NeTO: Neural Reconstruction of Transparent Objects with Self-Occlusion Aware Refraction-Tracing

Zongcheng Li\textsuperscript{1,*} Xiaoxiao Long\textsuperscript{2,*} Yusen Wang\textsuperscript{1} Tuo Cao\textsuperscript{1} Wenping Wang\textsuperscript{3} Fei Luo\textsuperscript{1} Chunxia Xiao\textsuperscript{1,†}

\textsuperscript{1}School of Computer Science, Wuhan University \quad \textsuperscript{2}The University of Hong Kong \quad \textsuperscript{3}Texas A&M University

Abstract

We present a novel method called NeTO, for capturing the 3D geometry of solid transparent objects from 2D images via volume rendering. Reconstructing transparent objects is a very challenging task, which is ill-suited for general-purpose reconstruction techniques due to the specular light transport phenomena. Although existing refraction-tracing-based methods, designed especially for this task, achieve impressive results, they still suffer from unstable optimization and loss of fine details since the explicit surface representation they adopted is difficult to be optimized, and the self-occlusion problem is ignored for refraction-tracing. In this paper, we propose to leverage implicit Signed Distance Function (SDF) as surface representation and optimize the SDF field via volume rendering with a self-occlusion aware refractive ray tracing. The implicit representation enables our method to be capable of reconstructing high-quality reconstruction even with a limited set of views, and the self-occlusion aware strategy makes it possible for our method to accurately reconstruct the self-occluded regions. Experiments show that our method achieves faithful reconstruction results and outperforms prior works by a large margin. Visit our project page at https://www.xxlong.site/NeTO/.

1. Introduction

Reconstructing 3D models of real-world objects has been one of the longstanding problems. It has been researched for decades in computer vision and graphics, which boosts the development of many applications, such as augmented reality, automatic driving, and robots. However, existing general-purpose multi-view reconstruction methods [15, 16, 17, 39, 47, 50] are only suitable for opaque objects whose surfaces are approximately Lambertian, and few of them can tackle transparent objects. The light paths passing through transparent objects are extremely complex and involve refractions and reflections.

Recently, some state-of-the-art methods have been proposed to reconstruct solid transparent objects in a non-intrusive manner, capturing refraction-tracing consistency with specially designed hardware systems, and have produced impressive results. This is achieved by optimizing correspondences between camera rays and locations on the background monitor [23] or enforcing consistency between

\*Equal contributions.
\†Corresponding author.
camera rays and refracted rays with a rotating background monitor [53]. However, those methods either adopt point cloud [53] or mesh [23] as surface representation, and the explicit representations are difficult to be optimized. As a result, the methods usually require a large number of views as input for optimization. Without enough images as input, the methods easily fail to reconstruct faithful geometry due to unstable optimization (see Figure 1).

More importantly, a critical issue still remains ignored, i.e., how to tackle the self-occluded parts of the objects. The widely-used refraction-tracing consistency assumes that a camera ray is only refracted twice (upon entering and upon exiting) on the object surfaces when the ray passes through a transparent object. However, the assumption is not always true when a camera ray passes through the self-occluded parts where the ray will be refracted by surfaces more than twice. As a result, mistakenly enforcing the refraction-tracing consistency on the self-occluded parts will unavoidably introduce errors in the optimization of reconstruction, which is a bottleneck to further enhance the reconstructed geometries.

In this work, we propose a novel method, called NeTO, for reconstructing high-quality 3D geometry of solid transparent objects. In contrast to prior works [23], we adopt implicit Signed Distance Function (SDF) as surface representation and leverage volume rendering [47] to enforce the refraction-tracing consistency. Moreover, we propose a simple but effective strategy to detect the self-occluded parts and avoid mistakenly enforcing constraints on these regions. The key idea is that we leverage the law of reversibility, that is, If the direction of a light beam is reversed, despite the number of times the beam is reflected or refracted, it will follow the same path, to identify whether a camera ray is reversible or not upon the assumption that the ray is refracted exactly twice.

To validate our method, we conduct experiments on DRT [23] dataset and our collected data with full views settings and various sparse views settings. The sparse setting selects one view from every n consecutive camera index, i.e., \(\{1, n + 1, 2n + 1, \ldots\}\), where n is termed as Sparsity. The extensive experiments show that our method enables the high-quality reconstruction of transparent objects and outperforms the previous methods. Our contributions can be summarized as follows:

- A novel neural surface reconstruction system is adopting implicit SDF as a representation for reconstructing transparent objects, thus enabling robust reconstruction optimization.
- A self-occlusion aware refraction-tracing strategy is introduced to accurately enforce the constraint, making it possible to recover geometries with fine details.
- Experimental results show that our method achieves SOTA results compared to prior works.

2. Related Work

2.1. Environment matting

Environment matting is introduced by [59], which extracts the environment matte and silhouettes from a series of projected horizontal and vertical stripe patterns. Subsequent works have been extended to multiple cameras [7], natural images [3, 52], wavelet domains [32], and frequency domain [35]. Meanwhile, it can be combined with compressive sensing theory [9] to reduce the number of used images. Unlike the above method, we adopt environment matting to capture environment matte and object masks and optimize the object geometries to fit them.

2.2. Transparent Object Reconstruction

Recovering the 3D geometry of transparent objects is a longstanding challenging problem [13]. To solve this difficult task, many works leverage specially designed hardware setups to provide more information encoding object
geometries, including polarization [8, 12, 26, 41], tomography [44], a moving point light source [4, 27], light field probes [51] and gray-coded patterns [23, 36, 53]. Some methods [40, 57] target the reconstruction of transparent objects with refractive or mirror-like surfaces. Other methods, including ours, focus on solid transparent objects where most camera rays will refract on the surfaces twice. To reconstruct the geometry of transparent objects, there are many types of correspondences proposed, like multi-view ray-ray correspondences [36, 45, 53], and ray-location correspondences [23]. DRT [23] proposed to extract per-view ray-location correspondences by using the EnvMatt algorithm [59], and utilize the differentiable rendering for progressively optimizing explicit meshes. Xu et al. [54] introduced ray-cell correspondence for reconstructing the full mode under natural light. Besides, Shao et al. [41] adopted polarimetric cues to reconstruct the full model of transparent objects [41].

Recently, data-driven-based methods have shown remarkable achievements in estimating the depth and normal maps of transparent objects [14, 38, 42]. Li et al. [14] first predicted the rough geometry of the transparent objects and then leveraged Pointnet++ [34] to further refine the rough geometry. However, due to the domain gap between the synthetic and real data, Li et al. [14] failed to reconstruct real objects unseen in its training dataset. More recently, Bemana et al. [2] leveraged NeRF for novel view synthesis of transparent objects and showed good performance to render novel views. However, since it targets novel view synthesis rather than reconstruction, it’s difficult to extract reliable geometry from the method. Unlike the above methods, We leverage volume rendering to simulate the refraction-tracing path for geometry optimization.

2.3. Neural Implicit Representation

Existing 3D representations can be roughly divided into four categories: voxel-based representations [6], point-based representations [1, 10], mesh-based representations [23, 46], and neural implicit representations [5, 24, 30, 37]. Recently, implicit neural representations have been applied to a variety of applications, including novel view synthesis [25, 58], camera pose estimation [18, 49], human [19, 33] and multi-view 3D reconstruction [11, 20, 21, 28, 29, 43, 47, 48, 55, 56], and achieved impressive successes. Recent research has shown that reconstructed results using implicit neural representation often produce higher quality than other 3D representations.

For the task of 3D reconstruction from 2D images, some works combine implicit neural representation with surface rendering techniques. These works typically require additional constraints for optimization, such as object masks. Moreover, inspired by the seminal work NeRF [49], more recent works apply volume rendering techniques to optimze the implicit neural representation encoded geometry. In this work, we adopt an implicit signed distance function as geometry representation and leverage the volume rendering technique proposed in NeuS [47] to optimize the geometries with the ray-location correspondences [23, 36, 53].

3. Method

3.1. Overview

We aim to reconstruct the surfaces $S$ of a solid transparent object from a set of posed object masks and the correspondences between the camera view rays and locations on the background under each viewpoint. We propose to adopt an implicit Signed Distance Function (SDF) as geometry representation and leverage the volume rendering technique proposed in NeuS [47] to enforce the refraction-tracing consistency, which enables stable and robust optimization. Moreover, we propose a simple but effective strategy to identify the rays passing through self-occluded parts and then exclude these rays during optimization to avoid mistakenly enforcing refraction-tracing consistency.

3.2. Preliminaries

Object capture setup. To reconstruct the transparent objects, we adopt the similar object capture system proposed in [23, 53]. The system consists of a static LCD monitor, a turntable, and a camera. The monitor displays hori-
Figure 4: The diagram of a ray with self-occlusion $r_e$ and a ray without self-occlusion $r_p$. The ray $r_p$ only refracts twice on the object surfaces, while the ray $r_e$ refracts on the surfaces more than twice due to self-occlusion. The rays with self-occlusion should be excluded in the optimization for high-quality reconstruction.

Figure 5: A example of self-occlusion checking strategy applied on the real Horse object.

Horizontal and vertical stripe patterns that form a Gray-coded background, and is placed behind the object and the camera. The transparent object is placed on the turntable, which is rotated in data acquisition to provide the static camera with multiple views of the object. The silhouette mask information and environment matte can be extracted from the patterns displayed on the monitor.

**Refraction-tracing consistency.** For general objects, feature points of the input images are extracted to establish correspondences for 3D reconstruction. However, for transparent objects, it’s difficult to extract reliable feature points to establish correspondences, so the prior works and ours leverage the environment matting technique to establish the relationship between object geometry and the observed images. As shown in Figure 3, a ray $r$ shooting from the camera center passes through the transparent object, which refracts twice on the object surfaces and then hits on the monitor at point $Q$. Since the gray-coded patterns are known, we can calculate the exact location of $Q$, and therefore we obtain a pair of camera ray $r$ and hit location $Q$. Our method is based on optimizing the correspondences between camera rays and the locations, which can also be named refraction-tracing consistency.

### 3.3. SDF-based refraction tracing

**Surface representation.** Unlike that the prior works adopt point clouds or meshes as geometry representations, we adopt Signed Distance Function (SDF) as surface representation. Specifically, the SDF field maps a point $x \in \mathbb{R}^3$ to its signed distance value to the surfaces, and the field is encoded by a Multi-layer Perceptrons (MLP) network. The surface $S$ of the object is represented by the zero-set of the signed distance function (SDF), that is, $S = \{ x \in \mathbb{R}^3 | g(x) = 0 \}$. The SDF field is initialized as a unit sphere. For convenience, we denote the shape being optimized as a virtual shape.

**Refraction-tracing.** As shown in Figure 3, given the current virtual shape, we first trace rays from the camera center that intersect and refract through the shape, and then optimize the SDF values of associated surface intersections according to the captured correspondences between the view rays and background locations (e.g., the ray $\tilde{c}q$ and the location $Q$ in Figure 3). We take a ray that only refracts on the surfaces exactly twice as an example, the ray first enters the virtual shape at point $p_1^v$, and then it exits the shape at point $p_2^v$. Finally, the simulated light path, shown in purple in Figure 3, hits on the background monitor at a virtual location $Q'$. Before the optimization of geometry converges, $Q'$ is generally different from the destination of the actual optical path passing through the real object, which is shown in orange, and finally hits on the background monitor at $Q$. The goal of optimization is to minimize the differences between the virtual hitting location and real hitting location, that is, $\Delta = ||Q - Q'||^2$.

To trace how the simulated light path interacts with the virtual shape and then penalize the location differences in the optimization, we leverage the SDF-based volume rendering technique [47] to calculate the exact locations of the two refraction intersections $p_1^v$ and $p_2^v$. The SDF-based surface rendering technique [56] can also be used for the intersection calculation, as discussed in Section 4.3, volume rendering yields more robust and stable optimization and leads to better reconstruction quality.
3.4. Self-occlusion handling

Since the objects to be reconstructed are solid, most camera rays will refract on the object surfaces exactly twice. When a ray passes through self-occluded regions, the refractions will be more complex, and the ray may refract on the surfaces more than twice. As shown in Figure 4, the light path without self-occlusion (blue line in Figure 4) has two refracted intersections with object surfaces, while the light path with self-occlusion (green line) has four refracted intersections. However, the prior works [23, 53] ignore the self-occlusion problem and assume that all the camera rays only refract exactly twice. As a result, for the rays that refract more than twice, the simulated light paths will be mistakenly approximated, thus introducing wrong supervision information into the geometry optimization.

Naive checking strategy. To tackle this problem, the key is to identify whether a ray refracts more than twice, and then exclude the ray in the optimization. A naive solution is to calculate the exact locations of the refraction intersections. As shown in Figure 4, when a ray passes through the self-occluded parts, we can leverage Snell’s law to obtain the directions of the refracted lights, and then calculate the locations of the intersections, $e_1, e_2, e_3, e_4$. However, we have to extensively conduct iterative sampling and network queries to find the points sampled in the refracted lights which are in the surfaces, which significantly increases the computational costs.

Proposed checking strategy. We, therefore, propose a simple but effective strategy to identify the rays that refract more than twice at low costs. The motivation is based on the law of reversibility, that is, If the direction of a light beam is reversed, it will follow the same path.

As shown in Figure 6, the procedure of the strategy is introduced below:

**Algorithm 1:** Self-occlusion checking strategy

1) Shoot a ray $r_p$, emitting from the camera center, and get its first forward intersection $p_f/e_f$.
2) Leverage Snell’s Law to obtain the refracted light line $\overrightarrow{p_f v_f e_f v_e}$, where $v_f/v_e$ is an infinite point on the line.
3) Shoot the reversed refracted light line $\overrightarrow{v_p p_f v_f e_f v_e}$ from $v_p/v_e$, and then obtain the backward intersection $p_b/e_b$.
4) Sample points on the line segment $\overrightarrow{p_f p_b e_f e_b}$, and then evaluate the SDF values of the points.
5) If there exist points with positive SDF values, the ray refracts more than twice; if not, the ray refracts exactly twice.

We can see the ray $r_p$ refracts the surfaces exactly twice, and there are no intersections on the line segment $\overrightarrow{p_f p_b}$, which indicates the light path $c \rightarrow p_f \rightarrow v_p$ is reversible. On the other hand, for the ray $r_e$, there exist two more intersections on the line segment $\overrightarrow{e_f e_b}$, which indicates that the light path $c \rightarrow e_f \rightarrow v_e$ is not reversible with the twice refraction assumption. Moreover, thanks to the properties of SDF (negative values inside and positive values outside), we can evaluate whether there exist any points with positive SDF values between the forward and backward intersections to identify the existence of self-occlusion.

Unlike that the naive checking strategy requires accurately finding the locations of all intersections, our proposed checking strategy only needs to identify whether there exist positive SDF values in a line segment with a short length. We provide an example of the self-occlusion checking strategy applied on a real Horse object in Figure 5, and our method can accurately identify the self-occluded regions (the overlapping legs of the horse).

3.5. Loss Functions

We optimize the SDF field by sampling a batch of rays with their ray-location correspondences and object masks $\{Q, M\}$, where $Q$ is the observed location on the background monitor, and $M \in \{0, 1\}$ is mask value. We sample $n$ points on the ray, and the batch size is $m$. The loss function is defined as:

$$\mathcal{L} = \omega_1 \mathcal{L}_{\text{Refraction}} + \omega_2 \mathcal{L}_{\text{Eikonal}} + \omega_3 \mathcal{L}_{\text{Mask}}$$  \hspace{1cm} (1)

**Refraction loss.** We minimize the difference between simulated background position $Q'$ and and its corresponding captured ground truth $Q$ (see Figure 3). The refraction
loss is defined as follows:

$$\mathcal{L}_{Refraction} = \sum_{i \in R} (||Q_i - Q'_i||^2)$$

(2)

where $R$ is the set containing ray paths that go through the object and refract on surfaces exactly twice.

With our proposed self-occlusion checking strategy, the invalid rays are excluded in the optimization, and the refraction loss gets rid of noises and makes reconstruction accurate.

**Mask loss.** Following the prior works [23, 53], the mask loss is also included and defined as:

$$\mathcal{L}_{mask} = BCE(M_k, O_k)$$

(3)

where $O_k$ is the sum of weights along the $k_{th}$ camera ray,

$M_k$ is the mask of the $k_{th}$ ray, and $BCE$ is the binary cross entropy loss.

**Eikonal loss.** We add an Eikonal loss to regularize the SDF field of the sampling point on the ray to have a unit norm of gradients. The loss term is defined as:

$$\mathcal{L}_{Eikonal} = \frac{1}{nm} \sum_{k,i} (||\nabla g(x_{k,i})||_2 - 1)^2$$

(4)

where $x_{k,i}$ is the $i_{th}$ sampled point at the $k_{th}$ ray, $g$ is the geometry function.
Table 3: Evaluation of reconstruction with sparsity = 4 (18 views). Compared with Li et al. [14] and DRT [23], our method achieves the best performance in all cases.

Table 4: Comparisons of reconstruction with full views. Our method obtains the best performance in all cases, and the full table is in the supplementary material.

Figure 8: We show two groups of full views reconstruction results generated by ours and DRT [23], respectively. Our method can faithfully reconstruct high-quality geometries with fewer errors.

4. Experiments

4.1. Experimental Setup

Datasets. We conduct evaluations on the DRT [23] dataset. The dataset contains eight transparent objects. Each transparent object contains 72 views with corresponding masks, ray-pixel correspondences, and intrinsic camera parameters. The view resolution is 960 × 1280 or 1080 × 1920. Ground truth 3D models are also provided for the transparent models.

Implementation Details. The geometry function \( g \) is modeled by an MLP, which consists of 8 hidden layers with a hidden size of 256. We use PyTorch [31] to implement our approach and use the Adam optimizer with a global learning rate \( 5 \times 10^{-4} \) for the network training. Our network architecture and initialization scheme are similar to those of prior works [47, 48]. We sample 512 rays per batch and train our model for 300k iterations on a single NVIDIA RTX 2080Ti GPU. We extract explicit mesh from the learned SDF field via a marching cube algorithm [22].

A hierarchical sampling strategy is used to sample points along a ray in a coarse-to-fine manner for volume rendering. We first uniformly sample 64 points along the ray, and then iteratively conduct importance sampling [47] to sample more points on top of coarse probability estimation for 4 times. The positional encoding is applied to the spatial location with 5 frequencies. The hyper-parameters used in the experiments are set as \( \omega_1 = 0.0001, \omega_2 = 0.1, \omega_3 = 0.1 \). Following the prior work [53], the IOR (index of refraction) of air is set to 1.0003 and the IOR of transparent material (glass) is set to 1.4723.

4.2. Comparisons

Baselines. We compare our method with the two state-of-the-art baselines: 1) DRT [23], the most related work to ours, which also optimizes the geometry by the ray-location correspondences but it adopts explicit mesh as surface representation. 2) A data-driven deep learning based approach Li et al. [2020] [14]. They generate a synthetic dataset of the transparent objects, and then learn geometric priors from the training data to reconstruct the objects.

Evaluation Protocols. To evaluate the quality of reconstructed models, we calculate the metrics, accuracy, completeness, precision, recall, and F-score between the reconstructed model and the ground truth model. Our method and DRT adopt the same dataset provided by DRT, so the input images and the ground truth models adopted by ours and DRT are the same. Although Li et al. experimented with transparent objects obtained from the same source, due to different manufacturing batches, there are slight differences...
between the shapes, and therefore their results are compared to a different set of ground truth models. For fairness, the ground truth models of the two types are reshaped into the same scale for evaluation. We evaluate the reconstruction results with sparse views and with full dense views.

**Reconstructions with sparse views.** We conduct experiments under various sparsity levels. The sparse setting selects a small proportion of the camera views by consecutively sampling a view $I_i$ from every $\text{sparsity} = n$ camera index, i.e., $\{1, n + 1, 2n + 1, \ldots\}$. The quantitative comparisons with $\text{sparsity} = 18, 8, 4$ are presented in Table 1, Table 2 and Table 3 respectively. As you can see, with sparse views, our results outperform DRT and Li et al. in terms of model completeness and accuracy. In addition to making quantitative comparisons, we render the model to visually observe the differences between our method and other methods. The qualitative comparisons are shown in Figure 7, and our method faithfully reconstructs the geometry with rich details, such as the leg of the Mouse and the eyes of the Monkey. The reconstruction results produced by [14] and [23] do not work well to reconstruct the rich geometric details and tend to over-smooth the surfaces.

**Reconstruction with full views.** When we use more views, e.g., full views (72 views), our reconstruction results and DRT are improved compared with reconstructions with sparse views. However, based on the quantitative results presented in Figure 8 and the qualitative results shown in Table 4, our method significantly outperforms the other methods in terms of completeness and accuracy, and the reconstructed models contain more rich details and have fewer errors. We further evaluate a self-collected real

**4.3. Ablation study and discussions.**

**Ablation study.** To better validate the effects of the self-occlusion checking strategy and the optimization terms, we conduct the ablation studies on the full method, method without self-occlusion checking, method without Eikonal loss term, and method without refraction loss term. The quantitative evaluation is shown in Table 5, and the qualitative evaluation is presented in Figure 10. The experiments demonstrate that $\mathcal{L}_{\text{Eikonal}}$ plays the most important role, which encourages the SDF field to be continuous and smooth. Without the refraction loss term, although our method can still reconstruct the rough shape relying on the silhouette information, the reconstruction becomes worse with larger errors. Thanks to our proposed self-occlusion checking strategy, the quality of the self-occluded parts is improved with fewer errors, like the legs of the Pig model. Furthermore, to better demonstrate the effectiveness of the self-occlusion checking strategy, we visualize the reconstruction error maps and the object surfaces affected by all self-occluded rays in Figure 11 (a self-occluded ray has multiple refractions on object surfaces, and thus a few rays will cause more affected surfaces), where the geometries are improved with self-occlusion check, especially on the affected regions.

**Different rendering techniques.** Both surface rendering [28] and volume rendering [47] are used in neural rendering-based reconstruction. Through experiments, we find that optimization using volume rendering is more robust and stable than using surface rendering. As shown in Figure 12, the reconstructed model using surface rendering is over-smoothing and lacks detailed geometries, while the reconstruction model using volume rendering achieves much better quality with rich details.

**5. Conclusion and Future Work**

We have presented NeTO, a novel neural rendering-based method for transparent object reconstruction, which
Figure 10: Qualitative ablation study on the Pig model. For better visualization, we measure and colorize the errors between the reconstructed models with the ground truth model. The reds indicate large errors, and the blues indicate small errors.

Figure 11: W/ and W/o Self-occluded check with full views.

adopts implicit signed distance function as surface representation and leverage volume rendering to enforce refraction-tracing consistency. With our proposed self-occlusion checking strategy, the reconstructed geometries of self-occluded parts are further improved. Our method significantly outperforms the state-of-the-art methods qualitatively and quantitatively by a large margin.

Although our method achieves high-quality reconstruction of transparent objects, the objects should be solid. This is because we adopt the ray-location correspondences, which assumes that most of the camera rays only refract on the object surfaces exactly twice. In the future, we would like to explore how to reconstruct hollow transparent objects, where refraction is more complex and most of the camera rays will refract on the surfaces more than twice.

6. Acknowledgments

This work is partially supported by NSFC (No.61972298).

References


