Transparent Object Reconstruction via Implicit Differentiable Refraction Rendering

Fangzhou Gao*
Tianjin University
China
gaofangzhou@tju.edu.cn

Lianghao Zhang*
Tianjin University
China
opoiiuiouiuy@tju.edu.cn

Li Wang
Tianjin University
China
li_wang@tju.edu.cn

Jiamin Cheng
Tianjin University
China
cjm@tju.edu.cn

Jiawan Zhang†
Tianjin University
China
jwzhang@tju.edu.cn

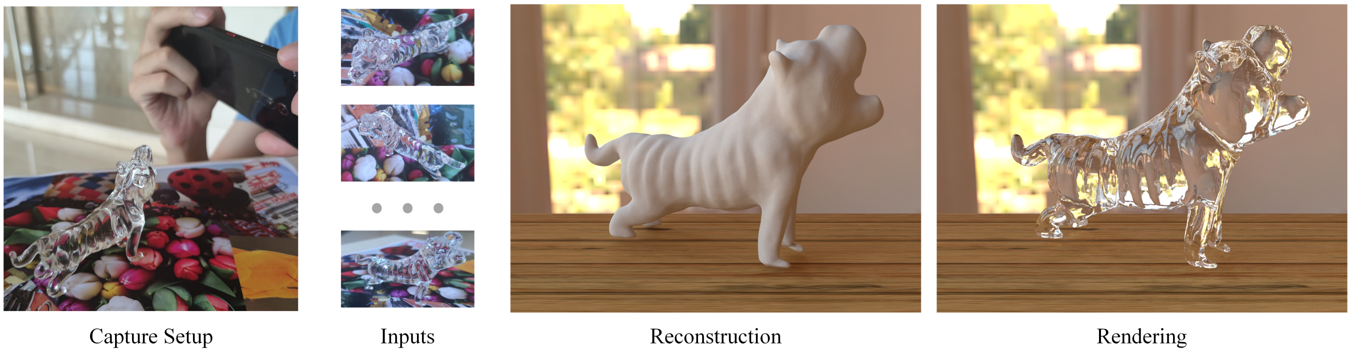


Figure 1: In this paper, we introduce a novel method to reconstruct transparent objects in natural scenes using straightforward setups. Specifically, for a transparent object positioned on an arbitrary planar surface, our method solely relies on its multi-view RGB images as input to achieve high-precision geometry reconstruction. On the right side, we present the reconstructed shape and render it as a transparent object in a new environment.

ABSTRACT

Reconstructing the geometry of transparent objects has been a long-standing challenge. Existing methods rely on complex setups, such as manual annotation or darkroom conditions, to obtain object silhouettes and usually require controlled environments with designed patterns to infer ray-background correspondence. However, these intricate arrangements limit the practical application for common users. In this paper, we significantly simplify the setups and present a novel method that reconstructs transparent objects in unknown natural scenes without manual assistance. Our method incorporates two key technologies. Firstly, we introduce a volume rendering-based method that estimates object silhouettes by projecting the 3D neural field onto 2D images. This automated process yields highly accurate multi-view object silhouettes from images captured in natural scenes. Secondly, we propose transparent object optimization through differentiable refraction rendering with the neural SDF field, enabling us to optimize the refraction ray based on color rather than explicit ray-background correspondence. Additionally, our optimization includes a ray sampling method to supervise the object silhouette at a low computational cost. Extensive experiments and comparisons demonstrate that our method produces high-quality results while offering much more convenient setups.

CCS CONCEPTS

• **Computing methodologies → Reconstruction**; **Mesh geometry models**.

KEYWORDS

Transparent Object, Multi-view Reconstruction, Neural Rendering

ACM Reference Format:
Fangzhou Gao, Lianghao Zhang, Li Wang, Jiamin Cheng, and Jiawan Zhang. 2023. Transparent Object Reconstruction via Implicit Differentiable Refraction Rendering. In *SIGGRAPH Asia 2023 Conference Papers (SA Conference*

*Equal contribution
†Corresponding authors.

SA Conference Papers '23, December 12–15, 2023, Sydney, NSW, Australia
© 2023 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 979-8-4007-0315-7/23/12…$15.00
https://doi.org/10.1145/3610548.3618236

Papers '23), December 12–15, 2023, Sydney, NSW, Australia. ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3610548.3618236

1 INTRODUCTION

The distinctive properties of transparent objects caused by refraction and reflection present a longstanding challenge in their reconstruction. The complex light path consisting of multiple segments impedes the correspondence matching used in most common approaches.

When reconstructing a transparent object, it is essential to take the environment of the object into account in order to analyze and constrain the refractive light path. Previous methods either model the environment as known environment lighting to feed to a pre-trained network [Li et al. 2020] or require unique patterns for inferring the correspondence between the camera ray and the background [Lyu et al. 2020; Wu et al. 2018; Xu et al. 2022]. However, these rigorous prerequisites result in elaborate setups in practice, such as capturing the environment map in advance or deploying monitors that display designed patterns. Besides, they all rely on manual annotation [Li et al. 2020; Xu et al. 2022] or a dark room [Lyu et al. 2020; Wu et al. 2018] to extract multi-view object silhouettes. Such intricate setups impose a heavy burden on common users and restrict practical application.

But it is challenging to simplify the setups due to two problems. Firstly, optimizing the multi-segment refraction light path is a severely ill-posed problem[Kutulakos and Steger 2008], which heavily relies on object silhouettes to provide additional constraints. However, simple image-based segmentation methods can hardly estimate accurate silhouettes that are consistent across views. Secondly, while optimizing the refractive ray based on color instead of explicit ray-background correspondence is a promising approach, supervising the color would exacerbate the ambiguity and result in self-intersections, folding, and high-frequency artifacts on explicit mesh representation even recent hybrid representation[Xu et al. 2022].

In this paper, we address these two problems and introduce a novel method that automatically reconstructs transparent objects in uncontrolled natural scenes from multi-view RGB images. Our approach significantly simplifies the setup required for transparent object reconstruction, making it as convenient as the multi-view stereo used for opaque objects. This achievement is attributed to the incorporation of two key technologies.

Firstly, we observe and analyze the shape-radiance ambiguity that arises when using neural volume rendering for transparent objects. While the shape of the transparent object remains inaccurate through volume rendering, we propose to project the 3D neural field back to each input view and determine whether a ray hit the transparent object. Through the projection, the positive samples in silhouettes can be recovered by the imperfect transparent surface and the negative samples are recovered by the well-recovered opaque surroundings. These multi-view silhouettes provide strong regularization for the following object reconstruction.

Secondly, we leverage the input images and the estimated silhouettes to reconstruct the transparent object, represented by a neural SDF field. The neural implicit representation avoids the discretization artifacts during optimization. The reconstruction primarily relies on the implicit differentiable refraction rendering with the neural SDF field. This approach allows us to optimize both the refractive light paths and the object's shape to be close to the real case by enforcing the reconstructed object to refract the same color as the input image. Additionally, the reconstruction also contains a ray sampling method that selects the most important rays to constrain the object silhouettes at a low computational cost.

In summary, we present, to our knowledge, the first method that reconstructs transparent objects in uncontrolled natural scenes with only multi-view RGB photographs as input. It greatly simplifies the setups for transparent object reconstruction, which is contributed to the following technology contributions:

- The projection method that estimates accurate multi-view 2D object silhouettes from the 3D neural field.
- The transparent object optimization through the differentiable refraction rendering with neural SDF field.
- The ray sampling for low-cost silhouette constraint.

Experimental evaluations conducted on synthetic and real data validate the superiority of our proposed method. Our approach achieves even better results than previous methods, with much more convenient setups.

2 RELATED WORK

In this section, we review the research on the reconstruction of transparent surfaces. Besides, we briefly review the multi-view reconstruction for opaque objects through neural rendering, which inspires us to adopt neural rendering to transparent object reconstruction.

2.1 Transparent Surface Reconstruction

Reconstructing a transparent surface is challenging due to its unique optical property. Various special hardware and setups were introduced, such as light field probe[Wetzstein et al. 2011], polarizing camera[Huynh et al. 2010; Miyazaki and Ikeuchi 2005], depth camera [Alt et al. 2013; Tanaka et al. 2016], and tomography[Trifonov et al. 2006]. More details can be found in the review [Ihrke et al. 2010]. Besides, a series of methods were proposed to reconstruct transparent surfaces with consumer cameras, which can be divided into reconstruction in controlled environments and in natural scenes.

Controlled Environment. Kutulakos et al. [2008] analyzed the light path triangulation for transparent surfaces. Based on it, a series of methods were proposed to reconstruct the single [Morris and Kutulakos 2011; Qian et al. 2017; Shan et al. 2012] and double transparent surfaces [Qian et al. 2016], with special patterns in the environment to provide the correspondence between a camera ray and the point on the environment.

Wu et al. [2018] proposed a method to reconstruct the complete 3D shape of a transparent object by utilizing a turntable and a monitor displaying Gray codes in a darkroom. This setup allowed them to obtain the correspondence for multiple views. Starting from the visual hull, the point cloud of the object is optimized by enforcing its normals to refract rays in the correct directions. Lyu et al. [2020] further improved it with mesh-based differentiable refraction ray tracing, which recovered more detailed shapes. Recently,

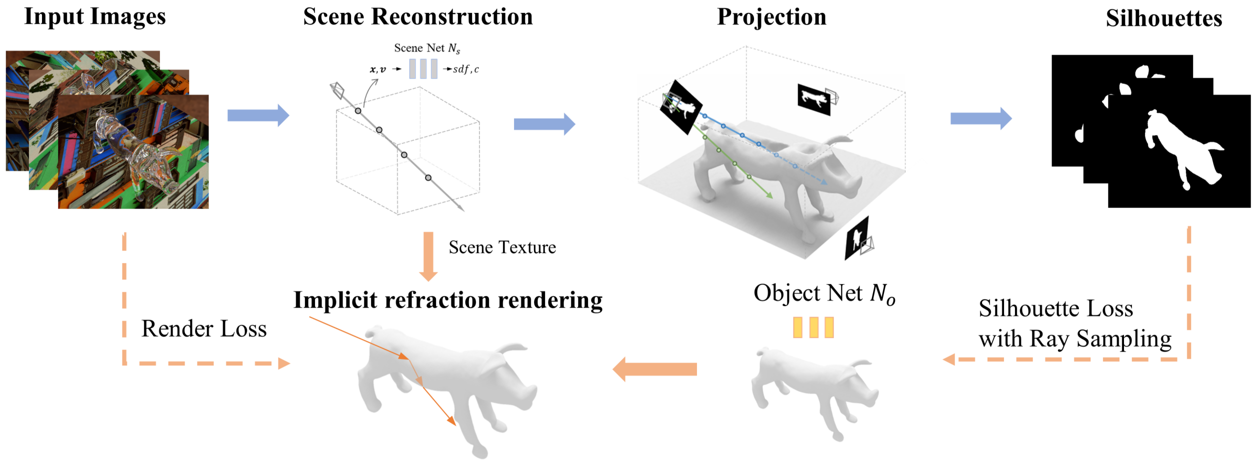


Figure 2: The overview of our method. We first adopt neural volume rendering to recover the entire scene and estimate the object silhouettes by projecting the neural field back to input views. Then we reconstruct the object's shape which refracts the same color as input images through implicit refraction rendering. We also utilize estimated silhouettes to regularize the object's shape with our designed ray sampling method.

a hybrid mesh-neural representation was proposed to recover detailed shapes under natural light, with an iPad displaying designed patterns[Xu et al. 2022].

In addition to analyzing the refraction, some methods utilized the specular reflection on the transparent surface with controlled light sources[Morris and Kutulakos 2007; Yeung et al. 2011].

In contrast to these methods, our method can reconstruct the transparent object in uncontrolled natural scenes.

Natural Scene. Morris et al. [2014] and Xiong et al. [2021] reconstructed both the fluid surface and the immersed scene. Stets et al. [2019] employed deep learning to predict the depth and normal of the transparent object from a single image. Furthermore, some neural-based methods were proposed to recover the depth map of transparent surfaces for robotic manipulation[Ichnowski et al. 2021; Sajjan et al. 2020; Zhu et al. 2021].

Li et al. [2020] introduced a physically-based network for reconstructing the complete shape of a transparent object under natural lighting conditions. They optimized the object shape in a latent feature space. However, their method relied on a pre-captured environment map and manually annotated multi-view silhouettes. In contrast, our method eliminates the need for pre-acquisition and manual annotation, making it more convenient and practical.

In addition, a refractive novel view synthesis method was proposed recently [Bemana et al. 2022]. But it aims to view synthesis and fails to produce realistic shapes for complex transparent objects. More discussions are included in Sec 4.

2.2 Multi-View Reconstruction with Neural Rendering

Recently, a series of differentiable rendering-based methods are proposed to recover the opaque shape and appearance as implicit neural field representation, which can restore the 3D content continuously at a low cost. According to the rendering techniques, these methods can be divided into surface rendering-based and volume rendering-based.

The surface rendering-based methods determine the radiance according to the intersection of the ray and the object surface, which can only backpropagate the gradients to a local region and requires object masks as supervision [Niemeyer et al. 2020; Yariv et al. 2020]. In contrast, Nerf [Mildenhall et al. 2020] and follow-ups [Darmon et al. 2022; Fu et al. 2022; Oechsle et al. 2021; Wang et al. 2021; Yariv et al. 2021] use volume rendering to aggregate radiance from all sampled points along the ray. These methods converge to better results and do not require masks.

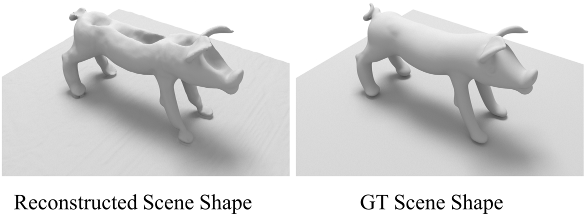
Inspired by these methods, we adapt neural rendering to refraction and reconstruct transparent objects from multi-view images.

3 METHOD

3.1 Overview

To reconstruct the transparent objects from RGB images, we assume the environment is richly textured to arise obvious refractive distortion as visual cues. We adopt differentiable refraction rendering to reconstruct the transparent object within an unknown natural scene. However, it is exceedingly challenging to directly recover the object's shape through refraction. On the one hand, the surroundings remain unknown, impeding refraction rendering. On the other hand, optimizing multi-segmented refractive light paths is a highly ill-posed problem that cannot converge to the true case without additional constraints.

Thus we first ignore refraction and reflection and treat the appearance of the transparent object as its own intrinsic color, as the same appearance model of the opaque surroundings. Assuming the transparent object is placed on a plane with an unknown appearance, we use the neural network N_s to represent the shape and appearance of the entire scene, including the transparent object and plane. Subsequently, we optimize the network N_s through neural



Reconstructed Scene Shape GT Scene Shape

Figure 3: The scene shape optimized by the neural volume rendering. The scene consists of an opaque plane with a transparent object placed on top of it. The former is well recovered while the latter is degenerated

volume rendering, which can accurately recover the appearance and shape of the opaque plane due to the constraints imposed by multi-view images. Regarding the transparent object, as illustrated in the upper section of Fig. 2, we discard its imprecise depth information and project the optimized neural field back to input views to acquire accurate multi-view object silhouettes. These silhouettes serve as strong constraints for object reconstruction in the subsequent phase.

With the recovered plane and silhouettes, we then consider the actual process of refraction and reconstruct the object that refracts the same color as the input image. To accomplish this, we employ a new object network denoted as N_o to fit the SDF field of the object. The neural implicit representation is continuous and free from the discretization artifacts [Niemeyer et al. 2020]. After initializing the network N_o with estimated silhouettes, as illustrated in the lower section of Fig. 2, we subsequently optimize N_o with rendering loss through the differentiable refraction rendering with the neural SDF field. The object's shape and refractive light path are optimized to the real case in order to refract the same color as the input. During optimization, we constrain the object silhouettes by calculating silhouette loss only for rays close to the silhouette edge, efficiently reducing the computational cost.

In the remainder of this paper, we first describe the preparation before the transparent object reconstruction, including scene reconstruction and object silhouette estimation (Sec 3.2) and then present the object reconstruction in detail (Sec 3.3).

3.2 Scene Reconstruction and Silhouette Estimation

In this step, we ignore the refraction and assume the straight rays. We use the scene network N_s to represent the shape and appearance of the entire scene. With position encoding, $N_s : (x, v) \longrightarrow s, c$ maps a 3D location x to its view-independent signed distance s and view-dependent color c. Subsequently, we employ the neural volume rendering in [Wang et al. 2021] to optimize N_s.

However, it approach alone does not yield precise geometry. While the opaque plane is adequately recovered, the shape of the transparent object severely degenerates, as shown in Fig. 3. This is primarily because the “own appearance“ of transparent objects changes rapidly with the view direction, which does not fully conform to the smooth BRDF assumption in neural volume rendering[Zhang et al. 2020]. As a result, a shape-radiance ambiguity arises.

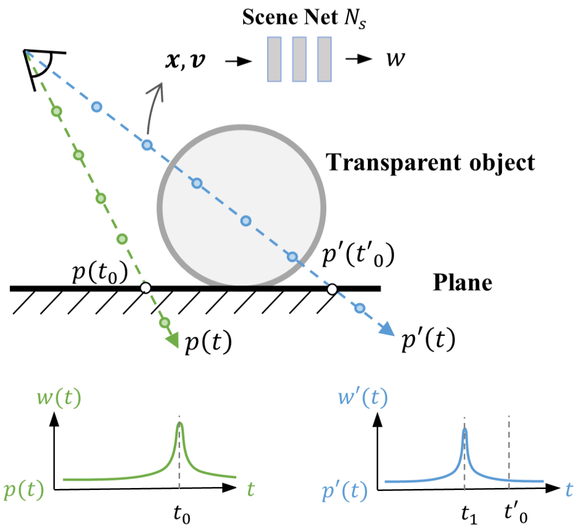


Figure 4: The illustration of the weight distributions of the ray $p(t)$ that hits the plane and the ray $p'(t)$ that hits the transparent object. $p(t_0)$ and $p'(t'_0)$ are their intersection with the plane, respectively. We show the weight distributions below.

Despite the inherent inaccuracy in the shape reconstruction, we conduct an analysis of the weight distributions of the rays and leverage it to estimate precise object silhouettes.

In volume rendering, for a ray represented as $p(t) = \boldsymbol{o} + t\boldsymbol{v}$, its color is calculated as:

$$C = \int_0^{+\infty} w(t)c(t)dt \tag{1}$$

, where $w(t)$ is the weight of the point, which in our method is calculated from the SDF value as in [Wang et al. 2021].

In our scene, where the transparent object is placed on an opaque plane, there are two kinds of rays: rays hit the plane and rays hit the transparent object, as shown in Fig. 4. For the ray $p(t)$ that hits the plane, since the plane is a simple opaque object with smooth BRDF, it should well converge close to the real case. The color of the ray should be solely determined by the color of the point on the plane. Let t_0 represent the depth of the intersection of the ray and plane, approximately, we have:

$$w(t) = \begin{cases} 0, t \neq t_0 \\ 1, t = t_0 \end{cases} \tag{2}$$

For the ray $p'(t)$ that hits the transparent object, due to refraction and reflection, its color significantly differs from the color of its intersection $p'(t'_0)$ with the plane. Therefore, it has to be fitted by the color of some point in front of the plane, which can be

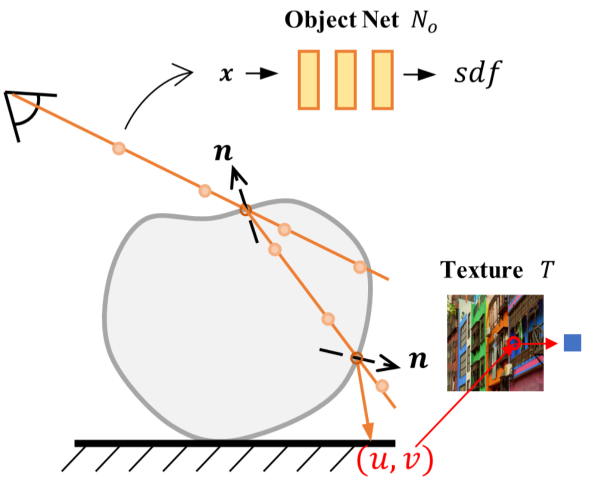


Figure 5: The illustration of our implicit refraction rendering. We locate the intersection of the ray and the implicit SDF field from the sample points and refract it twice until it hits the plane to fetch the rendered color from the plane texture.

approximately expressed as:

$$\exists t_1 < t_0', \quad w(t_1) = 1 \tag{3}$$

Note t_1 is usually not the accurate depth of the transparent surface since its rapidly-changing appearance as we motioned above.

Then we define a silhouette function $f_M(o,v)$ to calculate the integral of weights along a ray, while the weights of points on and below the plane are manually set to be zero.

$$f_M(o,v) = \int_0^{+\infty} w'(t)dt$$
$$w'(t) = \begin{cases} w(t), & t < t_0(\boldsymbol{o},\boldsymbol{v}) \\ 0, & t \geq t_0(\boldsymbol{o},\boldsymbol{v}) \end{cases} \tag{4}$$

where $t_0(\mathbf{o},\mathbf{v})$ calculates the depth of the intersection of the ray and the plane, according to the ray origin $\boldsymbol{o}$ and the direction $\boldsymbol{v}$.

It can be easily calculated that for rays hitting the plane $f_M = 0$, and for rays hitting the object $f_M = 1$. By setting a threshold σ, we can use f_M to determine whether a ray hits the object. The choice of threshold does not have a significant impact on the results. In our experiment, it is set as 0.4. By applying f_M to all rays, every pixel is determined and accurate multi-view silhouettes are estimated, as shown in the upper section of Fig. 2. For each view, we select the connected region with the largest area as the final result to eliminate noise. These estimated silhouettes provide strong constraints for the subsequent shape reconstruction.

3.3 Shape Reconstruction

We use a new object network N_o to represent the SDF field of the transparent object. In contrast to the scene net N_s, N_o only output the SDF value since we focus on the shape. We adopt the silhouette loss in [Wang et al. 2021] to optimize N_o with estimated silhouettes and add the eikonal term in [Gropp et al. 2020] as regularization.

The initial shape only constrained by silhouettes is not accurate enough. Thus we further consider the actual refraction light path and optimize the initial shape by enforcing its refracted color to be consistent with the input image. The supervision is mainly implemented through implicit refraction rendering and ray sampling for silhouette constraint.

3.3.1 Implicit Refraction Rendering. To differentiable render the refracted color with the SDF field, given a pixel in an image, we trace and refract the ray until it hits the plane to fetch the rendered color. We only consider the two-bounce refraction and do not consider the reflection since most reflected rays would not hit the plane.

Specifically, for a ray represented as $p(t) = o + tv$, we obtain the SDF values of a set of sampled points along the ray and then use the linear interpolation in [Fu et al. 2022] to locate the intersection $\hat{x}$ of the ray and object surface. We use the gradient of SDF at $\hat{x}$ as its normal vector [Niemeyer et al. 2020]:

$$n(\hat{x}) = \nabla S_{obj}(\hat{x},v)/||\nabla S_{obj}(\hat{x},v)||_2 \tag{5}$$

, then the refracted ray is calculated according to Snell's Law and the index of refraction (IoR) of the object, which is initialized manually as IoR_{init} and optimized along with object net N_o. We trace and refract the new ray again to get the ray coming out of the object, which is used to render the final color, as shown in Fig. 5.

During the ray tracing, the Fresnel term F_1, F_2 for twice refraction are calculated separately according to the Fresnel Equation:

$$\mathcal{F} = \frac{1}{2}(\frac{\eta_i l_i \cdot n - \eta_t l_t \cdot N}{\eta_i l_i \cdot n + \eta_t l_t \cdot N})^2 + \frac{1}{2}(\frac{\eta_t l_t \cdot N - \eta_i l_i \cdot n}{\eta_t l_t \cdot N + \eta_i l_i \cdot n})^2 \tag{6}$$

, where η_i, η_t are the refractive indices of the incident media and refraction media, respectively. And l_i, l_t are the incident and refracted ray directions, respectively.

We render each point on the plane separately from directly above through volume rendering.

For rendering the final color, we render the appearance of the plane from the top view through the volume rendering with the scene net N_c. For each point, the ray origin is set very close to directly above it to exclude the color of the transparent object. We store the rendered result as a view-independent explicit texture T^*, which can be blurred for a coarse-to-fine optimization. During optimization, the texture T for rendering is blurred as:

$$T = Gaussian(T^*, \sigma) \tag{7}$$

, where $Gassuian$ is the Gaussian blur and σ is the standard deviation of the blur kernel that decreases with the optimization. In our experiment, we set the size of the texture as 512 × 512.

Then for a ray coming out of the object, we calculate its intersection with the plane and fetch the texture as the background color c_b with bilinear interpolation. The rendered color c is further calculated as:

$$c = (1 - \mathcal{F}_1)(1 - \mathcal{F}_2)c_b \tag{8}$$

While our method can handle unknown and arbitrary planar textures, as long as they cause noticeable refraction distortions, a richly textured plane that reduces color ambiguity can facilitate the convergence of rays and yield more accurate results.

3.3.2 Ray Sampling for Silhouette Constraint. We continue to use estimated silhouettes to constrain the object's shape. However, calculating silhouette loss for all rays is inefficient. Since the final shape is close to decent initialization, most rays far away from the silhouette edges will have a low silhouette loss throughout the optimization. And the weights of the neural SDF can only be optimized along the rays[Zhang et al. 2022]. Thus these rays can not effectively supervise the shape of the object.

Instead of calculating silhouette loss for all rays, we sample the rays close to the silhouette edges, which are the most important rays for constraining the silhouette, to calculate the silhouette loss. It allows us to greatly reduce the number of rays used to compute the silhouette loss. In our experiment, it is easily implemented by the morphological gradient:

$$G_i = M_i \oplus b - M_i \ominus b \tag{9}$$

, where M_i is the ith silhouette, b is a structuring element which is a 5×5 square kernel in our experiment, $\oplus$ and $\ominus$ denote the dilation and erosion operations, respectively, and G_i is the sampled result. During optimization, we only use rays inside the G_i to compute the silhouette loss.

3.3.3 Loss. We optimize the object net S_o and the index of refraction with the following loss:

$$L = \lambda_{ren} L_{render} + \lambda_{sil} L_{silhouette} + \lambda_{reg} L_{regularize} \tag{10}$$

, where L_{render} is rendering loss, $L_{silhouette}$ is silhouette loss and $L_{regularize}$ is the regularization loss. and $\lambda_{ren}, \lambda_{sil}, \lambda_{reg}$ are 1.0, 1.0, 1.5, respectively.

rendering loss: The rendering loss measures the difference between the rendered color and the ground truth color in input images. For each iteration, we sample rays inside the estimated silhouettes to calculate the rendering loss. During implicit refraction rendering, the valid rays that can hit the plane without total reflection are recorded as a binary mask M_{ren}. Donating the pixel color of the ray p as $\widetilde{c}(p)$ and the rendered color as $c(p)$, the rendering loss is calculated as:

$$L_{render} = \frac{1}{|M_{ren}|} \sum_{p} M_{ren}(p) ||c(p) - \widetilde{c}(p)||_1 \tag{11}$$

Silhouette Loss: The silhouette loss provides extra constraints for optimization and effectively prevents a degraded result. Independent from the sampled rays used to calculate rendering loss, for each iteration, we sample rays inside G_i to calculate the silhouette loss that is the same as the one for shape initialization.

Regularization Loss: We calculate the Eikonal term in [Gropp et al. 2020] for both rays used to calculate rendering loss and silhouette loss to regularize the SDF field. We also add an l2 loss $||IoR - IoR_{init}||_2$ with weight 0.1 to regularize the index of refraction.

4 EXPERIMENT

We evaluate our method on both synthetic and real data, compared with the state-of-the-art method that reconstructs transparent objects in uncontrolled scenes [Li et al. 2020]. It is noticeable that their method requires much more complicated setups, including pre-capturing the environment map and manually annotating the object silhouettes for each view. Other transparent reconstruction methods focus on the reconstruction in a highly-controlled scene with designed patterns [Lyu et al. 2020; Wu et al. 2018; Xu et al. 2022], which are not suitable for a fair comparison with our method. We compare our method with [Lyu et al. 2020] on synthetic data in the supplementary materials for reference.

Table 1: The quantitative result of our method on synthetic data, including the number of input images, the silhouette error measured by mean absolute error(MAE), the shape reconstruction errors of our full method, the method without rendering loss, and the method without silhouette loss. The shape error is measured by the Chamfer distance and normalized by the bounding box diagonals.

Shape	Img. Num.	Sil. Error ($\times 10^{-3}$)	Shape Error ($\times 10^{-5}$)		
			Full	w/o Ren.	w/o Sil.
Dog	108	6.1	0.77	1.01	4.68
Pig	45	4.0	1.41	3.21	9.45
Cloud	108	3.5	3.25	5.03	14.07
Monkey	45	3.4	1.55	1.59	9.49

Recently, a refractive novel view synthetic method was proposed to approximate refraction with the eikonal field while only takes only RGB images as input [Bemana et al. 2022]. However, we experimentally found it fails to handle the total internal reflection that commonly occurs in complex transparent objects, and thus cannot be applied to transparent object reconstruction. Bemana et al.'s method produces unrealistic results for most objects in our experiment. We show their results and discuss them in detail in the supplementary materials.

4.1 Implement details

We follow the network structure and position encoding in [Wang et al. 2021]. We assume the region of interest is inside a unit sphere and handle the scene outside the sphere using NeRF++ [Zhang et al. 2020]. We maintain the same number of sample points as in [Wang et al. 2021] for the silhouette estimation. For the shape initialization with estimated silhouettes, we set iteration as 100k and also add the eikonal term to regularize the SDF field. The weights of the silhouette loss and eikonal loss are both 1.0. Regarding shape optimization with refraction rendering, we sample 16 points for coarse sampling and another 16 points for fine sampling. We sample 1024 rays for rendering loss and 256 rays for silhouette loss per batch and optimize the model for 300k iterations. The σ in Gaussian blur is initially set as 10 and is halved every 30k iterations. It takes about 4 hours on a single NVIDIA RTX4090 GPU for shape optimization and 11 hours for the whole process.

For real data acquisition, we capture approximately 40-50 RGB images for each object by circling around the object. Some images are shown in Fig. 9. Then we use COLMAP[Schonberger and Frahm 2016] to obtain the camera parameters. The position of the plane is calculated by using RANSAC to fit the sparse 3D points recovered in structure-from-motion.

4.2 Result on Synthetic Data

We render synthetic data in Mitsuba[Nimier-David et al. 2019] to evaluate our method. The shapes of transparent objects are from online sources and the data in [Lyu et al. 2020] and the IoR is set as

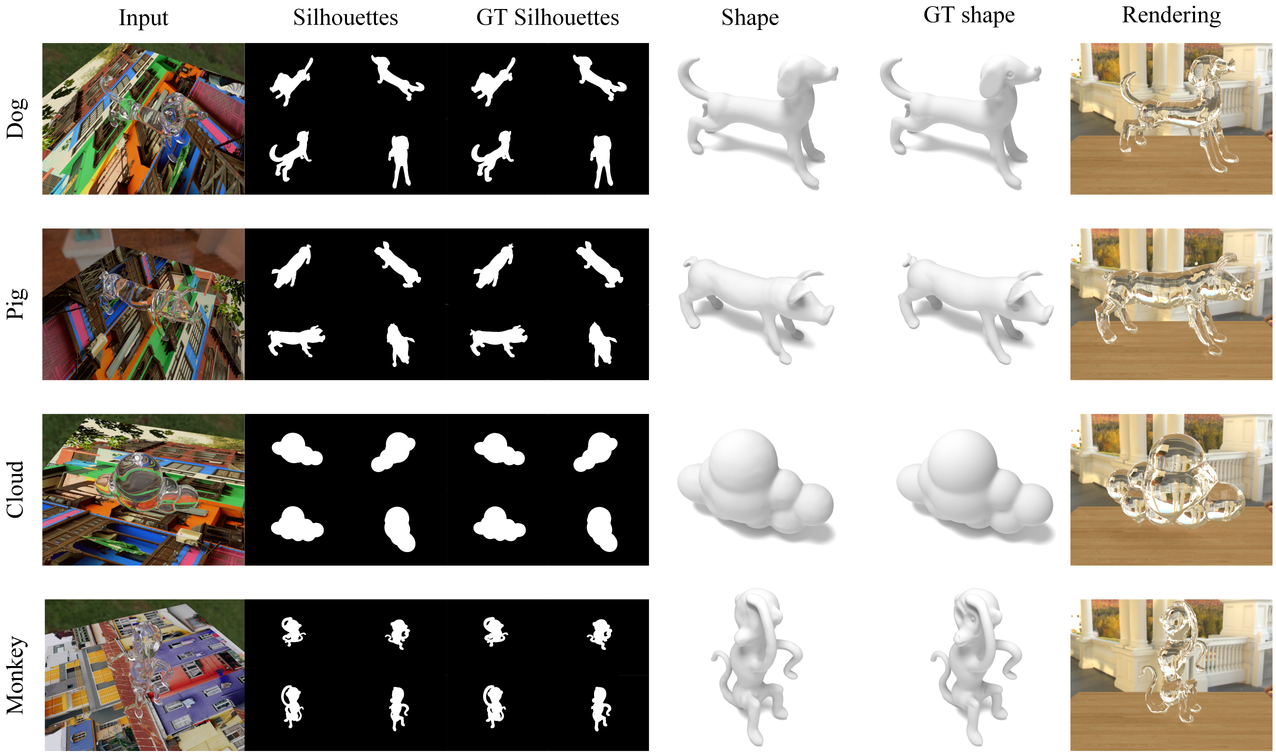


Figure 6: The results of our method on synthetic data. We show the some of estimated silhouettes and the recovered object shape, compared with the ground truth. The "Input" shows one of the input multi-view images. Our recovered silhouettes and shapes are both accurate. We also show the rendered images of our reconstruction results in a novel environment.

1.5. We initialize the IoR in our optimization with $IoR_{init} = 1.6$ to verify its ability to handle unknown IoR. The textures of the planar are from [Agustsson and Timofte 2017]. We extract explicit mesh from the object net N_o and compute the Chamfer distance with ground truth shape to measure the reconstruction error.

We present synthetic samples containing different numbers of images to demonstrate the generalization ability of our method. As shown in Fig. 6, our method produces high-quality results for all samples. The estimated multi-view silhouettes are close to the ground truth, which demonstrates the superiority of our silhouette estimation method. Besides, our method reconstructs accurate object shapes. The quantitative results are summarized in Tab. 1.

4.3 Result on Real Data

We evaluate our method on three kinds of real transparent objects, compared with [Li et al. 2020]. We use ICP to align the result with ground truth shapes, which are obtained by scanning these transparent objects with a scanner after painting them with DPT-5.

As shown in Fig. 7, our method reconstructs shapes that precisely preserve surface details and thin structures, like the folds of the tiger's belly, the tail and horn of the cow, and the tail and beard of the dragon. In contrast, Li et al.'s method produces overly smooth results that lack details. This may be attributed to that their method relies on data priors and suffers from the gap between the distribution of synthetic training data and real data. The quantitative results summarized in Tab. 2 also demonstrate the superiority of our method. Our method produces even better results than state-of-the-art while does not require silhouettes as input.

4.4 Ablation Study

We remove the rendering loss and silhouette loss separately on synthetic data to verify their effect. When removing the silhouette loss, the object shape cannot maintain the correct contour and exhibits significant errors, as the quantitative results in Tab. 1.

We also present the result without rendering loss in Fig. 8, compared with the shape recovered with full loss terms. As the normal results shown in Fig. 8, the "Cloud" shape contains completely concave regions (marked in the red box) that can not be reconstructed through silhouette constraints alone. Optimization without rendering loss leads to a flat reconstruction result, while optimization with rendering loss can accurately recover the shape by optimizing the refractive ray paths. In addition, a limited number of silhouettes is not sufficient to fully constrain the object's contour and the rendering loss can further optimize convex regions, as the "Dog" shape in Fig. 8. The quantitative results are summarized in Tab. 1.

Table 2: The quantitative result of our method on real data, compared with [Li et al. 2020]. The ground-truth silhouettes are obtained by rendering the ground-truth shape under real capturing conditions. The silhouette error is measured by mean absolute error(MAE) and the shape error is measured by the Chamfer distance and normalized by the bounding box diagonals.

Shape	Silhouette Error ($\times 10^{-3}$)	Shape Error ($\times 10^{-4}$)	
		Ours	Li et al.
Cow	6.00	0.25	1.85
Tiger	6.14	0.16	1.04
Dragon	12.46	0.37	5.11

5 LIMITATION AND FUTURE WORK

Although our method produces high-quality results, it has limitations in recovering the contact regions between transparent objects and the underlying plane, such as the feet of the objects in Fig. 7 and Fig. 10. This is mainly because there are fewer refraction distortions on these parts that are close to the plane, which can not be distinguished in scene reconstruction. In addition, the insufficient amount of refracted rays hitting the plane and the ambiguity of color supervision would disturb the optimization, making some details difficult to recover (the monkey eyes in Fig. 6 and the stripe on the cow chuck in Fig. 7).

6 CONCLUSION

In this paper, we present, to our knowledge, the first method that reconstructs transparent objects in an uncontrolled natural scene without object silhouettes as input, which greatly simplifies the setups for transparent object reconstruction. We project the neural field recovered by volume rendering back to 2D images to estimate accurate multi-view object silhouettes, and further perform implicit refraction rendering to reconstruct the detailed shape represented by a neural SDF field. Our method simplifies the setups, enabling complete data acquisition in just a few minutes. This greatly facilitates the practical application of transparent object reconstruction.

ACKNOWLEDGMENTS

This work was supported in part by National Key Research and Development Program of China (2022YFF0904301).

REFERENCES

Eirikur Agustsson and Radu Timofte. 2017. Ntire 2017 challenge on single image super-resolution: Dataset and study. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. 126–135.

Nicolas Alt, Patrick Rives, and Eckehard Steinbach. 2013. Reconstruction of transparent objects in unstructured scenes with a depth camera. In *2013 IEEE International Conference on Image Processing*. IEEE, 4131–4135.

Mojtaba Bemana, Karol Myszkowski, Jeppe Revall Frisvad, Hans-Peter Seidel, and Tobias Ritschel. 2022. Eikonal fields for refractive novel-view synthesis. In *ACM SIGGRAPH 2022 Conference Proceedings*. 1–9.

François Darmon, Bénédicte Bascle, Jean-Clément Devaux, Pascal Monasse, and Mathieu Aubry. 2022. Improving neural implicit surfaces geometry with patch warping. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 6260–6269.

Qiancheng Fu, Qingshan Xu, Yew Soon Ong, and Wenbing Tao. 2022. Geo-Neus: geometry-consistent neural implicit surfaces learning for multi-view reconstruction. *Advances in Neural Information Processing Systems* 35 (2022), 3403–3416.

Amos Gropp, Lior Yariv, Niv Haim, Matan Atzmon, and Yaron Lipman. 2020. Implicit geometric regularization for learning shapes. *arXiv preprint arXiv:2002.10099* (2020).

Cong Huynh, Antonio Robles-Kelly, and Edwin Hancock. 2010. Shape and refractive index recovery from single-view polarisation images. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. https://doi.org/10.1109/cvpr.2010.5539828

Jeffrey Ichnowski, Yahav Avigal, Justin Kerr, and Ken Goldberg. 2021. Dex-NeRF: Using a Neural Radiance Field to Grasp Transparent Objects.

Ivo Ihrke, Kiriakos N Kutulakos, Hendrik PA Lensch, Marcus Magnor, and Wolfgang Heidrich. 2010. Transparent and specular object reconstruction. In *Computer Graphics Forum*, Vol. 29. Wiley Online Library, 2400–2426.

Kiriakos N Kutulakos and Eron Steger. 2008. A theory of refractive and specular 3D shape by light-path triangulation. *International Journal of Computer Vision* 76 (2008), 13–29.

Zhengqin Li, Yu-Ying Yeh, and Manmohan Chandraker. 2020. Through the looking glass: Neural 3d reconstruction of transparent shapes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 1262–1271.

Jiahui Lyu, Bojian Wu, Dani Lischinski, Daniel Cohen-Or, and Hui Huang. 2020. Differentiable refraction-tracing for mesh reconstruction of transparent objects. *ACM Transactions on Graphics (TOG)* 39, 6 (2020), 1–13.

Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. 2020. *NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis*. 405–421. https://doi.org/10.1007/978-3-030-58452-8_24

Daisuke Miyazaki and Katsushi Ikeuchi. 2005. Inverse polarization raytracing: estimating surface shapes of transparent objects. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, Vol. 2. IEEE, 910–917.

Nigel JW Morris and Kiriakos N Kutulakos. 2007. Reconstructing the surface of inhomogeneous transparent scenes by scatter-trace photography. In *2007 IEEE 11th International Conference on Computer Vision*. IEEE, 1–8.

Nigel JW Morris and Kiriakos N Kutulakos. 2011. Dynamic refraction stereo. *IEEE transactions on pattern analysis and machine intelligence* 33, 8 (2011), 1518–1531.

Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. 2020. Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 3504–3515.

Merlin Nimier-David, Delio Vicini, Tizian Zeltner, and Wenzel Jakob. 2019. Mitsuba 2: A retargetable forward and inverse renderer. *ACM Transactions on Graphics (TOG)* 38, 6 (2019), 1–17.

Michael Oechsle, Songyou Peng, and Andreas Geiger. 2021. Unisurf: Unifying neural implicit surfaces and radiance fields for multi-view reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 5589–5599.

Yiming Qian, Minglun Gong, and Yee-Hong Yang. 2016. 3D Reconstruction of Transparent Objects with Position-Normal Consistency. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. https://doi.org/10.1109/cvpr.2016.473

Yiming Qian, Minglun Gong, and Yee-Hong Yang. 2017. Stereo-Based 3D Reconstruction of Dynamic Fluid Surfaces by Global Optimization. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. https://doi.org/10.1109/cvpr.2017.704

Shreeyak Sajjan, Matthew Moore, Mike Pan, Ganesh Nagaraja, Johnny Lee, Andy Zeng, and Shuran Song. 2020. Clear Grasp: 3D Shape Estimation of Transparent Objects for Manipulation. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*. https://doi.org/10.1109/icra40945.2020.9197518

Johannes L. Schonberger and Jan-Michael Frahm. 2016. Structure-from-Motion Revisited. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. https://doi.org/10.1109/cvpr.2016.445

Qi Shan, S. Agarwal, and B. Curless. 2012. Refractive height fields from single and multiple images. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*. https://doi.org/10.1109/cvpr.2012.6247687

Jonathan Stets, Zhengqin Li, Jeppe Revall Frisvad, and Manmohan Chandraker. 2019. Single-Shot Analysis of Refractive Shape Using Convolutional Neural Networks. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*. https://doi.org/10.1109/wacv.2019.00111

Kenichiro Tanaka, Yasuhiro Mukaigawa, Hiroyuki Kubo, Yasuyuki Matsushita, and Yasushi Yagi. 2016. Recovering Transparent Shape from Time-of-Flight Distortion. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. https://doi.org/10.1109/cvpr.2016.475

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

Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. 2021. NeuS: Learning Neural Implicit Surfaces by Volume Rendering for Multi-view Reconstruction. In *Advances in Neural Information Processing Systems*.

Gordon Wetzstein, David Roodnick, Wolfgang Heidrich, and Ramesh Raskar. 2011. Refractive shape from light field distortion. In *2011 International Conference on Computer Vision*. IEEE, 1180–1186.

Bojian Wu, Yang Zhou, Yiming Qian, Minglun Cong, and Hui Huang. 2018. Full 3D reconstruction of transparent objects. *ACM Transactions on Graphics (TOG)* 37, 4 (2018), 1–11.

Jinhui Xiong and Wolfgang Heidrich. 2021. In-the-wild single camera 3D reconstruction through moving water surfaces. In *Proceedings of the IEEE/CVF international conference on computer vision*. 12558–12567.

Jiamin Xu, Zihan Zhu, Hujun Bao, and Weiwei Xu. 2022. A hybrid mesh-neural representation for 3D transparent object reconstruction. *ACM Trans. Graph* 1, 1 (2022).

Lior Yariv, Jiatao Gu, Yoni Kasten, and Yaron Lipman. 2021. Volume rendering of neural implicit surfaces. *Advances in Neural Information Processing Systems* 34 (2021), 4805–4815.

Lior Yariv, Yoni Kasten, Dror Moran, Meirav Galun, Matan Atzmon, Basri Ronen, and Yaron Lipman. 2020. Multiview neural surface reconstruction by disentangling geometry and appearance. *Advances in Neural Information Processing Systems* 33 (2020), 2492–2502.

Sai-Kit Yeung, Tai-Pang Wu, Chi-Keung Tang, Tony F Chan, and Stanley Osher. 2011. Adequate reconstruction of transparent objects on a shoestring budget. In *CVPR 2011*. IEEE, 2513–2520.

Kai Zhang, Fujun Luan, Zhengqi Li, and Noah Snavely. 2022. Iron: Inverse rendering by optimizing neural sdfs and materials from photometric images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 5565–5574.

Kai Zhang, Gernot Riegler, Noah Snavely, and Vladlen Koltun. 2020. Nerf++: Analyzing and improving neural radiance fields. *arXiv preprint arXiv:2010.07492* (2020).

Mingjie Zhang, Xing Lin, Mohit Gupta, Jinli Suo, and Qionghai Dai. 2014. Recovering scene geometry under wavy fluid via distortion and defocus analysis. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*. Springer, 234–250.

Luyang Zhu, Arsalan Mousavian, Yu Xiang, Hammad Mazhar, Jozef van Eenbergen, Shoubhik Debnath, and Dieter Fox. 2021. RGB-D local implicit function for depth completion of transparent objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 4649–4658.

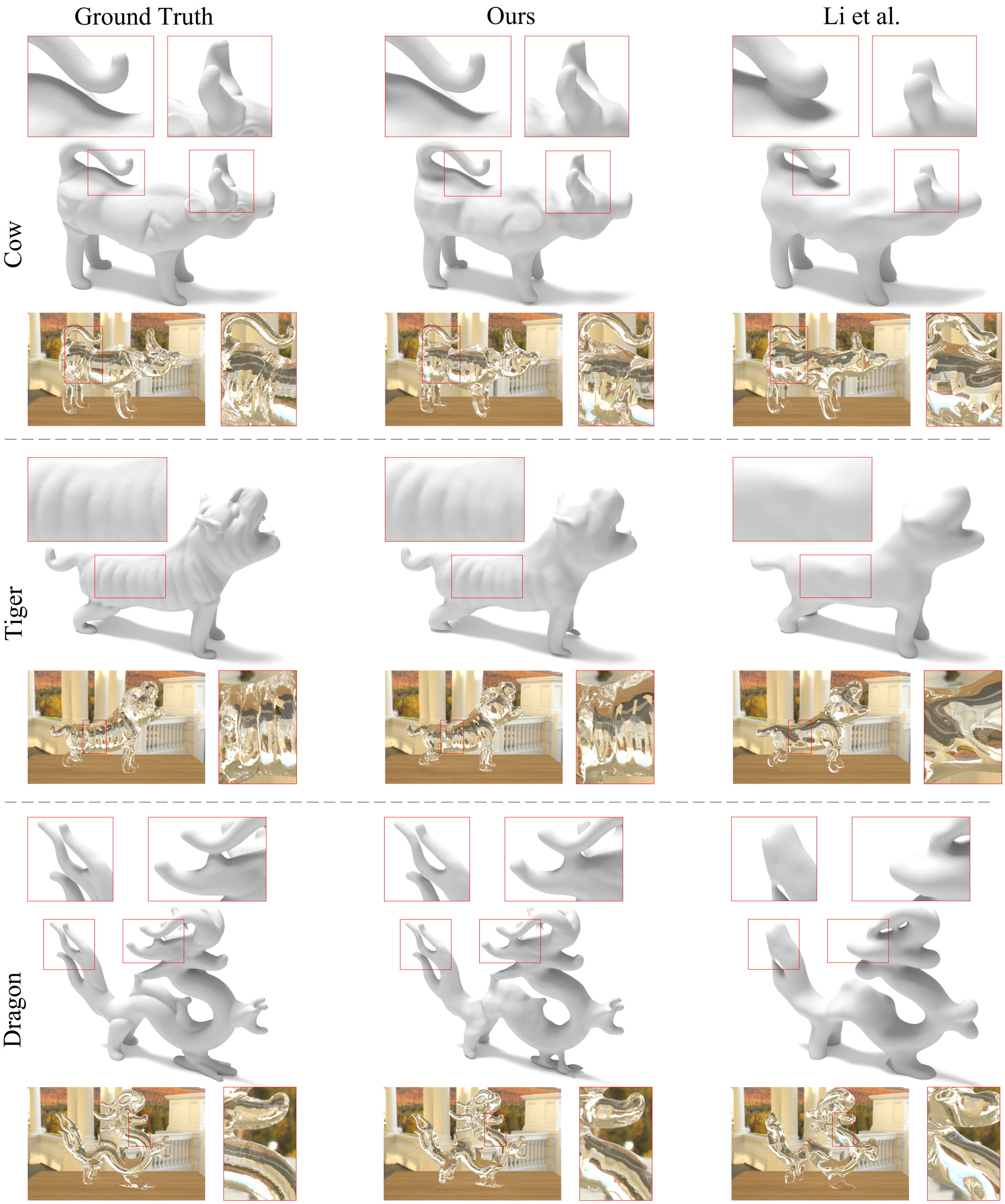


Figure 7: We present our reconstruction on real data and its rendered results for a comprehensive comparison. Compared with [Li et al. 2020], our method accurately captures the details of objects, as indicated by the red box, without the need for silhouettes as input.

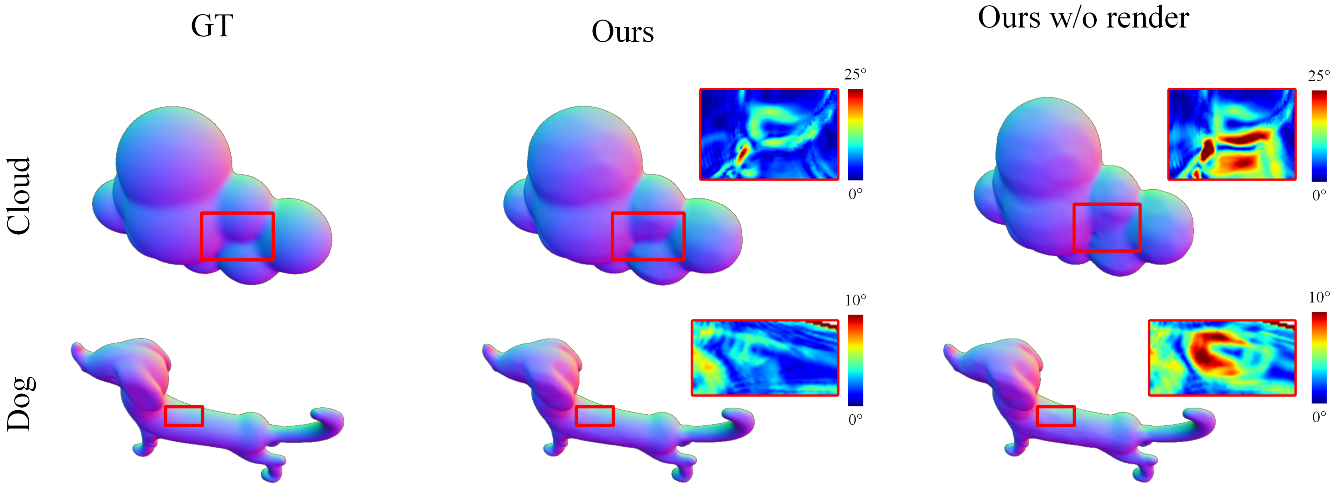


Figure 8: The visualized results of ablation study. We show the normal map of the ground truth shapes, results with our full method, and results without rendering loss. The visualized angle errors (degree measure) of the marked red boxes are enlarged and shown at the right top corner, with its color bar shown above. The rendering loss significantly improves the surface details.

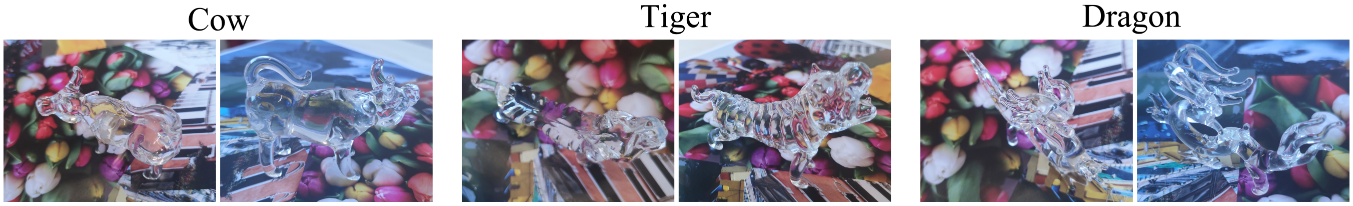


Figure 9: The captured images of real data. Images are cropped for better visualization.

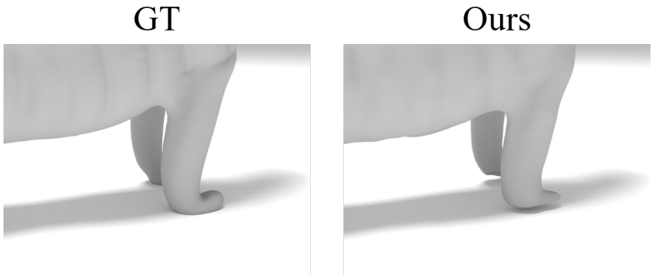


Figure 10: Failure case. The zoom-in display of the front feet of our reconstructed real object “Tiger“. Our method missing the parts close to the plane due to the lack of refraction distortions.