Differentiable Neural Surface Refinement for Modeling Transparent Objects

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

Abstract

Neural implicit surface reconstruction leveraging volume rendering has led to significant advances in multi-view reconstruction. However, results for transparent objects can be very poor, primarily because the rendering function fails to account for the intricate light transport induced by refraction and reflection. In this study, we introduce transparent neural surface refinement (TNSR), a novel surface reconstruction framework that explicitly incorporates physical refraction and reflection tracing. Beginning with an initial, approximate surface, our method employs sphere tracing combined with Snell's law to cast both reflected and refracted rays. Central to our proposal is an innovative differentiable technique devised to allow signals from the photometric evidence to propagate back to the surface model by considering how the surface bends and reflects light rays. This allows us to connect surface refinement with volume rendering, enabling end-to-end optimization solely on multi-view RGB images. In our experiments, TNSR demonstrates significant improvements in novel view synthesis and geometry estimation of transparent objects, without prior knowledge of the refractive index.

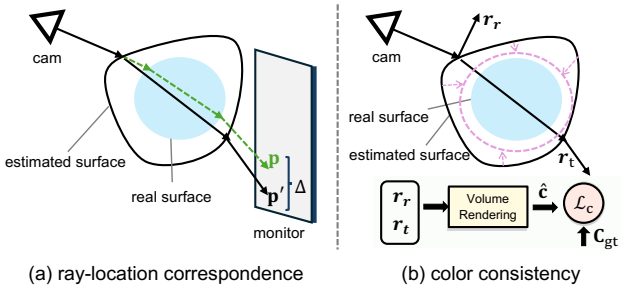

Figure 1. Comparison of ray-location correspondence versus color consistency for refining surfaces. **Left:** Previous methods use a controlled camera setup [16, 17] to establish ray-location correspondence, refining surfaces when the refractive index is known via the exact hitting point $\mathbf{p}$ of a light ray on a monitor. **Right:** Our approach eliminates the need for a controlled camera setup or a known refractive index. It employs ray tracing with an initial surface estimate to compute paths of reflected and refracted rays ($\mathbf{r_r}$ and $\mathbf{r_t}$). Color predictions from the opaque background are obtained by querying a pre-learned radiance field along these rays. Integrated with rendering and Fresnel equations, these predictions yield the $\hat{\mathbf{c}}$ color prediction, compared to the actual color $\mathbf{c}_{\text{gt}}$ using a photometric loss. This loss improves ray accuracy and reduces photometric loss, refining the surface estimate.

1. Introduction

For the task of surface reconstruction from images, the advent of neural implicit representations has led to significant advances in capturing complex surfaces with a precision that extends beyond traditional methods [10, 23, 35, 39, 40]. Despite this progress, most reconstruction methods operate under the assumption that light propagates in linear paths—an assumption that holds for a variety of materials but falls short when dealing with refractive elements such as glass. In such contexts, light takes non-linear paths, its course altered by the phenomena of refraction and reflection, presenting unique challenges in accurately reconstructing surfaces and rendering the view-dependent phenomena intrinsic to refractive materials.

The pursuit of reconstructing the 3D geometry of refractive objects has spurred the development of a variety of methods. To accurately model transparent objects, existing methods consider the curved light paths caused by refraction and reflection [1, 15, 24]. For instance, to achieve better novel view synthesis, Pan et al. [24] employ the Eikonal equation [13] to accurately compute these paths with known refractive indices and object geometries. Li et al. [15] combine rendering with cost volume layers to refine surface normals for detailed point cloud reconstruction. Furthermore, surface reconstruction of transparent objects benefits from controlled setups that are adept at capturing complex light interactions [6, 7, 16, 17, 20, 26, 32, 37]. These setups often employ custom-designed hardware, which are critical for obtaining ray–location correspondence supervision [16, 17, 26, 37]. For instance, as illustrated in Fig. 1 (left), when an estimated surface is considered, the calculated ray hits the monitor at point $\mathbf{p}'$. This point typically

has a measurable gap from the ground-truth point $\mathbf{p}$, as determined by the well-controlled camera and background setup. This gap is then utilized as a form of supervision, enabling the refinement of the estimated surface to more closely resemble the real one. However, their applicability is limited in real-world scenarios where transparent objects are captured under natural lighting conditions.

In this work, we explore an alternative form of supervision for surface refinement. Our observation is that the neural implicit surface (NeuS) model [35] can model the geometry and radiance of the surrounding environment with high fidelity, even though it is unable to accurately render transparent objects. This motivates us to leverage the learned background model to establish a supervision signal. As shown in Fig. 1 (right), we employ sphere tracing and Snell's Law on an initial object surface to compute the paths of reflected and refracted rays ($\mathbf{r_r}$ and $\mathbf{r_t}$). We then use a learned radiance field to render the predicted color $\hat{\mathbf{c}}$, by integrating along these rays with the rendering equation and then combining the reflected and refracted terms with the Fresnel equations. The rendered color is compared with the ground-truth color $\mathbf{c}_{\text{gt}}$ in a photometric color consistency loss. Optimizing this loss updates the surface estimate to better redirect the reflected and refracted rays to the correct locations in the opaque background.

To achieve this, we introduce a differentiable technique called transparent neural surface refinement (TNSR), which hinges on backpropagating the color consistency loss to the weights of the implicit surface network. By framing the computation of ray–surface intersections as a nested optimization problem, Proposition 4.6 of Gould et al. [11] can be used to analytically derive the required closed-form gradients. This connects the surface refinement with the volume rendering, enabling end-to-end training solely on multi-view RGB images. We combine our approach with the NeuS model [35] and demonstrate improvements in the synthesis of novel views and estimation of accurate geometries for transparent objects. Our contributions are:

- a method (TNSR) for reconstructing the geometry of transparent objects in unconstrained scenes, eliminating the need for complex camera setups and relying solely on photometric consistency for surface refinement; and
- a fully differentiable nested optimization strategy for computing piece-wise linear refracted and reflected rays that can be used for volume rendering, facilitating the propagation of a photometric error signal back to the parameters of the implicit surface network. It thus establishes a seamless, integrated connection between surface refinement and volume rendering.

2. Related Work

Neural 3D Reconstruction. Surface reconstruction from multi-view data remains a pivotal challenge in 3D vision. Broadly, reconstruction techniques bifurcate into explicit and implicit methods, contingent upon the surface representation. Explicit methods utilize representations like voxels [4, 29] and triangular meshes [2, 14], while implicit methods use neural network-parameterized implicit representations and have recently demonstrated notable prowess at surface reconstruction. Unlike their explicit counterparts, these representations are continuous, offering adaptive resolution capabilities. Prevailing neural implicit approaches include the occupancy function [18] and the signed distance function (SDF) [25]. Various methods, such as DeepSDF [25] and Point2Surf [9], have been introduced for implicit surface reconstruction from point clouds. In addition, MeshSDF [28] uses SDFs to create an end-to-end differentiable surface mesh parameterization, enabling topology changes within mesh structure. Leveraging neural implicit fields, reconstruction strategies like IDR [39] harness differentiable ray casting to delineate object surfaces via the zero-level set of SDF representations. Several hybrid approaches, including UNISURF [23], VolSDF [40], and NeuS [35], propose to combine implicit surface representations with volume rendering tehniques [19]. Moreover, many techniques (*e.g.*, patch-match [8, 10]) have been proposed to improve the NeuS model for surface reconstruction. However, these methods often presuppose straight-line light transport based on the emission and absorption model, which, while suitable for opaque surfaces, does not fully account for transparent objects where ray paths are altered by light refraction and reflection.

Reconstructing Transparent Objects. In the pursuit of reconstructing 3D geometries for refractive objects, various methods have been explored. Controlled setups have proven effective in capturing complex light interactions, such as moving point light sources [6, 21], light field probes [36], and gray-coded patterns [16, 17, 26, 37]. Their specially designed hardware enables the acquisition of detailed information such as ray–location correspondences [16, 26, 37], providing critical supervision for more accurate modeling. For example, turntable and gray-coded backgrounds are used in Li et al. [16] to capture ray–location correspondences. Han et al. [12] reconstruct transparent objects with unknown refractive indices by immersing them partially in a liquid to modify incident light paths, then recovering the object surface through triangulation of these altered paths. Li et al. [15] assume known environment illumination and refractive indices. They incorporate rendering and cost volume layers to model reflection and refraction, optimizing surface normals for precise point cloud reconstruction. Other works aim to reconstruct objects inside the refractive and reflective transparent object [27, 31]. In contrast, this work presents transparent neural surface refinement, which does not require a well-controlled camera setup but instead leverages color consistency to update the implicit surface.

Novel View Synthesis of Transparent Objects. Novel view synthesis has increasingly dealt with the simulation of curved light paths within refractive media [1, 24]. The approach of Pan et al. [24] involves using the Eikonal equation [13], assuming the refractive index and object geometry are known, to calculate how light bends due to refraction. Bemana et al. [1] proposed a method that does not rely on a known refractive index, instead applying multi-step ODE solvers to infer the refractive field. NEMTO [34] introduces a simplified model for refractive interaction by considering objects against an infinitely distant background. This abstraction permits the use of an MLP to efficiently predict color along a ray, focusing on the ray's exit direction. Additionally, MS-NeRF [41] partitions Euclidean space into virtual sub-spaces to model the light interactions. Unlike these works, our approach computes the refracted and reflected rays explicitly via a nested optimization scheme, allowing it to query a learned model of the surrounding environment and thereby obtain photometric cues as to the accuracy of the modeled transparent geometry.

3. Transparent Neural Surface Refinement

Problem Definition. Our goal is to reconstruct the geometry and view-dependent appearance of scenes containing solid transparent objects, from a set of N images with known camera poses and intrinsic matrices. The challenge arises from the unique properties of transparent objects, which distort appearance through light refraction and reflection. This distortion deviates from the typical straight-ray assumption in neural 3D reconstruction models, necessitating a more nuanced approach to accurately model the interaction of light with these objects.

3.1. Revisiting Neural Implicit Surfaces

For the initial stage of surface reconstruction in our pipeline, we use NeuS [35] as the core model, which effectively combines the capabilities of implicit surface [39] and neural field [19] approaches. This integration facilitates the learning of smooth surface geometries, which are helpful for our subsequent surface refinement processes.

Surface Representation. The geometry is implicitly encoded as a Signed Distance Function (SDF), which maps a spatial position $\mathbf{x} \in \mathbb{R}^3$ to its signed distance from a surface, where negative values indicate points inside a watertight surface. This mapping $\phi_{\text{SDF}} : \mathbb{R}^3 \to \mathbb{R}$ is parameterized by an MLP. The surface is defined by the zero level set of the SDF, given by $\mathcal{S} = \{\mathbf{x} \in \mathbb{R}^3 \mid \phi_{\text{SDF}}(\mathbf{x}) = 0\}$. The view-dependent appearance is estimated by an MLP parameterizing the function $\phi_{\text{color}} : \mathbb{R}^3 \times \mathbb{R}^3 \times \mathbb{R}^3 \times \mathbb{R}^D \to [0, 1]^3$, which predicts the color $\mathbf{c}$ of a 3D point from its position $\mathbf{x}$, view direction $\mathbf{d}$, normal $\mathbf{n}$, and geometry feature $\mathbf{g}$.

Volume Rendering. To render a pixel, a ray is cast from the camera center $\mathbf{x}_0$, extending in the direction $\mathbf{d}$ through the pixel center. The color for this pixel is obtained by integrating the color contributions at sampled points $\{\mathbf{x}_i = \mathbf{x}_0 + t_i\mathbf{d} \mid t_i > 0\}$ along the ray, given by

$$\hat{\mathbf{c}}(\mathbf{x}_0, \mathbf{d}) = \sum_{i=1}^{N} \alpha_i \mathbf{c}_i \prod_{j=1}^{i-1} (1 - \alpha_j), \tag{1}$$

where α_i represents the discrete opacity and $\mathbf{c}_i$ the color at point $\mathbf{x}_i$, respectively. The product term here is the accumulated transmittance up to that point.

To adapt volume rendering for use with an SDF network, NeuS [35] diverges from NeRF's approach of predicting density directly. Instead, NeuS derives an unbiased and occlusion-aware weighting function. This function is based on an opacity value computed in closed-form from the predicted signed distance, given by

$$\alpha_i = \max\left\{ \frac{\Phi_s(\phi_{\text{SDF}}(\mathbf{x}_i)) - \Phi_s(\phi_{\text{SDF}}(\mathbf{x}_{i+1}))}{\Phi_s(\phi_{\text{SDF}}(\mathbf{x}_i))}, 0 \right\}, \tag{2}$$

where $\Phi_s(x) = (1 + e^{-sx})^{-1}$ is the sigmoid function, parameterized by a learnable scalar s. As optimization progresses, $1/s$ approaches zero, yielding a steep transition in the function.

NeuS optimizes a photometric loss function, which compares the predicted color $\hat{\mathbf{c}}(\mathbf{x}_0, \mathbf{d})$ with the ground-truth color $\mathbf{c}_{\text{gt}}(\mathbf{x}_0, \mathbf{d})$ from the input images,

$$\mathcal{L}_{\text{c}} = \frac{1}{|\mathcal{R}|} \sum_{(\mathbf{x}_0, \mathbf{d}) \in \mathcal{R}} \left\| \hat{\mathbf{c}}(\mathbf{x}_0, \mathbf{d}) - \mathbf{c}_{\text{gt}}(\mathbf{x}_0, \mathbf{d}) \right\|^2, \tag{3}$$

where $\mathcal{R}$ denotes the set of training rays, and each ray is represented by its origin $\mathbf{x}_0$ and direction $\mathbf{d}$. To regularize the SDF, an Eikonal loss is applied at the N points sampled along the ray to encourage a unit-norm SDF gradient,

$$\mathcal{L}_{\text{e}} = \frac{1}{N} \sum_{i=1}^{N} (\|\nabla \phi_{\text{SDF}}(\mathbf{x}_i)\|_2 - 1)^2. \tag{4}$$

Surface Normals. For a point $\mathbf{x} \in \mathcal{S}$ on the zero level set, the surface normal is given by

$$\mathbf{n}(\mathbf{x}) = \frac{\nabla \phi_{\text{SDF}}(\mathbf{x})}{\|\nabla \phi_{\text{SDF}}(\mathbf{x})\|}. \tag{5}$$

To promote the smoothness of the reconstructed surface, we encourage the normals of surface points and their immediate neighbors to be similar:

$$\mathcal{L}_{\text{n}} = \frac{1}{|\mathcal{T}|} \sum_{\mathbf{x} \in \mathcal{T} \subset \mathcal{S}} \left\| \mathbf{n}(\mathbf{x}) - \mathbf{n}(\mathbf{x} + \boldsymbol{\epsilon}) \right\|, \tag{6}$$

where $\boldsymbol{\epsilon}$ is a small random 3D perturbation and $\mathcal{T}$ is a set of surface points. Following Oechsle et al. [23], surface points are retrieved by root finding along the ray.

The overall loss function of NeuS is

$$\mathcal{L} = \mathcal{L}_{\text{c}} + \lambda_{\text{e}} \mathcal{L}_{\text{e}} + \lambda_{\text{n}} \mathcal{L}_{\text{n}}, \tag{7}$$

with loss coefficients λ.

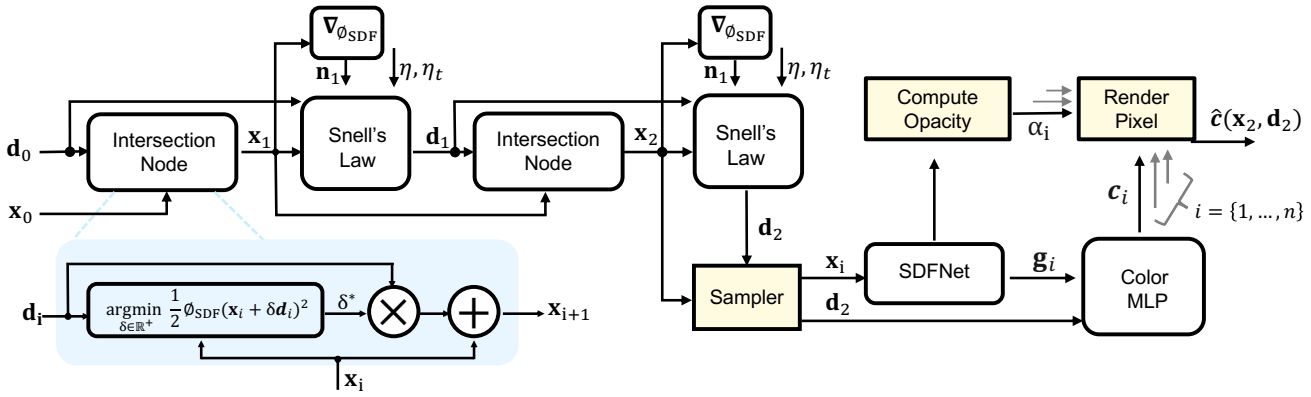

Figure 2. Flowchart of neural surface refinement for transparent object modeling. Our framework employs an SDF ϕ_{SDF} for the surface representation and incorporates physical refraction and reflection tracing. It starts with a ray from $\mathbf{x}_0$ in direction $\mathbf{d}_0$, treating the distance to the surface intersection as an optimization problem. Based on this, the intersection nodes calculate ray–surface intersections at $\mathbf{x}_1$ and $\mathbf{x}_2$. Utilizing Snell's law, the ray directions $\mathbf{d}_1$ and $\mathbf{d}_2$ are determined. The outgoing direction, along with intersection point $\mathbf{x}_2$, is used in standard NeuS volume rendering to produce the rendered color $\hat{\mathbf{c}}(\mathbf{x}_2, \mathbf{d}_2)$. The reflected color is computed similarly, using the first intersection point $\mathbf{x}_1$ and the reflected direction $\mathbf{d}_1^r$, and the two color contributions are combined using the Fresnel equations.

3.2. Surface Refinement

Overview. Our primary objective is to refine the surface geometry of a transparent object, initialized by an SDF that was optimized under the assumption that all surfaces were opaque. Starting from this initial surface representation, we generate reflected and refracted rays through ray tracing. Existing methods depend on precise camera setups to obtain ground-truth ray–location correspondences for guiding surface refinement [16, 26, 37]. Additionally, they assume a known refractive index. In comparison, our method uses NeuS's ability to precisely learn the geometry and appearance of opaque background surfaces, in order to define a color consistency loss for refracting and reflecting rays. This is achieved by rendering colors using the computed piecewise-linear rays and the learned radiance field, which can be compared with the ground-truth color.

3.2.1 Ray Tracing

We consider the light transport of both refraction and reflection to model transparent objects. As depicted in Fig. 3, we begin with an estimated surface of the object and apply a sphere tracing algorithm [39] to locate the first surface intersection point $\mathbf{x}_1$. After identifying this point, we apply Snell's Law to obtain the paths of light as it reflects off and refracts through the surface. Given the intersection point $\mathbf{x}_1$, the incoming ray direction $\mathbf{d}_0$, and the unit surface normal $\mathbf{n}_1$, the direction of the *reflected* ray is given by

$$\mathbf{d}_1^r = \mathbf{d}_0 - 2(\mathbf{n}_1^\top \mathbf{d}_0)\mathbf{n}_1, \tag{8}$$

and the *refracted* direction is given by

$$\mathbf{d}_1 = \frac{\eta}{\eta_t}\mathbf{d}_0 - \left(\frac{\eta}{\eta_t}\mathbf{n}_1^\top \mathbf{d}_0 + \sqrt{1 - \frac{\eta^2}{\eta_t^2}\left(1 - (\mathbf{n}_1^\top \mathbf{d}_0)^2\right)}\right)\mathbf{n}_1, \tag{9}$$

where η_t denotes the refractive index of the transparent object, and $\eta = 1.0003$ is that of air.

Upon establishing the refracted direction $\mathbf{d}_1$, sphere tracing is again employed to ascertain the second intersection point $\mathbf{x}_2$. From there, we apply Snell's Law again to compute the final outgoing direction $\mathbf{d}_2$. For computational efficiency, we only consider light paths with one reflection and (up to) two refractions, though the algorithm can be extended to accommodate additional reflections and refractions with only minor modifications.

Color Rendering. For transparent objects, the final radiance is a combination of the radiance of the reflected ray $\{\mathbf{x}_1 + t\mathbf{d}_1^r \mid t > 0\}$ and the radiance of the outgoing refracted ray $\{\mathbf{x}_2 + t\mathbf{d}_2 \mid t > 0\}$. We employ the Fresnel equations [3] to determine how much of the incident radiance at the surface point comes from reflection as compared to refractive transmission:

$$F_r = \frac{1}{2}\left(\frac{\eta \mathbf{d}_0^\top \mathbf{n}_1 - \eta_t \mathbf{d}_1^\top \mathbf{n}_1}{\eta \mathbf{d}_0^\top \mathbf{n}_1 + \eta_t \mathbf{d}_1^\top \mathbf{n}_1}\right)^2 + \frac{1}{2}\left(\frac{\eta_t \mathbf{d}_0^\top \mathbf{n}_1 - \eta \mathbf{d}_1^\top \mathbf{n}_1}{\eta_t \mathbf{d}_0^\top \mathbf{n}_1 + \eta \mathbf{d}_1^\top \mathbf{n}_1}\right)^2. \tag{10}$$

With the Fresnel reflectance F_r, the final radiance for a camera ray $(\mathbf{x}_0, \mathbf{d}_0)$ is given by

$$\hat{\mathbf{c}}(\mathbf{x}_0, \mathbf{d}_0) = F_r \hat{\mathbf{c}}(\mathbf{x}_1, \mathbf{d}_1^r) + (1 - F_r)\hat{\mathbf{c}}(\mathbf{x}_2, \mathbf{d}_2), \tag{11}$$

where $\hat{\mathbf{c}}(\mathbf{x}_1, \mathbf{d}_1^r)$ and $\hat{\mathbf{c}}(\mathbf{x}_2, \mathbf{d}_2)$, computed using Eq. 1, represent the rendered colors obtained from the reflected and refracted rays, respectively.

Searching for Refractive Index. We start with NeuS-initialized surface geometry, optimized assuming full opacity. We find the refractive index that maximizes the PSNR of novel views (Eq. 11). This process requires no extra training; it merely renders images using the pre-trained NeuS model while varying the refractive index (see Fig.7).

3.2.2 Differentiable Refinement

Consider the optimization problem that finds the distance to the first surface intersection (zero level crossing) along the ray from $\mathbf{x}_i$ with direction $\mathbf{d}_i$,

$$\delta_i(\boldsymbol{\theta}, \mathbf{x}_i, \mathbf{d}_i) \in \arg\min_{\delta \geqslant \epsilon} \quad \delta \qquad (12)$$
$$\text{subject to} \quad \phi_{\text{SDF}}(\mathbf{x}_i + \delta \mathbf{d}_i; \boldsymbol{\theta}) = 0,$$

and further than a small positive constant ϵ to avoid trivial solutions. Assuming δ_i exists, and noting that the inequality $\delta \geqslant \epsilon$ is inactive, then by Proposition 4.6 from Gould et al. [11] we can compute the derivatives as

$$\frac{\mathrm{d}\delta_i}{\mathrm{d}\boldsymbol{\theta}} = -\frac{1}{\mathbf{n}_{i+1}^{\mathsf{T}} \mathbf{d}_i} \frac{\partial}{\partial \boldsymbol{\theta}} \phi_{\text{SDF}}(\mathbf{x}_{i+1}; \boldsymbol{\theta}) \qquad (13)$$

$$\frac{\mathrm{d}\delta_i}{\mathrm{d}\mathbf{x}_i} = -\frac{\mathbf{n}_{i+1}}{\mathbf{n}_{i+1}^{\mathsf{T}} \mathbf{d}_i} \qquad (14)$$

$$\frac{\mathrm{d}\delta_i}{\mathrm{d}\mathbf{d}_i} = -\frac{\delta_i \mathbf{n}_{i+1}}{\mathbf{n}_{i+1}^{\mathsf{T}} \mathbf{d}_i}, \qquad (15)$$

where the intersection point is $\mathbf{x}_{i+1} = \mathbf{x}_i + \delta_i \mathbf{d}_i$ and its normal vector is $\mathbf{n}_{i+1} = \frac{\mathrm{d}}{\mathrm{d}\mathbf{x}} \phi_{\text{SDF}}(\mathbf{x}_{i+1})$. This amounts to implicit differentiation of $\phi_{\text{SDF}} = 0$. The full derivation and 6-line code implementation is provided in the appendix.

These derivatives work for any number of sequential refractions, where the next direction vector is computed by Snell's Law (Eq. 9), and subsequent intersection points $\mathbf{x}$ now depend on the previous intersection points. This can be implemented in PyTorch by defining an Autograd function, which allows PyTorch to handle backpropagation through the SDF parameters. After obtaining the intersection point $\mathbf{x}_j$ and direction vector $\mathbf{d}_j$ upon exiting the refractive medium, the color is integrated along this new outgoing ray $(\mathbf{x}_j, \mathbf{d}_j)$ and is compared to the ground-truth color. The error signal is propagated back to the SDF network parameters $\boldsymbol{\theta}$ via the derivatives defined above.

4. Experiments

4.1. Experimental Setup

Datasets. We evaluate our method on both real-world and synthetic datasets. For the real-world datasets, we use *Glass* and *Ball* from Bemana et al. [1] and follow their dataset split to report results. We use the camera poses provided by [1], which are computed with COLMAP.

For the synthetic datasets, we create four transparent objects (*Kitty* [38], *Optical Ball*, *Bottle*, and *Ellipsoid*). We use Blender with the physically-based rendering engine *Cycles* to generate photorealistic images. The material of the objects is set by *Principled BSDF* to appear as glass with various refractive indices. Note that we do *not* assume that the surrounding environment is infinitely far away, unlike many existing works [33, 34, 43]. Instead, we use near-object background panels to mimic real-world conditions closely. For each object, we generate a total of 250 images, distributed as 150 for training, 50 for validation, and 50 for testing. The camera views are randomly sampled on a full sphere of radius 4. The resolution of images is 800×800. Sample images from each scene are shown in Fig. 5.

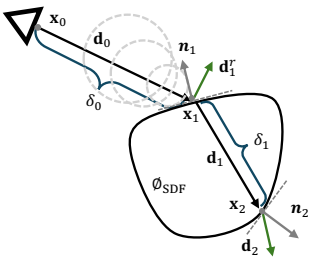

Figure 3. Ray tracing with a transparent object. Start with sphere tracing to locate the first intersection $\mathbf{x}_1$ of the ray with the signed distance function ϕ_{SDF}. Using $\mathbf{n}_1$, Snell's law finds $\mathbf{d}_1^r$ and $\mathbf{d}_1$. Tracing further yields the second intersection $\mathbf{x}_2$ and computes the exiting direction $\mathbf{d}_2$. For transparency, the camera ray $(\mathbf{x}_0, \mathbf{d}_0)$ combines radiance from the reflected ray $(\mathbf{x}_1, \mathbf{d}_1^r)$ and the exiting refracted ray $(\mathbf{x}_2, \mathbf{d}_2)$.

Implementation Details and Compared Methods. For novel view synthesis, we compare our proposed approach to three NeRF methods: TensoRF [5], Instant-NGP [22], and Nerfacto [30]. Moreover, we also include three refraction-specific methods for comparison: MS-NeRF [41], SampleNeRFRO [24], and Eikonal Fields [1]. We do not compare with NEMTO [34], since it assumes a known background or environment map at an infinite distance, and so cannot be used with real data. We use the publicly available implementations to run each method on all datasets. We do not include SampleNeRFRO [24] for comparison, as it assumes that the true geometry of transparent objects is known. For geometry reconstruction, we include UNISURF [23], VolSDF [40], and NeuS [35] for comparison. We use the implementation provided by SDFStudio [42] to run all the experiments. We build our method upon NeuS and adopt its default hyper-parameters for consistency: λ_{e} is set at 0.1 and λ_{n} is set at 0.005 in Eq. 7.

Evaluation Metrics. To assess the reconstructed geometry, we utilize the chamfer-L_1 distance as defined in [18] (lower values correlate with better geometric accuracy). The background does not constitute part of the ground-truth surface; consequently, we exclude the background when computing this metric. To evaluate the novel view synthesis results, we employ three visual quality metrics: peak signal-to-noise ratio (PSNR), structural similarity index (SSIM), and learned perceptual image patch similarity (LPIPS). A higher value of PSNR and SSIM indicates better visual quality, while a lower value of LPIPS signifies better perceptual similarity to the ground truth.

Model	Ball [1]			Glass [1]			Optical Ball			Bottle			Kitty			Ellipsoid		
	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS
TensoRF [5]	21.41	0.735	0.187	20.49	0.695	0.226	22.42	0.806	0.327	20.76	0.786	0.267	19.37	0.782	0.384	22.45	0.850	0.195
Instant-NGP [22]	21.56	0.790	0.121	21.42	0.748	0.148	20.93	0.800	0.330	20.48	0.784	0.296	19.51	0.783	0.362	23.61	0.886	0.146
Nerfacto [30]	21.67	0.797	0.113	**22.14**	0.774	0.121	20.73	0.794	0.300	20.94	0.828	0.196	19.14	0.782	0.309	23.39	0.894	0.128
MS-NeRF [41]	**22.35**	**0.810**	**0.105**	21.83	**0.781**	**0.119**	21.36	0.822	0.281	21.35	0.856	0.161	19.57	0.800	0.240	23.01	0.906	0.110
Eikonal Fields [1]	21.64	0.699	0.217	20.92	0.663	0.262	13.17	0.511	0.507	13.82	0.404	0.485	13.82	0.649	0.529	13.76	0.617	0.521
NeuS [35]	22.24	0.780	0.129	21.95	0.754	0.136	22.84	0.812	0.249	21.13	0.859	0.166	19.43	0.805	0.271	23.36	0.894	0.128
NeuS+Ref.	21.15	0.773	0.128	20.15	0.745	0.141	22.57	0.815	0.127	22.59	0.854	0.115	18.76	0.805	0.170	25.21	0.902	0.073
Ours	21.80	0.785	**0.105**	21.30	0.754	0.120	**24.07**	**0.826**	**0.112**	**23.20**	**0.866**	**0.084**	**19.61**	**0.812**	**0.163**	**25.24**	**0.915**	**0.061**

Table 1. Quantitative evaluation on the test set of six real datasets of refractive objects. We provide a comprehensive analysis of performance metrics on the test set, encompassing PSNR (↑), SSIM (↑), and LPIPS (↓), across various NeRF models: TensoRF [5], Instant-NGP [22], Nerfacto [30], MS-NeRF [41], Eikonal Fields [1], and ours. "NeuS+Ref." denotes rendering scenes using refracted and reflected rays analytically calculated based on surfaces learned by NeuS.

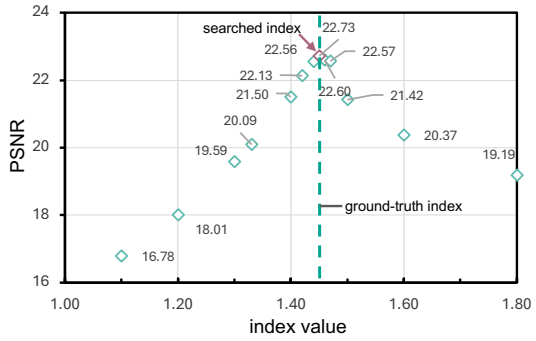

Figure 4. Refractive index estimation. We accurately search the index on *Optical Ball* with the NeuS+Ref. model, identifying the precise ground-truth index (in this case, 1.45).

Model	↓ Chamfer L_1 ($\times 10^{-3}$)			
	Optical Ball	Bottle	Kitty	Ellipsoid
Nerfacto [30]	99.67	56.37	31.79	5.16
UNISURF [23]	2.08	33.10	–	1.24
VolSDF [40]	1.98	32.01	–	1.15
NeuS [35]	1.96	31.05	19.28	1.05
Ours	**1.91**	**29.54**	**18.05**	**0.80**

Table 2. Quantitative evaluation on extracted meshes of synthetic datasets. We quantitatively compare the quality of reconstructed geometries by measuring the chamfer L_1 distance between ground-truth meshes and extracted meshes.

4.2. Evaluation Results

Effectiveness of Refractive Index Search. In Fig. 4, we showcase the effectiveness of our index search method on the *Optical Ball* using the NeuS+Ref. model. By accurately identifying the ground-truth index (1.45), we eliminate the need for assuming a known refractive index. Additionally, we plot the sensitivity of the PSNR to the refractive index.

Novel View Synthesis. We provide a quantitative comparison of our method against existing models in Tab. 1, evaluating the performance across six transparent objects. Compared with the NeuS+Ref. baseline, our method gains consistent improvements across the six datasets. This indicates that our TNSR method is effective at refining the initial surface of NeuS. Moreover, we observe that our method gains improvements over other baseline methods with respect to the LPIPS metric on *Ball* and *Glass*, albeit with a marginal reduction in PSNR and SSIM, these metrics being biased towards blurry results. We speculate that because our method does not consider the radiance of the ray segment inside the object, it yields sharper results than models like NeuS and Nerfacto. Moreover, while Eikonal Fields achieves competitive results on *Glass* and *Ball*, it is not effective on others.

We present a qualitative evaluation of our method against other methods in Fig. 5. NeuS accurately models environments but struggles with light interactions, resulting in noisy views of transparent objects, which appear blurry and foggy. By rendering scenes with refracted and reflected rays, NeuS+Ref. offers smoother visuals but still experiences jitter in object regions due to initial surface inaccuracies. MS-NeRF displays smooth outcomes in *Glass* and *Bottle*, but its performance is inconsistent in others. Eikonal Fields, despite delivering high-quality outcomes on *Ball* and *Glass*, fails to effectively estimate the refractive field in other scenarios, resulting in noisy outputs. In contrast, our method enhances the estimated surface of NeuS and accurately captures refraction and reflection, yielding more convincing and visually appealing renders.

Surface Reconstruction. First, our quantitative analysis in Tab. 2 evaluates the geometry enhancement from our transparent neural surface refinement. We compare with three SDF-based methods (UNISURF [23], VolSDF [40], and NeuS [35]), alongside Nerfacto [30], which exports meshes

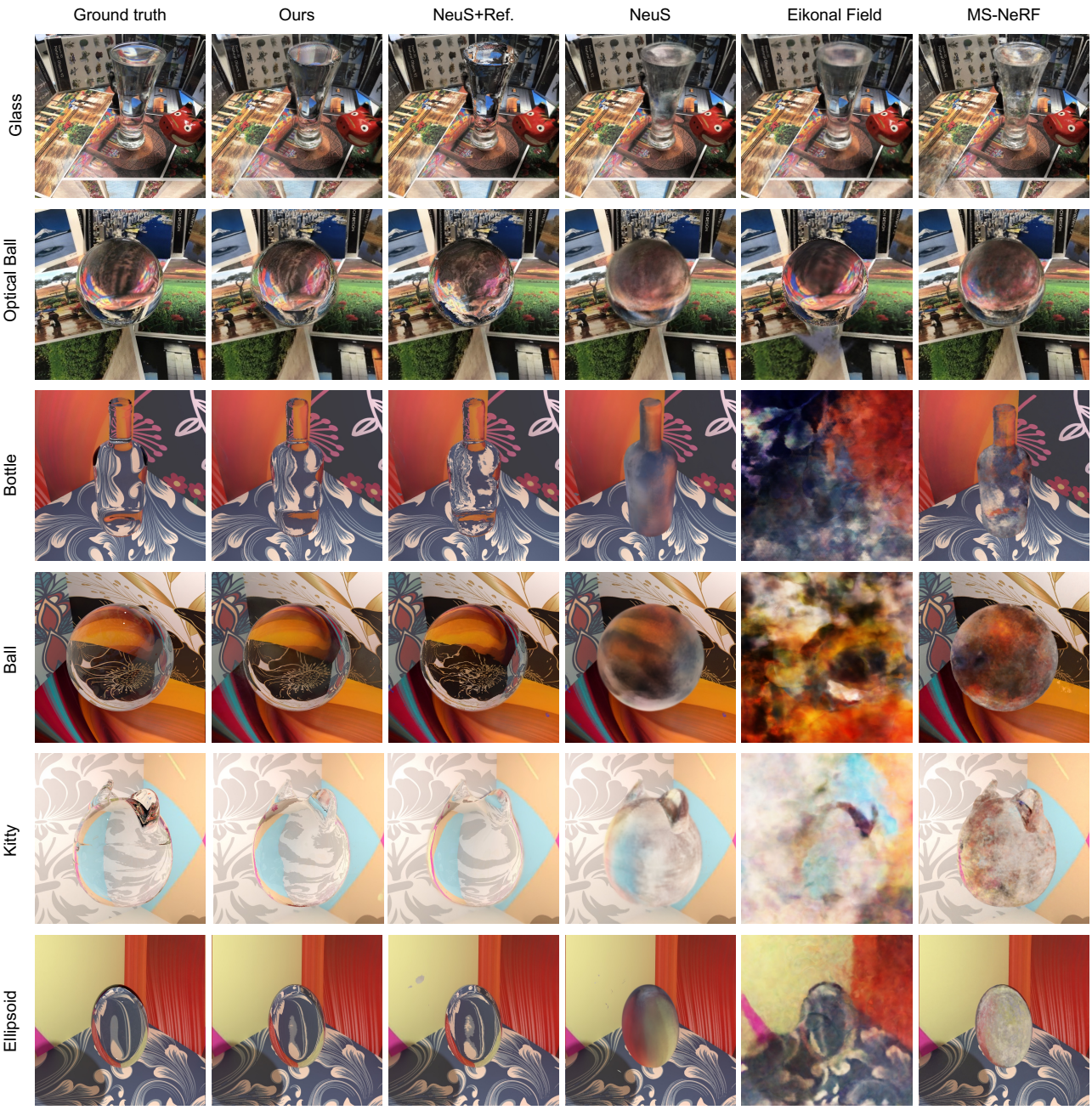

Figure 5. Qualitative comparison of novel view synthesis on transparent objects. Visual results are displayed across six distinct refractive datasets showcasing ground-truth novel views alongside renders from multiple methods, including our method, NeuS+Ref., NeuS [35], Eikonal Field [1], and MSNeRF [41]. Each row corresponds to a different transparent dataset. Our method is effective in handling refraction effects, leading to smoother and clearer outcomes in novel view synthesis compared to other models.

using Poisson surface reconstruction. Results indicate our method's superiority with a lower chamfer L_1 distance. For *Kitty*'s complex shape, both VolSDF and UNISURF fail in surface reconstruction. Additionally, Nerfacto exhibits higher chamfer L_1 distance, implying inferior reconstruction. In contrast, our method, compared to NeuS, achieves lower chamfer L_1 distance, showcasing its effectiveness. Second, qualitative comparison in Fig. 6 highlights differences between our method, NeuS, and Nerfacto. Nerfacto struggles with transparent objects, resulting in noisy sur-

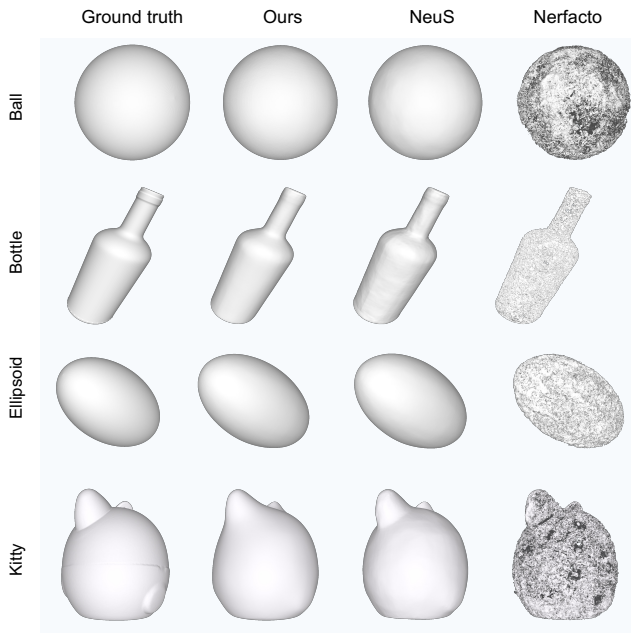

Figure 6. Visualization of 3D shape reconstruction across four refractive objects. We compare our method with NeuS [35], Nerfacto [30], and the ground-truth surfaces. The results indicate that our method produces smoother reconstructions of transparent objects, while NeuS's reconstructions have rough artifacts and Nerfacto's reconstructions have many holes.

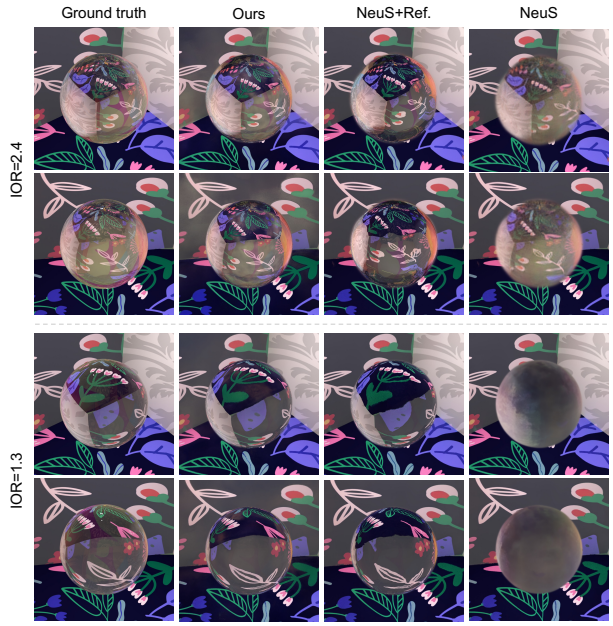

Figure 7. Qualitative comparison of novel view synthesis for transparent media with different refractive indices. We display two views for each method and show that our approach achieves more plausible results that closely align with ground truth, regardless of the material's optical properties.

faces. Conversely, our method offers smoother reconstructions compared to NeuS, suggesting improved modeling.

However, it is important to acknowledge that while our approach improves geometry, it has its limitations. Particularly in areas with multiple light refractions, like the ears of *Kitty*, our method yields imperfect results. Extending our optimization approach to handle multiple refractions is likely to address these issues.

Robustness to Index of Refraction (IoR). To demonstrate the robustness of our method, we further conduct tests on *Optical Ball* with IoRs of 2.4 and 1.3. As shown in Fig. 7, we observe that our method works well on these refractive materials, producing visually plausible novel view results. For instance, at an IoR of 2.4, our method is effective in managing strong reflections as well as modeling light refraction, when compared to NeuS+Ref and NeuS.

5. Conclusion

In this work, we introduce Transparent Neural Surface Refinement (TNSR), specifically designed to model the optical properties of transparent objects. TNSR integrates physical refraction and reflection tracing into the reconstruction process, effectively addressing the challenges posed by traditional volume rendering techniques in handling transparencies. Central to TNSR is its use of a differentiable technique that analytically computes gradients at ray–surface intersections, seamlessly linking surface refinement with volume rendering. This enables end-to-end training using only multi-view RGB images. Our experiments with TNSR have shown significant improvements in both novel view synthesis and the geometric accuracy of the modeled transparent objects, highlighting the efficacy of TNSR.

Limitations and Future Directions. Transparent neural surface refinement shows promise but has limitations. First, we only consider light paths with (up to) two refractions and one explicit reflection. While this already gives dramatically improved results, extending our approach to an arbitrary number of refractions would potentially help with complex objects. Our proposed nested optimization can naturally extend to additional bounces, which we will explore in future work. Second, our approach assumes that the scene is composed of fully transparent or opaque objects and cannot model translucent objects. In future work, we intend to extend our approach to handle translucent objects by allowing the radiance of the ray segment inside the object to contribute to the total radiance.

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

References

[1] Mojtaba Bemana, Karol Myszkowski, Jeppe Revall Frisvad, Hans-Peter Seidel, and Tobias Ritschel. Eikonal fields for refractive novel-view synthesis. In *ACM SIGGRAPH 2022 Conference Proceedings*, pages 1–9, 2022. 1, 3, 5, 6, 7

[2] Jean-Daniel Boissonnat and Bernhard Geiger. Three-dimensional reconstruction of complex shapes based on the delaunay triangulation. In *Biomedical image processing and biomedical visualization*, pages 964–975, 1993. 2

[3] Max Born and Emil Wolf. *Principles of optics: electromagnetic theory of propagation, interference and diffraction of light*. Elsevier, 2013. 4

[4] Adrian Broadhurst, Tom W Drummond, and Roberto Cipolla. A probabilistic framework for space carving. In *Int. Conf. Comput. Vis.*, pages 388–393, 2001. 2

[5] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. Tensorf: Tensorial radiance fields. In *Eur. Conf. Comput. Vis.*, 2022. 5, 6

[6] Tongbo Chen, Michael Goesele, and H-P Seidel. Mesostructure from specularity. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 1825–1832, 2006. 1, 2

[7] Zhaopeng Cui, Jinwei Gu, Boxin Shi, Ping Tan, and Jan Kautz. Polarimetric multi-view stereo. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 1558–1567, 2017. 1

[8] François Darmon, Bénédicte Bascle, Jean-Clément Devaux, Pascal Monasse, and Mathieu Aubry. Improving neural implicit surfaces geometry with patch warping. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 6260–6269, 2022. 2

[9] Philipp Erler, Paul Guerrero, Stefan Ohrhallinger, Niloy J Mitra, and Michael Wimmer. Points2surf learning implicit surfaces from point clouds. In *Eur. Conf. Comput. Vis.*, pages 108–124, 2020. 2

[10] Qiancheng Fu, Qingshan Xu, Yew Soon Ong, and Wenbing Tao. Geo-neus: Geometry-consistent neural implicit surfaces learning for multi-view reconstruction. In *Adv. Neural Inform. Process. Syst.*, pages 3403–3416, 2022. 1, 2

[11] Stephen Gould, Richard Hartley, and Dylan Campbell. Deep declarative networks. *IEEE Trans. Pattern Anal. Mach. Intell.*, 44(8):3988–4004, 2021. 2, 5

[12] Kai Han, Kwan-Yee K Wong, and Miaomiao Liu. Dense reconstruction of transparent objects by altering incident light paths through refraction. *Int. J. Comput. Vis.*, 126:460–475, 2018. 2

[13] Ivo Ihrke, Gernot Ziegler, Art Tevs, Christian Theobalt, Marcus Magnor, and Hans-Peter Seidel. Eikonal rendering: Efficient light transport in refractive objects. *ACM Trans. Graph.*, 26(3):59–es, 2007. 1, 3

[14] Patrick Labatut, J-P Pons, and Renaud Keriven. Robust and efficient surface reconstruction from range data. In *Comput. Graph. Forum*, pages 2275–2290, 2009. 2

[15] Zhengqin Li, Yu-Ying Yeh, and Manmohan Chandraker. Through the looking glass: neural 3d reconstruction of transparent shapes. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 1262–1271, 2020. 1, 2

[16] Zongcheng Li, Xiaoxiao Long, Yusen Wang, Tuo Cao, Wenping Wang, Fei Luo, and Chunxia Xiao. Neto: Neural reconstruction of transparent objects with self-occlusion aware refraction-tracing. In *Int. Conf. Comput. Vis.*, pages 18547–18557, 2023. 1, 2, 4

[17] Jiahui Lyu, Bojian Wu, Dani Lischinski, Daniel Cohen-Or, and Hui Huang. Differentiable refraction-tracing for mesh reconstruction of transparent objects. *ACM Trans. Graph.*, 39(6):1–13, 2020. 1, 2

[18] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 4460–4470, 2019. 2, 5

[19] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *Eur. Conf. Comput. Vis.*, pages 405–421, 2020. 2, 3

[20] Daisuke Miyazaki and Katsushi Ikeuchi. Inverse polarization raytracing: estimating surface shapes of transparent objects. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 910–917, 2005. 1

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

[22] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. Graph.*, 41(4):1–15, 2022. 5, 6

[23] Michael Oechsle, Songyou Peng, and Andreas Geiger. Unisurf: Unifying neural implicit surfaces and radiance fields for multi-view reconstruction. In *Int. Conf. Comput. Vis.*, pages 5589–5599, 2021. 1, 2, 3, 5, 6

[24] Jen-I Pan, Jheng-Wei Su, Kai-Wen Hsiao, Ting-Yu Yen, and Hung-Kuo Chu. Sampling neural radiance fields for refractive objects. In *SIGGRAPH Asia 2022 Technical Communications*, pages 1–4. 2022. 1, 3, 5

[25] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 165–174, 2019. 2

[26] Yiming Qian, Minglun Gong, and Yee Hong Yang. 3d reconstruction of transparent objects with position-normal consistency. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 4369–4377, 2016. 1, 2, 4

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

[28] Edoardo Remelli, Artem Lukoianov, Stephan Richter, Benoit Guillard, Timur Bagautdinov, Pierre Baque, and Pascal Fua. Meshsdf: Differentiable iso-surface extraction. In *Adv. Neural Inform. Process. Syst.*, pages 22468–22478, 2020. 2

[29] Steven M Seitz and Charles R Dyer. Photorealistic scene reconstruction by voxel coloring. *Int. J. Comput. Vis.*, 35: 151–173, 1999. 2

[30] Matthew Tancik, Ethan Weber, Evonne Ng, Ruilong Li, Brent Yi, Justin Kerr, Terrance Wang, Alexander Kristoffersen, Jake Austin, Kamyar Salahi, Abhik Ahuja, David McAllister, and Angjoo Kanazawa. Nerfstudio: A modular framework for neural radiance field development. In *ACM SIGGRAPH 2023 Conference Proceedings*, 2023. 5, 6, 8

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

[32] Borislav Trifonov, Derek Bradley, and Wolfgang Heidrich. Tomographic reconstruction of transparent objects. In *ACM SIGGRAPH 2006 Sketches*, pages 55–es. 2006. 1

[33] Dor Verbin, Peter Hedman, Ben Mildenhall, Todd Zickler, Jonathan T Barron, and Pratul P Srinivasan. Ref-nerf: Structured view-dependent appearance for neural radiance fields. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 5481–5490, 2022. 5

[34] Dongqing Wang, Tong Zhang, and Sabine Süsstrunk. Nemto: Neural environment matting for novel view and relighting synthesis of transparent objects. *arXiv preprint arXiv:2303.11963*, 2023. 3, 5

[35] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. In *Adv. Neural Inform. Process. Syst.*, pages 27171–27183, 2021. 1, 2, 3, 5, 6, 7, 8

[36] Gordon Wetzstein, David Roodnick, Wolfgang Heidrich, and Ramesh Raskar. Refractive shape from light field distortion. In *Int. Conf. Comput. Vis.*, pages 1180–1186, 2011. 2

[37] Bojian Wu, Yang Zhou, Yiming Qian, Minglun Cong, and Hui Huang. Full 3d reconstruction of transparent objects. *ACM Trans. Graph.*, 37(4):1–11, 2018. 1, 2, 4

[38] Jiankai Xing, Fujun Luan, Ling-Qi Yan, Xuejun Hu, Houde Qian, and Kun Xu. Differentiable rendering using rgbxy derivatives and optimal transport. *ACM Trans. Graph.*, 41 (6):1–13, 2022. 5

[39] Lior Yariv, Yoni Kasten, Dror Moran, Meirav Galun, Matan Atzmon, Basri Ronen, and Yaron Lipman. Multiview neural surface reconstruction by disentangling geometry and appearance. In *Adv. Neural Inform. Process. Syst.*, pages 2492–2502, 2020. 1, 2, 3, 4

[40] Lior Yariv, Jiatao Gu, Yoni Kasten, and Yaron Lipman. Volume rendering of neural implicit surfaces. In *Adv. Neural Inform. Process. Syst.*, pages 4805–4815, 2021. 1, 2, 5, 6

[41] Ze-Xin Yin, Jiaxiong Qiu, Ming-Ming Cheng, and Bo Ren. Multi-space neural radiance fields. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 12407–12416, 2023. 3, 5, 6, 7

[42] Zehao Yu, Anpei Chen, Bozidar Antic, Songyou Peng, Apratim Bhattacharyya, Michael Niemeyer, Siyu Tang, Torsten Sattler, and Andreas Geiger. Sdfstudio: A unified framework for surface reconstruction, 2022. 5

[43] Kai Zhang, Fujun Luan, Qianqian Wang, Kavita Bala, and Noah Snavely. Physg: Inverse rendering with spherical gaussians for physics-based material editing and relighting. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 5453–5462, 2021. 5