

# NeRRF: 3D Reconstruction and View Synthesis for Transparent and Specular Objects with Neural Refractive-Reflective Fields

Xiaoxue Chen<sup>\*1</sup>, Junchen Liu<sup>\*2</sup>, Hao Zhao<sup>1</sup>, Guyue Zhou<sup>1</sup>, and Ya-Qin Zhang<sup>1</sup>

**Abstract**—Neural radiance fields (NeRF) have revolutionized the field of image-based view synthesis. However, NeRF uses straight rays and fails to deal with complicated light path changes caused by refraction and reflection. This prevents NeRF from successfully synthesizing transparent or specular objects, which are ubiquitous in real-world robotics and A/VR applications. In this paper, we introduce the refractive-reflective field. Taking the object silhouette as input, we first utilize marching tetrahedra with a progressive encoding to reconstruct the geometry of non-Lambertian objects and then model refraction and reflection effects of the object in a unified framework using Fresnel terms. Meanwhile, to achieve efficient and effective anti-aliasing, we propose a virtual cone supersampling technique. We benchmark our method on different shapes, backgrounds and Fresnel terms on both real-world and synthetic datasets. We also qualitatively and quantitatively benchmark the rendering results of various editing applications, including material editing, object replacement/insertion, and environment illumination estimation. Codes and data are publicly available at [NeRRF](#).

**Index Terms**—Novel View Synthesis, Neural Rendering, Shape Reconstruction, Transparent Objects, Specular Objects.

## I. INTRODUCTION

Transparent and specular objects are omnipresent in the real world, for example, glass objects, mirrors, or liquids. Successfully modeling scenes containing these objects will benefit many augmented reality or virtual reality applications including relighting or scene editing. However, 3D reconstruction and novel view synthesis on such scenes is an ill-posed problem due to complex optical phenomena, especially refraction and reflection. Different from diffuse objects that scatter incident light, the appearance of a transparent/specular object is not determined by itself but by the surrounding environment. The light bends inside (or on the surface of) the object, and the distorted environment radiance accumulated in the outgoing ray becomes the object's appearance. Therefore, the RGB information of the transparent/specular object is highly view-dependent and not reliable, and common computer vision algorithms designed for opaque objects cannot be applied to these scenes. To this end, this study aims to model scenes containing non-Lambertian objects, reconstructing the 3D geometry of these objects and learning environment radiance from reflective or refractive surfaces.

In order to model refraction and reflection and engage in physically-based ray tracing, a representation of the object's

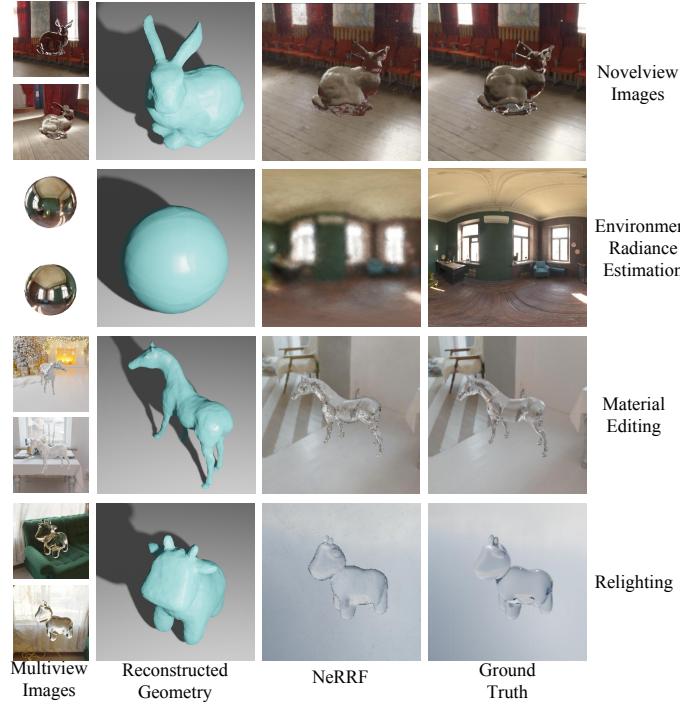


Fig. 1. Qualitative results of NeRRF. (a) Results on novel view synthesis. (b) The results of environment radiance estimation. We learn the radiance from only the reflection on the ball. (c) The results of material editing. We change the material of the horse from reflective to refractive. (d) The results of relighting. We relight the cow with a new illumination.

surface is necessary to be constructed, either explicitly or implicitly. Conventional 3D reconstruction techniques, such as laser scanners and standard depth sensors, are designed primarily for capturing opaque objects. Due to the intricate visual properties, the captured data of transparent objects using these sensors is distorted. Besides, as mentioned before, without known environment illumination, RGB information is also hard to utilize. Therefore, we choose a direct and robust input to reconstruct the 3D shape of both transparent and specular objects, which is the silhouette of the object. These silhouettes can be easily obtained with on-the-shelf instance segmentors[3][1]. Besides, although it's difficult for humans to precisely model non-Lambertian objects from images, it's easy to distinguish the boundaries of objects, so accurate silhouettes can also be acquired through manual annotation.

Nonetheless, how to utilize object masks is also a problem. This necessitates the underlying algorithm to have the

<sup>1</sup> Institute for AI Industry Research (AIR), Tsinghua University, China  
chenxx21@mails.tsinghua.edu.cn, zhaohao@air.tsinghua.edu.cn.

<sup>2</sup> Beihang University, China 20373790@buaa.edu.cn.

\* Equal contribution.

ability to project 3D objects into mask images in a differentiable manner. Consequently, to address this requirement, we resort to the field of differentiable rendering[32]. NDR [51], as a recent work in this domain, employs differentiable marching tetrahedrons to enable gradient-based optimization directly on the surface mesh. As inspiration, we leverage Deep Marching Tetrahedra (DMTet) [58], a differentiable hybrid shape representation, to render object mask images, which serves as the first stage. The object geometry is optimized by back-propagating the loss between the rendered masks and preprocessed silhouettes. Since there are only silhouettes as supervision, in order to avoid excessive details on the object surface, we also employ a progressive encoding that learns the object geometry in a coarse-to-fine manner. With the above algorithm, we can successfully reconstruct the geometry of transparent/specular objects. Experiments on diverse objects (Fig. 6) demonstrate the robustness and generalization of selecting silhouettes as input. With the reconstructed object’s shape, we are enabled to disentangle the geometry and appearance, which facilitates a concentrated investigation into the environmental radiance estimation.

Radiance estimation is a long-existing computer vision problem and neural radiance fields (NeRFs)[48] have recently broken its bottleneck. Despite the noticeable recent progress, current NeRF formulations are still limited to sampling points on straight rays for training and inference. This limitation prevents NeRF from modeling scenes containing transparent/specular objects, as these objects inevitably lead to complicated light path changes due to refraction or reflection. This limitation is evidenced by the poor results in the left column of Fig.9 (a) and (b). Nonetheless, given that the object’s shape has been estimated during the first stage, we can explicitly model the light path change in a physically-based manner, and equip NeRF with the capability of modeling **non-straight rays**, which serves as the second stage.

Our solution, which successfully synthesizes unseen views of transparent/specular objects as shown in Fig. 1, is named as **Neural Refractive-Reflective Field** (NeRRF). NeRRF explicitly incorporates the reflection equation and Snell’s law, the image formulation for transparent objects, into the ray-tracing process of NeRF. Due to the optical law that the relative fraction of reflection is determined by Fresnel’s equation, refraction and reflection of NeRRF are modeled in a **unified** manner based on the Fresnel term. Since both refraction and reflection are highly sensitive to the surface normal, and the normal extracted from the estimated geometry in the first stage may be inaccurate, we also propose a supersampling module based upon a **virtual cone** idea. This module addresses the high-frequency noise raised by the unsmooth estimated normals. Experiments demonstrate that this module brings better results while being tractable.

Through the two-stage pipeline, NeRRF demonstrates its capability to effectively disentangle a transparent/specular object’s geometry and appearance. This distinctive feature equips NeRRF to accommodate a range of AR or VR applications. As shown in Tab. I, to the best of our knowledge, NeRRF is the only methodology capable of concurrently addressing the challenges of modeling refraction and reflection while also

TABLE I  
METHOD COMPARISON.

Method	A	B	C	D	E	F	G
Eikonal [6]	✓			✓			
NeRFRO [54]	✓		✓	✓	✓		
NeTO, ... [37], [42], [73], [76], [68]	✓			✓	✓	✓	
NeRO, ...[64], [40], [51]		✓	✓	✓	✓	✓	
Ref-NeRF, ... [9], [66], [62]		✓	✓	✓		✓	✓
NeRF-DS, ...[77], [81], [24]		✓		✓			
IDR [80]		✓		✓	✓		
PhySG [84]		✓	✓	✓	✓	✓	
<b>NeRRF(ours)</b>	✓	✓	✓	✓	✓	✓	✓

The first group focus on modeling the refractive effects and the second group focus on the reflective effects. (A) model the refractive effects. (B) model the reflective effects. (C) allow environment radiance estimation while training. (D) novel view synthesis. (E) shape reconstruction. (F) relighting. (G) scene editing.

facilitating applications such as relighting and scene editing.

To evaluate the effectiveness of NeRRF, we build and contribute a benchmark with diverse shapes, backgrounds, and Fresnel terms, to the community. Experiments (Fig.9) on this benchmark conclusively demonstrate the superiority of NeRRF over the vanilla NeRF and other methods that can model refraction and reflection. In addition, we show many different applications of NeRRF, three of which are highlighted in Fig.1. In the second row of Fig.1, we achieve faithful environment radiance estimation results, which can potentially serve as a light probe [15]. In the third row of Fig.1, we edit the material of the horse by switching it from reflective to refractive. Besides, in the last row of Fig.1, we also demonstrate the result of the relighting of a cow, which has a similar appearance compared with the reference image generated using the rendering engine. These applications are all quantitatively evaluated on our benchmark, to facilitate future research.

Our main contributions are summarized as follows:

- We reconstruct the 3D shape of non-lambertian object leveraging a hybrid representation with only object silhouette as input. With the help of progressive encoding, our method can estimate the geometry of object precisely.
- We propose a novel refractive-reflective field that explicitly models light path changes caused by refraction and reflection, notably in a unified manner through a Fresnel term. It also benefits from an efficient and effective anti-aliasing module using virtual cones.
- We contribute a diverse benchmark and conclusively show the superiority of RRF over the vanilla NeRF and a strong NeRF upgrade that integrates ray tracing.
- We provide several different applications covering material editing, object replacement/insertion and environment radiance estimation. They are also quantitatively evaluated. Codes and data will be released.

## II. RELATED WORKS

**Novel view synthesis** is the task of synthesizing images or videos from unseen viewpoints given a set of input views. Early works[12], [36], [21] address image-based rendering

with view interpolation and light field techniques. Some other works resolve novel view synthesis by establishing better representations for 3D scenes like mesh-based[67], [10], [17] or volumetric representations[34], [57]. With the rise of deep learning, many learning-based methods [31], [20], [19], [41], [87], [47] emerge and synthesize photorealistic rendering by utilizing the differentiable rendering pipeline. Recently, the neural radiance field (NeRF) [48] has shown great success in novel view synthesis. It represents a 3D scene as a continuous 5D radiance field function parameterized by a neural network. However, training NeRF models requires a large set of images, so several works [82], [75], [18] are proposed to reduce the number of training views and improve generalization by utilizing image features or depth supervision. Besides, [53], [70] combine NeRF with implicit 3D representations which enable both surface and volume rendering using the same model. In addition, [4], [5] focus on training an anti-aliasing NeRF with the integrated positional encoding of conical frustums.

**Illumination estimation** has been a long-standing problem. [15] first use a mirrored sphere as a physical probe to measure the environment radiance. Subsequent works [16], [72], [55] use other known objects as light probes to capture the light. With the rising of deep learning, a series of works like [61], [83], [69] estimate the environment lighting from LDR RGB images using end-to-end neural networks. Besides, [13] generate high-quality HDRIs using a text-driven framework. Estimating the illumination can benefit many applications in augmented reality (AR) or virtual reality (VR). [39] disentangles outdoor scenes into illumination and other scene factors, and changes lighting effects to generate novel images. In addition, [60], [86] estimate an HDR environment map and show applications for augmented reality like object insertion.

**Modeling reflection and refraction.** In computer graphics, reflection and refraction are modeled by simulating the light paths using techniques like ray-tracing. Hence existing works achieve reconstructing shapes [35], [29], [49], [63], [43], [38], [25], [65], [74] or estimating poses [26], [28], [11], depths [14] and optical flow [78] of reflective objects by simulating the light transport passing refractive materials. Besides, [34], [88] leverage silhouette to reconstruct shapes of reflective or refractive objects, and many works like [79], [46], [44], [45] have been proposed to detect or segment non-Lambertian objects. For reflection, [52] models a shiny object by separating the view-dependent and view-independent components of the surface reflection. Other works model reflection by employing a parametric BRDF[33], or decomposing each image into a transmitted layer and a reflected layer[59], [23].

While NeRF [48] maps 3D coordinates to corresponding radiance, it's challenging to deal with complex view-dependent effects as reflection and refraction. For refraction, [42], [37], [73], [76] model the light path changing with a specific image capture setup such as a gray-coded background, to reconstruct the geometry of transparent shapes, which cannot be applied to images in the wild. [54], [6] follow the ray equation of Eikonal rendering [30] to conduct light transport in refractive objects. [68] introduces a ray bending network to predict the refracted ray, while the environment radiance is considered as a prior. For reflection, [40], [9], [66], [62], [51], [84], [80]

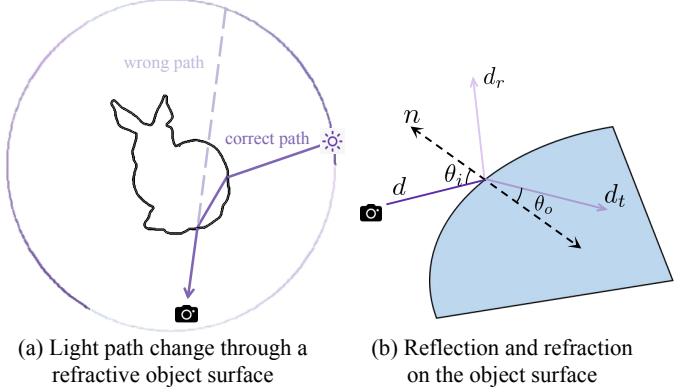


Fig. 2. The illustration of (a) light path change, and (b) physical process of reflection and refraction.

assume opaque surfaces with BRDF, and generate renderings of a non-Lambertian object. However, these methods lack the modeling of light-path changing that can't be applied in complex reflection and refraction. Instead, [64], [24] physically models view direction changes on the object's surface, which enables representing mirror reflection. Besides, [81] use a group of feature fields in parallel sub-spaces to represent scenes with reflection. [77] focus on dynamic specular objects and reformulate the neural radiance field to be conditioned on surface position and orientation in the observation space.

It is manifest from Tab. I that all the methods mentioned above have certain constraints when applied in practical scenarios. In light of these limitations, in this paper, we introduce NeRRF, a continuous neural reflective-refractive radiance field. As illustrated in Tab. I, NeRRF can not only reconstruct the geometry of non-Lambertian objects but also model the intricate phenomena of refraction and reflection simultaneously, which enables novel view synthesis and illumination estimation. Benefit from the disentanglement of geometry representation and radiance representation, our method can also be applied to a series of AR/VR applications including object editing, which needs to change the geometry or optical properties of object, or relighting, which needs to change the background.

### III. METHOD

#### A. Problem Formulation

As basic optical phenomena, reflection and refraction of non-Lambertian objects are ubiquitous in the real world, and at times, such objects can serve as a light probe [15] to measure the environment radiance. Given the distorted imaging on the surface of a transparent or specular object, humans can decompose the radiance based on object geometry, and infer scenes that are occluded or out-of-view. However, for machines, handling refraction and reflection still requires exploration.

In this paper, we explore how to learn environment radiance from reflective or refractive objects. Specifically, we focus on modeling a scene containing a non-Lambertian object using a physically correct representation. Given a set of posed images of the scene, we aim at reconstructing the non-Lambertian object and measuring the environment radiance, which enables

synthesizing novel view images that adhere to the laws of optical imaging, and also benefits many applications requiring exact modeling of reflection or refraction.

Recently, NeRF [48] has shown great success in novel view synthesis, which represents a 3D scene as a radiance field parameterized by a neural network  $f_\theta$ . Specifically, given a 3D coordinate  $x$  and a viewing direction  $d$ :

$$f_\theta(x, d) = (c, \sigma), \quad (1)$$

where  $\sigma$  is a volume density and  $c$  is the corresponding RGB color value  $c$ . Despite NeRF's view-dependent radiance formulation can represent non-Lambertian effects to some extent, NeRF is completely unable to deal with complicated light path changes. Both reflection and refraction are highly sensitive to view direction. For scenes containing non-Lambertian objects, the training images are inconsistent across multiple views, hence NeRF fails in learning correct representation of the scene and the synthesized images are also physically incorrect, as shown in the middle column of Fig. 2 (a).

Therefore, we first estimate the object geometry leveraging a differentiable hybrid shape representation. Then we employ a NGP variant as the baseline to learn environment radiance and propose a physical-based ray-tracing module that models refraction and reflection in a unified manner. With this module, we can successfully synthesize novel images by adjusting the material and geometry of the object, and recover out-of-view scenes from distorted reflection. Since the normal of object surface extracted from the estimated geometry may be inaccurate which leads to high-frequency noise in the rendering results, we also use a supersampling strategy to address aliasing. Fig. 3 presents an overview of our method.

### B. Shape Reconstruction of Non-Lambertian Object

Both refractions and reflections are highly sensitive to object geometry, particularly the surface normals. Thus having an accurate object shape is crucial for image synthesis. However, acquiring the geometry of non-Lambertian objects is quite challenging. Unlike opaque objects, estimating the normals and depths of non-Lambertian objects poses formidable challenges. Traditional 3D sensors, such as RGB-D cameras, are rendered ineffective due to the intricate visual properties on the object's surface, particularly the Fresnel effect. This optical phenomenon leads to distortions in the captured data, rendering it unfeasible to obtain accurate depth and normal maps from materials influenced by this effect. Meanwhile, the appearance of a transparent/specular object is also hard to model without known environment radiance due to the view-dependent effects.

Instead of estimating geometry from view-dependent RGB images or distorted depths, we directly recover the object shape from its multi-view silhouette, which is the most robust information extracted from non-Lambertian objects and can be easily obtained using on-the-shelf instance segmentors[3] as well.

1) *Shape Representation:* To learn the object geometry solely from the object mask, we require a differentiable rendering pipeline capable of rendering the geometry's silhouette.

This allows us to back-propagate the loss and optimize the object geometry using only the object mask loss. To do so, we integrate Deep Marching Tetrahedra[58] (DMTet), a differentiable hybrid surface representation in our geometry estimation pipeline. DMTet represents geometry as a signed distance field (SDF) defined on a deformable tetrahedral grid  $T$ . Each vertices  $v_i$  in the grid  $T$  are characterized by both the SDF value  $S_i \in R$  and the deformation  $\Delta v_i \in R^3$ . The grid is deformed as  $v'_i = v_i + \Delta v_i$  and can leverage all its resolution. Using the differentiable marching tetrahedra, explicit mesh can be extracted by computing the SDF level set conditioned in each deformed tetrahedron  $T_k \in T$ .

Benefited from the differentiable representation, we can render object mask images and back-propagate the loss to optimize the object geometry characterized in the per-vertex SDF value  $s_i$  and deformation value  $\Delta v_i$ . Given the estimated binary object mask  $\hat{M}$  and the ground truth mask  $M$ , the mask loss is formulated as:

$$\mathcal{L}_{\text{mask}} = \left\| M - \hat{M} \right\|_2 \quad (2)$$

2) *Progressive Encoding:* Despite the advantages, directly optimizing the per-vertex SDF values  $S_i$  and deformations  $\Delta v_i$  conditioned in parameter grid can introduce high-frequency noise in the surface mesh (refer to Sec. IV-G3). Compared with the per-vertex parameterized method used in [51], we employ a MLP with fewer frequency encoding to predict the SDF and deformation  $(s_i, \Delta v_i)$  of each tetrahedral vertex can reduce the surface high-frequency noise. However, reducing the surface high-frequency noise may result in the loss of important high-frequency surface details, which are critical for achieving high-quality surface reconstruction.

To strike a balance between reducing high-frequency noise and preserving shape details, we leverage a progressive encoding strategy inspired by SAPE [27], which progressively unveils higher frequency encoding channels as the training iterations increase. With an input position  $p$  and a set of encoding functions  $\{e_n\}_{n=1}^N$ , we applied a set of soft mask  $\{\alpha_n\}_{n=1}^N$  to each encoding. We use the following rule to config  $\alpha_n$  at the time step  $t$ :

$$\alpha_n(t) = \begin{cases} 1 & n \leq n_{\text{base}} \\ \text{clamp}\left(\frac{t-t_s(n)}{t_e(n)-t_s(n)}, 0, 1\right) & n > n_{\text{base}} \end{cases} \quad (3)$$

The lowest  $n_{\text{base}}$  encoding channels are always exposed, while the remaining encoding channels will be gradually revealed during  $[t_s(n), t_e(n)]$ . The values of  $t_s(n)$  and  $t_e(n)$  increase with the frequency  $n$ . Benefiting from the progressive encoding, our model is capable of learning the lower frequency features at earlier stages and higher frequency details later on, which significantly reduces the high-frequency noises.

3) *Mesh Extraction:* Benefit from the differentiable rendering pipeline, the implicit SDF based representation can be converted into explicit mesh using the marching tetrahedra algorithm[58]. Specifically, with vertex SDF value  $s_i$  and deformed vertex position  $v'_i$ , the surface can be detected based on the signs of  $s_i$ . For an edge  $e$  with 2 endpoints  $v_i$  and  $v_j$ , if  $\text{sign}(s_i) \neq \text{sign}(s_j)$ , the position with zero SDF value can

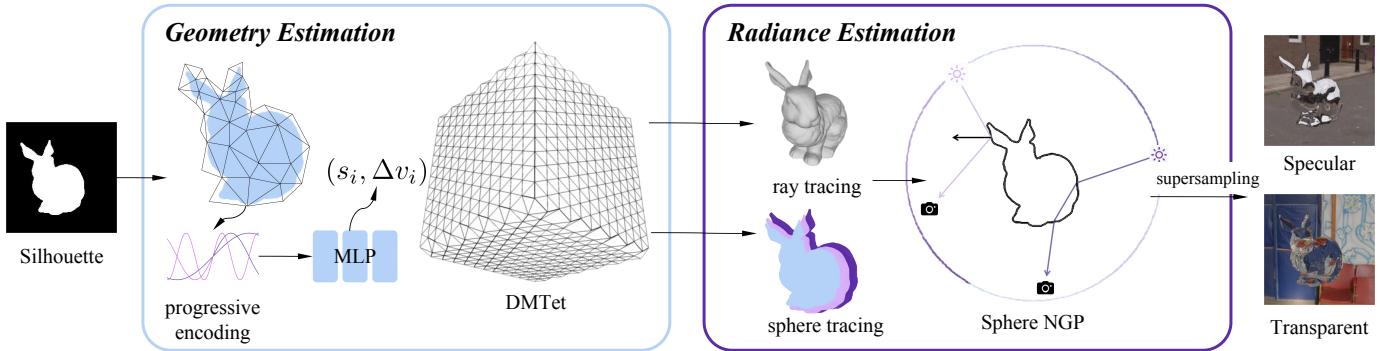


Fig. 3. Overall architecture. (a) Geometry estimation. With only the object mask as prior, we employ Deep Marching Tetrahedra[58], a hybrid shape representation to reconstruct the geometry of the object, in which we progressively encode multilayer perceptrons (MLP) to predict the signed distance field (SDF) and per-vertex offsets defined on a grid for both smoothness and high-frequency details. (b) Radiance estimation. We propose Sphere-NGP, a direction-agnostic NGP model for radiance estimation. When a ray intersects with an object, we use the Fresnel equation to calculate the outgoing direction and radiance of the light. Supersampling is also used to remove high-frequency artifacts.

be calculated using linear interpolation:  $\mathbf{v}_0 = \frac{\mathbf{v}'_{sj} - \mathbf{v}'_{si}}{s_j - s_i}$ . For each point  $v_0 \in V_0$  with the interpolated zero SDF value, we also applied IGR regularization to these points:

$$\mathcal{L}_{\text{eikonal}} = \mathbb{E}_{v_0 \in V_0} (\|\nabla f_\theta(v_0)\|_2 - 1)^2. \quad (4)$$

The importance of the Eikonal loss in regularizing the SDF field has been demonstrated in previous studies, as shown in [22]. When using the grid-based SDF values and deformations  $(s_i, \Delta v_i)$ , it is possible to calculate the SDF value of surface points through interpolation. However, this approach hinders the calculation of gradients for these surface points, thus preventing the application of the Eikonal loss. As a result, high-frequency noise on the object's surface can be observed, as illustrated in Fig. 13. Indeed, it's a crucial factor contributing to the excellent performance of our geometry reconstruction method.

The overall loss function in this stage is defined as the weighted sum of individual loss with  $\lambda$  representing the weight terms:

$$\mathcal{L} = \lambda_{\text{mask}} \mathcal{L}_{\text{mask}} + \lambda_{\text{eikonal}} \mathcal{L}_{\text{eikonal}}. \quad (5)$$

### C. Illumination Estimation with Refractive-Reflective Field

With the estimated shape in the first stage, we can successfully disentangle of object's geometry and appearance. Based on this, in this section, we focus on measuring the environment radiance from reflection/refraction with a neural radiance field.

Both synthesised and real world datasets are used in the experiment, while the synthesised images are generated by a non-Lambertian object and an HDRI lighting environment through Blender [7]. Following many works that use environment maps as the illumination representation[8], [38], we assume that the environment is infinitely far away from the object, so the radiance of the object's surface only depends on the direction  $d$  of the emergent ray in this scenario.

Targeted at estimating the environment map, we propose a grid-based representation named **Sphere-NGP**, which is a variation of Instant-NGP [50]. Given a ray  $r(t) = \mathbf{o} + td$ ,

Sphere-NGP takes only the view direction as input and encodes the direction to a trainable grid feature, then the feature is mapped to its radiance using an MLP  $f_\theta$ , which is formulated as:

$$\mathbf{c}(\mathbf{r}) = f_\theta(e(\mathbf{d})). \quad (6)$$

Where  $e(\cdot)$  is the multi-resolution hash encoding. Leveraging the advantages of the multi-resolution hash map, Sphere-NGP demonstrates the ability to render more high-frequency details within a shorter time. For additional benchmark results, refer to Sec. IV-G1 .

For each pixel in the image, the incident ray  $r(t) = \mathbf{o} + td$  passing through that pixel can be derived through camera calibration, where  $\mathbf{o}$  is the camera origin and  $\mathbf{d}$  is the corresponding view direction. Certain rays will intersect with the object's surface, causing them to undergo light path deflections. Fig. 2 (a) depicts the alterations in the light path when a ray traverses a refractive object, in which employing the incident ray directly could result in incorrect estimation of the environment radiance. Instead, we model reflection and refraction of rays through a physically-based ray-tracing module, and the proposed field which has a ray-tracing module is named as the **Neural Refractive-Reflective Field (NeRRF)**.

Given the geometry and material of a non-Lambertian object and the incident ray  $r(t)$ , we first calculate their closest intersection point  $\mathbf{p}$  and its corresponding normal  $\mathbf{n}$ . The estimated objects surface can be represented by either an mesh extracted from DMTet or by configuring the SDF level set through an implicit neural network  $f_\theta(\mathbf{p})$ . For the explicit representation, we employ ray tracing based on ray-mesh intersection to determine if the ray hits the mesh surface. And for the SDF-based representation, sphere tracing is applied to detect the intersection points alternatively.

The normal of intersection point is also crucial for confirming the outgoing direction. In the case of the mesh representation, the intersection point's normal is interpolated from the intersecting faces for smoothness. When employing the SDF representation with an MLP, the surface normal can be obtained through its first-order derivative:  $\mathbf{n} = \nabla_p f_\theta$ .

With the intersection point  $\mathbf{p}$  and corresponding normal  $\mathbf{n}$ , the outgoing ray direction  $\mathbf{d}_t$  on the surface of transparent object is determined by the Snell's law:

$$\cos\theta_o = \sqrt{1 - \lambda^2(1 - \cos^2\theta_i)}, \quad (7)$$

$$\mathbf{d}_t = -\lambda\mathbf{d} + (\lambda(\mathbf{d} \cdot \mathbf{n}) - \cos\theta_o)\mathbf{n}, \quad (8)$$

where  $\lambda$  is the index of reflection,  $\theta_i$  and  $\theta_o$  are incident angle and emergent angle respectively. The direction of reflected ray  $\mathbf{d}_r$  is formulated as:

$$\mathbf{d}_r = \mathbf{d} + 2\cos\theta_i\mathbf{n}. \quad (9)$$

For rays that hit the surface, we replace the incident ray  $r(t) = \mathbf{o} + t\mathbf{d}$  with either the reflected ray  $r_r(t) = \mathbf{p} + t\mathbf{d}_r$  or the refracted ray  $r_t(t) = \mathbf{p} + t\mathbf{d}_t$  at the intersection points, contingent upon the object's material. Then we proceed to trace the new ray until there are no new intersection points with the object. In practice, constrained by GPU memory limitations, we consider light paths with a maximum of two bounces for simplification.

When light strikes a refractive surface, both reflection and refraction of the light may occur, causing a difference between the radiance of the incident ray and the emergent ray. Fresnel equations provide the ratio of the reflected radiance to the incident radiance, characterized by a Fresnel term  $F$ :

$$F = \frac{1}{2} \left( \frac{\mathbf{d}_i \cdot \mathbf{n} - \lambda \mathbf{d}_t \cdot \mathbf{n}}{\mathbf{d}_i \cdot \mathbf{n} + \lambda \mathbf{d}_t \cdot \mathbf{n}} \right)^2 + \frac{1}{2} \left( \frac{\lambda \mathbf{d}_i \cdot \mathbf{n} - \mathbf{d}_t \cdot \mathbf{n}}{\lambda \mathbf{d}_i \cdot \mathbf{n} + \mathbf{d}_t \cdot \mathbf{n}} \right)^2 \quad (10)$$

$$R_r = F \cdot R_i, \quad R_t = (1 - F) \cdot R_i \quad (11)$$

where  $R_i$ ,  $R_r$ , and  $R_t$  denote the radiance of incoming, reflected, and refracted rays respectively. We consider the Fresnel effect only with the first intersection of the refractive surface. For reflective surfaces, we set  $F$  to 1 directly, adhering to the principle of energy conservation. The ultimate radiance with the incident ray  $r(t)$  intersecting with an object is:

$$c(r) = F \cdot c(r_r) + (1 - F) \cdot c(r_t) \quad (12)$$

For those incident rays that do not intersect with the non-Lambertian object, we directly follow Eq. 6 without any further processing. The training objective in this stage is formulated as:

$$\mathcal{L}_{RGB} = \frac{1}{|R|} \sum_{r \in R} \|c(r) - \hat{c}(r)\|^2 \quad (13)$$

where  $R$  denotes a set of rays, and  $\hat{c}(r)$  is the ground-truth RGB value of ray  $r$ .

**Point-based refractive-reflective fields.** The radiance representation introduced by NeRRF, as expressed in Eq. 6 is compatible with the dataset setting that uses an environment map as the illumination representation. In some cases of real-world dataset where the environment cannot be treated as infinitely distant, a radiance representation incorporating point-based considerations is necessary. Hence we introduce a variant NeRRF named NeRRF-P(point), which adopts the point-based radiance field based on Instant-NGP [50]. The radiance formulation of NeRRF-P can be expressed as:

$$c(r) = f_\theta(e(\mathbf{x}), \mathbf{d}). \quad (14)$$

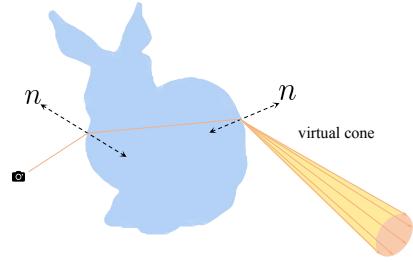


Fig. 4. Virtual cone supersampling illustration. After the ray is emitted from the object's surface, we employ a virtual cone for supersampling, enabling multiple sampling and anti-aliasing.

where  $x$  is the position of the sampling points on the ray, and  $d$  is the viewing direction. NeRRF-P also incorporates the ray-tracing/sphere-tracing module used in NeRRF, for dealing with the real world scenario.

#### D. Anti-aliasing with Supersampling

Given that the geometry estimation stage lacks direct normal supervision or a consistency constraint, focusing only on minimizing the mask loss as depicted in Eq. 2, the resulting reconstructed surface normals are not as accurate. This deficiency in accuracy translates to a lack of local coherence in the normals, consequently introducing undesirable high-frequency noise onto the surface. To reduce surface high-frequency artifacts, we model a ray as a light cone when the ray is shot out of the object's surface, as illustrated in Fig. 4. The radiance of the object's surface then becomes the integration of all incident radiance in the virtual cone, which can also reduce the aliasing. For the rays that do not intersect with the object, their accuracy is not affected by the inaccurate surface normals, for which we keep using Eq. 6 as the radiance formulation.

The axis of the virtual cone is the outgoing direction  $\mathbf{d}$ . In the cone, we randomly sample  $N$  directions on the uniform distribution of angle and obtain the corresponding sampled radiance as Eq. 6. To penalize rays that deviate significantly from the axis, we introduce a confidence index  $conf$  to quantify the reliability of each ray. Within the cone, we assume that the angle between sample direction and the cone axis adhere to a Gaussian distribution  $N(0, \sigma)$ , with  $\sigma$  being a hyperparameter. The confidence index  $conf$  is defined as the probability density function (pdf) value of the Gaussian distribution corresponding to the given deviation angle. The ultimate radiance of outgoing rays  $r_o$  is then represented as the weighted sum of radiance values for each ray, with the weight determined by the corresponding confidence index  $conf$ . The final radiance of the outgoing ray  $r'(t)$  is formulated as:

$$c(r') = \frac{\sum_{i=1}^{N_s} conf_i \cdot f_\theta(\gamma(\mathbf{d}_i))}{\sum_{i=1}^{N_s} conf_i}, \quad (15)$$

$\mathbf{d}_i$  is the sampled directions in the virtual cone. It is noteworthy that employing supersampling on the vanilla NeRF leads to increased complexity, while the complexity of our ray sampling strategy is the same as that of the vanilla NeRF, since we do not sample points.

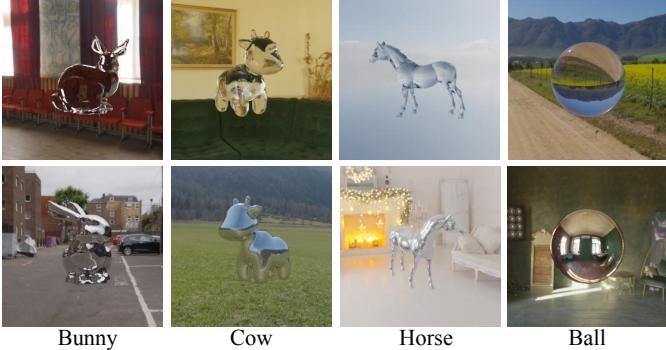


Fig. 5. Illustration of our Blender Synthetic Dataset.

## IV. EXPERIMENT

### A. Experiment protocol

1) *Datasets*: Our synthetic dataset is generated using Blender's physics engine [7] and physically-based ray-tracing Blender Cycles rendering engine, which is capable of simulating the physically accurate non-Lambertian effects exhibited by objects, including refraction, reflection, and Fresnel effects. We use 4 objects including cow, horse, bunny and ball. Each of these objects is utilized twice, once to represent specular properties and once for transparent properties. These objects are positioned within eight scenes derived from PolyHaven[2]. The index of refraction (IOR) for reflective materials is set within the range of 1.1 to 1.5. The generated images are illustrated in Fig. 5.

We conducted sampling of 89 camera poses on the upper hemisphere around the object. For each sampled camera pose, we gathered ground truth data that encompassed: (1) monocular RGB rendering, (2) segmentation mask of the non-Lambertian object, (3) the camera's pose, and (4) object material details, including whether the object is refractive or reflective and its corresponding index of refraction (IoR).

2) *Evaluation Details*: We conducted a comprehensive benchmark of our approach on the synthesized Blender dataset, utilizing a subset of the generated data for both training and testing. Specifically, we employed 50 images for training and 39 images for testing within each scene. During the geometry estimation stage, we assessed reconstruction quality using the Chamfer Distance (CD) between the predicted surface point cloud and the ground truth surface point cloud. For the environment estimation task, we employed standard image quality metrics, including PSNR, SSIM[71], and LPIPS[85] for all quantitative assessments to gauge the performance.

3) *Baselines*: We compare our method with several state-of-the-arts surface reconstruction methods, including IDR [80], PhySG [84], NDR [51], and NeRO [40] for geometry estimation task. For the novel view synthesis task, we employ NeRF [48] and NeRO [40] as our baseline methods. Note that we did not compare with some previous arts working on transparent objects mentioned in related works, mainly because they are not open-source [68] or have significant differences in experiment settings [37], [76] (requiring specific backgrounds, etc.).

### B. Implementation and training details

We conducted our implementation using the PyTorch[56] framework and employed a single NVIDIA GeForce RTX 3090 GPU for all experiments.

1) *Geometry estimation details*: In the stage of geometry estimation, we trained each object with the progressive encoding applied. We applied Fourier frequency encoding with 6 encoding channels, the input  $x$  and the lowest encoding channel comprising  $\sin(x), \cos(x)$  were constantly enabled. The remaining channels were progressively activated in 500 iteration following the Eq. 3. We employed the Adam optimizer with the learning rate of 0.001 to train the objects. We trained each objects for 3 epochs, with each epoch comprising 3000 iterations. For objects with more high-frequency details like the bunny, cow, and horse, we set the eikonal loss coefficient ( $\lambda_{eikonal}$ ) to 0.01 for the first two epochs and 0.001 for the subsequent 3000 iterations. In the case of the ball, which exhibits fewer high-frequency details,  $\lambda_{eikonal}$  was set to 0.1 for the first two epochs and 0.01 for the remaining epoch. For the baseline methods[40], [84] and [80] and [51], we adhered to their original training configurations.

2) *Radiance estimation details*: In the training of Sphere-NGP, we employed the Adam optimizer with a learning rate of 0.01. Each scene underwent four epochs, and each epoch involved 25,000 iterations. For training the NeRF backbone, we used the Adam optimizer with a learning rate of 0.005. Each object went through eight epochs, with each epoch comprising 25,000 iterations. This extensive training was necessary as NeRF can be harder to converge. Regarding other methods [80], [84], and [40], we adopted their original settings.

Additionally, we set the parameter  $\sigma$ , which determines the cone apex angle used for supersampling, to different values. For objects with specular surfaces (bunny, cow, horse), the apex angle was set to 20 degrees, while for the specular ball, it was set to 10 degrees. For transparent objects (bunny, cow, horse), the value was set to 10 degrees, and for the transparent ball, it was 5 degrees.

Our framework supports both SDF-based sphere tracing and mesh-based ray tracing. Since the mesh is extracted from DMTet during the geometry estimation stage, there is minimal difference in rendering results between the two methods. Notably, SDF-based sphere tracing offers significantly faster rendering performance.

### C. Geometry Reconstruction

We evaluate our capacity of geometry estimation for transparent/specular objects compared with previous arts, as shown in Tab. II. NeRRF's hybrid geometry representation demonstrates stable performance on four different objects and performs best in the average chamfer distance. This verifies our method's robustness, which is because our input only depends on the silhouettes and does not require other view-dependent inputs. The silhouette defines the boundaries of an object, which is crucial for reconstructing the object's shape, meanwhile, it is not affected by the material and appearance of the object. This allows our method to handle objects of different materials, including reflective and refractive, or even

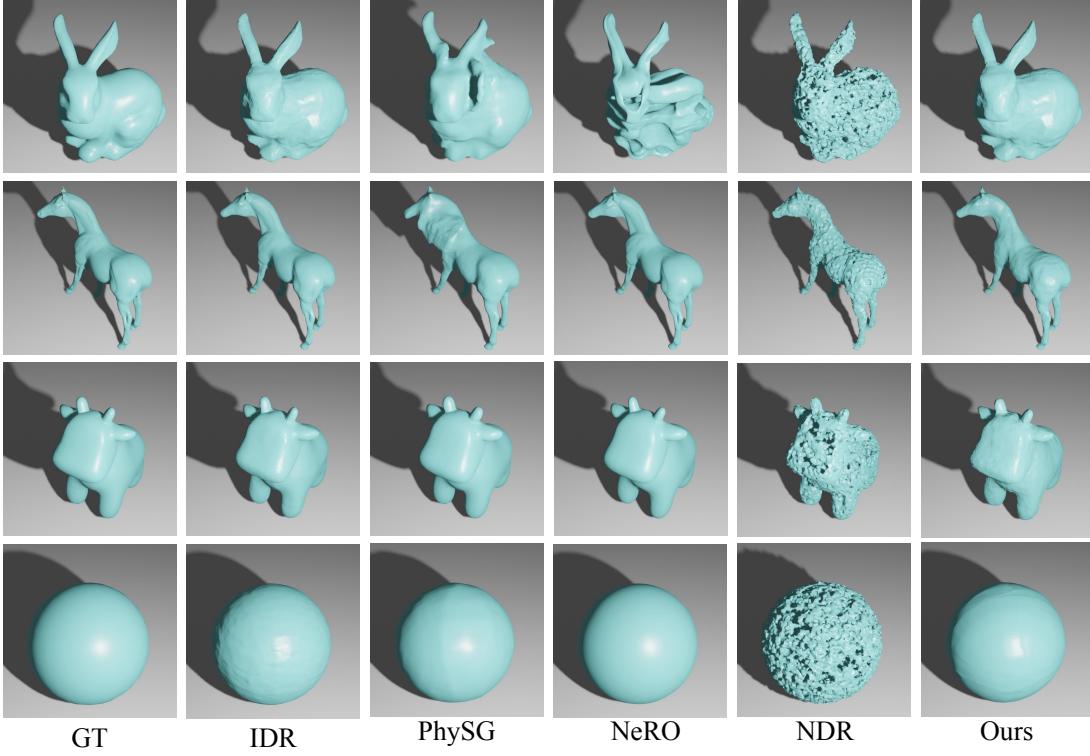


Fig. 6. Ground-truth and reconstructed surfaces of the Blender synthetic dataset. Our results are compared with IDR[80], PhySG[84], NeRO[40], and NDR[51]. Note that NeRO and NDR are only capable of reconstructing glossy surfaces, so we only include reflective objects for the two methods.

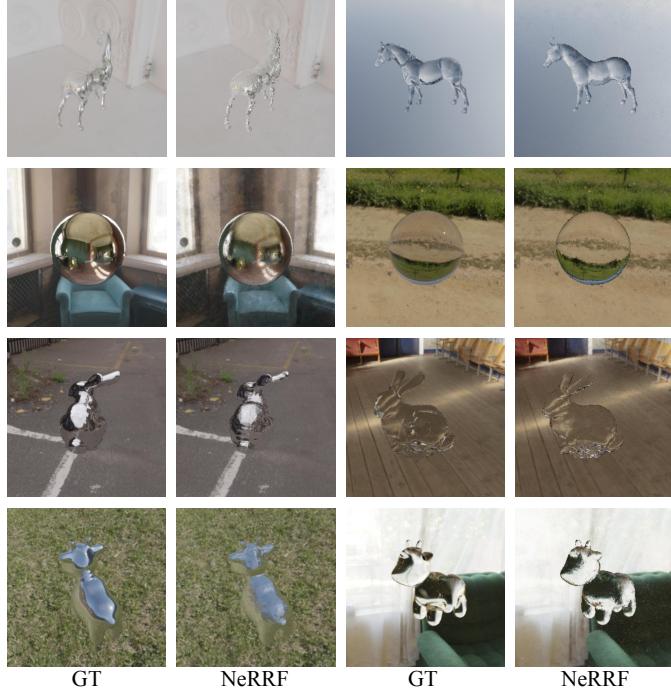


Fig. 7. More qualitative results of novel view synthesis.

be generalized to opaque objects, while NeRO and NDR are only capable of modeling glossy surfaces, as illustrated in Fig. 8 (a). In addition, Fig. 6 provides qualitative comparisons with other methods. As shown in the figure, compared to

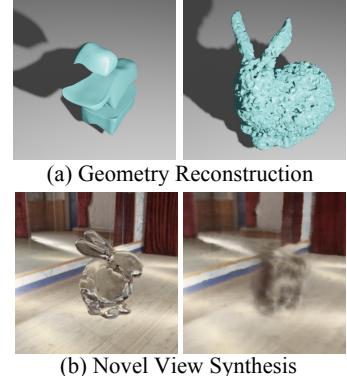


Fig. 8. Existing method limitation Existing method are unable to accurately reconstruct the geometry of transparent objects and generate novel view images. (a) Geometry Estimation The left side shows the reconstructed surface of a transparent cow, learned using NeRO, while the right side displays the surface of a transparent bunny, learned using NDR. (b) Novel View Synthesis The left side is the ground truth novel view image of transparent bunny while the right side shows the predicted image generated by NeRO.

ground truth, our method can accurately reconstruct the shape of objects with enough details. Besides, NeRO can perfectly reconstruct the reflective surfaces of the horse, cow, and ball in our blender synthetic dataset. However, it fails when dealing with the bunny. This also demonstrates that the mask may be vital for the geometry estimation of some non-Lambertian objects since NeRO doesn't require object masks as input.

TABLE II  
COMPARISON OF GEOMETRY RECONSTRUCTION METHODS

Method	Bunny	Cow	Horse	Ball	Avg
IDR[80]	<b>0.0032</b>	<u>0.0037</u>	0.0005	<u>0.0035</u>	0.0027
PhySG[84]	0.0209	0.0047	0.0171	<u>0.0035</u>	0.0116
NeRO[40]	0.0313	<b>0.0024</b>	<b>0.0002</b>	<b>0.0033</b>	0.0093
NDR[51]	0.0665	0.0555	0.0151	0.2339	0.0928
NeRRF(ours)	<u>0.0038</u>	<b>0.0024</b>	<u>0.0003</u>	0.0037	<b>0.0025</b>

We report the chamfer distance on ground truth mesh and extracted meshes on the blender synthetic dataset as a quantitative measure for reconstructed geometry quality.

TABLE III  
QUANTITATIVE RESULTS FOR NOVEL VIEW SYNTHESIS

Method	PSNR↑	Reflective		Refractive		
		SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
NeRF [48]	23.536	0.685	0.237	26.183	0.764	0.217
NeRFRO [54]	-	-	-	23.765	0.756	0.379
NeRO [40]	<b>30.839</b>	0.805	0.196	-	-	-
NeRRF	29.655	<b>0.901</b>	<b>0.071</b>	<b>28.780</b>	<b>0.862</b>	<b>0.148</b>

We compare our method with NeRF [48], NeRO [40] and NeRFRO [54].

#### D. Novel View Synthesis

As IDR and PhySG do not inherently model the background, we employed them for a qualitative evaluation of novel view synthesis without including the background. While IDR[80] and PhySG[84] can estimate the geometry of transparent objects, they fail to accurately model refraction effects and instead represent Fresnel effects as diffuse colors on the object’s surface, as showcased in Fig. 10.

We also conducted a benchmark of novel view images with backgrounds, utilizing the NeRF[48] and NeRO[40] as baselines for scenes with reflective objects, and NeRF and NeRFRO[54] for scenes with refractive objects. Notably, we did not assess NeRO on the refractive objects since it is designed specifically for generating novel view images of reflective objects and lacks the capability to model refractive effects, as illustrated in Fig. 8(b). Similarly, NeRFRO[54] was only evaluated on the reflective dataset for the same reason. The quantitative results are presented in Tab. III.

The quantitative results in Tab. III demonstrate that our method excels in synthesizing novel view images when compared to the two baseline methods, both for reflective and refractive object datasets. These results highlight the superior performance of our approach in handling the Fresnel effect. Our evaluation reveals that while NeRO[40] exhibits a stronger ability to model the geometry of reflective cow, horse, and ball, it faces challenges when dealing with the geometry of the bunny, resulting in less accurate reconstruction quality in comparison to our method. This observation highlights the robustness of our approach, particularly in scenarios involving complex object shapes or the presence of implicit boundaries between objects and the background. For the refractive objects, our method also outperform the baseline methods both qualitatively and quantitatively, as illustrated in Fig. 10 and Tab. III.

TABLE IV  
ESTIMATED ENVIRONMENT RADIANCE COMPARISON

Method	PSNR↑	SSIM↑	LPIPS↓
PhySG [84]	13.030	0.420	0.550
NeRRF	<b>16.468</b>	<b>0.479</b>	<b>0.433</b>

We quantitatively compare the estimated environment radiance (learned solely by the object surface radiance) with the ground truth environment radiance.

#### E. Environment Radiance Estimation

Smooth glossy surfaces have the potential to function as visual sensors, enabling the observation of scenes out of individual’s field of view. This has numerous practical applications in daily life, including tasks like estimating the environment behind a car through its rear-view mirror. Benefit from the decomposed geometry and radiance, NeRRF can effectively reconstruct scenes based on the outgoing radiance from the object surfaces. To validate this, we conduct an environment radiance estimation experiment, in which we mask out the background RGB information and learn the environment radiance solely from the object surface radiance. The input is illustrated in Fig. 11.

We compare NeRRF with PhySG, which can also learn the environment radiance from only the reflective surface, and the qualitative and quantitative results are shown in Fig. 11 and Tab. IV respectively. The performance of NeRRF outperforms PhySG in all the metrics as shown in the table. Besides, the visualizations in Fig. 11 demonstrate that NeRRF can successfully estimate the environment map using only the reflection on the ball, while the result of PhySG is blurry. This is because our physically-based ray tracing module can accurately model the environment radiance, while PhySG approximates the environment map using Spherical Gaussian coefficients which are limited in frequency.

#### F. Scene Editing and Relighting

Thanks to the disentanglement of object geometry and environment illumination in our approach, our method also possesses the capability to perform modifications on the input multi-view images. To demonstrate the practicability of our model, we conduct experiments on a set of image editing and relighting tasks. The qualitative results are presented in Fig. 12. Note that our rendered images are synthesized by the NeRRF renderer, while the reference images are the ones in the original scene which has the same camera calibration as the rendered images (not in the training set), and the ground-truth targets are generated using the Blender Cycles engine by directly editing the scene in Blender.

1) *Relighting*: NeRRF enables controllable light by disentangling object material, geometry, and environment illumination. To validate this, we also demonstrate relighting results in Fig. 12(a). By changing the background and maintaining the optical properties of the objects, our method can directly render relighting results of transparent or glossy objects, which is achieved by changing the NeRRF model to the one trained with target lighting.

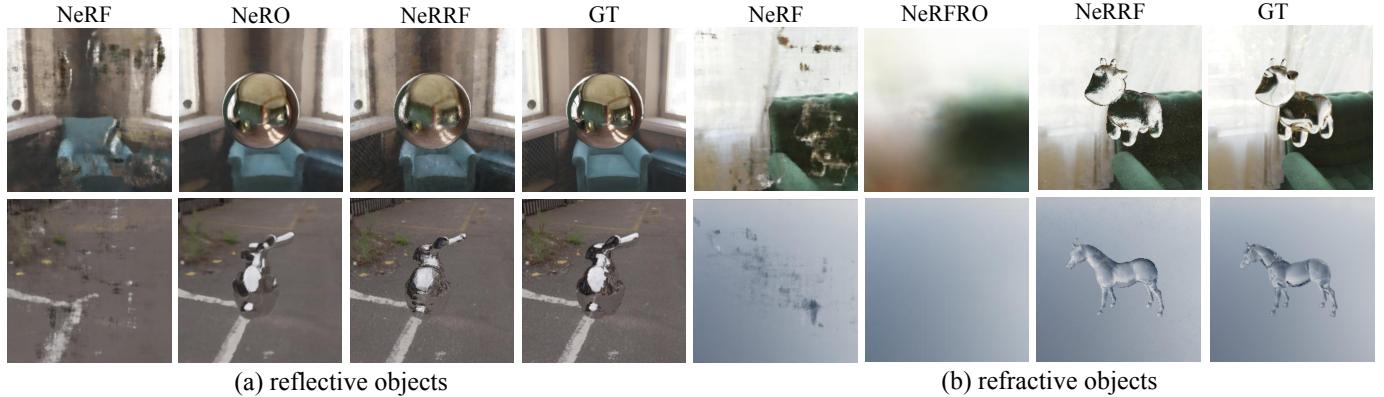


Fig. 9. Ground-truth and predicted novel view of all objects. The reflective objects are compared with NeRF[48] and NeRO[40] while the refractive objects are compared with NeRF[48] and NeRFRO[54].

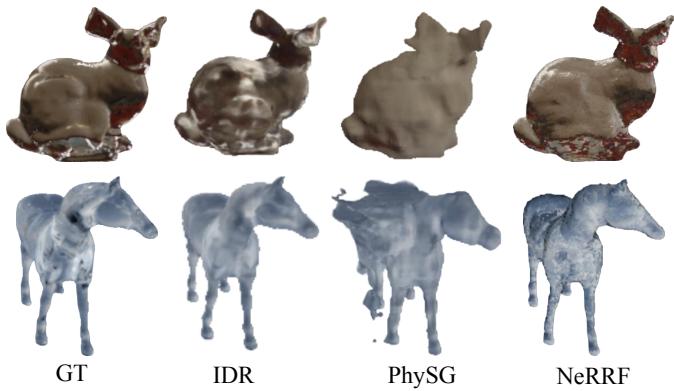


Fig. 10. Ground-truth and predicted novel view of refractive objects. Our results are compared with IDR[80] and PhySG[84].

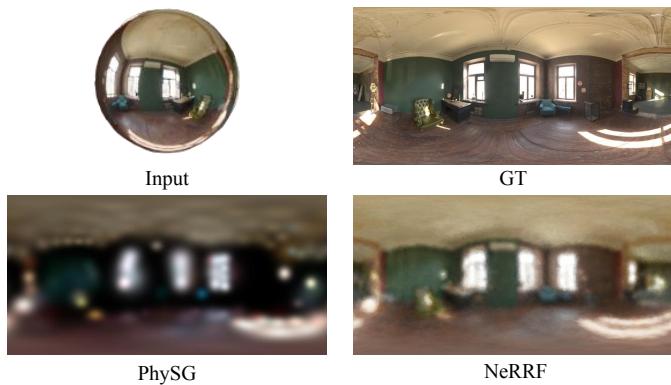


Fig. 11. Environment radiance estimation results. We generate environment map learned solely from the surface radiance of reflective ball. The result is compared with PhySG[84].

2) *Material editing*: Editing material properties of an object has applications for 3D shape design and visualization. We achieve material editing by straightforwardly changing the object material parameters involved in the ray-tracing process, and the results are provided in Fig. 12 (b). In the first row, We edit the material of a horse from reflective to transparent, and in the second row, we change the object's index of refraction from 1.1 to 1.2. The generated image is basically consistent

with the ground truth and conforms to the laws of optics, while the editing process is very simple.

**3) Object replacement:** Object replacement is a critical application for high-quality augmented reality. Given a set of posed images of the scene containing a non-Lambertian object, we aim at replacing the object with a novel one, and rendering images with physically-correct light transport effects. To achieve this, we first learn the environment radiance using our model with the given set of images, and then replace the object geometry used in the ray-tracing process with the geometry of the novel object. Fig. 12 (c) provides the qualitative results of object replacement, in the first row, we replace the metal bunny with a mental cow, and in the second row, we replace a transparent ball with a transparent bunny. As illustrated in the figure, our approach successfully models the non-Lambertian effects of the novel object and generates photorealistic renderings.

**4) Object insertion:** Considering the time costs of the ray-tracing process, we conduct all the above experiments on scenes that consist of only one non-Lambertian object, but this doesn't mean our method cannot be applied to multi-object scenes. We carry out an experiment to insert a novel non-Lambertian object, which is achieved by providing the geometry and material of the new object for the ray-tracing module. As demonstrated in Fig. 12 (d), we insert a reflective bunny into a scene with a reflective ball in the first row, while in the second row, we insert a transparent horse into the scene. The results reveal that our model is capable of modeling multiple non-Lambertian objects with high fidelity, which can benefit a variety of VR/AR applications.

## G. Ablation on Network Architecture

*1) Shape representation:* With the Deep Marching Tetrahedral(DMTet) as our shape representation, we embarked on two implementations: a MLP-based one and a grid-based one. In the grid-based scheme, vertex SDF values and deformations ( $s_i, \Delta v_i$ ) utilized in [58] do not allow for the interpolation of SDF level sets from the four tetrahedron vertex SDF values and the calculation of SDF gradients, preventing the application of the eikonal loss. Hence, in the ablation study comparing the MLP-based representation with

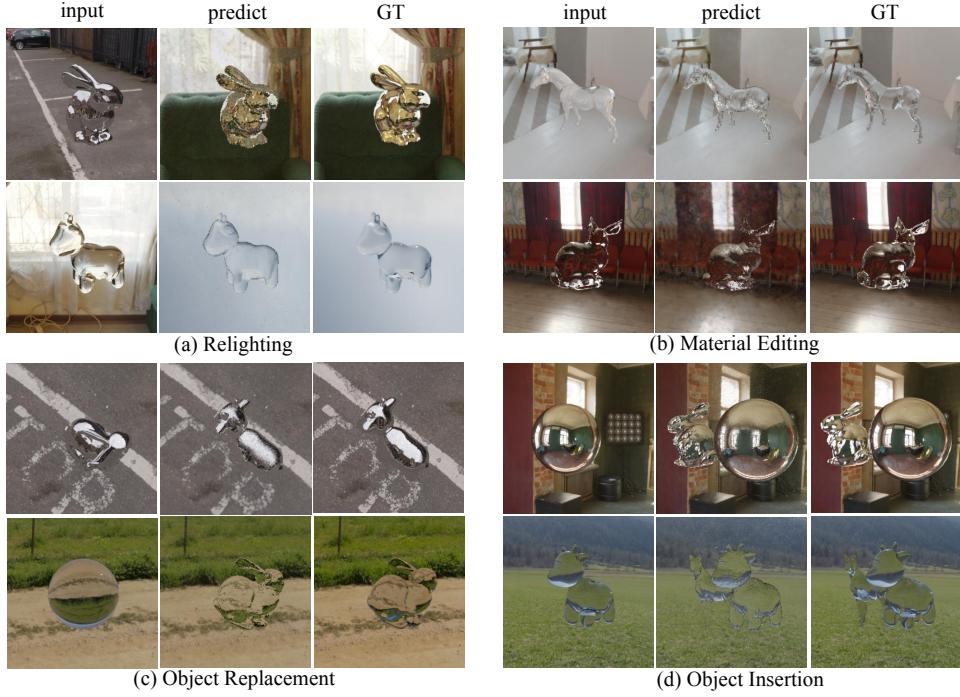


Fig. 12. Relighting and Editing Result. (a) Relighting We demonstrate the ability to change the background of a glossy bunny and a transparent cow. The new backgrounds are also rendered using NeRRF renderer. (b) Material Editing We showcase material editing by transforming a glossy horse into a transparent horse and changing the index of refraction (IOR) of a transparent bunny. These changes are achieved by editing the material properties in NeRRF. (c) Object Replacement We illustrate object replacement by transforming the glossy bunny into a glossy cow and the transparent ball into a transparent bunny, respectively. (d) Object Insertion We demonstrate object insertion by adding a bunny and a horse to two different scenes.

TABLE V  
ABLATION STUDY OF MLP-BASED DMTET.

Method	Bunny	Cow	Horse	Ball	Avg
Grid-based	0.0393	0.0341	0.0085	0.1611	0.0608
MLP-based	<b>0.0032</b>	<b>0.0024</b>	<b>0.0003</b>	<b>0.0037</b>	<b>0.0025</b>

We report the chamfer distance on ground truth mesh and extracted meshes as a quantitative measure for reconstructed geometry quality. The MLP-based method use a MLP to predict vertex SDF values and deformations  $(s_i, \Delta v_i)$ , while the grid-based method store them in a vertex grid array.

the grid representation, we opted not to apply the eikonal loss to the grid-based approach. All other settings remain the same. Tab. V and Fig. 13 presents the quantitative and qualitative results of the two implementations. As shown in Fig. 13, grid-based representation tends to generate surfaces riddled with holes, leading to a significantly higher chamfer distance compared to the MLP-based one. The presence of holes is also observed in the quantitative results for NDR[51] of Fig. 6. These holes persist even when relying on unreliable surface color supervision, indicating that the issue cannot be solely attributed to the inadequacy of color supervision. This underscores the critical role played by the MLP-based DMTet framework and the inclusion of the eikonal loss within our specific task domain.

2) *Radiance representation*: To evaluate the effectiveness of Sphere-NGP, we compare the rendering result on novel view synthesis using Sphere-NGP with two baselines: (1) NeRF-D: a variant NeRF with only direction as input and incorporates with our ray-tracing module; (2) NeRRF-P: a

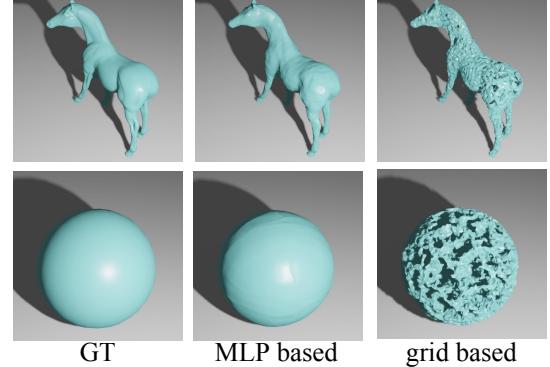


Fig. 13. Qualitative results on the ablation study of MLP-based DMTet. MLP-based method use a MLP to predict vertex SDF values and deformations  $(s_i, \Delta v_i)$ , while the grid-based method store  $(s_i, \Delta v_i)$  in a vertex grid array.

point-based reflective-refractive field with Instant-NGP as the backbone (not incorporates with the supersampling module due to memory efficiency). The experiment is conducted on two representative scenes: the reflective horse and the refractive cow, and the results are provided in III. As shown in the table, NeRRF outperforms NeRF-D with a significant margin, due to the multi-resolution hash encoding. It also achieves better performance compared with NeRRF-P. This is because point-based sampling is more difficult to converge, and the density learned is usually inaccurate, while our direction-only formulation Eq. 6 is more compatible with the Blender dataset. Nonetheless, NeRRF-P is still effective in more complicated

TABLE VI  
ABLATION STUDY OF THE SPHERE-NGP

Method	Reflective Horse			Refractive Cow		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
NeRF-D	32.619	0.931	0.077	24.574	0.780	0.185
NeRRF-P	32.160	0.934	0.091	24.604	0.761	0.168
NeRRF	<b>35.103</b>	<b>0.954</b>	<b>0.068</b>	<b>25.465</b>	<b>0.798</b>	<b>0.181</b>

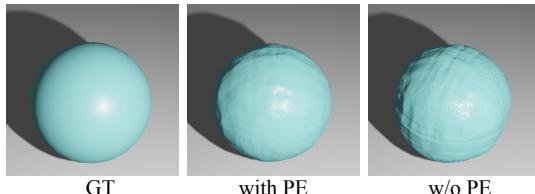


Fig. 14. Qualitative results on the effects of progressive encoding. 'PE' denotes the progressive encoding.

scenes where the environment cannot be seen as infinitely far, and our ray-tracing module serves as a plug-and-play module that can be applied to any NeRF-based method.

3) *Progressive encoding*: We conduct an ablation study to investigate the role of progressive encoding. Because the encoding channels are fully unveiled after 500 iterations in our experiment setting, we train each object for 1000 iterations to assess the impact of progressive encoding. As the quantitative results presented in Tab. VII indicate, the incorporation of progressive encoding leads to smaller chamfer distances between the predicted geometry and the ground truth geometry, demonstrating the effectiveness of progressive encoding. Additionally, visual inspection in Fig. 14 reveals that the sphere's surface appears smoother when progressive encoding is applied, in contrast to the version without progressive encoding. This indicates that progressive encoding facilitates the early learning of low-frequency features, which in turn aids in the removal of high-frequency noise.

#### H. Ablation on Supersampling

Tab. VIII investigates the effects of our supersampling strategy. For comparison, we train a model without supersampling module which is denoted as "w/o Supersampling". As Tab. VIII shows, the model with our supersampling strategy outperforms the "w/o Supersampling" in all scenes, which indicates that supersampling does help the learning of the scene representations. Fig. 15 provides the qualitative results of supersampling. As shown in the figure, the rendering result of w/o supersampling is noisy on the horse's surface, which is caused by the unsmooth surface normal extracted from

TABLE VII  
ABLATION STUDY OF PROGRESSIVE ENCODING.

Method	Bunny	Cow	Horse	Ball	Avg
w/o progressive encoding	0.0323	0.0227	0.0094	0.0047	0.0172
w/ progressive encoding	<b>0.0100</b>	<b>0.0213</b>	<b>0.0085</b>	<b>0.0040</b>	<b>0.0110</b>

We report the chamfer distance on ground truth mesh and extracted meshes as a quantitative measure for reconstructed geometry quality.

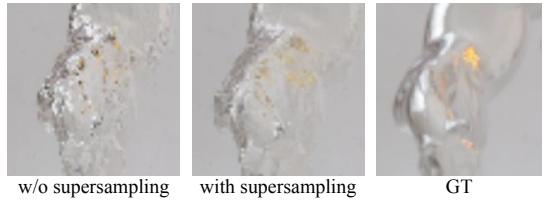


Fig. 15. Qualitative results on the effects of supersampling.

TABLE VIII  
ABLATION STUDY OF SUPERSAMPLING.

Object	w/o Supersampling			w/ Supersampling		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
Bunny-s	25.738	0.874	0.076	<b>26.583</b>	<b>0.878</b>	<b>0.064</b>
Cow-s	28.407	0.866	0.076	<b>29.254</b>	<b>0.878</b>	<b>0.057</b>
Horse-s	33.325	0.949	0.079	<b>35.102</b>	<b>0.954</b>	<b>0.068</b>
Ball-s	26.651	0.889	0.073	<b>27.680</b>	<b>0.894</b>	<b>0.090</b>
Bunny-t	28.132	0.859	<b>0.076</b>	<b>28.606</b>	<b>0.861</b>	0.142
Cow-t	24.779	0.796	<b>0.144</b>	<b>25.465</b>	<b>0.798</b>	0.181
Horse-t	30.169	<b>0.920</b>	0.185	<b>30.397</b>	<b>0.920</b>	<b>0.183</b>
Ball-t	30.250	0.862	0.082	<b>30.650</b>	<b>0.869</b>	<b>0.081</b>

the predicted geometry. Instead, the result with supersampling becomes significantly smoother. This is because the radiance on the object's surface is the average radiance sampled in the virtual cone, which is equivalent to averaging the surface normal under a certain distribution.

#### I. Real-world Application

To further validate the effectiveness of NeRRF, we also applied it into a real-world dataset which is provided by [6]. Considering the environment illumination settings in real-world scenarios, we use NeRRF-P for the experiment. The qualitative novel view images are given in Fig. 17. In this experiment, we access the object's masks through the Apple's built-in cutout tool [1]. While this tool may not provide precise masks for transparent objects, we observed that the predicted ball was unaffected by these artifacts. This demonstrates the robustness of our geometry reconstruction method, as illustrated in Fig. 16.

#### V. CONCLUSION

In this paper, we introduce NeRRF, a two-stage pipeline for geometry estimation and novel view synthesis of transparent and glossy objects. Through the use of a progressive encoder and eikonal loss applied to an MLP-based DMTet, our method can effectively reconstruct the geometry of non-Lambertian objects with only the supervision of object masks. NeRRF disentangles non-Lambertian object geometry from its highly view-dependent appearance, making it suitable for various tasks, including material editing, relighting, and environment illumination estimation, which could have implications for AR/VR applications. Our experiments have shown that NeRRF can successfully reconstruct object surfaces compared to state-of-the-art methods and generate high-quality novel view images. One limitation of our method is when learning geometry solely from the supervision of a mask loss may not be sufficient for accurately reconstructing some non-vertex

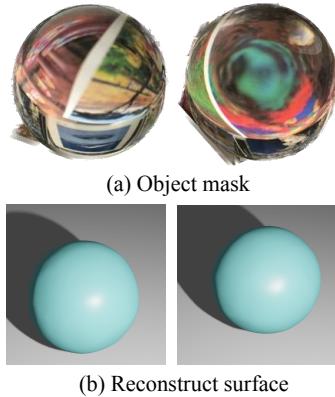


Fig. 16. Object mask and reconstructed surface. (a) Object mask obtained by the Apple's built-in cutout tool (b) Reconstructed surface which are not affected by the high-frequency noises of the object mask.



Fig. 17. Qualitative results of novel view synthesis on the real-world dataset provided by [6].

surfaces. Additionally, NeRRF relies heavily on the availability of masks of non-Lambertian objects, which may not always be obtainable using current segmentation models.

## REFERENCES

- [1] Apple built-in cutout tools. <https://support.apple.com/102460/>. 1, 12
- [2] Poly haven. <https://polyhaven.com/>. 7
- [3] Remove image background. <https://www.remove.bg/>. 1, 4
- [4] Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5855–5864, 2021. 3
- [5] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5470–5479, 2022. 3
- [6] Mojtaba Bemana, Karol Myszkowski, Jeppe Revall Frisvad, Hans-Peter Seidel, and Tobias Ritschel. Eikonal fields for refractive novel-view synthesis. In *ACM SIGGRAPH 2022 Conference Proceedings*, pages 1–9, 2022. 2, 3, 12, 13
- [7] Blender. Blender physics engine. [Online]. Available: <https://docs.blender.org/>. 2019. 5, 7
- [8] James F Blinn and Martin E Newell. Texture and reflection in computer generated images. *Communications of the ACM*, 19(10):542–547, 1976. 5
- [9] Mark Boss, Raphael Braun, Varun Jampani, Jonathan T Barron, Ce Liu, and Hendrik Lensch. Nerd: Neural reflectance decomposition from image collections. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12684–12694, 2021. 2, 3
- [10] Chris Buehler, Michael Bosse, Leonard McMillan, Steven Gortler, and Michael Cohen. Unstructured lumigraph rendering. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 425–432, 2001. 3
- [11] Matthew Cassidy, Jean Mélou, Yvain Quéau, François Lauze, and Jean-Denis Durou. Refractive multi-view stereo. In *2020 International Conference on 3D Vision (3DV)*, pages 384–393. IEEE, 2020. 3
- [12] Shenchang Eric Chen and Lance Williams. View interpolation for image synthesis. In *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, pages 279–288, 1993. 2
- [13] Zhaoxi Chen, Guangcong Wang, and Ziwei Liu. Text2light: Zero-shot text-driven hdr panorama generation. *ACM Transactions on Graphics (TOG)*, 41(6):1–16, 2022. 3
- [14] Zhihu Chen, Kwan-Yee K Wong, Yasuyuki Matsushita, Xiaolong Zhu, and Miaomiao Liu. Self-calibrating depth from refraction. In *2011 International Conference on Computer Vision*, pages 635–642. IEEE, 2011. 3
- [15] Paul Debevec. Rendering synthetic objects into real scenes: Bridging traditional and image-based graphics with global illumination and high dynamic range photography. In *Acm siggraph 2008 classes*, pages 1–10. 2008. 2, 3
- [16] Paul Debevec, Paul Graham, Jay Busch, and Mark Bolas. A single-shot light probe. In *ACM SIGGRAPH 2012 Talks*, pages 1–1. 2012. 3
- [17] Paul E Debevec, Camillo J Taylor, and Jitendra Malik. Modeling and rendering architecture from photographs: A hybrid geometry-and image-based approach. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 11–20, 1996. 3
- [18] Kang Deng, Andrew Liu, Jun-Yan Zhu, and Deva Ramanan. Depth-supervised nerf: Fewer views and faster training for free. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12882–12891, 2022. 3
- [19] John Flynn, Michael Broxton, Paul Debevec, Matthew DuVall, Graham Fyffe, Ryan Overbeck, Noah Snavely, and Richard Tucker. Deepview: View synthesis with learned gradient descent. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2367–2376, 2019. 3
- [20] John Flynn, Ivan Neulander, James Philbin, and Noah Snavely. Deep-stereo: Learning to predict new views from the world's imagery. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5515–5524, 2016. 3
- [21] Steven J Gortler, Radek Grzeszczuk, Richard Szeliski, and Michael F Cohen. The lumigraph. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 43–54, 1996. 2
- [22] Amos Gropp, Lior Yaniv, Niv Haim, Matan Atzman, and Yaron Lipman. Implicit geometric regularization for learning shapes. In *Proceedings of Machine Learning and Systems 2020*, pages 3569–3579. 2020. 5
- [23] Xiaojie Guo, Xiaochun Cao, and Yi Ma. Robust separation of reflection from multiple images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2187–2194, 2014. 3
- [24] Yuan-Chen Guo, Di Kang, Linchao Bao, Yu He, and Song-Hai Zhang. Nerfren: Neural radiance fields with reflections. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18409–18418, 2022. 2, 3
- [25] Kai Han, Kwan-Yee K Wong, and Miaomiao Liu. A fixed viewpoint approach for dense reconstruction of transparent objects. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4001–4008, 2015. 3
- [26] Sebastian Haner and Kalle Astrom. Absolute pose for cameras under flat refractive interfaces. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1428–1436, 2015. 3
- [27] Amir Hertz, Or Perel, Raja Giryes, Olga Sorkine-Hornung, and Daniel Cohen-Or. Sape: Spatially-adaptive progressive encoding for neural optimization. *arXiv preprint arXiv:2104.09125*, 2021. 4
- [28] Xiao Hu, François Lauze, Kim Steenstrup Pedersen, and Jean Mélou. Absolute and relative pose estimation in refractive multi view. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2569–2578, 2021. 3
- [29] Ivo Ihrke, Kiriakos N Kutulakos, Hendrik PA Lensch, Marcus Magnor, and Wolfgang Heidrich. Transparent and specular object reconstruction. In *Computer Graphics Forum*, volume 29, pages 2400–2426. Wiley Online Library, 2010. 3
- [30] Ivo Ihrke, Gernot Ziegler, Art Tevs, Christian Theobalt, Marcus Magnor, and Hans-Peter Seidel. Eikonal rendering: Efficient light transport in refractive objects. *ACM Transactions on Graphics (TOG)*, 26(3):59–es, 2007. 3
- [31] Nima Khademi Kalantari, Ting-Chun Wang, and Ravi Ramamoorthi. Learning-based view synthesis for light field cameras. *ACM Transactions on Graphics (TOG)*, 35(6):1–10, 2016. 3
- [32] Hiroharu Kato, Deniz Becker, Mihai Morariu, Takahiro Ando, Toru Matsuoka, Wadim Kehl, and Adrien Gaidon. Differentiable rendering: A survey. *arXiv preprint arXiv:2006.12057*, 2020. 2
- [33] Kihwan Kim, Jinwei Gu, Stephen Tyree, Pavlo Molchanov, Matthias Nießner, and Jan Kautz. A lightweight approach for on-the-fly

- reflectance estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 20–28, 2017. 3
- [34] Kiriakos N Kutulakos and Steven M Seitz. A theory of shape by space carving. *International journal of computer vision*, 38:199–218, 2000. 3
- [35] Kiriakos N Kutulakos and Eron Steger. A theory of refractive and specular 3d shape by light-path triangulation. In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, volume 2, pages 1448–1455. IEEE, 2005. 3
- [36] Marc Levoy and Pat Hanrahan. Light field rendering. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 31–42, 1996. 2
- [37] Zongcheng Li, Xiaoxiao Long, Yusen Wang, Tuo Cao, Wenping Wang, Fei Luo, and Chunxia Xiao. Neto: Neural reconstruction of transparent objects with self-occlusion aware refraction-tracing. *arXiv preprint arXiv:2303.11219*, 2023. 2, 3, 7
- [38] Zhengqin Li, Yu-Ying Yeh, and Manmohan Chandraker. Through the looking glass: Neural 3d reconstruction of transparent shapes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1262–1271, 2020. 3, 5
- [39] Andrew Liu, Shiry Ginosar, Tinghui Zhou, Alexei A Efros, and Noah Snavely. Learning to factorize and relight a city. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IV 16*, pages 544–561. Springer, 2020. 3
- [40] Yuan Liu, Peng Wang, Cheng Lin, Xiaoxiao Long, Jiepeng Wang, Lingjie Liu, Taku Komura, and Wenping Wang. Nero: Neural geometry and brdf reconstruction of reflective objects from multiview images. *arXiv preprint arXiv:2305.17398*, 2023. 2, 3, 7, 8, 9, 10
- [41] Stephen Lombardi, Tomas Simón, Jason Saragih, Gabriel Schwartz, Andreas Lehrmann, and Yaser Sheikh. Neural volumes: Learning dynamic renderable volumes from images. *arXiv preprint arXiv:1906.07751*, 2019. 3
- [42] Jiahui Lyu, Bojian Wu, Dani Lischinski, Daniel Cohen-Or, and Hui Huang. Differentiable refraction-tracing for mesh reconstruction of transparent objects. *ACM Transactions on Graphics (TOG)*, 39(6):1–13, 2020. 2, 3
- [43] Chenguang Ma, Xing Lin, Jinli Suo, Qionghai Dai, and Gordon Wetzstein. Transparent object reconstruction via coded transport of intensity. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3238–3245, 2014. 3
- [44] Haiyang Mei, Bo Dong, Wen Dong, Pieter Peers, Xin Yang, Qiang Zhang, and Xiaopeng Wei. Depth-aware mirror segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3044–3053, 2021. 3
- [45] Haiyang Mei, Bo Dong, Wen Dong, Jiaxi Yang, Seung-Hwan Baek, Felix Heide, Pieter Peers, Xiaopeng Wei, and Xin Yang. Glass segmentation using intensity and spectral polarization cues. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12622–12631, 2022. 3
- [46] Haiyang Mei, Xin Yang, Yang Wang, Yuanyuan Liu, Shengfeng He, Qiang Zhang, Xiaopeng Wei, and Rynson WH Lau. Don’t hit me! glass detection in real-world scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3687–3696, 2020. 3
- [47] Ben Mildenhall, Pratul P Srinivasan, Rodrigo Ortiz-Cayon, Nima Khademi Kalantari, Ravi Ramamoorthi, Ren Ng, and Abhishek Kar. Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. *ACM Transactions on Graphics (TOG)*, 38(4):1–14, 2019. 3
- [48] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021. 2, 3, 4, 7, 9, 10
- [49] Nigel JW Morris and Kiriakos N Kutulakos. Dynamic refraction stereo. *IEEE transactions on pattern analysis and machine intelligence*, 33(8):1518–1531, 2011. 3
- [50] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics (ToG)*, 41(4):1–15, 2022. 5, 6
- [51] Jacob Munkberg, Jon Hasselgren, Tianchang Shen, Jun Gao, Wenzheng Chen, Alex Evans, Thomas Müller, and Sanja Fidler. Extracting triangular 3d models, materials, and lighting from images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8280–8290, 2022. 2, 3, 4, 7, 8, 9, 11
- [52] Ko Nishino, Zhengyou Zhang, and Katsushi Ikeuchi. Determining reflectance parameters and illumination distribution from a sparse set of images for view-dependent image synthesis. In *Proceedings Eighth IEEE international conference on computer vision. ICCV 2001*, volume 1, pages 599–606. IEEE, 2001. 3
- [53] Michael Oechsle, Songyou Peng, and Andreas Geiger. Unisurf: Unifying neural implicit surfaces and radiance fields for multi-view reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5589–5599, 2021. 3
- [54] Jen-I Pan, Jheng-Wei Su, Kai-Wen Hsiao, Ting-Yu Yen, and Hung-Kuo Chu. Sampling neural radiance fields for refractive objects. In *SIGGRAPH Asia 2022 Technical Communications*, pages 1–4. 2022. 2, 3, 9, 10
- [55] Jeong Joon Park, Aleksander Holynski, and Steven M Seitz. Seeing the world in a bag of chips. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1417–1427, 2020. 3
- [56] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019. 7
- [57] Steven M Seitz and Charles R Dyer. Photorealistic scene reconstruction by voxel coloring. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1067–1073. IEEE, 1997. 3
- [58] Tianchang Shen, Jun Gao, Kangxue Yin, Ming-Yu Liu, and Sanja Fidler. Deep marching tetrahedra: a hybrid representation for high-resolution 3d shape synthesis. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. 2, 4, 5, 10
- [59] Sudipta N Sinha, Johannes Kopf, Michael Goesele, Daniel Scharstein, and Richard Szeliski. Image-based rendering for scenes with reflections. *ACM Transactions on Graphics (TOG)*, 31(4):1–10, 2012. 3
- [60] Gowri Somanath and Daniel Kurz. Hdr environment map estimation for real-time augmented reality. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11298–11306, 2021. 3
- [61] Shuran Song and Thomas Funkhouser. Neural illumination: Lighting prediction for indoor environments. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6918–6926, 2019. 3
- [62] Pratul P Srinivasan, Boyang Deng, Xiuming Zhang, Matthew Tancik, Ben Mildenhall, and Jonathan T Barron. Nerv: Neural reflectance and visibility fields for relighting and view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7495–7504, 2021. 2, 3
- [63] Kenichiro Tanaka, Yasuhiro Mukaigawa, Hiroyuki Kubo, Yasuyuki Matsushita, and Yasushi Yagi. Recovering transparent shape from time-of-flight distortion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4387–4395, 2016. 3
- [64] Kushagra Tiwary, Askhat Dave, Nikhil Behari, Tzofit Klinghoffer, Ashok Veeraraghavan, and Ramesh Raskar. Orca: Glossy objects as radiance field cameras. *arXiv preprint arXiv:2212.04531*, 2022. 2, 3
- [65] Tali Treibitz, Yoav Schechner, Clayton Kunz, and Hanumant Singh. Flat refractive geometry. *IEEE transactions on pattern analysis and machine intelligence*, 34(1):51–65, 2011. 3
- [66] Dor Verbin, Peter Hedman, Ben Mildenhall, Todd Zickler, Jonathan T Barron, and Pratul P Srinivasan. Ref-nerf: Structured view-dependent appearance for neural radiance fields. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5481–5490. IEEE, 2022. 2, 3
- [67] Michael Waechter, Nils Moehrle, and Michael Goesele. Let there be color! large-scale texturing of 3d reconstructions. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6–12, 2014, Proceedings, Part V 13*, pages 836–850. Springer, 2014. 3
- [68] Dongqing Wang, Tong Zhang, and Sabine Süsstrunk. Nemto: Neural environment matting for novel view and relighting synthesis of transparent objects. *arXiv preprint arXiv:2303.11963*, 2023. 2, 3, 7
- [69] Guangcong Wang, Yinuo Yang, Chen Change Loy, and Ziwei Liu. Stylelight: Hdr panorama generation for lighting estimation and editing. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XV*, pages 477–492. Springer, 2022. 3
- [70] Jiepeng Wang, Peng Wang, Xiaoxiao Long, Christian Theobalt, Taku Komura, Lingjie Liu, and Wenping Wang. Neuris: Neural reconstruction of indoor scenes using normal priors. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXII*, pages 139–155. Springer, 2022. 3
- [71] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. 7
- [72] Henrique Weber, Donald Prévost, and Jean-François Lalonde. Learning to estimate indoor lighting from 3d objects. In *2018 International*

- Conference on 3D Vision (3DV), pages 199–207. IEEE, 2018. 3
- [73] Bojian Wu, Yang Zhou, Yiming Qian, Minglun Gong, and Hui Huang. Full 3d reconstruction of transparent objects. arXiv preprint arXiv:1805.03482, 2018. 2, 3
- [74] Jinhui Xiong and Wolfgang Heidrich. In-the-wild single camera 3d reconstruction through moving water surfaces. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12558–12567, 2021. 3
- [75] Dejia Xu, Yifan Jiang, Peihao Wang, Zhiwen Fan, Humphrey Shi, and Zhangyang Wang. Sinnerf: Training neural radiance fields on complex scenes from a single image. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXII*, pages 736–753. Springer, 2022. 3
- [76] Jiamin Xu, Zihan Zhu, Hujun Bao, and Weiwei Xu. Hybrid mesh-neural representation for 3d transparent object reconstruction. arXiv preprint arXiv:2203.12613, 2022. 2, 3, 7
- [77] Zhiwen Yan, Chen Li, and Gim Hee Lee. Nerf-ds: Neural radiance fields for dynamic specular objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8285–8295, 2023. 2, 3
- [78] Jiaolong Yang, Hongdong Li, Yuchao Dai, and Robby T Tan. Robust optical flow estimation of double-layer images under transparency or reflection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1410–1419, 2016. 3
- [79] Xin Yang, Haiyang Mei, Ke Xu, Xiaopeng Wei, Baocai Yin, and Rynson WH Lau. Where is my mirror? In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8809–8818, 2019. 3
- [80] Lior Yariv, Yoni Kasten, Dror Moran, Meirav Galun, Matan Atzmon, Basri Ronen, and Yaron Lipman. Multiview neural surface reconstruction by disentangling geometry and appearance. *Advances in Neural Information Processing Systems*, 33:2492–2502, 2020. 2, 3, 7, 8, 9, 10
- [81] Ze-Xin Yin, Jiaxiang Qiu, Ming-Ming Cheng, and Bo Ren. Multi-space neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12407–12416, 2023. 2, 3
- [82] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4578–4587, 2021. 3
- [83] Fangneng Zhan, Changgong Zhang, Wenbo Hu, Shijian Lu, Feiying Ma, Xuansong Xie, and Ling Shao. Sparse needlets for lighting estimation with spherical transport loss. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12830–12839, 2021. 3
- [84] Kai Zhang, Fujun Luan, Qianqian Wang, Kavita Bala, and Noah Snavely. Physg: Inverse rendering with spherical gaussians for physics-based material editing and relighting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5453–5462, 2021. 2, 3, 7, 8, 9, 10
- [85] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018. 7
- [86] Yiqin Zhao and Tian Guo. Pointar: Efficient lighting estimation for mobile augmented reality. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIII 16*, pages 678–693. Springer, 2020. 3
- [87] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. arXiv preprint arXiv:1805.09817, 2018. 3
- [88] Xinxin Zuo, Chao Du, Sen Wang, Jiangbin Zheng, and Ruigang Yang. Interactive visual hull refinement for specular and transparent object surface reconstruction. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2237–2245, 2015. 3