.: Multiple axis-aligned filters for rendering of combined distribution effects. In *Computer Graphics Forum* (2017), vol. 36, Wiley Online Library, pp. 155–166. 3

[XBS*19] XU Z., BI S., SUNKAVALLI K., HADAP S., SU H., RAMAMOORTHI R.: Deep view synthesis from sparse photometric images. *ACM Transactions on Graphics (TOG) 38*, 4 (2019), 1–13. 5

[XSHR18] XU Z., SUNKAVALLI K., HADAP S., RAMAMOORTHI R.: Deep image-based relighting from optimal sparse samples. *ACM Transactions on Graphics (TOG) 37*, 4 (2018), 126. 5

[XZW*19] XU B., ZHANG J., WANG R., XU K., YANG Y.-L., TANG R.: Adversarial Monte Carlo denoising with conditioned auxiliary feature modulation. *ACM Transactions on Graphics (TOG) 38*, 6 (2019), 224. 3, 10

[YMRD15] YAN L.-Q., MEHTA S. U., RAMAMOORTHI R., DURAND F.: Fast 4d sheared filtering for interactive rendering of distribution effects. *ACM Transactions on Graphics (TOG) 35*, 1 (2015), 7. 3

[ZJL*15] ZWICKER M., JAROSZ W., LEHTINEN J., MOON B., RAMAMOORTHI R., ROUSSELLE F., SEN P., SOLER C., YOON S.-E.: Recent advances in adaptive sampling and reconstruction for monte carlo rendering. In *Computer Graphics Forum* (2015), vol. 34, Wiley Online Library, pp. 667–681. 2