Neural Complex Luminaires: Representation and Rendering

JUNQIU ZHU, Shandong University, China
YAOYI BAI, University of California, Santa Barbara, USA
ZILIN XU, Shandong University, China
STEVE BAKO, University of California, Santa Barbara, USA
EDGAR VELÁZQUEZ-ARMENDÁRIZ, Pure Storage, USA
LU WANG, Shandong University, China
PRADEEP SEN, University of California, Santa Barbara, USA
MILOŠ HAŠAN, Adobe Research, USA
LING-QI YAN, University of California, Santa Barbara, USA

Fig. 1. Our neural complex luminaire framework efficiently represents complicated light sources that are extremely costly to evaluate using traditional approaches due to the convoluted paths light takes through their geometry. Our approach easily integrates into standard Monte Carlo rendering systems including those using multiple importance sampling (MIS). Here we compare our results rendered with forward path tracing at 128 samples per pixel (spp) to the reference rendered using bidirectional path tracing (BDPT) at 1024 spp on two scenes with different luminaires. Our method is able to match both the luminaire appearance as well as the high frequency light patterns on the wall. Overall, we can attain similar quality and low-noise results with fewer samples per pixel (resulting in speedups of 15× and nearly 22×, respectively), without requiring the original luminaire geometry at runtime.

Complex luminaires, such as grand chandeliers, can be extremely costly to render because the light-emitting sources are typically encased in complex refractive geometry, creating difficult light paths that require many samples to evaluate with Monte Carlo approaches. Previous work has attempted to speed up this process, but the methods are either inaccurate, require the storage of very large lightfields, and/or do not fit well into modern path-tracing frameworks. Inspired by the success of deep networks, which can model complex relationships robustly and be evaluated efficiently, we propose to use a machine learning framework to compress a complex luminaire's lightfield into an implicit neural representation. Our approach can easily plug into conventional renderers, as it works with the standard techniques of path tracing and multiple importance sampling (MIS). Our solution is to train three networks to perform the essential operations for *evaluating* the complex luminaire at a specific point and view direction, *importance sampling* a point on the luminaire given a shading location, and *blending* to determine the transparency of luminaire queries to properly composite them with other scene elements. We perform favorably relative to state-of-the-art approaches and render final images that are close to the high-sample-count reference with only a fraction of the computation and storage costs, with no need to store the original luminaire geometry and materials.

CCS Concepts: • **Computing methodologies** → **Rendering**.

Additional Key Words and Phrases: neural rendering, complex luminaires

Authors' addresses: Junqiu Zhu, Shandong University, Jinan, Shandong, China, 250100, junqiuzhu@mail.sdu.edu.cn; Yaoyi Bai, University of California, Santa Barbara, Santa Barbara, CA, USA, 93106, yaoyibai@cs.ucsb.edu; Zilin Xu, Shandong University, Jinan, Shandong, China, 250100, zilin.xu@mail.sdu.edu.cn; Steve Bako, University of California, Santa Barbara, Santa Barbara, CA, USA, 93106, stevebako@ucsb.edu; Edgar Velázquez-Armendáriz, Pure Storage, Mountain View, CA, USA, 94041, evelazquez@acm.org; Lu Wang, Shandong University, Jinan, Shandong, China, 250100, luwang_hcivr@sdu.edu.cn; Pradeep Sen, University of California, Santa Barbara, Santa Barbara, CA, USA, 93106, psen@ece.ucsb.edu; Miloš Hašan, Adobe Research, San Jose, CA, USA, 95110, milos.hasan@gmail.com; Ling-Qi Yan, University of California, Santa Barbara, Santa Barbara, CA, USA, 93106, lingqi@cs.ucsb.edu.

0730-0301/2021/8-ART57
https://doi.org/10.1145/3450626.3459798

ACM Reference Format:
Junqiu Zhu, Yaoyi Bai, Zilin Xu, Steve Bako, Edgar Velázquez-Armendáriz, Lu Wang, Pradeep Sen, Miloš Hašan, and Ling-Qi Yan. 2021. Neural Complex Luminaires: Representation and Rendering. *ACM Trans. Graph.* 40, 4, Article 57 (August 2021), 12 pages. https://doi.org/10.1145/3450626.3459798

1 INTRODUCTION

Physically-based rendering is a heavily studied problem, and there has been significant progress in the complexity and quality of supported geometry and materials. However, the *light sources* in commonly rendered scenes still consist almost exclusively of simple light models such as point or area lights, diffusely emitting meshes, and environment maps. On the other hand, much more complex luminaires can be pervasively seen in everyday life, from clear light bulbs with exposed tungsten filaments, to the grand chandeliers in concert halls, to sophisticated light fixtures popular in home furnishings. Supporting such complex luminaires in practical renderers would provide an additional sense of realism and aesthetic value. While several previous methods focused on rendering complex luminaires, their solutions still have significant complexity, high computational or storage requirements, and/or reduced generality.

Why is it not easy to bring such complex luminaires into practical rendering systems? The difficulty comes not only from their complex geometries, but even more so from the complex light transport within them. From the point at which light is emitted to where it actually leaves the luminaire, light can take long, circuitous paths, often with multiple specular reflections and refractions. This configuration is challenging for common light transport methods, since we need to resolve not only the *forward* transport from the emitter outwards, but the *reverse* problem of connecting a shading point anywhere in the scene to the emitter. This prevents the use of direct illumination (next-event estimation) techniques, which are important for practical Monte Carlo rendering. Bidirectional estimators can be used instead, but do not usually perform as well as traditional next-event estimation techniques (if those are available).

Furthermore, it would be preferable to represent complex luminaires as "black boxes," such that no original geometric, material and emitter data needs to be stored to use them in a target renderer. This is desirable not only to make the performance independent of their geometric complexity, but also because special material and emitter models may be needed, and a luminaire manufacturer may prefer not to publicize the detailed internal design of their products.

To address this problem, previous work made several trade-offs. Far-field illumination profiles, such as the IES light profile standard, record measured intensity as the light exits a large sphere enclosing the complex luminaire, but are purely directional and completely ignore the non-zero size of the light, leading to inaccurate illumination when a close-by shading point is being lit. Moreover, they also do not address the problem of rendering the appearance of the luminaire when observed directly. Traditional discretized 4D surface light fields allow for direct viewing, but result in heavy storage. While the discretized lightfield representation can be importance-sampled [Lu et al. 2015], additional large data structures are required.

The approach of Velázquez et al. [2015] delivers the highest quality, but requires keeping fully detailed luminaire models, and does not integrate easily with modern path tracers. Recently, volumetric neural compressed representations received attention [Mildenhall et al. 2020], but these methods are expensive to query and generally designed for pure view synthesis, i.e., observing an object directly from different viewpoints. They do not contribute light to the surrounding scene and are not designed for efficient integration into full global illumination systems.

Our goal is to design a luminaire representation without any of these shortcomings (see Fig. 1 and the supplemental video). In order to facilitate the rendering of such luminaires in standard modern Monte Carlo path-tracing systems with direct illumination and multiple importance sampling (MIS), the representation needs to support three essential operations:

(1) *Evaluation:* Directly querying the complex luminaire from a specific direction. This operation allows rays to hit and query the luminaire, including for direct camera views when the luminaire itself is visible in the final image.
(2) *Importance sampling:* Given a shading point on an arbitrary scene surface, sampling a point on the complex luminaire that is visible to the shading point, ideally with probability density closely approximating the amount contributed towards the shading point. For MIS integration, we also need to be able to evaluate the probability density of this sampling.
(3) *Blending:* To properly composite the luminaire with other scene elements seen in the background, we need to know its transparency mask from a given direction.

We observe that neural networks are good candidates for such a representation to model luminaires: they are fast to evaluate, can model complex functional relationships accurately, achieve good compression ratios, can reasonably interpolate observed data, and to some extent, can even filter noisy training data.

Therefore, in this paper, we present a set of neural networks to implement the above operations. To model the luminaire, we first enclose it in a simple geometrical proxy, such as a sphere or cylinder. We then train an *evaluation network* to query the emitted lightfield on the surface of the proxy in any direction; this allows rays to intersect the luminaire and query its emission. Next, a *sampling network* will choose, for any point in space, a direction towards the luminaire with density proportional to the amount of light contribution to the point. This is critical for direct illumination in practical path tracing. Finally, we train a *transparency network* that gives, for any ray intersecting the proxy, the (approximate) amount of transmission of the background.

In summary, our approach can represent complex luminaires in a compact fashion that is specifically tailored for efficient rendering. We also show how to integrate these neural networks into a modern path-tracing system with multiple importance sampling (MIS). The details of the luminaires are well preserved, the data is highly compressed, and the rendering performance is greatly boosted compared to attempting a full light-transport simulation.

2 RELATED WORK

Complex light transport methods. Naturally, one could consider using general light transport approaches to render scenes with fully realized luminaire models. Path tracing with direct illumination [Kajiya 1986] is a very effective approach to render scenes in

which connections to light emitters can be made easily from most scene surfaces. The problem with applying this approach to complex luminaires is that the actual emitter is often hidden deep beneath complex refractive and reflective elements, and cannot be connected with direct path segments. Hitting the emitter by chance is the only possibility, making this method not practically viable.

Bidirectional techniques are capable of rendering direct lighting from complex luminaires, either through connecting light subpaths on surfaces directly to the camera (as in bidirectional path tracing [Veach 1997]), or by vertex merging with camera vertices (progressive photon mapping [Hachisuka et al. 2008] or vertex connection and merging [Georgiev et al. 2012; Hachisuka et al. 2012]). The problem with all approaches that trace subpaths from the light is that it takes a massive number of subpaths traced from the luminaire to achieve a photon (light vertex) density on scene surfaces that could compare with the number of direct connections normally made by a standard path tracer. These light subpaths are expensive, and many light vertices would still end up outside of camera view. Instead, our goal is to design a luminaire representation that does allow direct light connections to a simplified proxy around the light. Essentially, our goal is to turn a complex luminaire with sophisticated internal transport into a simple proxy with a surface lightfield and an associated sampling method.

Methods based on Metropolis light transport[Veach and Guibas 1997] can, to some extent, handle mutations of light subpaths from luminaires onto surfaces. However, if the luminaire is sufficiently complex, such mutations often fall off the specular manifold, reducing the efficiency of this approach. Finally, note that the light transport paths emitted from inside complex luminaires are independent of the scene they are used in, and can be precomputed (e.g., by our method). A general light transport method[Jakob and Marschner 2012] would instead keep retracing the same paths within the luminaire for every scene it is used in, which is fundamentally inefficient.

Luminaire-specific approaches. A classical approach to represent the far-field, directional illumination distribution from luminaires is to measure it with a goniophotometer, or simulate it by particle tracing. These distributions are the basis of the IES standard [2002]. Early work measured the lightfield at a virtual enclosing surface (proxy) as captured directly from a luminaire, to decouple geometry from emission function [Ashdown 1995; Ngai 1987]. In principle, these approaches represent a lightfield on the proxy geometry like ours, but their sampling density is too low to convincingly represent it. Planar representations of lightfields allow standard renderers to be re-purposed for generating lightfields from any scenes rendered from many different points of view [Gortler et al. 1996; Levoy and Hanrahan 1996]; however, these early methods were not focused on luminaires, and did not address direct illumination nor non-planar luminaire geometry.

Canned lightsources present a similar concept to precompute the emitted lightfield of a luminaire [Heidrich et al. 1998]; however, the resolution is low and data storage quite large. Lightfield compression has been explored using an assortment of classical techniques: wavelets, vector quantization, non-negative matrix factorization, and PCA [Chang et al. 2003, 2006; Chen et al. 2002].

Another way to represent the complex luminaire is a near-field rayset (essentially a set of lightfield samples in a suitable parameterization). Muschaweck [2011] and Ashdown and Rykowski [1998] use a rayset method to record the rays leaving the luminaire and capture them on a virtual bounding surface. Mas et al. [2008] transform the rayset to clusters to compress the rayset data, and do also support importance sampling. These methods handle smaller emitting elements (e.g., lightbulbs) well, but would be difficult to scale to more complex luminaires. Position-dependent importance sampling of complex luminaires [Lu et al. 2015] present an efficient method to importance-sample a discretized lightfield for direct illumination from the view of a given shading point. However, dense uncompressed lightfield storage is still required by that method. Directional photon mapping has been used for directly visualizing luminaires but requires many photons to be really useful; evaluating the illumination coming from the luminaire requires an expensive lookup [Kniep et al. 2009].

Velázquez et al. [2015] is probably the most comprehensive previous solution. The illumination from the luminaire is approximated by a large number of anisotropic point lights (APLs), each of which has a tabulated directional distribution. Direct viewing of the luminaire is instead implemented by keeping the full luminaire model, path-tracing it up to a maximum depth (e.g. 5 bounces), and looking up an approximate precomputed radiance volume if the emitter has not been hit. This method is not trivial to integrate in a path tracing framework with MIS, and to our knowledge this has never been done. While this technique can give high quality results, our method has several advantages, including not requiring full luminaire storage, faster queries, integration into the MIS framework, and more seamless compositing into scenes.

Deep learning in graphics. Deep learning has become wildly popular across a wide range of disciplines since its demonstration as a practical solution for image classification [Krizhevsky et al. 2012]. The field of computer graphics is no exception, and neural networks have been successfully used for a multitude of applications. The early works trained multilayer perceptrons (MLPs) to estimate global illumination [Ren et al. 2015, 2013] and for Monte Carlo (MC) denoising [Kalantari et al. 2015]. Since then, efficient convolutional neural networks (CNNs) have been applied for Gaussian material synthesis [Zsolnai-Fehér et al. 2018], predicting parameters in participating media [Ge et al. 2018] and subsurface scattering models [Vicini et al. 2019], generating normalized distribution functions for glints [Kuznetsov et al. 2019], and robust MC denoising [Bako et al. 2017; Chaitanya et al. 2017; Vogels et al. 2018].

Moreover, networks have been applied to importance sampling light paths by either modeling light distributions to sample from during rendering [Müller et al. 2019; Zheng and Zwicker 2019] or by reconstructing the incident radiance throughout the scene with a pre-trained network [Bako et al. 2019]. Kalantari et al. [2016] introduced deep learning for natural image view synthesis which has been steadily improved upon [Mildenhall et al. 2019; Wang et al. 2017] and even applied to video lightfield reconstruction [Bemana et al. 2019]. Deep networks have also been applied as compressed representations by overfitting to specific assets that can be efficiently

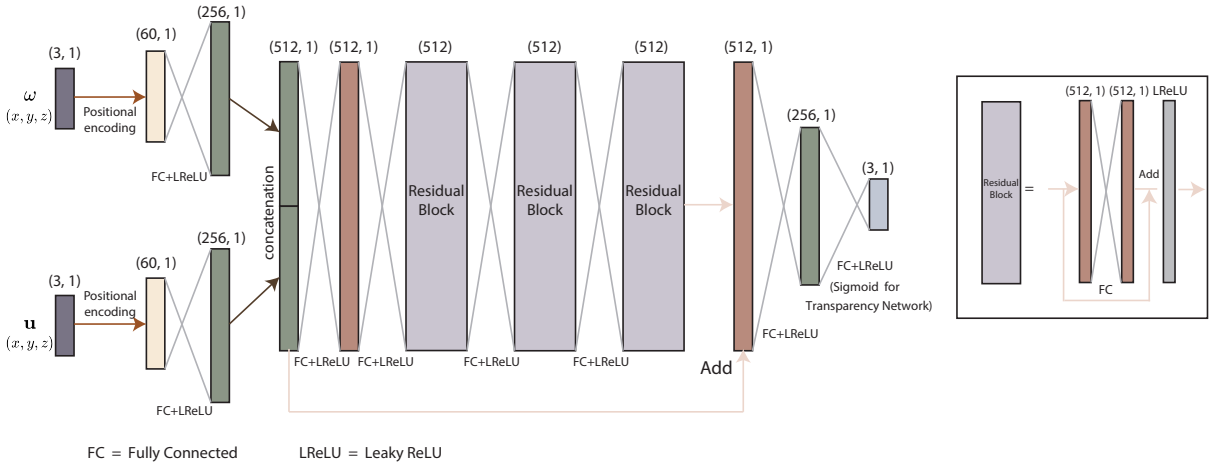

Fig. 2. The structure of our evaluation network with the residual blocks shown on the right. The transparency network uses the same architecture except for a sigmoid activation in the final layer. Note, we allow RGB output from the transparency network to handle transmission through colored glass.

stored and evaluated. For example, Davies et al. [2020] trained networks to overfit to the signed distance fields of specific models with unique shapes in order to accurately represent the data with relatively low storage costs. Our approach also applies learning, but it instead implicitly models and overfits to the lightfield of complex luminaires with a neural representation to estimate the incident radiance at a specific query point during rendering.

Neural rendering. Some learning-based methods either explicitly render the final image or estimate a component that is directly used in its computation. For example, networks were used to output the radiance for cloud rendering [Kallweit et al. 2017] and sky models [Satỳlmỳs et al. 2017] or for relighting images from sparse samples [Xu et al. 2018]. Rainer et al. [2019] use a compression network for a bidirectional texture function (BTF) to efficiently evaluate during rendering.

The work from Eslami et al. [2018] was the first to directly render a scene using a network. The network utilizes a small latent representation generated from a few views of the scene and a query view direction to generate the final image, without using any explicit geometry or textures. Follow-up work applied a similar framework for MC rendering and utilized additional auxiliary render information (e.g., shading normals and positions) to directly generate the output image [Granskog et al. 2020].

A recent approach learns the neural radiance fields [Mildenhall et al. 2020] of specific scenes to render novel views. Their network predicts the view dependent radiance and density of rays and combines them with a volume rendering model to generate the final image. We are inspired by the ability of this approach (and its many follow-ups) to efficiently represent complex scenes. We apply a related (though not volumetric) approach to representing complex luminaires, addressing the additional challenges of importance sampling and integration in full rendering systems.

3 NEURAL COMPLEX LUMINAIRES

3.1 Overview

Our goal is to evaluate complex luminaires quickly and accurately, which suggests a precomputed lightfield representation as a candidate solution. However, standard lightfield representations are substantially large and memory consuming, making them impractical for use in rendering systems. Fortunately, deep networks emerged as a powerful tool for approximating arbitrary functions; they are fast, can model complex relationships, and can interpolate and extrapolate input data well. Therefore, we use them to compress the precomputed lightfield data.

We consider a 4-dimensional lightfield representation $\mathcal{L}(\mathbf{u}, \boldsymbol{\omega})$, where $\mathbf{u}$ and $\boldsymbol{\omega}$ respectively represent positions and directions on a geometric proxy (scaffold) of the luminaire. The proxy bounds the luminaire and is typically simple (we use spheres and cylinders, but also a star-shaped polyhedron, in our results). We train a deep network to obtain a compact, compressed form of this lightfield.

To make this lightfield friendly to a path-tracing system with direct illumination and MIS, we need to address additional challenges: First, for evaluation, an efficient *point query* operation is needed to query the compressed representation given position $\mathbf{u}$ and direction $\boldsymbol{\omega}$ of a ray on the proxy. For importance sampling, given a shading point $\mathbf{s} \in \mathbb{R}^3$ anywhere in the scene, we need to sample a position $\mathbf{u}$ on the geometric proxy proportional to the slice of $\mathcal{L}$ as seen from $\mathbf{s}$. We should also be able to accurately estimate the probability density (pdf) of sampling a specific position on the geometric proxy by the above importance sampling method in order to compute the MIS weights correctly. Finally, we need to query luminaire's transparency for a given position $\mathbf{u}$ and direction $\boldsymbol{\omega}$. For all of these operations, full extraction of the compressed lightfield should be avoided, due to its high memory cost.

In what follows, we detail the design of our three neural networks that address the above challenges.

3.2 Evaluation Network

Network structure. The evaluation network takes in the queries $(\mathbf{u}, \boldsymbol{\omega})$ as inputs (where $\mathbf{u}$ is on the proxy and $\boldsymbol{\omega}$ points outwards) and outputs the lightfield query $\mathcal{L}(\mathbf{u}, \boldsymbol{\omega})$ as a 3-channel RGB value. Its structure is shown Fig. 2. It is a fully connected network with two notable features. We use fully connected residual blocks, which we found to give better accuracy than simple non-residual layers. We also use positional encoding [Mildenhall et al. 2020] to expand the input position $\mathbf{u}$ and the direction $\boldsymbol{\omega}$ from 3 to 60 dimensions each, to increase the network's ability to learn high-frequency functions.

Data preparation and training. To generate the synthetic data for training the evaluation network, a naïve approach would partition $\mathcal{L}$ into a 4D grid, then fill each grid cell by any appropriate Monte Carlo estimator. However, this is tedious due to the excessive computation and storage required.

Instead, we sample random viewing directions around the luminaire (using a low-discrepancy pseudo-random sequence), and render an image from each direction, using an orthographic camera, assuming black background (see Fig. 3). Each non-empty pixel of the image is converted into a training sample. Once we have enough images, we feed them as batches to the neural network training step (details in Sec. 4.1). We currently use a standard ℓ_2 loss on top of a $\log(x+1)$ transform to better handle a high dynamic range. We generate the image data on the CPU with the Mitsuba renderer [Jakob 2010], using the bidirectional integrator.

Furthermore, as noted by previous work [Lehtinen et al. 2018], the reference images do not have to be noise-free, because the network will learn to fit the underlying signal across examples and remove the noise. We observe that our compressed results are noise-free as shown in Fig. 4.

3.3 Importance Sampling Network

The importance sampling network enables support for direct illumination (next-event estimation), which is fundamental to practical rendering using path tracing. Specifically, at each shading point $\mathbf{s}$, we need to sample a direction $\mathbf{d}$ towards the luminaire; or equivalently, the corresponding visible point $\mathbf{u}$ on the geometric proxy. Ideally, the light sample should be drawn with probability density (pdf) proportional to the radiance $\mathcal{L}(\mathbf{u}, -\mathbf{d})$, or a close approximation of it. Furthermore, we need to be able to query the pdf $p(\mathbf{d})$ in solid angle measure, both to compute the corresponding sampling weight (the ratio $\mathcal{L}(\mathbf{u}, -\mathbf{d})/p(\mathbf{d})$), and for computing MIS weights.

Our solution is to train a network that will, for a given point $\mathbf{s}$, render a small color image of the luminaire as if viewed from a perspective camera at point $\mathbf{s}$, looking towards the center of the bounding sphere of the luminaire as shown in Fig. 5. The camera field of view is adjusted to perfectly fit the bounding sphere. The resulting small image can be easily importance sampled (assuming nearest-neighbor interpolation for its pixels), and the sample can be converted to the desired direction $\mathbf{d}$ and geometric proxy location $\mathbf{u}$. We found the resolution of 16×16 to work reasonably, though

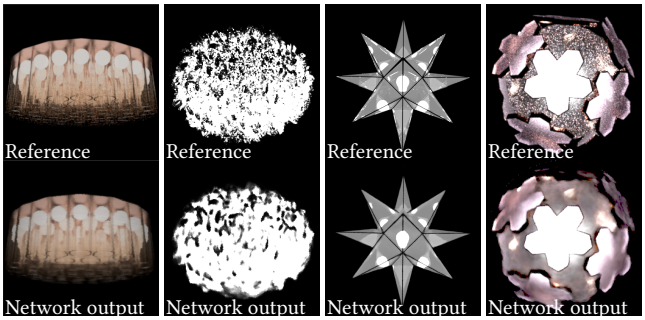

Fig. 4. Comparison between the reference training images (top) with the predicted lightfield renderings from our evaluation network (bottom). These images are rendered by orthogonal cameras as discussed in Fig. 3. The reference images are rendered at 128 spp using BDPT. Note how our results closely match the reference while reducing the noise.

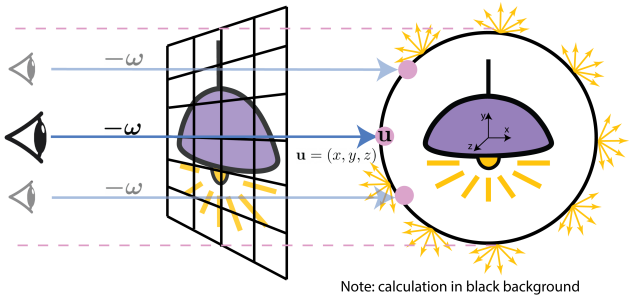

Fig. 3. Light evaluation configuration. The training data is generated through orthographic cameras from random directions that exactly cover the proxy. The camera rays in direction $-\boldsymbol{\omega}$ intersect with the proxy on various points $\mathbf{u}$, so each camera ray intersecting the proxy provides 3-channel RGB data as the output of the lightfield query $\mathcal{L}(\mathbf{u}, \boldsymbol{\omega})$. These training queries are shown as yellow arrows around the proxy of the luminaire.

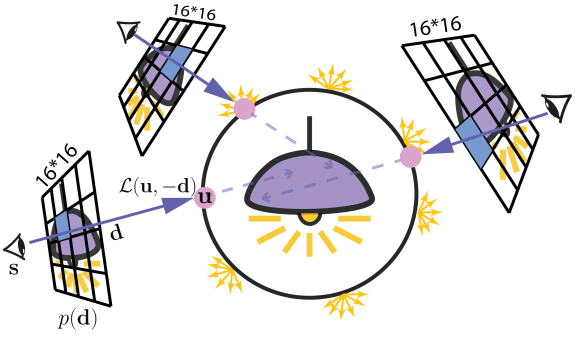

Fig. 5. To generate training data for the importance sampling network, we place a perspective camera at various positions around the proxy at different distances from its center. For each shading point $\mathbf{s}$, we render a 16 × 16 image, which the importance sampling network learns to estimate. To sample, we randomly choose a point on the rendered image (proportional to pixel values) to draw a direction $\mathbf{d}$ from the shading point. Next, we find a point $\mathbf{u}$ on the proxy and query the radiance $\mathcal{L}(\mathbf{u}, -\mathbf{d})$.

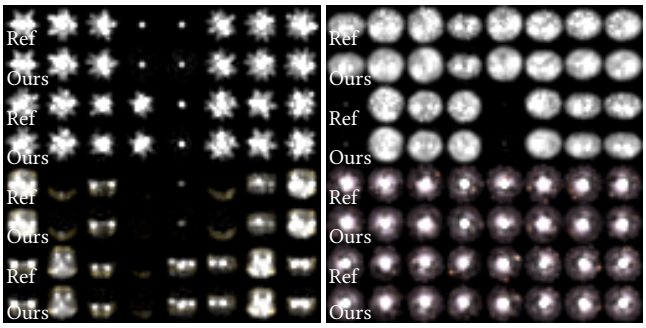

Fig. 6. Comparison of our importance sampling results with reference training data. The two images show comparisons between the 16×16 training data and the corresponding 16×16 predicted images from our importance sampling network.

we could also adapt the resolution to the specific luminaire. The resulting prediction quality is shown in Fig. 6.

Probabilities and weights. Furthermore, the pdf $p(\mathbf{d})$ can also be easily obtained from the small image and mapped into the solid angle measure. The sampling weight $\mathcal{L}(\mathbf{u}, -\mathbf{d})/p(\mathbf{d})$ can be computed directly from the image, giving a reasonable though biased estimate, because the value of $\mathcal{L}(\mathbf{u}, -\mathbf{d})$ comes from a low-resolution approximation. Alternatively, we could compute the radiance $\mathcal{L}(\mathbf{u}, -\mathbf{d})$ with a subsequent query to the evaluation network (Sec. 3.2), giving a less biased sampling of the compressed lightfield luminaire (though, of course, still subject to the approximations of the compression itself). We find that the simplest approach of computing the sampling weight from the low-res image already gives good results. The sampling weight computation is also the reason why we predict a color version of the small image; if we only used it to define the pdf, we could just learn the pixel values as scalars.

Note that with MIS, the pdf $p(\mathbf{d})$ is needed not only by the importance sampling process. When BSDF sampling results in a ray hitting the luminaire's proxy, the pdf is also required to compute

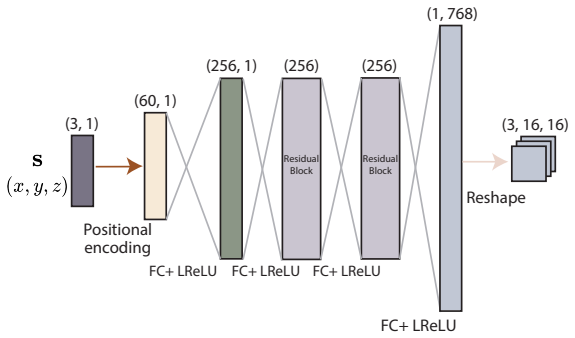

Fig. 7. The structure of our importance sampling network, which has a smaller size but larger output space relative to the evaluation network. We keep RGB information to be able to compute sampling weights (not just PDFs) directly from the network output.

the weights given by MIS heuristics. This indicates that we need to run the sampling network to compute $p(\mathbf{d})$ in this situation.

Network and training. Similar to the generation process of the data to train the evaluation network, we also dynamically generate the training data to feed the importance sampling network. We randomly select a shading point $\mathbf{s}$, then render an image of the complex luminaire using a perspective camera looking from $\mathbf{s}$ to the light. We also render the images directly.

Our importance sampling network is smaller than the evaluation network, as illustrated in Fig. 7. Another difference is the larger output space, which is reshaped into the desired 16×16 RGB image. We use an ℓ_2 loss as well as an overall mean loss.

3.4 Transparency Network

With the two neural networks introduced above, we are already able to render a scene lit by a complex luminaire within a path-tracing framework with MIS. However, there is one additional issue to be dealt with: compositing our luminaire into the scene.

A lightfield query that returns zero for a given pair $(\mathbf{u}, \boldsymbol{\omega})$ implies that there is no light emission from the luminaire along this ray, but that ray could still continue through the proxy back into the scene, and cannot be simply assumed to return zero radiance. The ray could pass the proxy and completely miss the luminaire, or the light path could pass through glass parts of the luminaire and exit back into the scene. These issues could theoretically be solved by keeping the luminaire geometry, but we find this solution undesirable due to the high storage cost and potential privacy concerns of releasing detailed luminaire geometry by manufacturers. Alternatively, we could consider compressing an 8D reflectance field in addition to the

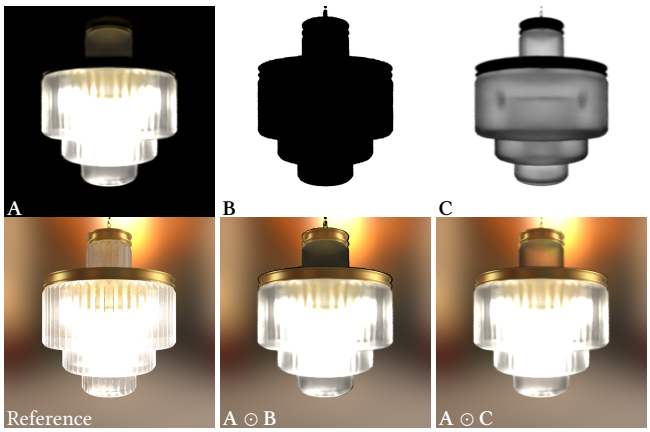

Fig. 8. Compositing a luminaire into the scene using our transparency network and optional real geometry components. **A**: The luminaire inside the proxy. **B**: A binary mask computed from the true geometry. **C**: Our RGB transparency mask (grayscale for this luminaire, but supports color for other luminaires). **A** ⊙ **B**: Result after applying the binary mask. **A** ⊙ **C**: Result after applying our RGB transparency network. Our network gives a softer, more pleasing result than the binary mask, while not requiring the full geometry. Furthermore, the bottom row shows our optional feature of keeping the reflective geometry of some opaque luminaire components (here made of brass), and rendering them in a traditional way.

emissive 4D lightfield, but the high dimensionality makes training networks for it much harder than for 4D lightfields.

Therefore, we use a small approximation: we compute RGB transparency of rays passing through the luminaire from different directions, calculating the transmission throughput of paths through the complex luminaire but ignoring the fact that refraction may have changed their direction. We allow for colored transparency to handle transmission through colored glass and similar materials. The result is another 4D function, which we compress using another neural network with the same structure as the evaluation network (Fig. 2), and trained using an analogous data generation approach (Sec. 3.2).

Fig. 8 shows a comparison between no transparency mask, binary transparency mask and our RGB transparency mask. Our approximate mask works well in practice, without requiring the original luminaire geometry. Nevertheless, we allow for optionally keeping *some* luminaire geometry, such as opaque reflective parts (metals or plastics). These parts are generally simple and easy to render using standard light transport, so optionally handling them in a classical way is reasonable. This is technique is used in the figure for the brass parts of the luminaire.

4 IMPLEMENTATION AND TRAINING DETAILS

4.1 Data Generation and Training

As our networks are trained with synthetic luminaires, the amount of data is theoretically unlimited, and we can choose how much to render. For the evaluation network, each image has a resolution of 256×256. We use 256 samples per pixel using the BDPT integrator on the CPU, which is much slower than the training process on the GPU. The transparency network is trained the same way, though the rendering spp can be lower (128). For the importance sampling network, the images are rendered with a perspective camera and have a resolution of 16×16, 4096 spp with BDPT.

We implemented our networks in the PyTorch framework. For each training step, we use 16 images as a batch. We apply the Adam solver with moment parameters $\beta_1 = 0.5$, $\beta_2 = 0.999$. The setting of the learning rate is 1×10^{-4}. To handle the high dynamic range, the ground truth is transformed to $\log(x+1)$ [Bako et al. 2017], and we further normalized the input to the range $[0, 1]$. We use ℓ_2 loss for all networks. For the sampling network, we additionally apply a loss on the mean of the sampling map to ensure the preservation of energy:

$$\mathcal{L}(y, \hat{y}) = \frac{1}{N}\sum_{i}^{N} \|\mathcal{T}(y_i) - \hat{y_i}\|_2 + \|\frac{1}{N}\sum_{i}^{N} y_i - \frac{1}{N}\sum_{i}^{N} \mathcal{T}^{-1}(\hat{y_i})\|_2$$

in which $\mathcal{T}(x) = \log(x+1)$, $\mathcal{T}^{-1}(x) = e^x - 1$, N is the total number of pixels in the sampling map, $\hat{y}$ is the predicted sample map, and y is the ground truth.

Generally, it is sufficient to use 200k unique images (256×256) for the evaluation and transparency networks, except for the White Feather Light, upon which we use 350k. We use 1 million images (16×16) for the importance sampling network. The training time is related to the complexity of the light. From our experience, the evaluation and transparency network will converge in 15-24 hours, while the sampling network takes about 12 hours using an NVIDIA RTX 3090 GPU.

4.2 Renderer Integration

At render time, at each shading point, we first perform light sampling so that both a sample on the light and the pdf for sampling anywhere on the light will be known. We convert the 16×16 images to grayscale, normalize them as a probability distribution and do importance sampling on the image; the original RGB image is then used to compute the RGB sampling weight. This is followed by BSDF sampling. If the indirect ray hits the complex luminaire, we query the eval network for the radiance and the transparency network to determine how much light goes through, otherwise we continue the standard process of tracing the ray for further bounces. We also look up the corresponding probability from the 16×16 grayscale image, which will be used to calculate balanced MIS weights for light and BRDF sampling.

4.3 GPU Acceleration

To efficiently evaluate our networks, we use NVIDIA TensorRT [NVIDIA 2021] to implement GPU batch inference inside the Mitsuba renderer. We revise the integrator and block rendering functionality in Mitsuba to make it work with TensorRT batch inference. During path tracing, we record in a buffer the positional and directional data whenever the emitter needs evaluation, importance sampling and transparency values. The corresponding throughput and BSDF values and then recorded in another buffer for final pixel color reconstruction. After path tracing, the buffers will be sent to GPU to perform inference. Each network inference will only be performed once for each network and block. Afterwards, the recorded throughput and BSDF values are used to calculate for the final color for each pixel. Note that paths can always be continued without waiting for the network evaluation since indirect rays are based on BSDF sampling and are independent of the luminaire.

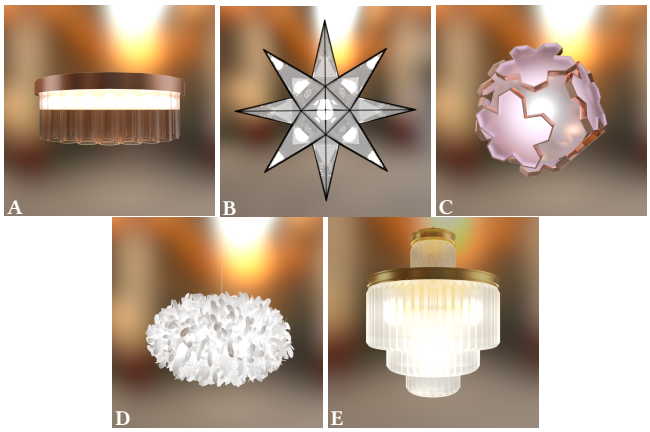

Fig. 9. The five complex luminaires used in the results. **A**: Statler Luminaire, **B**: Moravian Star Light, **C**: Crystal Cherry Blossom Chandelier, **D**: White Feather Light, **E**: Vintage Modern Crystal Chandelier.

Table 1. The performance details of five complex luminaires and their corresponding scenes. The reference results are rendered using bidirectional path tracing (BDPT) with 1024 samples per pixel, while our results are rendered with 128 spp to attain a similar noise level. Compared to the reference, our method achieves significant speedup. We also compare the storage of our method to the reference and the APL method by Velázquez et al. [2015] in the right three columns. Note that both the APL method and our method perform at a constant storage cost.

Luminaire	Scene	Ref. Samples per Pixel	Ref. Time (mins)	Our Samples per Pixel	Our Time (mins)	Speed Up	Ref Storage (MB)	APL Storage (MB)	Our Storage (MB)
Statler Luminaire	Elevator Room	1024	27.47	128	3.91	7.03×	21.5	384.0	17.0
Moravian Star Light	Loft Apartment	1024	42.38	128	3.49	12.14×	17.5	384.0	17.0
Crystal Cherry Blossom Chandelier	Bedroom	1024	83.55	128	3.85	21.70×	31.2	384.0	17.0
White Feather Light	Bedroom	1024	98.95	128	2.75	35.98×	219.0	384.0	17.0
Vintage Modern Crystal Chandelier	Livingroom	1024	65.41	128	4.31	15.18×	49.7	384.0	17.0

5 RESULTS, VALIDATION, AND COMPARISONS

In this section, we show the behavior of our approach on five different complex luminaires (see Fig. 9) embedded in four different scenes. We discuss the performance and storage of our method in comparison to previous work and analyze our advantages and limitations. We run on an Intel 20-core i9-10900K machine with an NVIDIA 3090 GPU for training and an NVIDIA 2080Ti GPU for inference. For data generation and comparisons, we use bidirectional path tracing (BDPT) as implemented in Mitsuba [Jakob 2010]. Additionally, we compare our method with the APL method of Velázquez et al. [2015] using code provided by the authors. The performance details are given in Table. 1. The storage of our method is constant for each luminaire, it costs 17 MB in ONNX format (evaluation and transparency network occupy 7.6 MB each, the importance network takes up 1.8 MB). Velázquez et al. [2015] method (APL) takes up 384 MB for each light. In this paper, the 3D luminaire models take up from 17.2 MB (Moravian Star Light) to 168 MB (White Feather Light). The size of pure 3D models still takes up more storage, despite containing only geometry and no useful light transport data. More results can be found in the supplementary video.

Statler Luminaire. This luminaire is a cylinder-shaped emitter with a metal base on top and 51 glass tubes at the bottom. The emitters are located at the top of these glass tubes, resulting in many specular paths. Note that this luminaire is similar, though not identical, to the one used in the previous work from Velázquez et al. Fig. 10 shows a comparison between BDPT and our method for the Statler Luminaire inside the *Elevator Room scene*. The rendering time of the reference image using BDPT at 1024 spp is 27.5 minutes, while our method takes only 3.9 minutes and achieves an equal noise level using far fewer samples per pixel (128 spp).

Moravian Star Light. This is a star-shaped luminaire with an emitter in the middle that is covered with rough glass. The proxy of this luminaire is its star-shaped polyhedral bounding volume, showing the ability to handle proxies beyond spheres and cylinders. Fig. 11 shows a comparison between BDPT using 1024 spp and our method inside the *Loft Apartment scene*. We achieve a 12× speedup

Fig. 10. Comparison with BDPT which takes significantly more samples to converge since many paths fail to connect with paths from the hidden emitters at the top of the glass tubes. Our approach can query the luminaire directly without having to evaluate costly paths resulting in relatively less noise with fewer samples and a 7× speed up in render time.

Fig. 11. The complex star shape encloses a central emitter, requiring all non-zero radiance paths to pass through rough glass. Our result converges 12× faster than BDPT since it avoids having to compute such paths at runtime.

Fig. 12. The glass petals of the luminaire obfuscate light paths and create high frequency shadows on the walls our method preserves. Moreover, we capture the direct appearance more accurately than the state-of-the-art method from Velázquez et al. and demonstrate a speedup of nearly 22× relative to BDPT.

relative to BDPT, while using only a fraction of the samples and generating significantly less noise. Moreover, the luminaire's direct appearance and illumination is correctly captured with our method.

Crystal Cherry Blossom Chandelier. This luminaire has a glass sphere shell with an emitter at the center and several flower-shaped glass "petals" on the surface around it. These petals are composed of rough glass surrounded by a metal contour that produce flower-shaped shadows on the surrounding environment (e.g., on the walls and ceilings). We use this luminaire to study the performance of all three networks on high-frequency data. Fig. 13 compares our method with BDPT and the state-of-the-art approach from Velázquez et al. when placing this luminaire inside a diffuse Cornell Box. We use far fewer samples to achieve a similar noise level, while still producing the correct flower-shaped patterns and high frequency highlights on the walls. On the other hand, the result from Velázquez et al. has artifacts in the shadows on the ceiling and fails to capture the directly visible appearance of the luminaire. Moreover, our method is still beneficial in highly specular scenes, as shown in Fig. 14, where we produce accurate highlights with relatively less noise using fewer samples. The difference images compared to the references (rendered using BDPT with 1024 spp) are on the bottom right of Fig. 13 and Fig. 14.

We also placed the luminaire into the more complex *Bedroom scene* for further comparisons (see Fig. 12). The rendering time of BDPT with 1024 spp is 83.6 minutes, while our method needs only 128 spp and 3.9 min total to achieve similar quality. We can see clearly from Fig. 12 that the flower-shaped shadow has been clearly preserved in both Velázquez et al. and our results, yet our method better captures the direct appearance and renders more than an

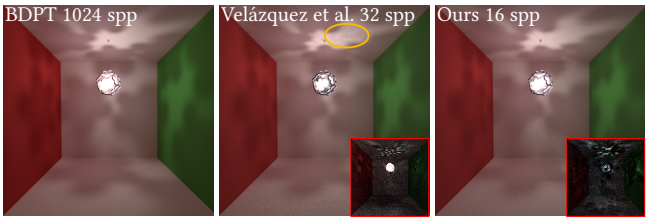

Fig. 13. We compare favorably against the state-of-the-art approach from Velázquez et al. Our approach captures the high frequency shadows on the ceiling without artifacts as well as the luminaire's direct appearance. The difference (16×) to the reference is shown on the bottom right of each image.

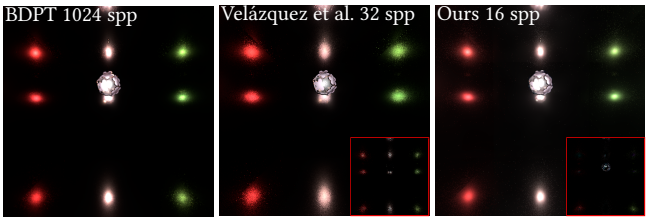

Fig. 14. Our approach works well even in a highly specular setting, generating images with relatively less noise using fewer samples. The difference (16×) to the reference is shown on the bottom right of each image.

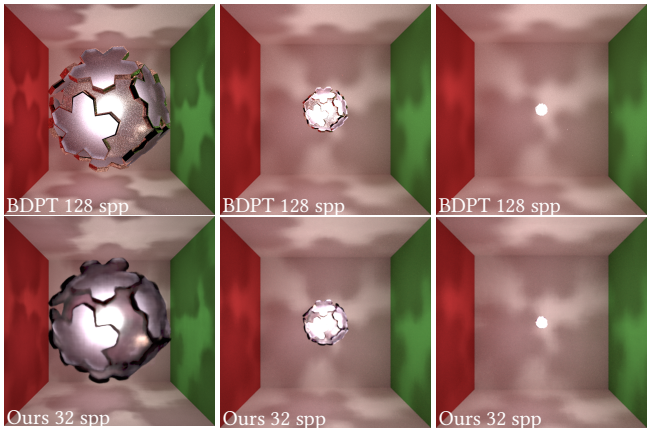

Fig. 15. We scale the Crystal Cherry Blossom Chandelier inside a diffuse Cornell box. Our method consistently produces similar patterns on the walls compared to the references, regardless of the distances between the luminaire and the shading points. Note that in the right column, the light is small enough, thus the rendering can be considered far-field.

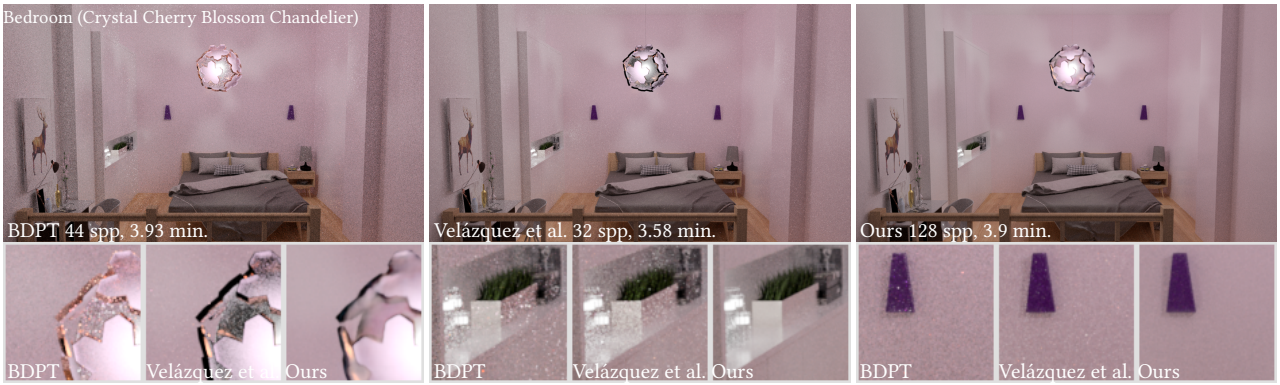

Fig. 16. Equal-time comparisons between BDPT, Velázquez et al., and our method. The fast performance of our method leads to the highest sample count and the best rendering quality, preserving the luminaire's appearance and the high frequency shadows on the walls.

order of magnitude faster. Overall, the reference and our result have similar noise levels, yet we generated our image with far fewer samples per pixel, required less time and storage, and could use a simpler light transport algorithm (standard path tracing).

To validate the reliability of our method when the luminaire has various distances to the illuminated surfaces, we place the Crystal Cherry Blossom Chandelier inside the diffuse Cornell box and scale the luminaire up and down in size. As shown in the first column of Fig. 15, when the surfaces are close to the luminaire (near-field), the petal-shaped pattern of our result is in accordance with the reference rendered with BDPT. Similarly, we investigate the accuracy of our method in the far-field. As shown in the third column of Fig. 15, when the size of the luminaire shrinks, it also moves away from the surfaces, and the effect becomes far-field. However, the pattern on the walls and the intensity of the luminaire still match the reference rendered with BDPT.

Fig. 17. Non-transparent feathers on a glass ellipsoid shell cover two emitters resulting in extremely complex light transport. BDPT still contains residual noise relative to our approach despite its 36× longer render time.

Additionally, we perform equal-time comparisons between BDPT, Velázquez et al., and our method. As shown in Fig. 16, our method attains the best quality and the lowest noise level.

White Feather Light. This luminaire contains two light bulbs at the center that are covered with a glass ellipsoid shell. On the external surface, there are many non-transparent feather-shaped geometries. Due to its higher geometric complexity, it causes complex light transport and is capable of creating intricate shadows on the surrounding environment, making it particularly difficult for previous methods to obtain a noise-free result. In Fig. 17, we demonstrate a noise-free result with a 36× speedup relative to BDPT, which after 1024 spp still has not completely converged.

Vintage Modern Crystal Chandelier. This luminaire is a chandelier that contains many rough glass components. The specular light paths inside are intricate, which makes it difficult to render using traditional approaches. We use the *Living Room scene* to compare the overall performance with BDPT and Velázquez et al. in Fig. 18, which take 65.4 minutes and 61.8 minutes, respectively. Our method attains a similar noise level as the reference in only 4.3 minutes and with substantially fewer samples. Furthermore, when compared to BDPT, we can clearly see that the scene has been properly illuminated and the luminaire's appearance is correctly preserved.

As with the *Crystal Cherry Blossom Chandelier*, Velázquez et al. is not able to capture the transmission of light back into the scene, which occurs along the glass in the middle region (see central crop). Another difference in the direct appearance occurs along the gold band at the top of the luminaire. Our result applies the optional feature of including the opaque luminaire component geometry (here made of brass), and thus it handles such reflections with more accuracy. Note, including this geometry is also possible in Velázquez et al. (with some modification), but this would not affect the aforementioned issue with transmission, as that is a fundamental limitation of their approach.

Limitations. The limitations of our method include some amount of blurring in the evaluation and further bias in importance sampling, due to using small predicted images to derive the sampling weight. The higher the resolution of these images, the less bias is present,

Fig. 18. Many rough glass components surrounding the internal emitters require intricate light paths resulting in 15× longer render times for convergence with BDPT. Meanwhile, Velázquez et al. have significantly longer render times, more noise, and more severe artifacts in the direct appearance compared to our approach. Note, our result includes the opaque luminaire component (here made of brass), which is rendered normally. Velázquez et al. could use the same strategy for the gold band, but the transmission along the glass in the middle region of the luminaire is missing due to fundamental limitations of the method.

but we found our current framework was practical and sufficient for the results shown here. Moreover, these issues are common to most neural rendering approaches, and it is unlikely that an unbiased approach can be constructed in any way other than with the full light transport resulting from the original luminaire geometry.

Another limitation is due to the approximations made in compositing. The dark region on the top band of the luminaire in Fig. 10 is because we model a static luminaire that does not capture the external light bounces reflected from the scene. This can be addressed by modeling a full 8D reflectance field or, alternatively, by keeping the metallic components as real geometry. We opt for the latter, as we also demonstrate in Fig. 18, since such regions are relatively cheap to compute with standard renderers.

Finally, creating our representation requires relatively expensive synthetic training data creation through bidirectional path tracing. It would be interesting to explore whether a similar quality can be achieved from fewer generated images, perhaps by introducing additional priors on the resulting lightfields.

6 CONCLUSIONS AND FUTURE WORK

We have introduced a compact representation of complex luminaires suitable for integration into practical path-tracing renderers with multiple importance sampling. The representation is based on a simplified geometric proxy and three deep neural networks, handling the compression and evaluation of the luminaire's lightfield, its importance sampling, and the transparency of the luminaire when compositing it into a scene. The total size of the neural network weights is small, resulting in a compact black-box representation independent of the original luminaire's geometry and materials. The details of the luminaires are well preserved and the rendering performance is significantly improved compared to a general light transport simulation. Future work can focus on further increasing the apparent spatial and angular resolution of the compressed results, more accurate compositing of the luminaire into the scene (perhaps by considering a full 8D reflectance field), and support for editing the luminaire parameters.

ACKNOWLEDGMENTS

This work has been partially supported by the National Key R&D Program of China under grant No.2020YFB1709200, the National Natural Science Foundation of China under grant No.61872223, the Shandong Provincial Natural Science Foundation of China under grant No.ZR2020LZH016. P. Sen was partially supported by NSF grants #16-19376 and #19-11230.

REFERENCES

Ian Ashdown. 1995. Near-Field Photometry: Measuring and Modeling Complex 3-D Light Sources. In *ACM SIGGRAPH Course Notes*. 1–15.

Ian Ashdown and Ron Rykowski. 1998. Making Near-Field Photometry Practical. *Journal of the Illuminating Engineering Society of North America* 27, 1 (1998), 67–79. https://doi.org/10.1080/00994480.1998.10748212

Steve Bako, Mark Meyer, Tony DeRose, and Pradeep Sen. 2019. Offline Deep Importance Sampling for Monte Carlo Path Tracing. *Computer Graphics Forum* (2019).

Steve Bako, Thijs Vogels, Brian McWilliams, Mark Meyer, Jan Novák, Alex Harvill, Pradeep Sen, Tony DeRose, and Fabrice Rousselle. 2017. Kernel-Predicting Convolutional Networks for Denoising Monte Carlo Renderings. *ACM Transactions on Graphics* 36, 4 (July 2017).

Mojtaba Bemana, Karol Myszkowski, Hans-Peter Seidel, and Tobias Ritschel. 2019. Neural View-Interpolation for Sparse LightField Video. *arXiv preprint arXiv:1910.13921* (2019).

Chakravarty R. A. Chaitanya, Anton Kaplanyan, Christoph Schied, Marco Salvi, Aaron Lefohn, Derek Nowrouzezahrai, and Timo Aila. 2017. Interactive Reconstruction of Noisy Monte Carlo Image Sequences using a Recurrent Autoencoder. *ACM Transactions on Graphics* (July 2017).

Chuo-Ling Chang, Xiaoging Zhu, Prashant Ramanathan, and Bernd Girod. 2003. Inter-View Wavelet Compression of Light Fields with Disparity-Compensated Lifting. In *Proc. of VCIP*. SPIE, 694–706.

Chuo-Ling Chang, Xiaoqing Zhu, Prashant Ramanathan, and Bernd Girod. 2006. Light Field Compression Using Disparity-Compensated Lifting and Shape Adaptation. *IEEE Transactions on Image Processing* 15, 4 (2006), 793–806.

Wei-Chao Chen, Jean-Yves Bouguet, Michael H. Chu, and Radek Grzeszczuk. 2002. Light Field Mapping: Efficient Representation and Hardware Rendering of Surface Light Fields. *ACM Trans. Graph.* 21, 3 (2002), 447–456.

Thomas Davies, Derek Nowrouzezahrai, and Alec Jacobson. 2020. Overfit Neural Networks as a Compact Shape Representation. arXiv:2009.09808 [cs.GR]

SM Ali Eslami, Danilo Jimenez Rezende, Frederic Besse, Fabio Viola, Ari S Morcos, Marta Garnelo, Avraham Ruderman, Andrei A Rusu, Ivo Danihelka, Karol Gregor, et al. 2018. Neural scene representation and rendering. *Science* 360, 6394 (2018), 1204–1210.

Liangsheng Ge, Beibei Wang, Lu Wang, and Nicolas Holzschuch. 2018. A Compact Representation for Multiple Scattering in Participating Media using Neural Networks. In *ACM SIGGRAPH 2018 Talks*. Vancouver, Canada, 1–2.

Iliyan Georgiev, Jaroslav Křivánek, Tomáš Davidovič, and Philipp Slusallek. 2012. Light transport simulation with vertex connection and merging. *ACM Trans. Graph.* 31, 6

(2012), 192:1–192:10.
Steven J. Gortler, Radek Grzeszczuk, Richard Szeliski, and Michael F. Cohen. 1996. The Lumigraph. In *Proc. of SIGGRAPH*. 43–54.
Jonathan Granskog, Fabrice Rousselle, Marios Papas, and Jan Novák. 2020. Compositional Neural Scene Representations for Shading Inference. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)* 39, 4 (July 2020).
Toshiya Hachisuka, Shinji Ogaki, and Henrik Wann Jensen. 2008. Progressive Photon Mapping. *ACM Trans. Graph.* 27, 5 (2008), 130:1–130:8.
Toshiya Hachisuka, Jacopo Pantaleoni, and Henrik Wann Jensen. 2012. A path space extension for robust light transport simulation. *ACM Trans. Graph.* 31, 6 (2012), 191:1–191:10.
Wolfgang Heidrich, Jan Kautz, Philipp Slusallek, and Hans-Peter Seidel. 1998. Canned Lightsources. In *Rendering techniques' 98*. 293–300.
Wenzel Jakob. 2010. Mitsuba Physically Based Renderer. http://www.mitsuba-renderer.org
Wenzel Jakob and Steve Marschner. 2012. Manifold Exploration: A Markov Chain Monte Carlo Technique for Rendering Scenes with Difficult Specular Transport. *ACM Transactions on Graphics (TOG)* 31, 4, Article 58 (2012), 58:1–58:13 pages.
James T. Kajiya. 1986. The Rendering Equation. *SIGGRAPH Comput. Graph.* 20, 4 (Aug. 1986), 143–150.
Nima Khademi Kalantari, Steve Bako, and Pradeep Sen. 2015. A Machine Learning Approach for Filtering Monte Carlo Noise. *ACM Transactions on Graphics* 34, 4, Article 122 (July 2015), 12 pages.
Nima Khademi Kalantari, Ting-Chun Wang, and Ravi Ramamoorthi. 2016. Learning-based View Synthesis for Light Field Cameras. *ACM Transactions on Graphics* 35, 6, Article 193 (Nov. 2016), 10 pages.
Simon Kallweit, Thomas Müller, Brian McWilliams, Markus Gross, and Jan Novák. 2017. Deep Scattering: Rendering Atmospheric Clouds with Radiance-Predicting Neural Networks. *ACM Transactions on Graphics* 36, 6, Article 231 (Nov. 2017), 11 pages.
S. Kniep, S. Häring, and M. Magnor. 2009. Efficient and Accurate Rendering of Complex Light Sources. *Computer Graphics Forum* 28, 4 (June 2009), 1073–1081.
Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*. 1097–1105.
Alexandr Kuznetsov, Miloš Hašan, Zexiang Xu, Ling-Qi Yan, Bruce Walter, Nima Khademi Kalantari, Steve Marschner, and Ravi Ramamoorthi. 2019. Learning Generative Models for Rendering Specular Microgeometry. *ACM Transactions on Graphics* 38, 6, Article 225 (Nov. 2019), 14 pages.
Jaakko Lehtinen, Jacob Munkberg, Jon Hasselgren, Samuli Laine, Tero Karras, Miika Aittala, and Timo Aila. 2018. Noise2Noise: Learning Image Restoration without Clean Data. In *Proceedings of the 35th International Conference on Machine Learning*, Vol. 80.
Marc Levoy and Pat Hanrahan. 1996. Light Field Rendering. *Proc. of SIGGRAPH* (1996).
Heqi Lu, Romain Pacanowski, and Xavier Granier. 2015. Position-Dependent Importance Sampling of Light Field Luminaires. *IEEE Transactions on Visualization and Computer Graphics* 21, 2 (feb 2015), 241–251.
Albert Mas, Ignacio Martín, and Gustavo Patow. 2008. Compression and Importance Sampling of Near-Field Light Sources. *Computer Graphics Forum* 27, 8 (Dec. 2008), 2013–2027. https://doi.org/10.1111/j.1467-8659.2008.01180.x
Ben Mildenhall, Pratul P. Srinivasan, Rodrigo Ortiz-Cayon, Nima Khademi Kalantari, Ravi Ramamoorthi, Ren Ng, and Abhishek Kar. 2019. Local Light Field Fusion: Practical View Synthesis with Prescriptive Sampling Guidelines. *ACM Transactions on Graphics* (2019).
Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. 2020. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. In *ECCV 2020*.
Thomas Müller, Brian McWilliams, Fabrice Rousselle, Markus Gross, and Jan Novák. 2019. Neural importance sampling. *ACM Transactions on Graphics* 38, 5 (2019).
Julius Muschaweck. 2011. What's in a ray set: moving towards a unified ray set format, In Illumination Optics II. *Proc. SPIE* 8170. https://doi.org/10.1117/12.896757
P.Y. Ngai. 1987. On near-field photometry. *Journal of the Illuminating Engineering Society* 16, 2 (1987), 129–136.
NVIDIA. 2021. NVIDIA TensorRT. https://developer.nvidia.com/tensorrt
Gilles Rainer, Wenzel Jakob, Abhijeet Ghosh, and Tim Weyrich. 2019. Neural BTF Compression and Interpolation. In *Computer Graphics Forum*, Vol. 38. 235–244.
Peiran Ren, Yue Dong, Stephen Lin, Xin Tong, and Baining Guo. 2015. Image based relighting using neural networks. *ACM Transactions on Graphics* 34, 4 (2015), 111.
Peiran Ren, Jinpeng Wang, Minmin Gong, Stephen Lin, Xin Tong, and Baining Guo. 2013. Global Illumination with Radiance Regression Functions. *ACM Transactions on Graphics* 32 (July 2013).
Todd Saemish, P. Ericson, G. Hauser, E. Gibson, R. Heinisch, C. Loch, and IESNA. 2002. ANSI/IESNA Standard File Format for the Electronic Transfer of Photometric Data and Related Information.
Pỳnar Satỳlmỳs, Thomas Bashford-Rogers, Alan Chalmers, and Kurt Debattista. 2017. A machine-learning-driven sky model. *IEEE computer graphics and applications* 37, 1 (2017), 80–91.
Eric Veach. 1997. *Robust Monte Carlo Methods for Light Transport Simulation*. Ph.D. Dissertation. Stanford University.
Eric Veach and Leonidas J. Guibas. 1997. Metropolis Light Transport. In *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '97)*. ACM Press/Addison-Wesley Publishing Co., USA, 65–76. https://doi.org/10.1145/258734.258775
Edgar Velázquez-Armendáriz, Zhao Dong, Bruce Walter, and Donald P. Greenberg. 2015. Complex Luminaires: Illumination and Appearance Rendering. *ACM Transactions on Graphics* 34, 3 (May 2015), 26:1–26:15.
Delio Vicini, Vladlen Koltun, and Wenzel Jakob. 2019. A Learned Shape-Adaptive Subsurface Scattering Model. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)* 38, 4 (July 2019), 126:1–126:15.
Thijs Vogels, Fabrice Rousselle, Brian McWilliams, Gerhard Röthlin, Alex Harvill, David Adler, Mark Meyer, and Jan Novák. 2018. Denoising with Kernel Prediction and Asymmetric Loss Functions. *ACM Transactions on Graphics* 37, 4, Article 124 (2018), 124:1–124:15 pages.
Ting-Chun Wang, Jun-Yan Zhu, Nima Khademi Kalantari, Alexei A. Efros, and Ravi Ramamoorthi. 2017. Light Field Video Capture Using a Learning-Based Hybrid Imaging System. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)* 36, 4 (2017).
Zexiang Xu, Kalyan Sunkavalli, Sunil Hadap, and Ravi Ramamoorthi. 2018. Deep image-based relighting from optimal sparse samples. *ACM Transactions on Graphics* 37, 4 (2018), 126.
Quan Zheng and Matthias Zwicker. 2019. Learning to importance sample in primary sample space. In *Computer Graphics Forum*, Vol. 38. Wiley Online Library, 169–179.
Károly Zsolnai-Fehér, Peter Wonka, and Michael Wimmer. 2018. Gaussian material synthesis. *ACM Transactions on Graphics* 37, 4 (2018), 76:1–76:14.