REF^2-NeRF: Reflection and Refraction aware Neural Radiance Field

Wooseok Kim[1], Taiki Fukiage[2] and Takeshi Oishi[1]

***Abstract*— Recently, significant progress has been made in the study of methods for 3D reconstruction from multiple images using implicit neural representations, exemplified by the neural radiance field (NeRF) method. Such methods, which are based on volume rendering, can model various light phenomena, and various extended methods have been proposed to accommodate different scenes and situations. However, when handling scenes with multiple glass objects, e.g., objects in a glass showcase, modeling the target scene accurately has been challenging due to the presence of multiple reflection and refraction effects. Thus, this paper proposes a NeRF-based modeling method for scenes containing a glass case. In the proposed method, refraction and reflection are modeled using elements that are dependent and independent of the viewer's perspective. This approach allows us to estimate the surfaces where refraction occurs, i.e., glass surfaces, and enables the separation and modeling of both direct and reflected light components. The proposed method requires predetermined camera poses, but accurately estimating these poses in scenes with glass objects is difficult. Therefore, we used a robotic arm with an attached camera to acquire images with known poses. Compared to existing methods, the proposed method enables more accurate modeling of both glass refraction and the overall scene.**

I. Introduction

3D reconstruction from 2D images is a well-established technique; however, there is still room for improvement in terms of modeling scenes that involve transparent objects, e.g., glass. Glass is commonly used in the real world; thus, there is a great demand for modeling scenes containing transparent objects [1]–[5]. Unfortunately, common image sensors cannot observe transparent objects directly, and such objects produce various effects, e.g., reflections and refractions, which prevent conventional photogrammetry methods from modeling such scenes correctly.

Recent advances in implicit neural representations [6] of 3D scenes have made it possible to model and synthesize novel views of scenes including reflection and refraction effects [7]. Prior to the introduction of the neural radiance field (NeRF) technique, researchers studied these photometric effects as physical occurrences and developed techniques to replicate scenes based on the physical principles. However, modeling real-world scenes based on physical models from images is an ill-posed problem that requires various constraints. In contrast, NeRF-based methods allow neural networks to learn these complex phenomena to synthesize new views effectively and enable geometry modeling. The ability of these methods to model transparent objects, metallic objects, objects in transparent medias, and objects in liquids without the need for complex models or constraints was a major development in the field.

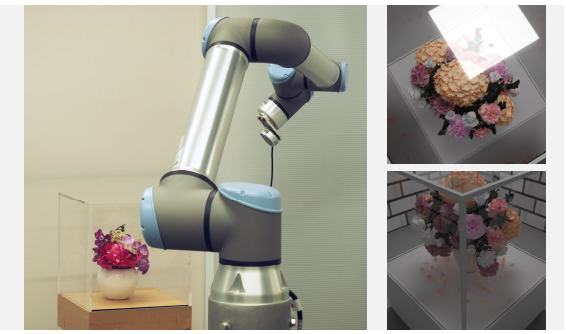


Fig. 1: Image acquisition system and example images of a scene including a glass case and objects. Images of those scenes contain effects of light ray reflection and refraction, which vary depending on viewpoint.

However, even with these techniques, modeling scenes with multiple transparent surfaces remains a challenge. Scenes frequently feature glass objects, as demonstrated by the showcase depicted in Fig. 1. When such objects appear in a scene, they generate multiple reflections and refractions along the viewer's line of sight, which increases the task complexity of generating neural fields that capture the corresponding objects accurately. In such scenes, camera tracking is also difficult, making it even more challenging to refine camera poses while training NeRF.

Thus, in this paper, we propose a neural modeling method that considers the characteristics of scenes containing multiple glass surfaces, particularly objects enclosed in a glass case. The proposed method introduces two networks to handle refraction and reflection independently, and it decomposes the view-independent and view-dependent components of these effects. In the case of refraction, the view-independent component is the refraction point, and the view-dependent component is modification of the ray's direction. Accordingly, the proposed method learns to synthesize new viewpoint images and estimates the position and magnitude of the refraction within the given scene. An additional network decomposes the direct and reflection components in geometric and photometric aspects.

The contributions of this paper can be summarised as follows:

- We propose a neural network-based approach for modeling scenes with multiple glass surfaces, focusing par-

[1]Wooseok Kim and Takeshi Oishi are with Institute of Industrial Science, The University of Tokyo, 153-8505, Tokyo, Japan {kim, oishi}@cvl.iis.u-tokyo.ac.jp

[2]Taiki Fukiage is with NTT Communication Science Laboratories, Kanagawa, Japan taiki.fukiage@ntt.com

ticularly on objects inside a glass case. This approach involves using two separate networks to handle the effects of refraction and reflection.
- We introduce a framework that handles refraction and reflection efficiently by learning the view-dependent and view-independent components separately.
- The proposed method decomposes direct and reflection components in geometric and photometric terms and estimates refraction position and magnitude in the scene.
- We demonstrate that our method works for simulation datasets and real scenes captured by a robotic arm-based measurement system.

II. Related work

In the following, we briefly review previous studies related to the multiview reconstruction and learning of scenes that include transparent objects, e.g., glass.

A. Multi-view method for transparent object

Common image-based 3D reconstruction methods based on the structure from motion and multiview stereo (MVS) [8] [9] [10] [11] techniques estimate the surface geometry using triangulation from feature points or patches. MVS technology has matured recently, and its use is common in various practical applications. In addition, the development of deep learning has made them more robust and accurate. However, strong reflection affects feature and texture matching, and refraction distorts the estimated surface shape, which makes it difficult to reconstruct such scenes using conventional MVS methods.

Methods have also been proposed to model transparent objects and objects behind or within them. For example, several methods operate under a controlled environment, e.g., using background patterns [12] [13] and polarization [14] [15]. However, these methods attempt to estimate the surface shape of the target transparent object; thus, they are unsuitable for modeling regions or objects behind transparent objects. A popular scene where it is difficult to control the environment and target behind or within transparent objects is modeling underwater environments. For underwater scenes, the refraction of light rays occurs at the surface between the lens and the water [16]–[20]. In other words, multiple reflections and refractions are not considered or require prior ray calibration to model more complex scenes.

Some methods employ supervised learning approaches [21]; however, the results are dependent on the available training data, and such methods do not handle complex light effects as well as other model-based methods.

B. Neural Radiance Fields (NeRF)

The NeRF technique [6] can represent various optical phenomena because the trained neural network data, density, and color fields are based on volume rendering [22]. NeRF and its variants optimize the fields represented by implicit neural functions to reduce the similarity error between the input and rendered images. A well-trained NeRF model allows us to reconstruct 3D scenes or synthesize novel views from the neural models. However, the original NeRF has some drawbacks; thus, various methods have been proposed to improve NeRF in terms of acceleration [23]–[25], synthesized image quality enhancement [26]–[28], and robustness against various conditions [29]–[34]. Originally, NeRF was designed to realize novel view synthesis; however, more accurate shape reconstruction is achievable using, for example, signed distance function (SDF)-based approaches [35] [36]. There are also various approaches to efficiently take images and reconstruct 3D scenes with the implicit neural representations using ground-based or aerial robots [37]–[42].

NeRF has shown promising results; however, similar to common 3D reconstruction methods, it assumes a straight light ray and generates poor results when refraction is present in the input images. In addition, reflections in a scene appear as if another scene exists behind the observable reflective and transparent objects, and those reflections may not be visible depending on the viewpoint from which the scene is observed. Note that this phenomenon contradicts the view-consistency assumption of NeRF.

C. NeRF for reflective scenes

Several NeRF variants have been proposed to handle scenes that include reflective surfaces. For example, NeRFReN [43] handles reflections by assuming reflective plane surfaces and separating the scene into transmitted and reflected components utilizing two rendering paths. In addition, NeuS-HSR [44] also estimates an auxiliary plane that separates the reflection components, which facilitate reconstruction of an object inside a glass case. The Neural Transmitted Radiance Fields method [45] detects recurring edges in the input images to optimize transmission and reflection components independently.

The MS-NeRF [46] technique separates the input scene into multiple spaces and estimates density fields and feature fields for these spaces. Here, the feature fields reduce the number of estimated parameters while estimating the color map and weights of the spaces. The neural point catacaustics method [47] introduces a pointwise neural warp field that represents the reflection from curved surfaces, which makes it possible to render the reflected points that are separated from the primary point cloud.

The aforementioned methods can model scenes containing reflective objects effectively; however, they do not consider refraction by transparent objects, and they do not handle scenes containing multiple transparent objects.

D. NeRF for refractive scenes

The LB-NeRF method [48] addresses scenes in which objects are present inside a refractive medium. The LB-NeRF method handles refraction by simplifying it as an offset from a straight light ray. By adding the offsets to each sampled point's position prior to training NeRF's MLP, the LB-NeRF method models canonical space without refraction effects. Other methods based on the physical properties of reflective and refractive medium [49]–[52] require additional

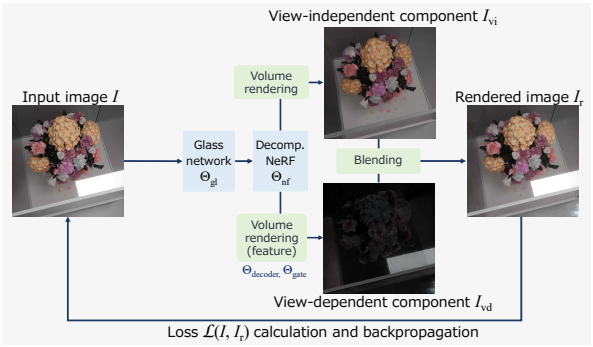


Fig. 2: Overview of the proposed framework. Glass network MLP models refraction occurred by transparent object and adjusted each sampled position. Then, we decompose the scene into view-dependent and view-independent components to separate reflection from input images and model both.

information, e.g., a known image pattern, the refractive index, or a mask image of the refraction, as a clue to detect and estimate the refraction present in an image.

A number of NeRF-based methods have been proposed for reflective or transparent objects [7] [53] [54] [32]; however, it is difficult to apply these methods to the target scene considered in this study for the reasons described above.

III. Preliminary and Overview

In this section, we summarize the refraction and reflection effects and provide an overview of the proposed framework, which is based on these effects.

A. Refraction and reflection effects

Refraction is a phenomenon whereby the direction of light changes due to differences in the refractive index between mediums. Note that refraction also changes according to the incident angle, and it follows Snell's law. Thus, the glass surface where refraction occurs is independent of the viewpoint; however, the amount of refraction depends on the viewpoint. In addition, when light passes through a glass plate, it passes through the parallel boundary in the air-to-glass and glass-to-air order; thus, the ray is parallel to the original ray and is shifted according to the incident angle and the thickness of the glass.

Reflection occurs on the surface of the glass and, similar to refraction, is a phenomenon whereby the path of the light changes. Here, the reflected light intensity is largely distributed in the direction opposite to the incident angle relative to the surface normal, which means that the reflected light intensity is strongly dependent on the viewpoint. Reflections, like mirrors, create a mirror object, i.e., it appears as if the same object is behind the glass. However, it differs from a mirror in that the mirror object is semitransparent because some of the light is reflected on the glass surface, and the remaining light is transmitted into the glass. In other words, in addition to the view-independent objects, view-dependent semitransparent objects can be assumed to exist in the scene.

B. Overview of proposed framework

Based on the above considerations, the proposed framework is designed to estimate the refraction effect and separate the view-independent and view-dependent components present in the given scene. Figure 2 shows an overview of the proposed method. In the proposed method, we assume that the input is multiple images $\{I_k\}(k = 1, 2, ..., n)$ taken from different viewpoints, similar to existing NeRF variants. Here, n is the number of input images, and we omit the index k in this section for simplicity.

The proposed framework primarily comprises two independent MLPs, i.e., $\Theta_{\rm gl}$ and $\Theta_{\rm nf}$, to handle the refractive and reflective components independently, respectively. The former is referred to as the glass network. The glass network represents the refraction points and the amount of refraction, which give the parallel shift of the ray for sampling points in the latter network. The latter is the main NeRF network, which decomposes the direct and reflection components. Here, $\Theta_{\rm nf}$ represents the fields to render an image $I_{\rm vi}$ of a direct component and an image $I_{\rm vd}$ of a reflection component. The density and feature fields of the view-dependent component generate the image through the decoder and gate MLPs: $\Theta_{\rm dc}$, $\Theta_{\rm gt}$ [46]. The blended image of $I_{\rm vi}$ and $I_{\rm vd}$ is the output image $I_{\rm r} = I_{\rm vi} \oplus \alpha I_{\rm vd}$, where α is the blending parameter derived from the network.

The training process optimizes both networks while minimizing the loss $\mathcal{L}$ between the input and rendered images as follows:

$$\hat{\Theta}_{\rm gl}, \hat{\Theta}_{\rm nf}, \hat{\Theta}_{\rm dc}, \hat{\Theta}_{\rm gt} = \underset{\Theta_{\rm gl}, \Theta_{\rm nf}, \Theta_{\rm dc}, \Theta_{\rm gt}}{\arg\min} \mathcal{L}(I, I_{\rm r}). \tag{1}$$

IV. Proposed Framework

This section describes the proposed framework and its implementation in detail. Figure 3 shows the network architecture of the proposed framework and the rendering flow.

A. Glass network

The glass network is employed to estimate the location of the glass surface where the refraction occurs and the amount of refraction. The method used to simplify and express the refraction as an offset is similar that utilized in the LB-NeRF technique [48]. In contrast to LB-NeRF, the proposed method introduces the view-independent density field and view-dependent offset field to estimate the refraction surfaces and the offsets simultaneously.

Here, for a point $\boldsymbol{x}_i \in \mathbb{R}^3, (i = 1, 2, ..., N)$ sampled along a ray $\boldsymbol{r}$, the glass network $\Theta_{\rm gl}$ estimates the glass density $\sigma_{\rm gl}$, which indicates the degree to which that point is involved in the refraction. N is the number of sampled points. $\Theta_{\rm gl}$ also outputs the offset vector $\Delta\boldsymbol{x}_i \in \mathbb{R}^3$, which represents the magnitude and direction of the refraction arising from the view direction $\boldsymbol{d}_i \in \mathbb{R}^3$ and position $\boldsymbol{x}_i$. In other words, the glass network takes the encoded $\boldsymbol{x}$ and $\boldsymbol{d}$ as inputs and outputs $\sigma_{\rm gl}$ and $\Delta\boldsymbol{x}$. This process is expressed as follows:

$$\mathcal{F}_{\Theta_{\rm gl}} : \Gamma(\boldsymbol{x}), \Gamma(\boldsymbol{d}) \rightarrow \sigma_{\rm gl}(\boldsymbol{x}), \Delta\boldsymbol{x}(\boldsymbol{x}, \boldsymbol{d}), \tag{2}$$

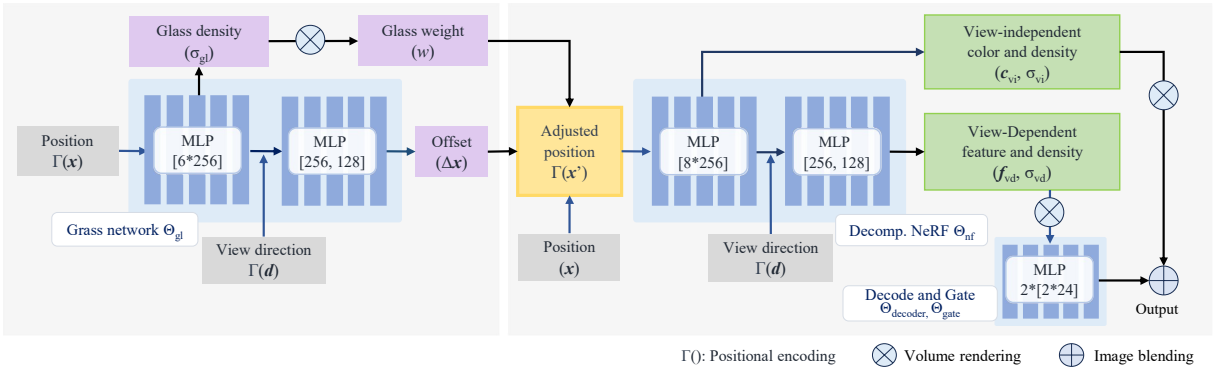


Fig. 3: Network architecture of the proposed method. The glass network outputs the glass density and offset, which modify the ray by the refraction effect through glass walls. Here, the glass density is view-independent, and the offset is view-dependent. The NeRF network takes the adjusted position as input and outputs the view-independent and view-dependent densities and color or feature. The feature renderer provides the corresponding feature map, and the decoder and gate MLPs convert the rendered feature map to a view-dependent image with a blending weight. Finally, the image blending module generates the image by composing the rendered view-independent and view-dependent images. The training process minimizes the loss calculated from the composed image and the input image while optimizing the MLPs.

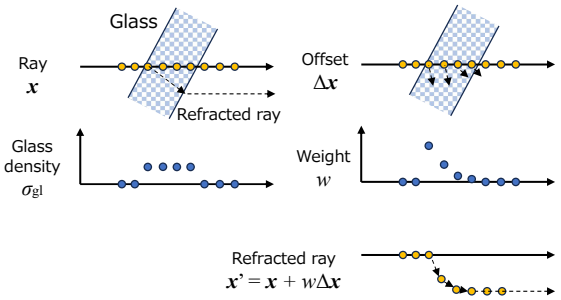


Fig. 4: Structure of the proposed method to express light refraction as volume rendering using glass density to estimate the offset. Here, refraction is simplified as a parallel translation in 3D space occurring on the glass surface. We estimate the path of the light considering refraction by accumulating the vectors of this translation.

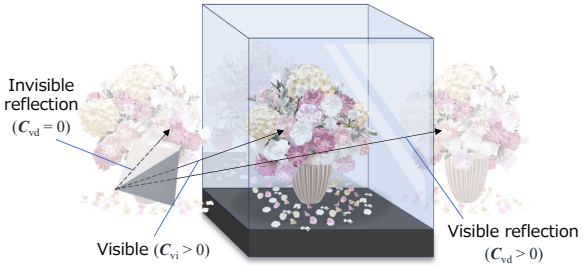


Fig. 5: A proposal method structure that divides the scene into two fields. Elements that do not change depending on the viewpoint, e.g., objects and backgrounds in the scene, are represented in the view-independent field. Elements that do change depending on the viewpoint, e.g., reflections caused by glass and reflections from light sources, are represented in the view-dependent field, where the density changes based on the viewpoint.

where Γ represents the positional encoding.

The sampling points are adjusted after refraction using the offset vectors, as shown in Fig. 4. Similar to NeRF's volume rendering [22], in the proposed method, the refraction weight of each sampling point is calculated using the glass density $\sigma_{\mathrm{gl},i}$ and the distance between adjacent sampling points δ_i as follows:

$$w_i = T_i(1 - exp(-\sigma_{\mathrm{gl},i}\delta_{\mathrm{gl},i})), \tag{3}$$

$$T_i = exp(-\sum_{j=1}^{i-1} \sigma_{\mathrm{gl},j}\delta_{\mathrm{gl},j}). \tag{4}$$

As a result, we obtain the amount of ray shifting that represents the distance each sampling point moves from its original coordinate by adding the weighted offset cumulatively along the ray, and we estimate the adjusted position of the sampling points $\boldsymbol{x}'$ as follows:

$$\boldsymbol{x}'_i = \boldsymbol{x}_i + \sum_{j=1}^{i} w_j \Delta \boldsymbol{x}_j. \tag{5}$$

B. Decomposition NeRF

We assume that an input scene can be separated into the view-independent component, which does not change based on the viewpoint, and the view-dependent component, which does change based on the viewpoint, as shown in Fig. 2. We then define two NeRF-like fields representing each component.

1) View independent NeRF: The view-independent components are represented using density $\sigma_{\mathrm{vi}} \in \mathbb{R}$ and color $\boldsymbol{c}_{\mathrm{vi}} \in \mathbb{R}^3$, similar to the conventional NeRF method. However, the view-independent components do not require the

view direction; thus, in the proposed method, we only use a former part of $\Theta_{\rm nf}$, which takes the position $\boldsymbol{x}'$ as input. By performing volume rendering with the $(\sigma_{\rm vi}, \boldsymbol{c}_{\rm vi})$ of the points sampled on a ray $\boldsymbol{r}$, we obtain the view-independent color $\boldsymbol{C}_{\rm vi}$ of a pixel corresponding to that ray as follows:

$$\boldsymbol{C}_{\rm vi}(\boldsymbol{r}) = \sum_{i=1}^{N} T_{{\rm vi},i}(1 - exp(-\sigma_{{\rm vi},i}\delta_{{\rm vi},i}))\boldsymbol{c}_{{\rm vi},i}. \quad (6)$$

Note that the calculation of $T_{{\rm vi},i}$ is the same as given in Eq. 4.

2) View dependent NeRF: The view-dependent components are separated using the feature field approach [46]. MS-NeRF introduced the feature field, which extracts multiple spaces explicitly as the density and feature fields, and it functions effectively for scenes with several mirrors. However, multiple glass plates generate reciprocal reflections of objects and light sources; thus, representing all spaces individually with a number of spaces is a highly complex task.

In the proposed method, we address this problem using a single view-dependent feature field in addition to the previously describe view-independent NeRF. The feature field is represented using $\Theta_{\rm nf}$, which estimates the view-dependent density $\sigma_{\rm vd}$ and the θ-dimensional feature vector $\boldsymbol{f}_{\rm vd}$ from an adjusted position $\boldsymbol{x}'$ and view direction $\boldsymbol{d}$ as follows:

$$\mathcal{F}_{\Theta_{\rm nf}} : \Gamma(\boldsymbol{x}'), \Gamma(\boldsymbol{d}) \rightarrow \sigma_{\rm vi}(\boldsymbol{x}'), \boldsymbol{c}_{\rm vi}(\boldsymbol{x}'), \sigma_{\rm vd}(\boldsymbol{x}', \boldsymbol{d}), \boldsymbol{f}_{\rm vd}(\boldsymbol{x}', \boldsymbol{d}). \quad (7)$$

We obtain the feature vector $\boldsymbol{F}_{\rm vd}$ corresponding to a ray $\boldsymbol{r}$ by volume rendering along the ray for $\sigma_{\rm vd}$ and $\boldsymbol{f}_{\rm vd}$, In addition, we estimate the color $\boldsymbol{C}_{\rm vd}$ using a decoder MLP $\Theta_{\rm dc}$ and determine the blending parameter α using a gate MLP $\Theta_{\rm gt}$ [46]. Here, $\boldsymbol{C}_{\rm vd}(\boldsymbol{r})$ represents the reflection component corresponding to the pixel of the ray.

$$\boldsymbol{F}_{\rm vd}(\boldsymbol{r}) = \sum_{i=1}^{N} T_i(1 - exp(-\sigma_{{\rm vd},i}\delta_{{\rm vd},i}))\boldsymbol{f}_i, \quad (8)$$

$$\mathcal{F}_{\Theta_{\rm dc}} : \boldsymbol{F}_{\rm vd}(\boldsymbol{r}) \rightarrow \boldsymbol{C}_{\rm vd}(\boldsymbol{r}), \quad (9)$$

$$\mathcal{F}_{\Theta_{\rm gt}} : \boldsymbol{F}_{\rm vd}(\boldsymbol{r}) \rightarrow \alpha(\boldsymbol{r}). \quad (10)$$

C. Optimization

Finally, we find the color $\boldsymbol{C}(\boldsymbol{r})$ of the pixel corresponding to the ray as follows:

$$\boldsymbol{C}(\boldsymbol{r}) = \boldsymbol{C}_{\rm vi}(\boldsymbol{r}) + \alpha(\boldsymbol{r})\boldsymbol{C}_{\rm vd}(\boldsymbol{r}). \quad (11)$$

We train the MLPs $\{\Theta_{\rm gl}$,$\Theta_{\rm nf}$, $\Theta_{\rm dc}$, and $\Theta_{\rm gt}\}$ by evaluating the rendered pixel color $\boldsymbol{C}(\boldsymbol{r})$ with that of the input image $\bar{\boldsymbol{C}}(\boldsymbol{r})$. Here, we utilize the same loss function utilized in the conventional NeRF method, i.e., the summation of L2 distances, which is expressed as follows:

$$\mathcal{L}_{\rm render} = \sum_{\boldsymbol{r}\in\boldsymbol{R}} \|\bar{\boldsymbol{C}}(\boldsymbol{r}) - \boldsymbol{C}(\boldsymbol{r})\|^2, \quad (12)$$

where $\boldsymbol{R}$ is a batch of sampled rays. In addition, we introduce L2 regularization loss $\mathcal{L}_{\rm offset}$ to train the glass network in a stable manner. This regularization loss prevents the neural network from learning biased offset and parallelly shifted scenes.

$$\mathcal{L}_{\rm offset} = \sqrt{\sum_{\boldsymbol{x},\boldsymbol{d}} (\Delta\boldsymbol{x})^2} \quad (13)$$

The entire loss $\mathcal{L}$ is an integration of these two loss functions:

$$\mathcal{L} = \mathcal{L}_{\rm render} + \epsilon\mathcal{L}_{\rm offset}, \quad (14)$$

where ϵ is a small number, which was set to $\epsilon = 10^{-5}$ in our experiments.

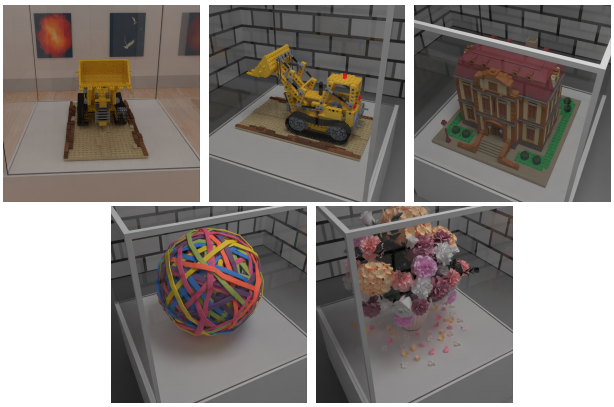


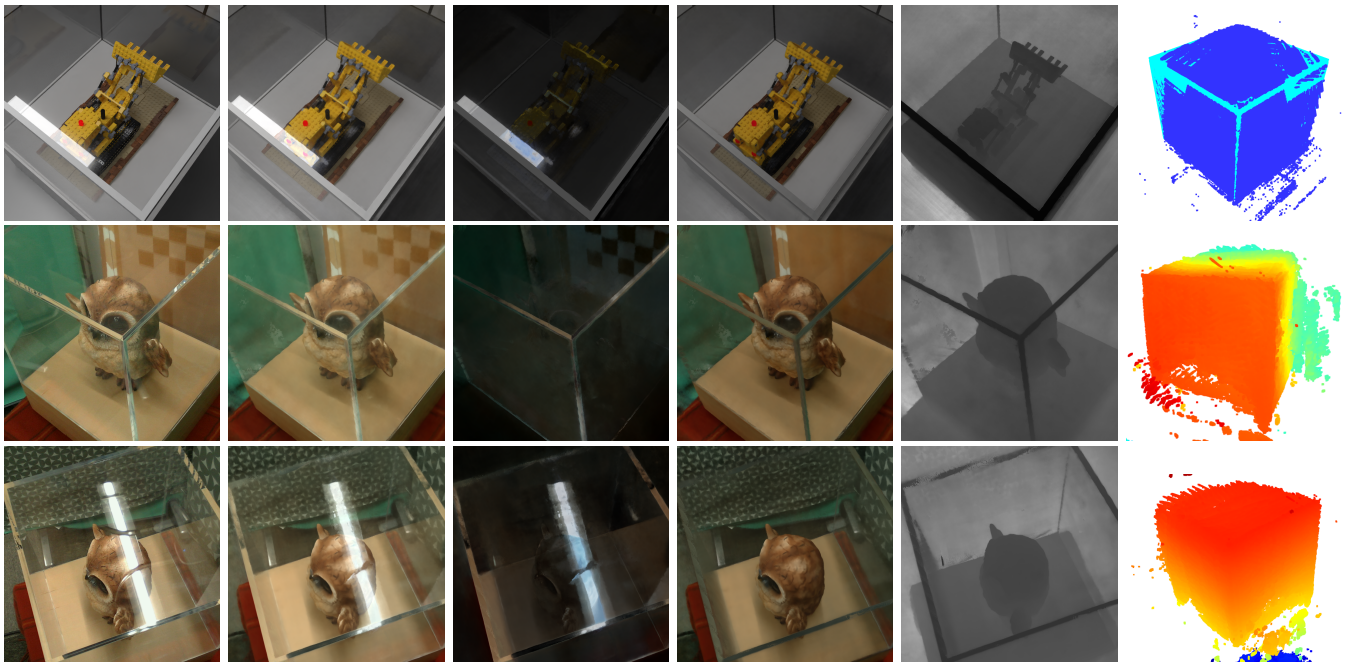
Fig. 6: Example images of simulation dataset. Each image (from left-top to right-bottom) shows input images for Lego (Gallery), Lego, House, Color Ball, and Flower, respectively.

Fig. 7: Example images of real-world dataset. Each image (from left-top to right-bottom) shows input images for Owl (simple bg), Owl (textured bg), Owl (white bg), Lion (white bg), Dog (white bg), respectively.

V. Experiment

A. Experimental dataset

1) Simulation dataset: In this study, we generated a simulation dataset using Blender [55]. The experimental dataset includes several scenes containing a glass showcase with the size of 50 cm × 50 cm × 50 cm surrounded by walls with textures. Here, the thickness of the glass is 1 cm, and the refractive index is 1.45. Inside this showcase is an object in {Lego, House, Color Ball, Flower}. We also generated a dataset of Lego object placed in art gallery,



(a) Ground truth (b) Rendered image (c) View-dependent component (d) View-independent component (e) View-independent depth (f) Refraction points (glass surface)

Fig. 8: Result images of our proposal method modeling test datasets. The top row shows results from the simulation dataset and the bottom two rows show results from the real-world dataset.(a) Ground-truth image of scene viewed from a test pose. (b) The image rendered by the proposed method. (c) The view-dependent component and (d) view-independent component modeled by the proposed method. (e) Depth map image of the view-independent component. (f) Points where refraction greater than a threshold occurred as a point cloud.

which is surrounded by wall with paintings. We placed a ceiling light with area in the scene and utilized Blender's BSDF model to render the scene with physical simulations of both the light and the glass. One set of a scene comprised 200 training images, and both the test and validation sets contained 25 images each with their respective intrinsic and extrinsic camera parameters.

2) Real world dataset: We developed an image-capturing system to take images using a camera, Realsense L515, and a robotic arm, UR-10e, as shown in Fig. 1. Note that though L515 is an RGB-D camera, we did not use the depth information for 3D reconstruction. Visual camera tracking methods struggle to function correctly when glass occupies a large portion of each image, presenting challenges in accurately estimating camera poses. Therefore, we used the robot arm to take the images from known camera poses [56], [57]. We utilized 170 images, taken as uniformly as possible, covering a quarter sphere primarily near the arm for the training. The case size is 30 cm $\times$ 30 cm $\times$ 30 cm, and the thickness of the glass is 8 mm.

B. Implementation and training

We implemented the proposed method based on NeRF-Pytorch [58]. The training process sampled the rays corresponding to 1,024 pixels from a randomly selected training image in each iteration. In addition, each ray initially sampled 128 points at uniform intervals. In the coarse-to-fine strategy of NeRF, an additional 64 points are sampled in a hierarchical manner for segments with higher densities inferred from the coarse model. The proposed method samples an additional 32 points using the glass density and 32 points using the view-independent density. This results in a total of 192 sampling points being input into the fine model. We set the feature vector's dimension θ to 64. We performed a total of 200,000 training iterations on an Nvidia RTX 4080 graphics processing unit, which took approximately 10 hours for a single scene.

C. Evaluation

Figure 8 shows examples of the images obtained by the proposed method using test images and corresponding camera poses. Here, each column (from left to right) shows the ground-truth image (i.e., the test image), the rendered image, the view-dependent component, the view-independent component, the corresponding depth image, and the point-cloud of refraction points.

The view-dependent images with specular reflection components demonstrate that the reflections were extracted correctly by the proposed method. In addition, the view-dependent image also shows that the reflection component was removed effectively. The depth image was estimated correctly, which indicates that the refraction was estimated well, and the reflective component, which is problematic for shape estimation, was removed effectively.

TABLE I: Results of each Dataset and methods (simulation dataset)

Method	Lego			House			Color Ball			Flower			Lego (Gallery)		
	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
NeRF [6]	29.0111	0.8893	0.2984	30.9108	0.9029	0.2951	29.2773	0.9054	0.2766	27.8217	0.8557	0.3351	31.7363	0.9184	0.2772
Mip-NeRF [26]	28.9192	0.8923	0.2979	30.6958	0.9100	0.2838	29.9715	0.9211	0.2346	27.9082	0.8711	0.3055	32.1508	0.9267	0.2565
Ref-NeRF [32]	28.2375	0.8679	0.3357	29.6852	0.8836	0.3262	29.0698	0.8944	0.2880	27.3446	0.8396	0.3642	31.3128	0.9104	0.3011
LB-NeRF [48]	29.8683	0.9094	0.2711	30.9116	0.9028	0.2980	30.0377	0.9151	0.2525	28.4501	0.8683	0.3198	32.4674	0.9324	0.2564
MS-NeRF [46]	31.0450	0.9060	0.2817	33.3059	0.9231	0.2822	31.6074	0.9230	0.2594	29.1936	0.8740	0.3197	33.0800	0.9274	0.2665
Proposal Method	**33.1834**	**0.9357**	**0.2073**	**35.3665**	**0.9483**	**0.1929**	**33.0759**	**0.9453**	**0.1842**	**30.6254**	**0.8958**	**0.2637**	**35.3476**	**0.9507**	**0.1913**

TABLE II: Results of each Dataset and methods (real-world dataset)

Method	Owl (simple bg)			Owl (textured bg)			Owl (white bg)			Lion (white bg)			Dog (white bg)		
	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
LB-NeRF [48]	26.1937	0.8388	0.4037	25.2232	0.7725	0.4484	26.1833	0.8649	0.4370	26.3987	0.8427	0.4562	26.5699	0.8577	0.4543
MS-NeRF [46]	**27.5344**	**0.8599**	**0.3496**	**26.8352**	**0.8006**	0.4048	**28.2428**	**0.8748**	**0.3984**	28.5226	0.8722	0.4177	**27.7415**	**0.8691**	**0.4191**
Proposal Method	26.7532	0.8541	0.3823	26.6856	0.8001	**0.3877**	27.4028	0.8721	0.4041	**28.6274**	**0.8793**	**0.4162**	27.2331	0.8656	0.4336

The proposed method estimates the points where refraction occurs explicitly; thus, we can evaluate the accuracy of the estimated refractive surface by using the simulation dataset to determine how close these points are to the original glass surface. We determined that the estimated glass surface point where the offset $\Delta x \times w$ is greater than a threshold value (0.01 cm in this experiment) when rendering test images. The estimated glass surface is shown in Fig. 8 (f). The average errors of the estimated glass surfaces for Lego, House, Color Ball, Flower, Lego (Gallery) were 0.4657 cm, 0.4728 cm, 0.3553 cm, 0.2693 cm, 0.9662 cm, respectively. Although there are some outliers, it can be seen that reasonably good results were obtained by the proposed method.

D. Comparative evaluation

In this evaluation, the proposed method was compared with several NeRF-based methods. Here, we applied the original NeRF [6], Mip-NeRF [26], Ref-NeRF [32], LB-NeRF [48], and MS-NeRF [46] methods to the simulation dataset constructed in this study. Note that no open source code is available for LB-NeRF; thus, we implemented its structure by adding a 3D offset estimated from a concatenation of the 3D point's positions and view direction using a fully-connected neural network with seven layers containing 256 nodes. In this evaluation, evaluation metrics commonly used for image comparison were used to assess the compared methods, i.e., the peak signal-to-noise Ratio (PSNR), the structural similarity index measure (SSIM) [59], and the learned perceptual image patch similarity (LPIPS) [60].

Table I shows the results on the simulation dataset. As can be seen, the proposed method outperformed all compared methods on each dataset. Among the compared method, the MS-NeRF method obtained comparatively superior results. Note that the MS-NeRF method does not clearly determine the decomposed spaces for nonmirror surfaces; thus, it enhances the image synthesis precision by separating the intense reflective components associated with glass.

Table II shows the results on the real-world dataset. Based on the results of the simulation data, Table II contains only comparisons with LB-NeRF and MS-NeRF. In the context of novel view synthesis, the proposed method and MS-NeRF had similar scores. Due to the difference in the base method, the performance is slightly different depending on the background, illumination, and target object. However, as mentioned above, the proposed method is superior to other methods because it can separate the reflection components and estimate the refraction position and magnitude.

VI. Conclusion

In this paper, we have proposed a method that utilizes implicit neural representations to model scenes with objects enclosed in a glass case. The proposed method distinguishes between the refraction and reflection effects by learning them with view-independent and view-dependent components, and by determining the refraction points indicative of the glass surfaces simultaneously. The proposed method was evaluated experimentally, and the experimental results demonstrate that the proposed method is proficient in terms of separating the components that vary with viewpoint from those that do not. On the other hand, as explained in Sec. V-C, the separation of view-dependent and view-independent components may not be perfect. Improving the accuracy of component separation is one of the future works.

References

[1] U. Klank, D. Carton, and M. Beetz, "Transparent object detection and reconstruction on a mobile platform," in *ICRA*, 2011, pp. 5971–5978.

[2] S. Sajjan, M. Moore, M. Pan, G. Nagaraja, J. Lee, A. Zeng, and S. Song, "Clear grasp: 3d shape estimation of transparent objects for manipulation," in *ICRA*, 2020, pp. 3634–3642.

[3] H. Fang, H.-S. Fang, S. Xu, and C. Lu, "Transcg: A large-scale real-world dataset for transparent object depth completion and a grasping baseline," *IEEE RA-L*, vol. 7, no. 3, pp. 7383–7390, 2022.

[4] J. Jiang, G. Cao, T.-T. Do, and S. Luo, "A4t: Hierarchical affordance detection for transparent objects depth reconstruction and manipulation," *IEEE RA-L*, vol. 7, no. 4, pp. 9826–9833, 2022.

[5] Y. Tang, J. Chen, Z. Yang, Z. Lin, Q. Li, and W. Liu, "Depthgrasp: Depth completion of transparent objects using self-attentive adversarial network with spectral residual for grasping," in *IROS*, 2021, pp. 5710–5716.

[6] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," in *ECCV*, 2020.

[7] J. Ichnowski*, Y. Avigal*, J. Kerr, and K. Goldberg, "Dex-NeRF: Using a neural radiance field to grasp transparent objects," in *CoRL*, 2020.

[8] S. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski, "A comparison and evaluation of multi-view stereo reconstruction algorithms," in *CVPR*, vol. 1, 2006, pp. 519–528.

[9] Y. Furukawa and J. Ponce, "Accurate, dense, and robust multiview stereopsis," *IEEE PAMI*, vol. 32, no. 8, pp. 1362–1376, 2009.

[10] J. L. Schonberger and J.-M. Frahm, "Structure-from-motion revisited," in *CVPR*, 2016, pp. 4104–4113.

[11] C. Griwodz, S. Gasparini, L. Calvet, P. Gurdjos, F. Castan, B. Maujean, G. De Lillo, and Y. Lanthony, "Alicevision meshroom: An open-source 3d reconstruction pipeline," in *ACM MM*, 2021, pp. 241–247.

[12] J. Lyu, B. Wu, D. Lischinski, D. Cohen-Or, and H. Huang, "Differentiable refraction-tracing for mesh reconstruction of transparent objects," *ACM TOG*, vol. 39, no. 6, nov 2020.

[13] B. Wu, Y. Zhou, Y. Qian, M. Cong, and H. Huang, "Full 3d reconstruction of transparent objects," *ACM TOG*, vol. 37, no. 4, pp. 1–11, 2018.

[14] D. Miyazaki, M. Kagesawa, and K. Ikeuchi, "Transparent surface modeling from a pair of polarization images," *IEEE PAMI*, vol. 26, no. 1, pp. 73–82, 2004.

[15] Y. Lyu, Z. Cui, S. Li, M. Pollefeys, and B. Shi, "Reflection separation using a pair of unpolarized and polarized images," in *NeurIPS*, vol. 32, 2019.

[16] X. Qiao, A. Yamashita, and H. Asama, "Underwater structure from motion for cameras under refractive surfaces," *Journal of Robotics and Mechatronics*, vol. 31, no. 4, pp. 603–611, 2019.

[17] C. Beall, B. J. Lawrence, V. Ila, and F. Dellaert, "3d reconstruction of underwater structures," in *IROS*, 2010, pp. 4418–4423.

[18] A. Jordt, K. Köser, and R. Koch, "Refractive 3d reconstruction on underwater images," *Methods in Oceanography*, vol. 15, pp. 90–113, 2016.

[19] F. Chadebecq, F. Vasconcelos, G. Dwyer, R. Lacher, S. Ourselin, T. Vercauteren, and D. Stoyanov, "Refractive structure-from-motion through a flat refractive interface," in *ICCV*, 2017, pp. 5325–5333.

[20] T. Masuda, "3d shape reconstruction of plant roots in a cylindrical tank from multiview images," in *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, 2019, pp. 2149–2157.

[21] Z. Li, Y.-Y. Yeh, and M. Chandraker, "Through the looking glass: neural 3d reconstruction of transparent shapes," in *CVPR*, 2020, pp. 1262–1271.

[22] N. Max, "Optical models for direct volume rendering," *IEEE TVCG*, vol. 1, no. 2, pp. 99–108, 1995.

[23] T. Müller, A. Evans, C. Schied, and A. Keller, "Instant neural graphics primitives with a multiresolution hash encoding," *ACM TOG*, vol. 41, no. 4, pp. 1–15, 2022.

[24] S. Fridovich-Keil, A. Yu, M. Tancik, Q. Chen, B. Recht, and A. Kanazawa, "Plenoxels: Radiance fields without neural networks," in *CVPR*, 2022, pp. 5501–5510.

[25] A. Yu, R. Li, M. Tancik, H. Li, R. Ng, and A. Kanazawa, "Plenoctrees for real-time rendering of neural radiance fields," in *ICCV*, 2021, pp. 5752–5761.

[26] J. T. Barron, B. Mildenhall, M. Tancik, P. Hedman, R. Martin-Brualla, and P. P. Srinivasan, "Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields," in *ICCV*, 2021, pp. 5855–5864.

[27] J. T. Barron, B. Mildenhall, D. Verbin, P. P. Srinivasan, and P. Hedman, "Mip-nerf 360: Unbounded anti-aliased neural radiance fields," in *CVPR*, 2022, pp. 5470–5479.

[28] C. Wang, X. Wu, Y.-C. Guo, S.-H. Zhang, Y.-W. Tai, and S.-M. Hu, "Nerf-sr: High quality neural radiance fields using supersampling," in *ACM MM*, 2022, pp. 6445–6454.

[29] A. Yu, V. Ye, M. Tancik, and A. Kanazawa, "pixelnerf: Neural radiance fields from one or few images," in *CVPR*, 2021, pp. 4578–4587.

[30] C.-H. Lin, W.-C. Ma, A. Torralba, and S. Lucey, "Barf: Bundle-adjusting neural radiance fields," in *ICCV*, 2021, pp. 5741–5751.

[31] R. Martin-Brualla, N. Radwan, M. S. Sajjadi, J. T. Barron, A. Dosovitskiy, and D. Duckworth, "Nerf in the wild: Neural radiance fields for unconstrained photo collections," in *CVPR*, 2021, pp. 7210–7219.

[32] D. Verbin, P. Hedman, B. Mildenhall, T. Zickler, J. T. Barron, and P. P. Srinivasan, "Ref-nerf: Structured view-dependent appearance for neural radiance fields," in *CVPR*, 2022, pp. 5481–5490.

[33] K. Park, U. Sinha, J. T. Barron, S. Bouaziz, D. B. Goldman, S. M. Seitz, and R. Martin-Brualla, "Nerfies: Deformable neural radiance fields," in *ICCV*, 2021, pp. 5865–5874.

[34] Z. Wang, S. Wu, W. Xie, M. Chen, and V. A. Prisacariu, "Nerf–: Neural radiance fields without known camera parameters," *arXiv preprint arXiv:2102.07064*, 2021.

[35] P. Wang, L. Liu, Y. Liu, C. Theobalt, T. Komura, and W. Wang, "Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction," *NeurIPS*, 2021.

[36] Y. Wang, Q. Han, M. Habermann, K. Daniilidis, C. Theobalt, and L. Liu, "Neus2: Fast learning of neural implicit surfaces for multi-view reconstruction," in *ICCV*, 2023.

[37] J. Zeng, Y. Li, Y. Ran, S. Li, F. Gao, L. Li, S. He, J. Chen, and Q. Ye, "Efficient view path planning for autonomous implicit reconstruction," in *ICRA*, 2023, pp. 4063–4069.

[38] S. Lee, L. Chen, J. Wang, A. Liniger, S. Kumar, and F. Yu, "Uncertainty guided policy for active robotic 3d reconstruction using neural radiance fields," *IEEE RA-L*, vol. 7, no. 4, pp. 12 070–12 077, 2022.

[39] L. Jin, X. Chen, J. Rückin, and M. Popović, "Neu-nbv: Next best view planning using uncertainty estimation in image-based neural rendering," in *IROS*, 2023, pp. 11 305–11 312.

[40] Y. Ran, J. Zeng, S. He, J. Chen, L. Li, Y. Chen, G. Lee, and Q. Ye, "Neurar: Neural uncertainty for autonomous 3d reconstruction with implicit neural representations," *IEEE RA-L*, vol. 8, no. 2, pp. 1125–1132, 2023.

[41] D. Yan, J. Liu, F. Quan, H. Chen, and M. Fu, "Active implicit object reconstruction using uncertainty-guided next-best-view optimization," *IEEE RA-L*, 2023.

[42] S. Zhou, S. Xie, R. Ishikawa, K. Sakurada, M. Onishi, and T. Oishi, "Inf: Implicit neural fusion for lidar and camera," in *IROS*, 2023, pp. 10 918–10 925.

[43] Y.-C. Guo, D. Kang, L. Bao, Y. He, and S.-H. Zhang, "Nerfren: Neural radiance fields with reflections," in *CVPR*, 2022, pp. 18 409–18 418.

[44] J. Qiu, P.-T. Jiang, Y. Zhu, Z.-X. Yin, M.-M. Cheng, and B. Ren, "Looking through the glass: Neural surface reconstruction against high specular reflections," in *CVPR*, 2023, pp. 20 823–20 833.

[45] C. Zhu, R. Wan, and B. Shi, "Neural transmitted radiance fields," in *NeurIPS*, vol. 35. Curran Associates, Inc., 2022, pp. 38 994–39 006.

[46] Z.-X. Yin, J. Qiu, M.-M. Cheng, and B. Ren, "Multi-space neural radiance fields," in *CVPR*, June 2023, pp. 12 407–12 416.

[47] G. Kopanas, T. Leimkühler, G. Rainer, C. Jambon, and G. Drettakis, "Neural point catacaustics for novel-view synthesis of reflections," *ACM TOG*, vol. 41, no. 6, nov 2022.

[48] T. Fujitomi, K. Sakurada, R. Hamaguchi, H. Shishido, M. Onishi, and Y. Kameda, "Lb-nerf: light bending neural radiance fields for transparent medium," in *ICIP*, 2022, pp. 2142–2146.

[49] J.-I. Pan, J.-W. Su, K.-W. Hsiao, T.-Y. Yen, and H.-K. Chu, "Sampling neural radiance fields for refractive objects," in *SIGGRAPH Asia 2022 Technical Communications*, 2022, pp. 1–4.

[50] Z. Wang, W. Yang, J. Cao, Q. Hu, L. Xu, J. Yu, and J. Yu, "Neref: Neural refractive field for fluid surface reconstruction and rendering," in *ICCP*, 2023, pp. 1–11.

[51] J. Tong, S. Muthu, F. A. Maken, C. Nguyen, and H. Li, "Seeing through the glass: Neural 3d reconstruction of object inside a transparent container," in *CVPR*, June 2023, pp. 12 555–12 564.

[52] Y. Zhan, S. Nobuhara, K. Nishino, and Y. Zheng, "Nerfrac: Neural radiance fields through refractive surface," in *ICCV*, October 2023, pp. 18 402–18 412.

[53] J. Kerr, L. Fu, H. Huang, Y. Avigal, M. Tancik, J. Ichnowski, A. Kanazawa, and K. Goldberg, "Evo-nerf: Evolving nerf for sequential robot grasping of transparent objects," in *CoRL*, 2022.

[54] Q. Dai, Y. Zhu, Y. Geng, C. Ruan, J. Zhang, and H. Wang, "Graspnerf: Multiview-based 6-dof grasp detection for transparent and specular objects using generalizable nerf," in *ICRA*, 2023, pp. 1757–1763.

[55] "blender.org - home of the blender project - free and open 3d creation software." [Online]. Available: https://www.blender.org/

[56] Y. Wang, S. James, E. K. Stathopoulou, C. Beltrán-González, Y. Konishi, and A. Del Bue, "Autonomous 3-d reconstruction, mapping, and exploration of indoor environments with a robotic arm," *IEEE RA-L*, vol. 4, no. 4, pp. 3340–3347, 2019.

[57] H. Aanæs, R. R. Jensen, G. Vogiatzis, E. Tola, and A. B. Dahl, "Large-scale data for multiple-view stereopsis," *IJCV*, vol. 120, pp. 153–168, 2016.

[58] L. Yen-Chen, "Nerf-pytorch," https://github.com/yenchenlin/nerf-pytorch/, 2020.

[59] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE TIP*, vol. 13, no. 4, pp. 600–612, 2004.

[60] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *CVPR*, 2018, pp. 586–595.