Ray Deformation Networks for Novel View Synthesis of Refractive Objects

Weijian Deng[1] Dylan Campbell[1] Chunyi Sun[1] Shubham Kanitkar[2]
Matthew E. Shaffer[2] Stephen Gould[1]
[1]The Australian National University [2]RIOS Intelligent Machines

Abstract

Neural Radiance Fields (NeRF) have demonstrated exceptional capabilities in creating photorealistic novel views using volume rendering on a radiance field. However, the intrinsic assumption of straight light rays within NeRF becomes a limitation when dealing with transparent or translucent objects that exhibit refraction, and therefore have curved light paths. This hampers the ability of these approaches to accurately model the appearance of refractive objects, resulting in suboptimal novel view synthesis and geometry estimates. To address this issue, we propose an innovative solution using deformable networks to learn a tailored deformation field for refractive objects. Our approach predicts position and direction offsets, allowing NeRF to model the curved light paths caused by refraction and therefore the complex and highly view-dependent appearances of refractive objects. We also introduce a regularization strategy that encourages piece-wise linear light paths, since most physical systems can be approximated with a piece-wise constant index of refraction. By seamlessly integrating our deformation networks into the NeRF framework, our method significantly improves rendering refractive objects from novel views.

1. Introduction

Refractive objects—transparent objects with significantly different indices of refraction to air, like glass and plastics—are ubiquitous in the real world, and capturing their appearance accurately is essential for achieving visual realism in virtual and augmented reality (VA/AR) applications. Light refraction is the change in the direction of a light ray upon entering a different medium at an oblique angle. It is caused by one side of the wavefront changing speed before the other, since light travels at different speeds in different media. This intrinsic property gives rise to complex light paths through refractive objects, making their appearance challenging to model compared to light transmission in scenes with only opaque objects.

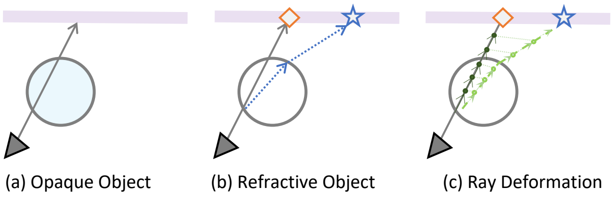

Figure 1. Casting a ray through a scene with an opaque or transparent (refractive) object. (a) Existing NeRF methods learn the density field based on light transport along *straight* paths. (b) However, when light paths intersect refractive objects, they may *curve* (dashed line), depending on the angle of incidence. Volume rendering with a straight-path assumption may assign color and density information to incorrect positions in the 3D volume. This limitation poses challenges for accurately learning the density and radiance field. (c) To address this issue, we propose to bend the light rays by predicting position and direction offsets for sample points along the rays. This approach enables us to model refracted light paths and obtain improved novel view synthesis and geometry estimation results for refractive objects.

NeRF [28] and related models [2, 3, 53] are highly effective at generating photorealistic novel views. This is achieved by leveraging volume rendering on a radiance field, which is parameterized by a neural network that maps the position and view direction to the corresponding density and view-dependent color of the volume element. Existing NeRF methods learn the density field under the assumption that light is transported along a single straight line, following the emission and absorption model (Fig.1 (a)). However, this assumption falls short when dealing with the unique characteristics of refractive objects, which inherently bend the light rays (Fig.1 (b)). Applying straight light rays for volume rendering in such cases may lead to color information being assigned to the wrong 3D position. This inherent limitation prevents conventional NeRF techniques from modeling the intricate, highly view-dependent appearance of refractive objects effectively. Consequently, this leads to suboptimal results in novel view synthesis and geometry estimation of refractive objects.

Forward rendering of refractive objects is well understood, leveraging principles like Snell's law of refraction.

Existing methods for modeling refractive objects often employ controlled setups for acquiring light paths [12, 13, 17, 21, 24, 31, 43, 52, 56, 57]. For instance, Lyu *et al*. [24] utilize turntables and static structured backlights for geometry reconstruction. Alternatively, environment matting techniques estimate a background deformation caused by the refractive object, enabling seamless compositing onto diverse backgrounds [10, 11, 63, 65]. Recent advances in NeRF-based refractive object modeling have considered curved light paths [4, 36, 55]. Pan *et al*. [36] compute curved paths using the Eikonal equation [18] with known refractive index and object geometry. NEMTO [55] assumes an infinitely distant background, linking the final radiance of intersecting camera rays solely to their exiting direction. Using this assumption, NEMTO utilizes an MLP to only predict the exiting direction of each ray for color prediction.

In contrast, this work addresses the challenge of novel view synthesis for refractive objects *without* making assumptions about known geometry, refractive index, controlled setups, or infinitely distant background. To enhance the novel view synthesis capabilities of NeRF, we introduce an approach centered around learning a deformation field specifically tailored for refractive objects, enabling the flexible bending of light rays. Our approach involves two deformation networks that predict shifts in position and direction for sample points along the rays. This leads to a new light path that accurately captures the curved trajectory caused by refraction, as depicted in Fig 1 (c). With this capability, NeRF can effectively model the intricate and view-dependent appearance of refractive objects. It is important to note that without knowing the geometry and refractive index, it is challenging to accurately determine the ray deformation of this highly under-constrained problem. To address this, we introduce a collinearity regularization term, justified by the prevalence of piece-wise constant refractive indices in natural scenes. By seamlessly integrating deformation networks into the standard NeRF framework, our approach achieves significant improvements in novel view synthesis for scenes containing refractive objects.

2. Related Work

Novel View Synthesis. The objective of novel view synthesis is to generate images of a scene from arbitrary camera viewpoints. Existing methods in this field commonly employ either a geometric or image-based 3D representation to facilitate the rendering of novel views. Mesh-based approaches, for instance, utilize surface representations and have been employed for modeling both Lambertian (diffuse) [54] and non-Lambertian scenes [6]. Moreover, volume-based representations, including voxel grids [19, 20] and multi-plane images [27, 41, 64], are utilized to achieve this goal. Recently, coordinate-based neural networks have gained popularity due to their flexibility in representing scenes without the constraints of fixed voxel grids. They take coordinates as input and outputs various spatial properties (*e.g*., occupancy [26, 35, 40], signed distance fields [37, 59, 62], or radiance [28]). NeRF [28] uses a multi-layer perceptron (MLP) to represent a scene as a radiance field and generates high-quality rendered novel views. Many extensions of NeRF have been proposed, such as acceleration [7, 8, 14, 32], scene scale [2, 3, 25, 34], ambiguity reduction [1–3], and specular surface rendering [15, 49, 53]. This work models the intricate view-dependent appearance of refractive objects, considering light ray bending caused by refraction and internal reflection.

Dynamic Neural Radiance Fields. To reconstruct a 3D dynamic scene from monocular RGB camera footage, there are various attempts to extend NeRF to dynamic scenes. One prominent technique involves the utilization of a learned deformation field that maps the coordinates from each input image onto a canonical template coordinate space. For instance, warping-based methods [38, 39, 42, 51] learn how the 3D structure of the scene is deformed and then warp the 3D radiance field of each frame to the canonical frame. Moreover, flow-based methods utilize flow estimation techniques to infer correspondence of 3D points between frames [22, 58]. This work does not aim to represent dynamic scenes with deformation fields. Instead, we introduce deformation networks to learn the bending of rays for a better representation of the refractive object.

Refractive Object Modeling. To recover the 3D geometry of refractive objects, several works build up controlled setups to obtain more information, including polarization [13, 17, 30], tomography [52], moving point light sources [12, 31], light field probes [56], and gray-coded patterns [21, 24, 43, 57]. For example, Li *et al*. [21] use gray-coded backlight and turntable to learn Sign Distance Function (SDF) that achieves refraction-tracing consistency. Han *et al*. [16] reconstruct transparent objects with an unknown refractive index by partially immersing them in a liquid, which alters the incident light path. The object surface is then recovered by triangulating these modified light paths. Li *et al*. [23] assume known environment illumination and refractive index. They incorporate rendering and cost volume layers to model reflection and refraction, optimizing surface normals for precise point cloud reconstruction. The technique of environment matting [10, 11, 63, 65] effectively captures the refraction of environmental light by transparent objects. It estimates the deformation caused by the refractive object on the background, thereby facilitating seamless compositing onto a variety of backgrounds. Other works aim to reconstruct objects inside the refractive and reflective transparent object [44, 50].

Recent advances in novel view synthesis have explored curved light paths through refractive objects [4, 36]. Pan *et* *al*. [36] calculate bending paths using the Eikonal equation [18] with known refractive index and object geometry to model refraction. On the other hand, Bemana *et al*. [4] tackle the challenge without assuming a known refractive index, using multi-step ODE solvers to learn the refractive field. Moreover, NEMTO [55] presumes an infinitely-distant background, where the final radiance of each camera ray intersecting the refractive object *solely depends on its exit direction*. Leveraging this, NEMTO employs an MLP to predict the outgoing ray direction, facilitating color computation for each ray. Unlike the above approaches, our work *does not* assume known geometry, refractive index, or an infinitely-distant background. We propose deformation networks that predict light paths (both direction and positions) traversing the transparent object. Furthermore, for enhanced modeling of reflective and refractive objects, the multi-space NeRF (MS-NeRF) [29] decomposes Euclidean space into virtual sub-spaces. In contrast, our method takes a more direct route by explicitly bending rays to effectively handle refraction.

3. Modeling Refraction by Ray Deformation

Given N RGB images of a scene that may contain one or more (partially) refractive objects, this work aims to estimate the underlying geometry and render images from novel camera views. The primary challenge is that a refractive object adopts its appearance from the surrounding environment through the refraction and (internal) reflection of light rays that traverse the object. However, in the context of NeRF modeling, the conventional assumption is that light rays propagate in straight paths. To address this limitation, our work proposes a ray deformation network, which facilitates the flexible bending of rays through the refractive object without assuming a known geometry, and an appropriate regularization strategy to constrain the model. This allows our NeRF-like model to accurately represent the complex view-dependent appearances that arise from the refraction and reflection properties of transparent objects.

3.1. NeRF Preliminaries

NeRF [28] utilizes a continuous field of volume elements that emit and absorb light to model both the appearance and geometry of a scene. At any given 3D position $\mathbf{x} \in \mathbb{R}^3$, NeRF calculates the density $\sigma(\mathbf{x})$ and geometric representation $\mathbf{g}(\mathbf{x})$ by employing a spatial MLP Ψ_s: $[\sigma(\mathbf{x}), \mathbf{g}(\mathbf{x})] = \Psi_s(\gamma(\mathbf{x}))$, where γ denotes the positional encoding. Additionally, NeRF incorporates a directional MLP Ψ_v to predict the emitted light color $\mathbf{c}(\mathbf{x}, \mathbf{d})$ by a particle located at position $\mathbf{x}$ from direction $\mathbf{d}$. The directional MLP takes the geometric representation $\mathbf{g}(\mathbf{x})$ and the view direction $\mathbf{d}$ as inputs: $\mathbf{c}(\mathbf{x}, \mathbf{d}) = \Psi_v(\gamma(\mathbf{d}), \mathbf{g}(\mathbf{x}))$.

To render each pixel of a camera view using NeRF, the two MLPs are queried at sample points $\mathbf{x}_i = \mathbf{o} + t_i\mathbf{d}$ along a ray. This ray originates from the camera's center of projection $\mathbf{o}$ with direction $\mathbf{d}$. The MLPs return densities $\{\sigma_i\}$ and colors $\{\mathbf{c}_i\}$ corresponding to the sampled points. They are then alpha-composited using numerical quadrature to determine the final color of the pixel associated with the ray:

$$\hat{\mathbf{C}}(\mathbf{o}, \mathbf{d}) = \sum\nolimits_i w_i \mathbf{c}_i \,, \tag{1}$$

where $w_i = e^{-\sum_{j<i} \sigma_j (t_{j+1} - t_j)} \left(1 - e^{-\sigma_i (t_{i+1} - t_i)}\right)$.

The parameters of the two MLPs are optimized by minimizing the difference between the predicted color $\hat{\mathbf{C}}(\mathbf{o}, \mathbf{d})$ and the ground truth color $\mathbf{C}_{\text{gt}}(\mathbf{o}, \mathbf{d})$ of each pixel, which is extracted from the input image. The following photometric loss function expresses this optimization:

$$\mathcal{L}_c = \frac{1}{|\mathcal{R}|} \sum_{(\mathbf{o},\mathbf{d}) \in \mathcal{R}} \|\hat{\mathbf{C}}(\mathbf{o}, \mathbf{d}) - \mathbf{C}_{\text{gt}}(\mathbf{o}, \mathbf{d})\|^2 \,, \tag{2}$$

where $\mathcal{R}$ represents all the training rays, each denoted by an ordered pair $(\mathbf{o}, \mathbf{d})$.

Predicting Normals. According to Snell's law, the refracted ray direction depends on input direction, interface normal and refractive index. We thus use normal vectors for our ray deformation networks. Following [5, 47, 53], a spatial MLP Ψ_n is employed to predict a normal vector $\mathbf{n}_i$ for each position $\mathbf{x}_i$ along the ray. It takes as input the geometric feature representation: $\mathbf{n}_i = \Psi_n(\mathbf{g}(\mathbf{x}_i))$. The predicted normal vector $\mathbf{n}_i$ is supervised by the underlying density gradient normal $\mathbf{n}'_i$ along the ray:

$$\mathcal{L}_n = \sum\nolimits_i w_i \|\mathbf{n}_i - \mathbf{n}'_i\|^2, \tag{3}$$

where $\mathbf{n}'_i = -\nabla\sigma(\mathbf{x}_i)/\|\nabla\sigma(\mathbf{x}_i)\|$ and w_i is the weight of the ith sample along the ray as defined in Eq. 1.

3.2. Ray Deformation Network

3.2.1 Ray Bending

In the case of a refractive object, if we have prior knowledge about its physical material and geometry, it is straightforward to perform analytical ray tracing to determine the path of the light. However, obtaining such detailed information about the refractive object is often challenging and not readily available. Additionally, environmental matting methods [10, 63, 65] can capture the reflection and refraction of transparent objects but are constrained by the requirement of controlled environments or complex camera setups with structured background lighting [21, 24, 43]. In light of the above limitations, we propose a ray deformation network that facilitates the flexible bending of light rays, enabling effective handling of both refraction and reflection.

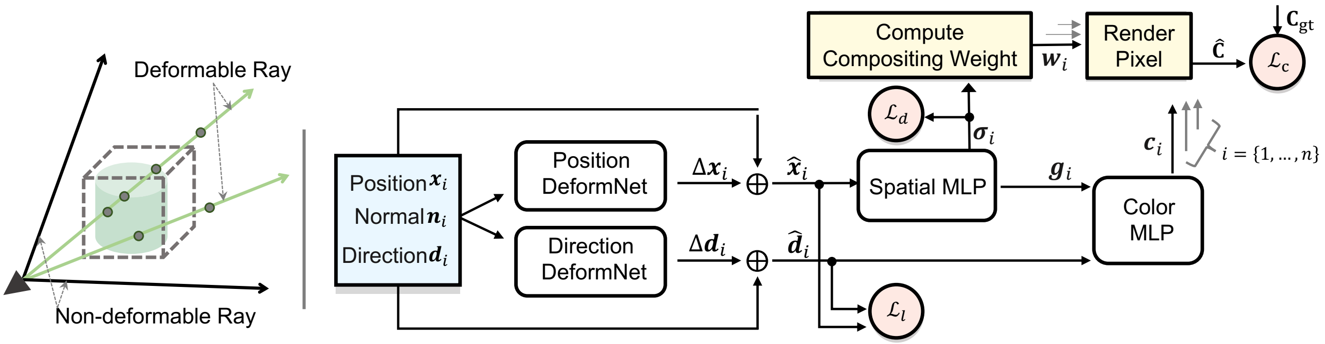

Figure 2. Flowchart of our framework for modeling refractive objects. Our proposed framework combines the conventional NeRF networks (spatial MLP and color MLP) with two deformation networks for position and direction. The model deems any camera ray that intersects the cuboid, which is assumed to cover all refractive objects, to be potentially deformable. Each sample point $\mathbf{x}_i$ on a deformable ray is processed by the deformation networks, taking its position $\mathbf{x}_i$, view direction $\mathbf{d}_i$, and normal vector $\mathbf{n}$ as inputs to compute offsets in position $\Delta\mathbf{x}_i$ and direction $\Delta\mathbf{d}_i$. From the updated position $\hat{\mathbf{x}}_i$ and direction $\hat{\mathbf{d}}_i$ vectors, the spatial MLP computes density σ_i and geometric representation g_i. Then, the color MLP takes as inputs $\mathbf{g}_i$ and $\hat{\mathbf{d}}_i$ and outputs color $\mathbf{c}_i$. After calculating densities and colors for all sample points along the deformable ray, they are integrated following volumetric rendering to obtain the rendered pixel color $\hat{\mathbf{c}}$. The photometric loss $\mathcal{L}_c$ (Eq. 1) is used for supervision. To discourage non-physical ray deformations, collinearity regularization $\mathcal{L}_l$ (Eq. 5) is introduced. Lastly, a near-camera density penalty $\mathcal{L}_d$ (Eq. 6) is applied to remove artifacts associated with refractive objects.

Deformable Rays. We select light rays that interact with the refractive object as deformable rays, and our ray deformation network is tailored specifically to handle these rays. Given the lack of prior knowledge about the precise geometry of the refractive object, we opt to use a cuboid to approximate and localize the region occupied by the refractive object. By determining whether a ray intersects the cuboid, we identify it as a deformable ray. To acquire the cuboid, we project *roughly annotated bounding boxes* of objects from the 2D training images back into 3D space, utilizing the known camera poses. While we use a cuboid for simplicity, more sophisticated techniques like the visual hull based on the segmentation masks [20, 23, 46] could be employed to achieve a more precise localization.

Flexible Sampling for Ray Bending. We grant the sample points along the deformable rays additional flexibility to better model the complex appearance of the refractive object. As shown in Fig. 2, we leverage a direction deformation network Ψ_d and a position deformation network Ψ_p to manipulate the *direction* and *position* of individual sample points along the deformable ray. Formally, for each point $\mathbf{x}$ sampled after the first intersection of a deformable ray with the cuboid, the direction deformation network Ψ_d predicts its rotation shift $\Delta\mathbf{d}$, and the position deformation network Ψ_p predicts the position offset $\Delta\mathbf{x}$. The inputs of Ψ_d and Ψ_p are the encoded position $\gamma(\mathbf{x})$, encoded view direction $\gamma(\mathbf{d})$, and encoded normal vector $\gamma(\mathbf{n})$ at every sample point $\mathbf{x}$:

$$\begin{aligned} \Delta\mathbf{d} &\leftarrow \Psi_d(\gamma(\mathbf{x}), \gamma(\mathbf{d}), \gamma(\mathbf{n})), \\ \Delta\mathbf{x} &\leftarrow \Psi_p(\gamma(\mathbf{x}), \gamma(\mathbf{d}), \gamma(\mathbf{n})). \end{aligned} \tag{4}$$

With the displacements of $\Delta\mathbf{x}$ and $\Delta\mathbf{d}$, we transform the original sample point to its new position $\hat{\mathbf{x}} = \mathbf{x} + \Delta\mathbf{x}$ associated with new direction $\hat{\mathbf{d}} = (\mathbf{d} + \Delta\mathbf{d})/\|\mathbf{d} + \Delta\mathbf{d}\|$. Having obtained the updated position $\hat{\mathbf{x}}$ and direction $\hat{\mathbf{d}}$, the standard NeRF networks (spatial MLP and color MLP) compute the corresponding color c and density σ. Once densities and colors are computed for all sample points along the deformable ray, they are integrated using volumetric rendering principles (Eq. 1). For non-deformable rays, the deformable networks are inactive. Standard NeRF networks directly use original positions and view directions to predict colors and densities for volumetric rendering.

3.2.2 Learning the Model

Our deformation networks are designed without relying on assumptions about known geometry or refractive index, which makes them versatile for modeling various refractive objects. However, the absence of prior knowledge poses a challenge in determining how the rays bend, impeding the networks from learning reasonable position and direction offsets. This, in turn, impacts the learning of the radiance field. To overcome this challenge, we introduce two regularization strategies tailored to enhance both the stability and accuracy of the ray deformation process.

Collinearity Regularization. In nature, the refractive index at each point in a scene is piece-wise constant to a good approximation. By Snell's law, this means that refracted light rays are piece-wise linear. We draw inspiration from this physical phenomenon and introduce collinearity regu-

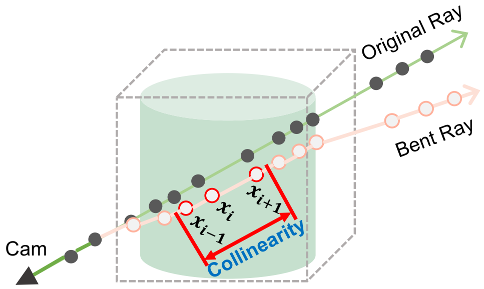

Figure 3. Illustration of collinearity regularization. It encourages adjacent sample points to be collinear after deformation so that the bent ray prefers piece-wise linear configurations.

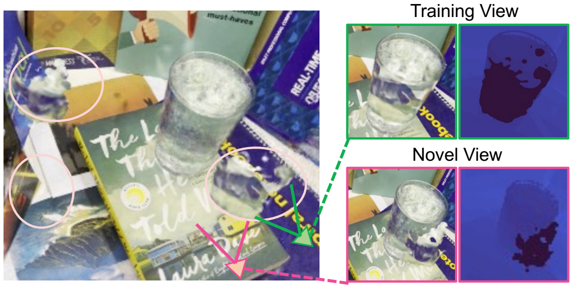

Figure 4. Illustration of near-camera outliers. For refractive objects, NeRF tends to generate outliers *near* the camera (highlighted with ellipses) in order to minimize the photometric loss. However, this leads to inaccurate novel views. We propose penalizing near-camera density to remove such outliers.

larization. It encourages the ray to be as linear as possible, avoiding jagged paths, for all rays that intersect the cuboid, *i.e.*, those that are labeled as deformable. As shown in Figure 3, for the set of K deformed sample points $\hat{\mathbf{x}}_i^r$ on each ray r in the training set $\mathcal{R}$, we apply collinearity regularization between each point and its neighbors using the cosine distance, given by (for $Z = (K-2)|\mathcal{R}|$)

$$\mathcal{L}_l = \frac{1}{Z}\sum_{r\in\mathcal{R}}\sum_{i=2}^{K-1}\left(1-\frac{(\hat{\mathbf{x}}_i^r-\hat{\mathbf{x}}_{i-1}^r)}{\|\hat{\mathbf{x}}_i^r-\hat{\mathbf{x}}_{i-1}^r\|}^{\mathsf{T}}\frac{(\hat{\mathbf{x}}_{i+1}^r-\hat{\mathbf{x}}_i^r)}{\|\hat{\mathbf{x}}_{i+1}^r-\hat{\mathbf{x}}_i^r\|}\right). \tag{5}$$

Near-Camera Density Penalty. The appearance of refractive objects varies significantly with the view direction: slightly changing the view angle can lead to significantly different colors for the same surface position. Such multi-view inconsistency introduces learning difficulties for NeRF. Fig. 4 illustrates how this affects standard NeRF models: they generate outliers near the camera to minimize the photometric error during training. While this behavior helps NeRF reproduce the training images, it stymies its ability to generalize, introducing artifacts in the rendered novel views. Inspired by this observation, we introduce regularization to discourage such near-camera artifacts —a free-space penalty. For the set of K sample points $\{\mathbf{x}_i = \mathbf{o} + t_i\mathbf{d}\}_{i=1}^K$ on each ray in the training set $\mathcal{R}$, we apply a density penalty given by

$$\mathcal{L}_d = \frac{1}{K|\mathcal{R}|}\sum_{(\mathbf{o},\mathbf{d})\in\mathcal{R}}\sum_{i=1}^{K}\sigma_i \mathbb{1}(t_i<\delta), \tag{6}$$

where δ is the distance along the ray up to which the penalty is applied. Empirically, we use $\delta = 0.3$ in the experiments. Note that, this regularization has shown helpful in few-shot NeRF [33,60], and this work further demonstrates its effectiveness in reducing artifacts caused by refractive objects.

Overall Loss. We base our model on the Nerfacto model [48], a well-designed NeRF method that combines advances from several published works, and apply our deformation networks to bend the deformable rays. The overall loss function of our framework combines color loss $\mathcal{L}_c$ (Eq. 2), normal regularization $\mathcal{L}_n$ (Eq. 3), collinearity regularization $\mathcal{L}_l$ (Eq. 5), and the near-camera density penalty $\mathcal{L}_d$ (Eq. 6) :

$$\mathcal{L} = \mathcal{L}_c + \lambda_1\mathcal{L}_n + \lambda_2\mathcal{L}_d + \lambda_3\mathcal{L}_l, . \tag{7}$$

The first two items are inherited from Nerfacto, while the latter two are introduced by this work. The coefficients λ_i correspond to the weight assigned to each loss term.

4. Experiments

4.1. Setup

Datasets. We collect data for four real scenes with different refractive objects (Cup-A/B/C/D), with views sampled (approximately) on a hemisphere. Each dataset is randomly split into training, validation, and testing sets with 90, 20, and 90 images, respectively. Camera poses are computed using COLMAP, and the image size is 960 × 540. Additionally, we use two real datasets (Glass and Ball) from Bemana *et al*. [4] and follow their dataset split to report results. Sample images from each scene are shown in Figure 5.

Compared Methods and Implementation Details. For our evaluation, we compare our proposed approach to three baseline NeRF methods: TensoRF [9], Instant-NGP [32], and Nerfacto [48]; as well as three refraction-specific methods: MS-NeRF [29], SampleNeRFRO [36], and Eikonal Fields [4]. SampleNeRFRO [36] assumes that the geometry of the refractive object and its refractive index is known, and so can only be evaluated on datasets where these are available. We do not compare with NEMTO [55], since it assumes a known background or environment map at an infinite distance, and so cannot be used with real data. We

Table 1. Quantitative evaluation on the test set of six real datasets of refractive objects. We provide a comprehensive analysis of performance metrics on the test set, encompassing PSNR (↑), SSIM (↑), and LPIPS (↓), across various NeRF models: TensoRF [9], Instant-NGP [32], MS-NeRF [29], Nerfacto [48], Eikonal Fields [4], SampleNeRFRO [36], and ours. Our method demonstrates consistently better performance across all six datasets. †Assumes known geometry/masks and refractive indices, so cannot be evaluated on the Cup datasets.

Model	Ball [4]			Glass [4]			Cup-A			Cup-B			Cup-C			Cup-D		
	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS
TensoRF	21.41	0.735	0.187	20.49	0.695	0.226	25.96	0.856	0.184	23.12	0.823	0.230	27.57	0.888	0.1671	24.13	0.825	0.212
Instant-NGP	21.56	0.790	0.121	21.42	0.748	0.148	23.43	0.842	0.189	22.76	0.827	0.184	26.03	0.894	0.127	23.51	0.838	0.176
Nerfacto	21.67	0.797	0.113	22.14	0.774	0.121	23.24	0.846	0.168	21.37	0.808	0.209	25.69	0.893	0.114	22.67	0.835	0.177
MS-NeRF	22.35	0.810	0.105	21.83	0.781	0.119	27.43	0.890	0.113	24.83	0.859	0.142	28.84	0.910	0.099	25.51	0.870	0.137
SampleNeRFRO†	21.49	0.679	0.270	21.11	0.630	0.317	–	–	–	–	–	–	–	–	–	–	–	–
Eikonal Fields	21.64	0.699	0.217	20.92	0.663	0.262	26.11	0.832	0.214	25.27	0.818	0.242	24.62	0.811	0.282	24.33	0.777	0.256
Ours	**23.30**	**0.822**	**0.092**	**23.54**	**0.795**	**0.103**	**29.33**	**0.894**	**0.104**	**27.04**	**0.867**	**0.128**	**30.11**	**0.916**	**0.093**	**27.09**	**0.871**	**0.137**

utilize the implementations in Nerfstudio [48] to run each NeRF model on all datasets, except for Pan *et al*. [36] and Eikonal Fields [4], where we use their implementations.

Our method is built upon Nerfacto, and we adopt its default hyper-parameters for consistency: λ_1 is set at 0.001. We set λ_2 and λ_3 in Eq. 7 to 0.01, by a coarse search on the validation set, for the near-camera density penalty and piece-wise linear regularization, respectively. For the deformation network, we use a simple 3-layer MLP architecture, similar to the normal-predicting MLP in Verbin *et al*. [53].

Evaluation Metrics. To evaluate the synthesis results, we employ three visual quality metrics: Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index (SSIM), and Learned Perceptual Image Patch Similarity (LPIPS). A higher value of PSNR and SSIM indicates better visual quality, while a lower value of LPIPS signifies better perceptual similarity to the ground truth.

4.2. Results

Quantitative Evaluation. In Table 1, we provide a thorough evaluation of our proposed novel view synthesis technique across an assortment of six refractive datasets. The results indicate that our method performs favorably compared to other NeRF models.

First, traditional NeRF models, like TensoRF, Instant-NGP, and Nerfacto, are grounded in the assumption of linear ray paths, which restricts their ability to accurately account for the intricate refraction behavior of transparent objects. This limitation becomes evident in their comparatively diminished performance across all six datasets, compared with our approach. For instance, on the Ball dataset, our method improves performance by 1.89 over TensoRF, 1.74 over Instant-NGP, and 1.63 over Nerfacto in terms of PSNR, with similar gains with respect to the other metrics. Second, our method performs competitively with three NeRF models that are designed for refraction (*i.e*., MS-NeRF [29], Eikonal Fields [4], and SampleNeRFRO [36]). While these approaches are better able to model the refractive objects than the traditional NeRF methods, our method exhibits the best quantitative performance.

We think that the improvements of our method over other NeRF models can be attributed to the integration of deformation networks, which enable a more suitable representation of ray refraction. The aforementioned comparative analysis collectively indicates the promising potential of our method in novel view synthesis for refractive objects.

Qualitative Evaluation. A qualitative evaluation of our proposed approach alongside baseline models is presented in Figure 5. It becomes apparent that conventional methods (Instant-NGP, and Nerfacto) yield visually noisy renderings of refractive objects, characterized by the presence of numerous tiny particles within the object region. MS-NeRF demonstrates smoother results on some datasets (*e.g*., Cup-A and Cup-C), but is inconsistent (*e.g*., Ball and Cup-B). In comparison, our method reliably achieves the best results across all datasets. Eikonal Fields showcases high-quality results on Ball and Cup-C, yet falters in effectively modeling the remaining datasets. In contrast, the refractive object regions in our renderings appear smoother and cleaner across all datasets. It is, however, worth noting that while our approach shows promise in accurately modeling refraction, there exists room for improvement in terms of preserving fine-grained details. Supplementary techniques such as diffusion models [61] or discriminators [45] may enhance detail plausibility in our results.

A qualitative comparison of 3D reconstruction is also shown in Figure 6, where we compare our method with MS-NeRF and Nerfacto. It shows that our method provides more complete and smoother reconstructions of refractive objects compared to the other methods. This indicates that our ray deformation approach allows the network to better model the geometry of refractive objects.

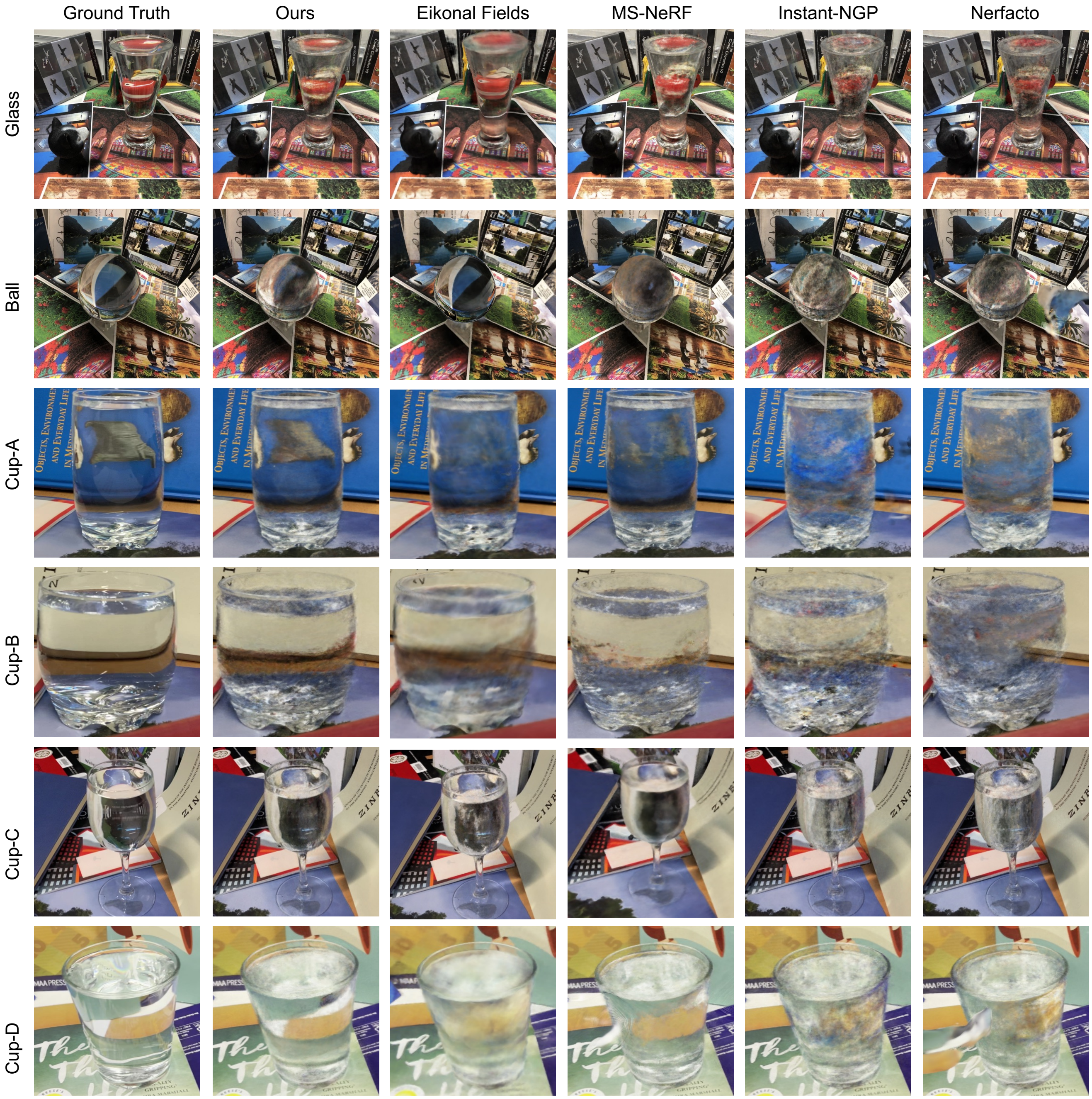

Figure 5. Qualitative comparison of novel view synthesis with different NeRF models on refractive objects. Visual results are displayed across six distinct refractive datasets. The sequence, from left to right, displays the ground-truth novel view, followed by renders from our method, Eikonal Fields, MS-NeRF, Instant-NGP, and Nerfacto. Each row corresponds to a different dataset. Our method outperforms other models in handling refraction effects, resulting in smoother and cleaner novel view synthesis outcomes for refractive objects.

Ablation Study. Building upon the baseline method Nerfacto, our method adds the ray deformation networks and two regularization strategies. To verify that each component has a salutary effect on the performance of the model, we perform an ablation study on the six refractive datasets and report the average results in Table 2. First, we remove the near-camera density penalty (A) and observe a significant performance drop, indicating that translucent material incorrectly placed near the camera is a significant mode of error for refractive objects. Second, we remove the ray de-

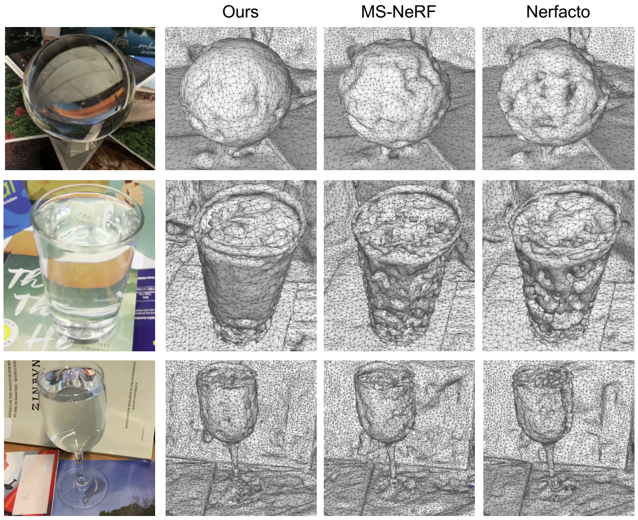

Figure 6. Visualisation of 3D shape reconstruction across three refractive objects. We present a qualitative comparison among our method, MS-NeRF, and Nerfacto. Our results illustrate that our approach produces comparatively more comprehensive and smoother reconstructions of refractive objects.

Table 2. Ablation study. We report the average metrics across the six refractive object datasets. A significant drop in performance is observed when removing any of the components, indicating that each contributes to the model's effectiveness.

Method	PSNR (↑)	SSIM (↑)	LIPIS (↓)
Nerfacto	22.80	0.826	0.150
# Ours	26.73	0.861	0.109
A w/o near-camera penalty	24.14	0.837	0.143
B w/o ray deformation	23.51	0.836	0.133
C w/o collinearity regularization	25.05	0.821	0.172

formation networks entirely (B), and observe that this has the biggest effect on model performance. Third, we remove the collinearity regularization (C), allowing the deformation networks to predict non-physical jagged or high curvature light trajectories. This has a smaller effect on performance.

Case Study I: Modeling Translucent Colored Objects. Our model accounts for the color contributions arising from sample points within translucent objects. This enables our model to handle varying levels of transparency. To verify this, we change the color of the liquid inside the object. In Figure 7 (a), we use two datasets: one with green liquid and the other with deep red liquid. We show that our method can handle both cases and achieve good novel view synthesis. Moreover, our approach consistently outperforms MS-NeRF and Nerfacto in terms of test set PSNR.

Case Study II: Modeling Partially Refractive Objects. Our method is not restricted by the assumption that the object of interest is entirely transparent. This flexibility enables our method to handle scenarios where opaque objects are contained within a transparent medium. To explore this potential, we introduce a dataset where a pencil is submerged in a cup of water. As depicted in Figure 7 (b), our method yields high-quality results, outperforming MS-NeRF by 1.11 in terms of test set PSNR.

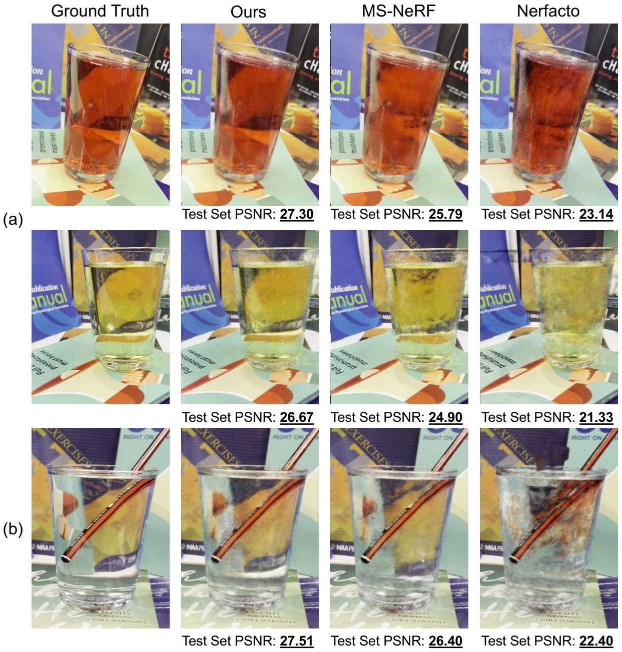

Figure 7. Case studies. (a) Modeling translucent, colored objects. Our model is able to inherently handle a range of transparency levels and colors. (b) Modeling partially refractive objects. Our model has the flexibility to model transparent objects that contain opaque components, viewed under refraction.

5. Conclusion

This work addresses the intricacies associated with the complex, view-dependent characteristics of transparent objects, encompassing the bending of light rays due to refraction. In contrast to conventional assumptions of straight rays, our approach employs a deformation network to learn the refractive behavior of light rays as they pass through a scene. We also introduce a regularization strategy that encourages the light paths to be piece-wise linear, since real-world scenes can be well-approximated by piece-wise constant refractive indices. By incorporating the deformation process into NeRF modeling, our approach achieves high-quality novel view synthesis and geometry estimation for scenes with refractive objects.

Acknowledgment We thank all reviewers and ACs for their constructive comments. This work was generously supported through research collaboration with RIOS Intelligent Machines.

References

[1] 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 *ICCV*, pages 5855–5864, 2021. 2

[2] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *CVPR*, pages 5470–5479, 2022. 1, 2

[3] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Zip-nerf: Anti-aliased grid-based neural radiance fields. *arXiv preprint arXiv:2304.06706*, 2023. 1, 2

[4] 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, 5, 6

[5] Mark Boss, Raphael Braun, Varun Jampani, Jonathan T Barron, Ce Liu, and Hendrik Lensch. Nerd: Neural reflectance decomposition from image collections. In *ICCV*, pages 12684–12694, 2021. 3

[6] 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. 2

[7] Eric R Chan, Connor Z Lin, Matthew A Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas J Guibas, Jonathan Tremblay, Sameh Khamis, et al. Efficient geometry-aware 3d generative adversarial networks. In *CVPR*, pages 16123–16133, 2022. 2

[8] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. Tensorf: Tensorial radiance fields. In *ECCV*, pages 333–350, 2022. 2

[9] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. Tensorf: Tensorial radiance fields. In *European Conference on Computer Vision (ECCV)*, 2022. 5, 6

[10] Guanying Chen, Kai Han, and Kwan-Yee K Wong. Tom-net: Learning transparent object matting from a single image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9233–9241, 2018. 2, 3

[11] Guanying Chen, Kai Han, and Kwan-Yee K Wong. Learning transparent object matting. *International Journal of Computer Vision*, 127(10):1527–1544, 2019. 2

[12] Tongbo Chen, Michael Goesele, and H-P Seidel. Mesostructure from specularity. In *CVPR*, pages 1825–1832, 2006. 2

[13] Zhaopeng Cui, Jinwei Gu, Boxin Shi, Ping Tan, and Jan Kautz. Polarimetric multi-view stereo. In *CVPR*, pages 1558–1567, 2017. 2

[14] Sara Fridovich-Keil, Alex Yu, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. In *CVPR*, pages 5501–5510, 2022. 2

[15] Yuan-Chen Guo, Di Kang, Linchao Bao, Yu He, and Song-Hai Zhang. Nerfren: Neural radiance fields with reflections. In *CVPR*, pages 18409–18418, 2022. 2

[16] Kai Han, Kwan-Yee K Wong, and Miaomiao Liu. Dense reconstruction of transparent objects by altering incident light paths through refraction. *International Journal of Computer Vision*, 126:460–475, 2018. 2

[17] Cong Phuoc Huynh, Antonio Robles-Kelly, and Edwin Hancock. Shape and refractive index recovery from single-view polarisation images. In *CVPR*, pages 1229–1236, 2010. 2

[18] 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. 2, 3

[19] Abhishek Kar, Christian Häne, and Jitendra Malik. Learning a multi-view stereo machine. In *NIPS*, volume 30, 2017. 2

[20] Kiriakos N Kutulakos and Steven M Seitz. A theory of shape by space carving. *International journal of computer vision*, 38:199–218, 2000. 2, 4

[21] 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

[22] Zhengqi Li, Simon Niklaus, Noah Snavely, and Oliver Wang. Neural scene flow fields for space-time view synthesis of dynamic scenes. In *CVPR*, pages 6498–6508, 2021. 2

[23] Zhengqin Li, Yu-Ying Yeh, and Manmohan Chandraker. Through the looking glass: neural 3d reconstruction of transparent shapes. In *CVPR*, pages 1262–1271, 2020. 2, 4

[24] 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

[25] Ricardo Martin-Brualla, Noha Radwan, Mehdi SM Sajjadi, Jonathan T Barron, Alexey Dosovitskiy, and Daniel Duckworth. Nerf in the wild: Neural radiance fields for unconstrained photo collections. In *CVPR*, pages 7210–7219, 2021. 2

[26] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *CVPR*, pages 4460–4470, 2019. 2

[27] 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. 2

[28] 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. In *ECCV*, pages 405–421, 2020. 1, 2, 3

[29] Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, Peter Hedman, Ricardo Martin-Brualla, and Jonathan T. Barron. MultiNeRF: A Code Release for Mip-NeRF 360, Ref-NeRF, and RawNeRF, 2022. 3, 5, 6

[30] Daisuke Miyazaki and Katsushi Ikeuchi. Inverse polarization raytracing: estimating surface shapes of transparent objects. In *CVPR*, pages 910–917, 2005. 2

[31] Nigel JW Morris and Kiriakos N Kutulakos. Reconstructing the surface of inhomogeneous transparent scenes by scatter-trace photography. In *ICCV*, pages 1–8, 2007. 2

[32] 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. 2, 5, 6

[33] Michael Niemeyer, Jonathan T Barron, Ben Mildenhall, Mehdi SM Sajjadi, Andreas Geiger, and Noha Radwan. Regnerf: Regularizing neural radiance fields for view synthesis from sparse inputs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5480–5490, 2022. 5

[34] Michael Niemeyer and Andreas Geiger. Giraffe: Representing scenes as compositional generative neural feature fields. In *CVPR*, pages 11453–11464, 2021. 2

[35] Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision. In *CVPR*, pages 3504–3515, 2020. 2

[36] 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, 5, 6

[37] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *CVPR*, pages 165–174, 2019. 2

[38] Keunhong Park, Utkarsh Sinha, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Steven M Seitz, and Ricardo Martin-Brualla. Nerfies: Deformable neural radiance fields. In *ICCV*, pages 5865–5874, 2021. 2

[39] Keunhong Park, Utkarsh Sinha, Peter Hedman, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Ricardo Martin-Brualla, and Steven M Seitz. Hypernerf: A higher-dimensional representation for topologically varying neural radiance fields. *arXiv preprint arXiv:2106.13228*, 2021. 2

[40] Songyou Peng, Michael Niemeyer, Lars Mescheder, Marc Pollefeys, and Andreas Geiger. Convolutional occupancy networks. In *ECCV*, pages 523–540, 2020. 2

[41] Eric Penner and Li Zhang. Soft 3d reconstruction for view synthesis. *ACM Transactions on Graphics (TOG)*, 36(6):1–11, 2017. 2

[42] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-nerf: Neural radiance fields for dynamic scenes. In *CVPR*, pages 10318–10327, 2021. 2

[43] Yiming Qian, Minglun Gong, and Yee Hong Yang. 3d reconstruction of transparent objects with position-normal consistency. In *CVPR*, pages 4369–4377, 2016. 2, 3

[44] Jiaxiong Qiu, Peng-Tao Jiang, Yifan Zhu, Ze-Xin Yin, Ming-Ming Cheng, and Bo Ren. Looking through the glass: Neural surface reconstruction against high specular reflections. In *CVPR*, pages 20823–20833, 2023. 2

[45] Barbara Roessle, Norman Müller, Lorenzo Porzi, Samuel Rota Bulò, Peter Kontschieder, and Matthias Nießner. Ganerf: Leveraging discriminators to optimize neural radiance fields. *arXiv preprint arXiv:2306.06044*, 2023. 6

[46] Nagabhushan Somraj, Adithyan Karanayil, and Rajiv Soundararajan. Simplenerf: Regularizing sparse input neural radiance fields with simpler solutions. *arXiv preprint arXiv:2309.03955*, 2023. 4

[47] 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 *CVPR*, pages 7495–7504, 2021. 3

[48] Matthew Tancik, Ethan Weber, Evonne Ng, Ruilong Li, Brent Yi, Justin Kerr, Terrance Wang, Alexander Kristoffersen, Jake Austin, Kamyar Salahi, Abhik Ahuja, David McAllister, and Angjoo Kanazawa. Nerfstudio: A modular framework for neural radiance field development. In *ACM SIGGRAPH 2023 Conference Proceedings*, SIGGRAPH '23, 2023. 5, 6

[49] Kushagra Tiwary, Akshat Dave, Nikhil Behari, Tzofi Klinghoffer, Ashok Veeraraghavan, and Ramesh Raskar. Orca: Glossy objects as radiance-field cameras. In *CVPR*, 2023. 2

[50] Jinguang Tong, Sundaram Muthu, Fahira Afzal Maken, Chuong Nguyen, and Hongdong Li. Seeing through the glass: Neural 3d reconstruction of object inside a transparent container. In *CVPR*, pages 12555–12564, 2023. 2

[51] Edgar Tretschk, Ayush Tewari, Vladislav Golyanik, Michael Zollhöfer, Christoph Lassner, and Christian Theobalt. Non-rigid neural radiance fields: Reconstruction and novel view synthesis of a dynamic scene from monocular video. In *ICCV*, pages 12959–12970, 2021. 2

[52] Borislav Trifonov, Derek Bradley, and Wolfgang Heidrich. Tomographic reconstruction of transparent objects. In *ACM SIGGRAPH 2006 Sketches*, pages 55–es. 2006. 2

[53] 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 *CVPR*, pages 5481–5490, 2022. 1, 2, 3, 6

[54] Michael Waechter, Nils Moehrle, and Michael Goesele. Let there be color! large-scale texturing of 3d reconstructions. In *ECCV*, pages 836–850. Springer, 2014. 2

[55] 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, 5

[56] Gordon Wetzstein, David Roodnick, Wolfgang Heidrich, and Ramesh Raskar. Refractive shape from light field distortion. In *ICCV*, pages 1180–1186, 2011. 2

[57] Bojian Wu, Yang Zhou, Yiming Qian, Minglun Gong, and Hui Huang. Full 3d reconstruction of transparent objects. *arXiv preprint arXiv:1805.03482*, 2018. 2

[58] Wenqi Xian, Jia-Bin Huang, Johannes Kopf, and Changil Kim. Space-time neural irradiance fields for free-viewpoint video. In *CVPR*, pages 9421–9431, 2021. 2

[59] Qiangeng Xu, Weiyue Wang, Duygu Ceylan, Radomir Mech, and Ulrich Neumann. Disn: Deep implicit surface network for high-quality single-view 3d reconstruction. In *NeurIPS*, 2019. 2

[60] Jiawei Yang, Marco Pavone, and Yue Wang. Freenerf: Improving few-shot neural rendering with free frequency reg-

ularization. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2023. 5

[61] Chun-Han Yao, Amit Raj, Wei-Chih Hung, Yuanzhen Li, Michael Rubinstein, Ming-Hsuan Yang, and Varun Jampani. Artic3d: Learning robust articulated 3d shapes from noisy web image collections. *arXiv preprint arXiv:2306.04619*, 2023. 6

[62] Lior Yariv, Yoni Kasten, Dror Moran, Meirav Galun, Matan Atzmon, Basri Ronen, and Yaron Lipman. Multiview neural surface reconstruction by disentangling geometry and appearance. In *NeurIPS*, pages 2492–2502, 2020. 2

[63] Sai-Kit Yeung, Chi-Keung Tang, Michael S Brown, and Sing Bing Kang. Matting and compositing of transparent and refractive objects. *ACM Transactions on Graphics (TOG)*, 30(1):1–13, 2011. 2, 3

[64] 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. 2

[65] Douglas E Zongker, Dawn M Werner, Brian Curless, and David H Salesin. Environment matting and compositing. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 205–214, 1999. 2, 3