

A Learned Radiance-Field Representation for Complex Luminaires

Jorge Condor^{1,2}  and Adrian Jarabo¹ 

¹Universidad de Zaragoza - I3A, Spain

²Università della Svizzera Italiana, Switzerland



Figure 1: A living room scene containing a learned radiance field-based complex luminaire. From left to right: Reference path-traced scene with explicitly-modeled complex luminaire (32768 samples per pixel, 7.2h); the same scene rendered with 64 samples per pixel in 1.3 minutes; and our method with 32 samples per pixel, 52.1 seconds. We leverage learned volumetric radiance fields to obtain high quality representations of complex luminaires.

Abstract

We propose an efficient method for rendering complex luminaires using a high quality octree-based representation of the luminaire emission. Complex luminaires are a particularly challenging problem in rendering, due to their caustic light paths inside the luminaire. We reduce the geometric complexity of luminaires by using a simple proxy geometry, and encode the visually-complex emitted light field by using a neural radiance field. We tackle the multiple challenges of using NeRFs for representing luminaires, including their high dynamic range, high-frequency content and null-emission areas, by proposing a specialized loss function. For rendering, we distill our luminaires' NeRF into a plenoctree, which we can be easily integrated into traditional rendering systems. Our approach allows for speed-ups of up to 2 orders of magnitude in scenes containing complex luminaires introducing minimal error.

CCS Concepts

- **Computer graphics** → Neural Rendering;
- **Machine Learning** → Neural Radiance Fields;

1. Introduction

Complex luminaires are ubiquitous in real-world scenes, from isolated light bulbs with tiny coils embedded in dielectric bulbs, to complex chandeliers made by thousands of small glass pieces that scatter the light of the (potentially many) emitters enclosed. While a common approach is to reduce the complexity of light sources to simple distant or area lights that can be compactly evaluated and sampled [BNJ15; GUK*17; Pet21], in many scenarios including architectural visualization such simplifications result into a significant loss in realism. However, as the light source increases in complexity these approaches become unfeasible: the complex light paths connecting the potentially multiple individual light sources

in the luminaire, and the surfaces being illuminated by simply random chance is unlikely in the best of cases. Thus, computing the illumination from a complex luminaire would require many thousands of samples to converge, even using complex Markov-Chain Monte Carlo methods [JM12].

Precomputing the radiance field of the luminaires is an effective approach for avoiding the sampling of difficult paths in run time [VDWG15; ZBX*21]. It allows to encode all light paths exiting the luminaire as a five-dimensional proxy function, which is fast to access during rendering: to determine the illumination at a point, it is only needed to integrate over the proxy function. Unfortunately, accurately precomputing a luminaire requires a very

dense storage of the five-dimensional function modeling the emission of luminaires, which might become unfeasible due to memory constraints, or to combine the precomputed proxy with the real complex geometry of the scene.

In this work we tackle this problem by leveraging the potential of recent advances on volumetric representations of appearance, and in particular on the compact and efficient representation of radiance fields (RFs) [MST*20; TTM*21]. Radiance Fields, including *neural radiance fields* (NeRF), have emerged as a suitable representation of very complex three-dimensional scenes by encoding the scene as a lightweight volumetric function. The key characteristic of this volumetric function, in contrast to previous voxelization-based representations of appearance [Ney98; LN17], is that this function is learned from a (relatively) sparse set of views. RFs allow to compactly encode highly-complex spatio-directional representations of 3D scenes, which can be efficiently rendered by standard ray-marching through the volumetric proxy.

Unfortunately, off-the-shelf RFs are not adequate for our particular task, and modeling a luminaire using a RF, and integrating it into a Monte Carlo rendering engine, introduces several challenges, including: 1) Luminaires have high dynamic range (HDR), as opposed to the usual low dynamic range (LDR) targeted by NeRFs, and 2) exhibit high angular frequency; and 3) they require handling both opaque and transparent elements for integration in synthetic scenes. We solve these challenges using a training loss designed for handling HDR content, and a combination of a linear volumetric representation of the RF inspired in correlated media [VJK21; JAG18].

We integrate our RF-based luminaire in a physically-based rendering using an efficient octree-based representation of the radiance field, demonstrating speed-ups of up to two orders of magnitude, compared with traditional path tracing and volumetric path tracing methods with little error.

2. Related Work

Volumetric representations of appearance Volumetric approaches have been successfully used to approximate the appearance of complex geometries in a wide array of different applications, from trees [LN17; Ney98], cloth and hair [ACG*17; KSZ*16; SKZ11; ZJMB11], or particulate media (sugar, salt, etc) [MPH*15; MWM07; MPG*16]. These works rely on computing (potentially heterogeneous) bulk scattering parameters by using anisotropic radiative transfer [JAM*10; HCD15]. Zhao et al. [ZWDR16] proposed a method for downsampling this type of volumetric appearances by optimizing their directional scattering. Recently, Vicini et al. [VJK21] improved the accuracy of volumetric representations of surface-based scenes by using a non-exponential correlated volumetric representation [JAG18; BRM*18], allowing to represent a larger range of media (correlated or otherwise) with a higher level of detail. All these works assume scattering media with no emission. Our work builds upon these ideas, using a volumetric representation for approximating the appearance of complex luminaries.

Light and Radiance Fields Light fields [WMJ*17] are vector functions that encode the radiance carried by a particular ray de-

fined in a four-dimensional space, and are a compact form of rendering complex scenes based on images. Recently, with the widespread of deep learning methods, a number of works have been proposed to encode these light fields in compact neural-based representations [LSS*19; SZW19; MSO*19].

Radiance fields (RF) [MST*20] encode the ray-to-radiance mapping by using a volumetric approach, in which each point of the volume space encodes its directional radiance. In rendering time, a volume rendering-based approach is used, by ray marching the volumetric representation of the scene and gathering the radiance at each point in the ray. This directional radiance is learned by using a set of images of the scene. In order to efficiently encode the RF, Mildenhall et al. [MST*20] proposed to use a neural network (a neural radiance field, or NeRF). Since its inception, NeRFs have received an outstanding attention; we refer to recent surveys on neural rendering for an overview of methods building upon and generalizing NeRF [TTM*21]. Unfortunately, NeRFs require evaluating a neural network for each ray marching step, which might be expensive. Yu et al. [YLT*21] proposed to distill the NeRF into a primal-space octree encoding the directional radiance in spherical harmonics. This allowed them to prune non-contributing empty regions, and perform very fast rendering of surface-based radiance fields. We extend this approach to encode the radiance field of complex luminaires.

Complex Luminaries The most common approach for efficiently rendering complex luminaries is to approximate the luminaire by a simpler proxy geometry, and encode its complexity by precomputing the near-field outgoing emission from the proxy. Early works [Ash95; Nga87] tabulated the complex five-dimensional emission function. Heidrich et al. [HKSS98] proposed to encode the emission using light fields, following an approach similar to image-based rendering techniques Rayset-based approaches [AR98; MMP08; Mus11] follow a similar idea, but removing the structure of the outgoing emission, by capturing the spatio-directional distribution of radiance at the proxy. Lu et al. [LPG14] proposed a method for importance sampling light field-based representations of luminaires. All these works focus on modeling the emission of light sources in the scenes, but cannot represent their appearance unless a very high resolution is used to represent the emission.

More recently, Velázquez-Armendáriz et al. [VDWG15] proposed a hybrid representation of the luminaire, representing its illumination as a set of directional point light sources which can be used for lighting the scene, while its appearance is rendered by using the full geometry of the luminaire and a precomputed volume emission. In contrast, our approach seamlessly models both illumination and appearance.

Finally, the closest method to ours was proposed by Zhu et al. [ZBX*21], that leverage neural rendering techniques for encoding the appearance and illumination from complex luminaires. They use two multilayer perceptrons (MLPs) to learn the full light field of the luminaire approximated as a simple proxy, as well as its transparency for integration in the synthetic scene. While the results are of high-quality, they require several thousands of input images for training, the results are slightly oversmoothed, and the evaluation is expensive, requiring a dedicated GPU for evaluating

the MLPs. Our work is similar in spirit, but instead leverages recent learned volumetric radiance fields for encoding the luminaire, resulting in fast and high-quality rendering of luminaires. It implicitly encodes transparency for matting, requires less training data, and does not need to evaluate a neural network in rendering time.

3. Problem Statement

Our goal is to evaluate a complex luminaire as efficiently as possible within any given scene, so that it can be used for both modeling its appearance (e.g. as seen from the camera) and its illumination, making it easy to integrate in any Monte Carlo-based rendering engine.

We start by following previous works on luminaire modeling [HKSS98; ZBX*21]: These works partially (or totally) remove the geometric complexity and precompute the light transport within the luminaire.

We first make the assumption that it will always be rendered from the outside; this is a reasonable assumption for most applications, even for close-up shots, as long as we don't place the camera inside the actual luminaire (i.e. inside the dielectric bulbs). This allows us to simplify the geometry of the luminaire as a simple proxy geometry bounding the luminaire. Then, the appearance of the luminaire can be encoded in a per-ray basis as a 5D function L_o modeling the outgoing radiance exiting the proxy at position \mathbf{x}_l in direction ω_o as

$$L_o : \mathbb{R}^3 \times \mathbb{R}^2 \rightarrow \mathbb{R}; \quad (\mathbf{x}_l, \omega_o) \mapsto L_o(\mathbf{x}_l, \omega_o). \quad (1)$$

In order to model $L_o(\mathbf{x}_l, \omega_o)$ we need to account for two terms: The emitted field of the luminaire $L_e(\mathbf{x}_l, \omega_o)$, that encodes all light paths starting at the luminaire and outgoing in (\mathbf{x}_l, ω_o) , and the light from the scene that might pass through the luminaire (due to empty space within the proxy), which is crucial for integrating the luminaire in the scene. Therefore, we model $L_o(\mathbf{x}_l, \omega_o)$ as

$$L_o(\mathbf{x}_l, \omega_o) = L_e(\mathbf{x}_l, \omega_o) + \alpha(\mathbf{x}_l, \omega_o) L_i(\mathbf{x}_l - \omega_o \cdot t_1, \omega_o), \quad (2)$$

with $\alpha(\mathbf{x}_l, \omega_o)$ the transparency along a ray with origin at \mathbf{x}_l and direction $-\omega_o$. $L_i(\mathbf{x}, \omega)$ is the incoming radiance at the proxy in point \mathbf{x} from direction ω , and t_1 is the intersection distance of the ray with origin at \mathbf{x}_l and direction $-\omega_o$ with the proxy geometry. This approach is illustrated in Figure 2 (1).

Thus, the key problem is how to model the emission and transparency functions L_e and α respectively, such that a) they can be efficiently evaluated in rendering time, b) they are compact from a storage perspective, c) preserve their spatio-directional high frequency, and d) can deal with the potentially large dynamic range of the luminaires. Zhu et al. [ZBX*21] proposed to use two specialized image-based neural rendering systems for each of these terms, at the cost of requiring a large training set.

However, we observe that modern volumetric-based *radiance fields* (RF) excel at learning such 5D mapping (ray to radiance), even in very complex scenes with a relatively small training set. Moreover, since RF are based on volume rendering, they implicitly encode the transparency term $\alpha(\mathbf{x}_l, \omega_o)$. Therefore it is natural to pose our problem by using a neural representation of the luminaire radiance field.

Challenges Unfortunately, directly applying a state-of-the-art RF to model synthetic complex luminaires for rendering does not provide good results, and there are several challenges that need to be addressed:

- **High Dynamic Range.** Learned radiance fields have mainly focused on capturing LDR scenes. However, in a physically-based rendering engine, we need to account for high-dynamic-range radiance. Extending these methods to work on HDR is challenging, since the areas with the highest radiance will dominate the gradient and thus the training.
- **Null Emission.** Since we are modeling isolated objects that will be integrated into synthetic scenes, we need to account for zero-radiance light rays. In contrast, in natural scenes these absolute-black areas are uncommon. In a general machine learning context, dark pixels generate very small gradients during training, which makes them a difficult training target as the error is dominated by the more radiant parts.
- **Matting.** In order to blend our luminaire with the rest of the scene, we will need to let through the proxy those rays that actually hit the convex geometry but not the underlying light source. This requires us to incorporate not only the emission from a ray, but also the amount of light that passes through the proxy of the luminaire (i.e. the *transparency*). A highly accurate transparency (or opacity) model is necessary for a good integration of the luminaire within the scene, which sometimes NeRF struggles with.

We address these issues by introducing a loss function targeting the particularities of luminaires' radiance fields. In the following, we detail our model.

4. Our Model

As discussed earlier, our approach leverages recent works on learned volumetric radiance fields for modeling the appearance and illumination of luminaires. These are learned from a sparse set of HDR training views around the luminaire. These radiance fields are then encoded into an efficient octree-based structure, that allows us to efficiently query our luminaires during render time. In the following, we detail the different elements of our model.

4.1. Physical Model

Volumetric radiance fields [MSO*19] represent the scene's spatio-directional radiance (light) field as the volume density and directional emitted radiance at any point in space, by using traditional volume rendering techniques. For each outgoing point and direction \mathbf{x} and ω , we define a ray $\mathbf{r}(t) = \mathbf{x} - \omega t$, and compute the emitted field of the luminaire $L_e(\mathbf{r}) := L_e(\mathbf{x}, \omega)$ as

$$L_e(\mathbf{r}) = \int_0^{t_1} T(\mathbf{r}, t) \sigma(\mathbf{r}(t)) \Phi(\mathbf{r}(t), \omega) dt, \quad (3)$$

with t_1 farthest intersection distance with the proxy geometry, $\sigma(\mathbf{x})$ and $\Phi(\mathbf{x}, \omega)$ are the density and emission at point \mathbf{x} towards direction ω , and $T(\mathbf{r}, t)$ is the transmittance.

While transmittance is in general modeled using the classic exponential Beer-Lambert law, Vicini et al. [VJK21] showed that non-exponential light transport [JAG18; BRM*18] is a better-suited model for radiance fields, specially when encoding surfaces due

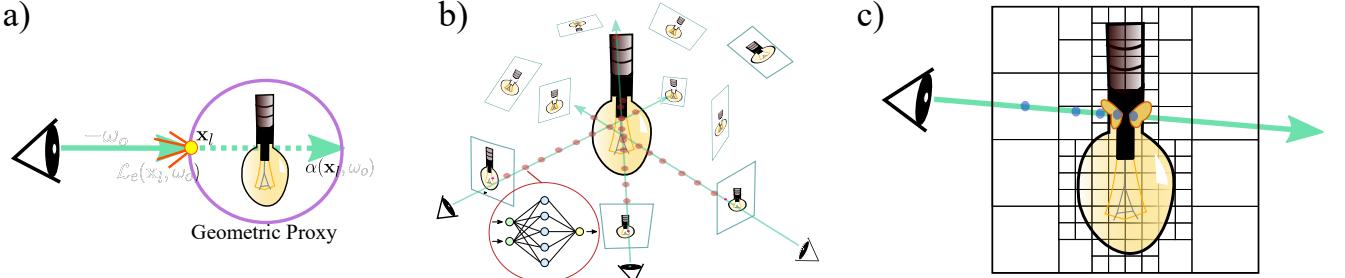


Figure 2: Overview: a) We approximate the appearance of a complex luminaire by using a proxy geometry and encoding the emitted field $L_e(\mathbf{x}, \omega)$ as a five-dimensional function in the proxy surface, plus a transparency function $\alpha(\mathbf{x}, \omega)$ for integrating the luminaire. Previous approaches [ZBX*21] directly learn the emitted field and transparency function, requiring thousands of training images to converge. b) We exploit recent advances in neural radiance fields (NeRFs) to first encode the emitted field using a multi-layer perceptron (MLP) using a volumetric rendering approach [MST*20], using a custom loss function for its training specifically tailored to accurately model the emitted field of complex luminaires. c) To avoid neural queries on render time, we project the angular domain of the NeRF on spherical harmonics, and the spatial domain in an octree, following Yu et al.[YLT*21]. This allows us to significantly limit the spatial queries, leading to a very efficient volumetric representation which is easy to integrate in any render engine.

to their implicit correlation. Therefore, we opt for Vicini's linear transmittance, defined as

$$T(\mathbf{r}, t) = \max \left(0, 1 - \int_0^t \sigma(\mathbf{r}(s)) ds \right). \quad (4)$$

Finally, we model the transparency $\alpha(\mathbf{r}) := \alpha(\mathbf{x}, \omega)$ as

$$\alpha(\mathbf{r}) = 1 - T(\mathbf{r}, t_1). \quad (5)$$

We numerically approximate Equations (3) and (4) by deterministically ray-marching the ray inside the proxy geometry, following the standard procedure in NeRF and other RF-related papers. This operation is fully differentiable, which is well-suited for learning the scene from a set of views.

4.2. Encoding the Radiance Field

As shown in Equation (3), we model the RF as a heterogeneous medium defined by its density $\sigma(\mathbf{x})$ and directional emission $\Phi(\mathbf{x}, \omega)$. The vast body of works on learned-RF representations use a neural network for jointly modeling $\sigma(\mathbf{x})$ and $\Phi(\mathbf{x}, \omega)$ (NeRFs); unfortunately, querying a NN might be expensive, and introduces additional complexity in terms of integration with a rendering engine. Instead, we follow the approach by Yu and colleagues [YLT*21], and project the trained NeRF into a voxelized octree representation of the radiance field (a *plenoctree*), which is compact, CPU friendly, and easy to integrate in any rendering engine.

We first train a NeRF that captures the full dynamic range of the light source. The transition from the common LDR to HDR introduces several practical problems in terms of encoding and training. We solve that by using a training loss that introduces a regularizer that copes with the potentially large radiance magnitudes in HDR. In addition, given the importance of a good transparency term $\alpha(\mathbf{r})$, we directly supervise the opacity during training.

Then, after training, we project the density $\sigma(\mathbf{x})$ and emission

$\Phi(\mathbf{x})$ in a plenoctree. To ease such projection, instead of learning the whole 5-D spatio-directional function, the neural network learns the spherical harmonics (SH) expansion of $\Phi(\mathbf{x}, \omega)$ defined by the SH coefficients $\bar{\Phi}_l^m(\mathbf{x})$, with l and m the SH level and band of the coefficient respectively.

In the following, we describe our neural network architecture, loss function and training details of our NeRF-based luminaires. Then, we discuss the projection to a plenoctree, and how we integrate it into a path tracer.

Network architecture The network takes as input the position and direction, which are augmented using Fourier-based positional encoding [MST*20], and outputs the density $\sigma(\mathbf{x})$ and the $(l_{\max} + 1)^2$ SH coefficients of the directional emission $(\bar{\Phi}_l^m(\mathbf{x}))$ for each color channel, with l_{\max} the number of SH levels. Following previous work [YLT*21], we use a fully-connected 8-layered 256-neurons-wide multilayer perceptron, with an extra output layer of 128 neurons returning $3(l_{\max} + 1)^2$ coefficients of the SH expansion of $\Phi(\mathbf{x}, \omega)$ (Figure 3). We use ReLU as the activation function between the inner layers.

In order to get the final emitted color $\Phi(\mathbf{x}, \omega)$, we also apply a non-linear activation function over the SH-projection, as

$$\Phi(\mathbf{x}, \omega) = \mathcal{S} \left(\sum_{l=0}^{l_{\max}} \sum_{m=-l}^l \bar{\Phi}_l^m(\mathbf{x}) Y_l^m(\omega) \right), \quad (6)$$

with $Y_l^m(\omega)$ the SH basis projected in direction ω , and $\mathcal{S}(\cdot)$ an extended-range sigmoid (a sigmoid multiplied by the maximum scene radiance). This allows us to compute losses in a bounded $[0, 1]$ range. This proved much more stable than using unbounded activation functions.

4.3. Loss function

Following the original work of NeRFs [MST*20], we use two different models for supervising the learning of both the density and

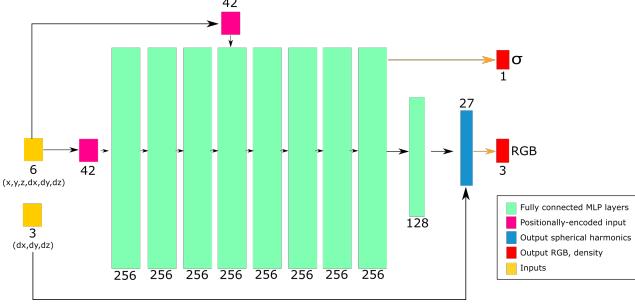


Figure 3: Our architecture. Following previous works on NeRF, we use a densely connected 8-layered 256-neurons-wide multilayer perceptron (MLP), with an extra output layer of 128 neurons. We use ReLU as the activation function between the inner layers and a extended-range sigmoid for the final transformation from SH to RGB values.

emission, by comparing against the training HDR image. In addition, we incorporate a term for actively supervising the transparency $\alpha(r)$. Therefore, our loss function is

$$\mathcal{L} = \frac{1}{|\mathcal{R}|} \sum_{r \in \mathcal{R}} [\mathcal{L}_c(r) + \mathcal{L}_f(r) + \mathcal{L}_\alpha(r)], \quad (7)$$

where \mathcal{R} is the set of $|\mathcal{R}|$ sampled rays, $\mathcal{L}_c(r)$ and $\mathcal{L}_f(r)$ are the HDR color loss for the coarse and fine models, respectively, and $\mathcal{L}_\alpha(r)$ is the transparency loss.

HDR color loss Using standard mean squared error (MSE) in high-dynamic range data results in severe problems in areas with low radiance, since MSE is dominated by errors in brighter regions (e.g. directly visible light emitters). In order to achieve an even learning across the whole dynamic range, we regularize the MSE by the square of the approximate luminance of the pixel, in the spirit of Lehtinen and colleagues [LMH*18]. In essence, Lehtinen et al. proposed to use a mapping similar to Reinhard's global tonemapping operator [RSSF02][†]. However, unlike their approach, we also need to multiply the estimated pixel luminance by the maximum radiance of the luminaire L_{\max} to compensate the fact that we are estimating color in the $[0,1]$ range. We found that keeping the sigmoid activation function for color and extending its range to the maximum radiance of the luminaire enabled a more stable learning of the emission than using exponential functions to unbound the output, which by definition struggle with producing values close to the absolute 0. This results in an HDR color error loss for the coarse model (the loss for the fine model is analogous) defined as

$$\mathcal{L}_c(r) = \left\| \frac{\widehat{L}_e^c(r) - L_e(r)}{\lambda \widehat{L}_e^c(r) + \epsilon} \right\|_2^2, \quad (8)$$

with $\widehat{L}_e^c(r)$ the color prediction by the coarse network, $L_e(r)$ the emission ground truth, and ϵ is a regularizing empirical term (we

[†] Reinhard's global tonemapping has the form $M(y) = \frac{\log(\lambda y + \epsilon)}{L_{\max}}$ and derivative $M'(y) = \frac{\lambda}{(\lambda y + \epsilon)^{-1}}$.

found that $\epsilon = 0.01$ works well for all our tested luminaires). In practice, we found that any decaying function can be used as a regularizer to the MSE loss, balancing the supervision of low and high radiance areas. For example, we also had success with exponentially-decaying regularizing functions, but these introduced an additional hyperparameter (the decay of the exponential function) that required manual tuning for each luminaire.

Transparency loss While the color metric above removes most problems with HDR content, we found that for scenes with isolated objects (i.e. no backgrounds) with HDR radiance, the training convergence would sometimes depend on supervising the transparency. Most importantly, achieving perfect predictions of the transparency of each ray is necessary for a successful integration of the model within a traditional rendering engine later on, enabling perfect blending of the luminaires with the rest of the scene. We therefore introduce a direct supervision of alpha masks through a MSE loss,

$$\mathcal{L}_\alpha(r) = \left\| \widehat{\alpha}^f(r) - \alpha(r) \right\|_2^2, \quad (9)$$

with $\widehat{\alpha}^f(r)$ the transparency estimated by the fine network. Note that we only need to supervise opacity in the fine model, as it is the one that will define the final opacity of a ray. Since all our data is synthetic we have ground truth alpha masks available for supervision.

4.4. Training Details

We used JAX [BFH*18] and its machine learning API, FLAX [HLO*20] for training. We trained our models with a batch size of 1024 rays, each of them sampled at $N_c = 64$ positions in the coarse model, and $N_f = 128$ additional coordinates in the fine one. We set $l_{\max} = 2$ levels of spherical harmonics. We use a 100 images as our training set, and a further 100 images for testing. All three components of the loss (coarse color supervision, fine color supervision, and alpha supervision) were given the same weight. We used Adam [KB14] combined with an exponentially decaying learning rate, starting at $lr = 5 \times 10^{-4}$ and decaying to $lr = 5 \times 10^{-6}$ by the end of training. Our models were trained for 700k-1.1M iterations, which took between 1 and 2 days on an Nvidia RTX 2080Ti.

5. Implementation details

We integrate our RF-based complex luminaires in Mitsuba [Jak10], as a new Emitter plugin attached to the proxy geometry (a sphere or a box) returning the emission $L_e(r)$ (Equation (3)). We also implement a null-like BSDF that returns the transparency $\alpha(r)$ (Equation (5)). Both the Emitter and the null BSDF raymarch through the volumetric representation of the radiance field, querying the density $\sigma(x)$ and directional emission $\Phi(x, \omega)$.

As discussed above, we do not query the trained neural network. Instead, we project the density and emission values into a plenocloud [YLT*21], i.e. an octree-based voxelized representation of the radiance field. This results into massive savings in terms of performance, as well as a much simpler integration in Mitsuba. Querying the octree is much cheaper than querying a neural network, and we only need to query the few intersected voxels by each ray until a

surface is hit (opacity saturates), massively reducing the number of samples placed over empty space which significantly impact the performance of NeRFs, in which a set number of samples for each ray is used.

Creating the plenoctree We extract the octree from the neural volume by densely sampling the density $\sigma(\mathbf{x})$ and the SH coefficients of the emission $\Phi_l^m(\mathbf{x})$. We first place a single sample in the center of each voxel to obtain their directionally-invariant density, and then prune voxels whose density is below an empirically set threshold (0.01). This allows us to prune empty space and most NeRF artifacts. Then, for each dense voxel, we throw an additional 256 positional samples uniformly placed within the volume defined by the voxel and average them to avoid aliasing at discontinuities. The maximum depth of the octree directly controls the quality of the extracted representation, with higher depths enabling finer detail, while also requiring more storage space. An octree using a maximum depth of 10 (fine quality), containing the representation of a single luminaire, has a memory footprint of around 1.7GBs.

Sampling In our current implementation we uniformly sample the proxy geometry, which is admittedly suboptimal and does not leverages the precomputed data of the luminaires radiance field. Importance sampling according to the RF would provide a significant variance reduction, although it remains an avenue of future work.

6. Evaluation and Results

6.1. Datasets

We generate our datasets in Mitsuba using a volumetric path tracer. Each luminaire is rendered using an orthographic camera placed at a 100 pseudo-random directions, generated using a 2D Halton sequence on the sphere around the luminaire. Each view is rendered with a resolution of 800^2 , making a total of 64M rays per luminaire.

We render five different luminaires with varying geometric complexity, number of emitters, and dynamic ranges (with maximum radiance ranging from 4.5 in STAR to 40 in CLUSTER). We intended to create a datasets of luminaires with a size and variety similar to Zhu et al.'s [ZBX*21], with some luminaires showing caustic paths (e.g. the light bulbs in both DALLAS and CLUSTER), as well as diffuse appearance due to multiple scattering. The rendering time of each dataset depends on its complexity, but they took between 12 and 36h each using 32 threads in a workstation with a dual Intel Xeon Gold processor. Unless stated otherwise, our results are generated using an order-2 SH expansion ($l_{\max} = 2$, 9 SH coefficients), with a maximum octree depth of 10 levels (spatial resolution of 1024^3).

6.2. Evaluation

Here we analyze the reconstruction error introduced by our RF approximation of the luminaires, as well as the impact of our design choices on the final error. In all cases, we report the average PSNR and SSIM for all test views.

Figure 4 shows example reconstructions for all our datasets, with the average reconstruction errors listed in Table 1. We refer to the supplemental video for animations of the luminaires, to assess temporal stability of our reconstructions.

	PSNR	SSIM
DALLAS	31.5072	0.9374
PORTICA	28.6694	0.8594
CLUSTER	27.7260	0.8872
NEOCLASSICAL	26.8784	0.8829
STAR	27.8027	0.9119

Table 1: Reconstruction error. PSNR and SSIM error metrics for all datasets used in our work, averaged over all test views of the luminaires.

Loss ablation study We analyze the effect of the different terms of our loss function (HDR regularizer, transparency supervision) in Figure 5, for the NEOCLASSICAL luminaire. We compare them against the baseline NeRF loss, which uses MSE. Our custom loss enables stable, high quality learning of the full dynamic range, and a correct volume density representation. Table 2 includes the numerical error for this experiment. While we found that for some luminaires the transparency supervision results in a slightly worse emission reconstruction, the improvement on transparency is significant. Note that a good transparency estimation is crucial for high-quality integration of the luminaire in synthetic scenes.

	PSNR L_e	SSIM L_e	RMSE α
MSE	19.8547	0.6987	0.3617
MSE+reg	28.1237	0.9062	0.1853
MSE+reg+ α loss	26.8784	0.8829	0.0358

Table 2: Loss ablation study. Error metrics demonstrating the effect of each term in our metric for the NEOCLASSICAL dataset. Note that the opacity loss (α loss) slightly introduces error on the reconstruction of the luminaire emission L_e , but dramatically improves the reconstruction of the transparency α .

Performance vs Quality In Table 3 we compare the performance of increasing the level of spherical harmonics when modeling the directional emission $\Phi(\mathbf{x}, \omega)$. The effect of increasing from $l_{\max} = 2$ to $l_{\max} = 4$ is marginal in error, but significantly increases rendering time, memory and storage requirements. Training directly our model using a band-limited SH-based angular representation of $\Phi(\mathbf{x}, \omega)$, allows us to preserve high-frequencies on the ray domain (e.g. occlusions), despite the local $\Phi(\mathbf{x}, \omega)$ is low frequency.

In order to further increase performance, we discard the integration of samples that have a density value below a certain threshold σ_{\min} , and consider a ray saturated when transparency reaches another threshold α_{\max} . This allows us to reduce the computational overhead of multiplying spherical harmonic weights by the sampled tree value for every intersected voxel, and avoids unnecessarily querying the octree more times than those really needed to obtain an accurate color value. We have empirically set the values of these heuristics to $\sigma_{\min} = 0.1$ and $\alpha_{\max} = 0.9$ respectively. Our method introduces an average overhead of 20% per sample over explicitly handling the luminaires.

Transmittance model We compare the impact of using a linear



Figure 4: Reconstruction examples. Example test views of the reconstruction of our dataset using our models. From left to right: DALLAS, NEOCLASSICAL, PORTICA, CLUSTER, and STAR. Average reconstruction errors for all datasets can be found in Table 1.

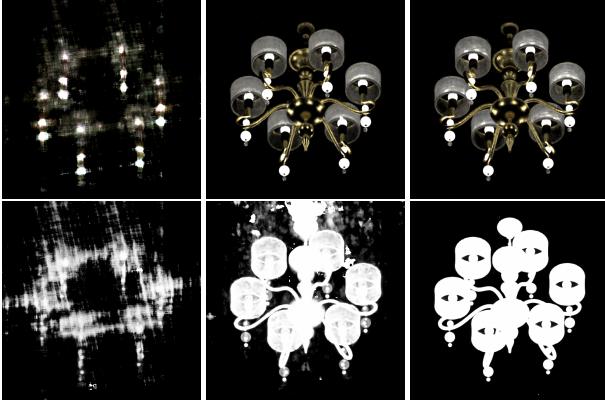


Figure 5: Loss ablation study. We analyze the effect of our metric on the NEOCLASSICAL luminaire, in the reconstruction of both the emission L_e (top) and transparency α (bottom). From left to right: LDR MSE, MSE+regularization, MSE+regularization+opacity loss. The error metrics can be found in Table 2.

SH levels	PSNR	SSIM
$l_{\max} = 2$ (9 coeff.)	27.7260	0.8872
$l_{\max} = 3$ (16 coeff.)	27.6958	0.8860
$l_{\max} = 4$ (25 coeff.)	27.7367	0.8859

Table 3: Effect of number of SH levels. PSNR and SSIM error metrics as a function of the maximum SH levels used to model the directional emission $\Phi(\mathbf{x}, \omega)$ in the CLUSTER dataset.

transmission model vs the classic exponential transmittance (Table 4). The linear model slightly outperforms the exponential one on every tested dataset, both in quality and rendering time. The faster decay of the linear transmittance reduces the number of queries to the octree, resulting in rendering times 5% faster.

6.3. Results

Here we show the results of integrating our RF-based volumetric representation of complex luminaires in Mitsuba. We compare our method against using the explicit light source as geometry, using path tracing (PT) or volumetric path tracing (VPT) in scenes that require the modelling of participating media (some of the tinted

	PORTICA		DALLAS	
	PSNR	SSIM	PSNR	SSIM
Exponential	28.43737	0.8566	31.4002	0.9371
Linear	28.6694	0.8594	31.5072	0.9374

Table 4: Transmittance model comparison. Comparison between the classic exponential transmittance and the linear model used for our luminaires. We consistently found the linear model outperforms the exponential one in the context of modelling luminaires.

glasses are modelled as dielectric pieces with homogeneous mediums inside them). All our renders have been computed in a workstation with a dual Intel Xeon Gold processor (18x2 hyper-threaded cores), using 32 threads. Reference images are rendered with VPT or PT on the explicit luminaires representation, using 32768 samples per pixel. Figures 6 shows results of our volumetric complex luminaires integrated in synthetic scenes; our method allows high-quality approximation of complex luminaires with a small overhead. In Figures 1 and 7 we compare our method against explicitly modeling and rendering the luminaires using path tracing (PT or VPT). Our method shows less variance and higher quality at equal time, while for equal RMSE, our method outperforms explicit PT by almost two orders of magnitude (Table 5), with minimal visual differences with respect to the reference.

7. Discussion & Limitations

Comparison with Zhu et al. [ZBX*21] While it is difficult to compare between different systems, our model offers similar quality and speedups as neural complex luminaires, with a significantly smaller training dataset. Table 6 compares the reconstruction error of Zhu et al.’s method with ours in the DALLAS dataset with the same high-resolution 100 training images. We obtain a $\times 3$ improvement in terms of PSNR. Zhu et al.’s method in general requires three to four orders of magnitude more training data for achieving comparable PSNR. Rendering-wise, our model is easy to integrate in any CPU-based rendering pipeline, without the need of complex neural inference methods over a GPU, and introduces a comparable penalty cost as Zhu et al.’s when run on a GPU. Finally, our model is two orders of magnitude heavier in terms of memory storage, although Zhu’s requires a significant amount of GPU memory for running.

		spp	RMSE	Time	Speedup
PORTICA Living Room	PT	4096	7.1211	84 min	96.2
	Ours	32	6.8146	52.4 s	
DALLAS Bedroom	PT	1024	7.9824	24.6 min	17.57
	Ours	32	7.9228	1.4 min	
CLUSTER Living Room 2	VPT	1024	7.5267	19 min	95.8
	Ours	4	7.4239	11.9 s	
NEOCLASSICAL Bedroom	PT	1024	7.2304	14.1 min	38.11
	Ours	16	7.3782	22.2 s	
STAR Living Room 2	VPT	512	2.6439	7.8 min	8.83
	Ours	32	2.7113	53 s	

Table 5: Equal-Quality Results. We compare our method against traditional path tracing or volumetric path tracing, depending on the luminaire. At equal RMSE, our method shows significant speedups, of up to two orders of magnitude, depending on the rendering complexity of the modelled luminaire. Corresponding renders of this table can be found in Figure 7



Figure 6: Converged results. We render some of our scenes with more samples (4096) using our method to showcase their quality. Notice how variance is smaller in these scenes than in the reference renders used for equal error measurements, at a fraction of the rendering time.



Figure 7: Equal-time and equal-quality comparison. From left to right: reference render; our method (see Table 5 for samples per pixel and rendering time), equal-time explicit renders; and equal-quality (RMSE) explicit rendering. See Table 5 for error and rendering times. Notice how our approach is significantly faster than explicit PT or VPT while introducing minimal error in the scene.

	PSNR	SSIM	Render Time (s)
[ZBX*21]	10.96	0.611	0.93
Ours	31.50	0.937	0.11

Table 6: Comparison with Zhu et al. [ZBX*21]. Peak signal to noise ratio (PSNR) and structural similarity (SSIM) of Zhu et al.’s neural luminaires [ZBX*21] and our method for the DALLAS dataset (100 images, 800 × 800). We also report rendering time on an unoptimized standalone GPU rendering implementation for both methods.

Comparison with Velazquez et al. [VDWG15] As opposed to the method by Velazquez et al., our method is able to both handle emission and appearance seamlessly, making it easier to integrate in a render engine, while providing high-quality render and illumination of the luminaires.

Limitations – Precomputation cost Precomputation times are rather long (2-3 days including both data generation and training), which is an issue of most complex luminaire rendering methods. While part of such cost comes from rendering the dataset, our training is still lengthy, and includes both training a NeRF and projecting it to an octree, following Plenoctrees [YLT*21]. A faster approach is to train directly the SH-based octree [FTC*22], which requires a total variation regularization to avoid aliasing at discontinuities, as well as fine per-scene parameter tuning for best performance.

Limitations – Sampling Currently, our sampling routine is a sub-optimal naïve surface-based sampling strategy. Investigating novel sampling techniques that take advantage of implicit learned representations of the luminaires is a promising avenue for future work.

Limitations – Illumination from other luminaires Our method only captures the light transport from paths inside the luminaire, and ignore inter-reflections from the rest of the scene (i.e. assumes emitting-only luminaires). Accounting for paths outside the luminaire would require adding an additional term encoding the light transport response of the luminaire as a function of the incident light field.

Conclusions In this work we have introduced a volumetric representation for complex luminaires that is both fast and capable of achieving high levels of quality. We have extended neural radiance fields to fit the particular needs of modelling complex luminaires (a high dynamic range, accurate transparency, and capable of modelling null emission, among others). Then, we have successfully exploited our volumetric representation within traditional rendering pipelines, showing significant speedups, and using significantly less input data and computational resources than previous methods.

Acknowledgements We want to thank the authors of [ZBX*21] for sharing their code with us. This work has been partially supported by the European Research Council (ERC) under the EU Horizon 2020 research and innovation programme (project CHAMELEON, grant No 682080), the EU MSCA-ITN programme (project PRIME, grant No 956585) and the Spanish Ministry of Science and Innovation (project PID2019-105004GB-I00).

Jorge Condor acknowledges support from a grant from the I3A Institute (Beca TFM + Practicas).

References

- [ACG*17] ALIAGA, CARLOS, CASTILLO, CARLOS, GUTIÉRREZ, DIEGO, et al. “An Appearance Model for Textile Fibers”. *Computer Graphics Forum* 36.4 (2017) 2.
- [AR98] ASHDOWN, IAN and RYKOWSKI, RON. “Making Near-Field Photometry Practical”. *Journal of the Illuminating Engineering Society* 27 (July 1998), 67–79 2.
- [Ash95] ASHDOWN, IAN. “Near-Field Photometry: Measuring and Modeling Complex 3-D Light Sources”. *SIGGRAPH* 1995. 1995 2.
- [BFH*18] BRADBURY, JAMES, FROSTIG, ROY, HAWKINS, PETER, et al. *JAX: composable transformations of Python+NumPy programs*. Version 0.2.5. 2018 5.
- [BNJ15] BITTERLI, BENEDIKT, NOVÁK, JAN, and JAROSZ, WOJCIECH. “Portal-masked environment map sampling”. *Computer Graphics Forum* 34.4 (2015), 13–19 1.
- [BRM*18] BITTERLI, BENEDIKT, RAVICHANDRAN, SRINATH, MÜLLER, THOMAS, et al. “A radiative transfer framework for non-exponential media”. *ACM Trans. Graph.* 37.6 (2018), 225 2, 3.
- [FTC*22] FRIDOVICH-KEIL AND YU, TANCIK, MATTHEW, CHEN, QIN-HONG, et al. “Plenoxels: Radiance Fields without Neural Networks”. *CVPR*. 2022 9.
- [GUK*17] GUILLÉN, IBÓN, UREÑA, CARLOS, KING, ALAN, et al. “Area-preserving parameterizations for spherical ellipses”. *Computer Graphics Forum*. Vol. 36. 4. Wiley Online Library. 2017, 179–187 1.
- [HDCD15] HEITZ, ERIC, DUPUY, JONATHAN, CRASSIN, CYRIL, and DACHSBACHER, CARSTEN. “The SGGX microflake distribution”. *ACM Trans. Graph.* 34.4 (2015), 1–11 2.
- [HKSS98] HEIDRICH, WOLFGANG, KAUTZ, JAN, SLUSALLEK, P., and SEIDEL, HANS-PETER. “Canned Lightsources”. *Rendering Techniques*. 1998 2, 3.
- [HLO*20] HEEK, JONATHAN, LEVSKAYA, ANSELM, OLIVER, AVITAL, et al. *Flax: A neural network library and ecosystem for JAX*. Version 0.4.0. 2020 5.
- [JAG18] JARABO, ADRIAN, ALIAGA, CARLOS, and GUTIERREZ, DIEGO. “A radiative transfer framework for spatially-correlated materials”. *ACM Trans. Graph.* 37.4 (2018), 1–13 2, 3.
- [Jak10] JAKOB, WENZEL. *Mitsuba renderer*. 2010 5.
- [JAM*10] JAKOB, WENZEL, ARBREE, ADAM, MOON, JONATHAN T, et al. “A radiative transfer framework for rendering materials with anisotropic structure”. *ACM Trans. Graph.* 29.4 (2010), 1–13 2.
- [JM12] JAKOB, WENZEL and MARSCHNER, STEVE. “Manifold exploration: A markov chain monte carlo technique for rendering scenes with difficult specular transport”. *ACM Trans. Graph.* 31.4 (2012), 1–13 1.
- [KB14] KINGMA, DIEDERIK P. and BA, JIMMY. “Adam: A Method for Stochastic Optimization”. *arXiv e-prints*, arXiv:1412.6980 (Dec. 2014), arXiv:1412.6980. arXiv: 1412.6980 [cs.LG] 5.
- [KSZ*16] KHUNGURN, PRAMOOK, SCHROEDER, DANIEL, ZHAO, SHUANG, et al. “Matching Real Fabrics with Micro-Appearance Models”. *ACM Trans. Graph.* 35.1 (Dec. 2016). ISSN: 0730-0301 2.
- [LMH*18] LEHTINEN, JAAKKO, MUNKBERG, JACOB, HASSELGREN, JON, et al. “Noise2Noise: Learning Image Restoration without Clean Data”. *CoRR* abs/1803.04189 (2018). arXiv: 1803 . 04189. URL: <http://arxiv.org/abs/1803.04189> 5.
- [LN17] LOUBET, GUILLAUME and NEYRET, FABRICE. “Hybrid mesh-volume LoDs for all-scale pre-filtering of complex 3D assets”. *Computer Graphics Forum* 36.2 (2017), 431–442 2.

- [LPG14] LU, HEQI, PACANOWSKI, ROMAIN, and GRANIER, XAVIER. “Position-dependent importance sampling of light field luminaires”. *IEEE Transactions on Visualization and Computer Graphics* 21.2 (2014), 241–251 [2](#).
- [LSS*19] LOMBARDI, STEPHEN, SIMON, TOMAS, SARAGIH, JASON, et al. “Neural Volumes: Learning Dynamic Renderable Volumes from Images”. *ACM Trans. Graph.* 38.4 (July 2019). ISSN: 0730-0301 [2](#).
- [MMP08] MAS, ALBERT, MARTÍN, IGNACIO, and PATOW, GUSTAVO. “Compression and Importance Sampling of Near-Field Light Sources”. *Comput. Graph. Forum* 27 (Dec. 2008), 2013–2027 [2](#).
- [MPG*16] MÜLLER, THOMAS, PAPAS, MARIOS, GROSS, MARKUS, et al. “Efficient Rendering of Heterogeneous Polydisperse Granular Media”. *ACM Trans. Graph.* 35.6 (Nov. 2016), 168:1–168:14. ISSN: 0730-0301 [2](#).
- [MPH*15] MENG, JOHANNES, PAPAS, MARIOS, HABEL, RALF, et al. “Multi-Scale Modeling and Rendering of Granular Materials”. *ACM Trans. Graph.* 34.4 (July 2015). ISSN: 0730-0301 [2](#).
- [MSO*19] MILDENHALL, BEN, SRINIVASAN, PRATUL P., ORTIZ-CAYON, RODRIGO, et al. “Local Light Field Fusion: Practical View Synthesis with Prescriptive Sampling Guidelines”. *ACM Trans. Graph.* (2019) [2](#), [3](#).
- [MST*20] MILDENHALL, BEN, SRINIVASAN, PRATUL P., TANCIK, MATTHEW, et al. “NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis”. *ECCV*. 2020 [2](#), [4](#).
- [Mus11] MUSCHAUER, JULIUS. “What’s in a ray set: moving towards a unified ray set format”. *Illumination Optics II*. Ed. by KIDGER, TINA E. and DAVID, STUART. Vol. 8170. Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series. Oct. 2011, 81700N, 81700N [2](#).
- [MWM07] MOON, JONATHAN T., WALTER, BRUCE, and MARSCHNER, STEPHEN R. “Rendering Discrete Random Media Using Precomputed Scattering Solutions”. *EGSR*. EGSR’07. Grenoble, France: Eurographics Association, 2007, 231–242. ISBN: 9783905673524 [2](#).
- [Ney98] NEYRET, FABRICE. “Modeling, Animating, and Rendering Complex Scenes Using Volumetric Textures”. *IEEE Trans. Vis. Comput. Graph.* 4.1 (1998), 55–70 [2](#).
- [Nga87] NGAI, P.Y. “On Near-Field Photometry”. *Journal of the Illuminating Engineering Society* 16.2 (1987), 129–136 [2](#).
- [Pet21] PETERS, CHRISTOPH. “BRDF importance sampling for polygonal lights”. *ACM Trans. Graph.* 40.4 (2021), 1–14 [1](#).
- [RSSF02] REINHARD, ERIK, STARK, MICHAEL, SHIRLEY, PETER, and FERWERDA, JAMES. “Photographic Tone Reproduction for Digital Images”. *SIGGRAPH*. New York, NY, USA, 2002, 267–276. ISBN: 1581135211 [5](#).
- [SKZ11] SCHRÖDER, KAI, KLEIN, REINHARD, and ZINKE, ARNO. “A Volumetric Approach to Predictive Rendering of Fabrics”. *Computer Graphics Forum* (2011). ISSN: 1467-8659 [2](#).
- [SZW19] SITZMANN, VINCENT, ZOLLMÖFER, MICHAEL, and WETZSTEIN, GORDON. “Scene Representation Networks: Continuous 3D-Structure-Aware Neural Scene Representations”. *arXiv e-prints*, arXiv:1906.01618 (June 2019), arXiv:1906.01618. arXiv: [1906](#) . [01618](#) [cs.CV] [2](#).
- [TTM*21] TEWARI, AYUSH, THIES, JUSTUS, MILDENHALL, BEN, et al. “Advances in neural rendering”. *arXiv preprint arXiv:2111.05849* (2021) [2](#).
- [VDWG15] VELÁZQUEZ-ARMENDÁRIZ, EDGAR, DONG, ZHAO, WALTER, BRUCE, and GREENBERG, DONALD P. “Complex luminaires: Illumination and appearance rendering”. *ACM Trans. Graph.* 34.3 (2015), 1–15 [1](#), [2](#), [9](#).
- [VJK21] VICINI, DELIO, JAKOB, WENZEL, and KAPLANYAN, ANTON. “A non-exponential transmittance model for volumetric scene representations”. *ACM Trans. Graph.* 40.4 (2021), 1–16 [2](#), [3](#).
- [WMJ*17] WU, GAOCHANG, MASIA, BELEN, JARABO, ADRIAN, et al. “Light field image processing: An overview”. *IEEE Journal of Selected Topics in Signal Processing* 11.7 (2017), 926–954 [2](#).
- [YLT*21] YU, ALEX, LI, RUILONG, TANCIK, MATTHEW, et al. “PlenOctrees for Real-time Rendering of Neural Radiance Fields”. *ICCV*. 2021 [2](#), [4](#), [5](#), [9](#).
- [ZBX*21] ZHU, JUNQIU, BAI, YAOYI, XU, ZILIN, et al. “Neural complex luminaires: representation and rendering”. *ACM Trans. Graph.* 40.4 (2021), 1–12 [1](#)–[4](#), [6](#), [7](#), [9](#).
- [ZJMB11] ZHAO, SHUANG, JAKOB, WENZEL, MARSCHNER, STEVE, and BALA, KAVITA. “Building Volumetric Appearance Models of Fabric Using Micro CT Imaging”. *ACM Trans. Graph.* 30.4 (July 2011). ISSN: 0730-0301 [2](#).
- [ZWDR16] ZHAO, SHUANG, WU, LIFAN, DURAND, FRÉDO, and RAMAMOORTHI, RAVI. “Downsampling scattering parameters for rendering anisotropic media”. *ACM Trans. Graph.* 35.6 (2016), 1–11 [2](#).