Polarimetric Inverse Rendering for Transparent Shapes Reconstruction

Mingqi Shao, Chongkun Xia, Dongxu Duan, and Xueqian Wang, Member, IEEE

Abstract—The acquisition of transparent 3D shapes will facilitate many multimedia and computer vision tasks, such as game/movie production and virtual environment applications. In this work, we propose a novel method for detailed reconstruction of transparent objects by exploiting polarimetric cues. Most of existing transparent shapes reconstruction methods usually lack sufficient constraints and suffer from the over-smooth problem. Hence, we introduce polarization information as a complementary cue. Specifically, we employ the implicit representation for object’s geometry with a neural network, while the polarization render is capable of differentiably rendering the object’s polarization images from given illumination configuration. However, direct comparison of rendered polarization images to the real-world captured images will have additional errors due to the transmission in the transparent object. To make the polarimetric cues technically feasible on transparent shapes reconstruction, the concept of reflection percentage which represents proportion of the reflection component is introduced as the weight of the polarization loss. Based on controllable environment setup, we build a polarization dataset containing several solid and smooth transparent objects to verify our method. Experimental results show that our method is capable of recovering detailed shapes and improving reconstruction quality of transparent objects.

I. INTRODUCTION

ACQUIRING objects’ 3D models plays an important role to facilitate many multimedia and computer vision tasks, such as game/movie production [1], [2], [3], AR/VR application [4], [5], [6] and virtual environment interaction [7], [8]. However, the acquisition of transparent 3D shapes has always been a challenge since their visual appearance is determined by the light paths with both reflected and refracted light. The complex optical characteristic of transparent shapes makes the observation of RGB camera or commonly used depth camera contain significant error, leading to the poor performance of traditional 3D reconstruction methods such as active camera scanning and multi-view stereo.

Recently, the neural inverse rendering based on neural implicit representation [9], [10], [11], [12] has achieved remarkable performance in the task of learning 3D shapes from 2D images of opaque objects. However, high complexity of transparent objects’ light paths couples the color of transparent surface with its geometry, environment light, and viewing direction, leading to the difficulty of applying neural inverse rendering methods that exploit photometric consistency constraint, e.g., IDR (implicit differentiable renderer) [11], to transparent shapes. In addition, reconstructed shapes that only use the silhouette constraint suffer from the over-smooth problem. Li et al. [13] proposed a physical-based neural network to handle the complex optical characteristic of transparent shapes under natural lighting conditions, however, it still suffers from the same over-smooth problem. Therefore, additional cues are necessary to be introduced to recover detailed shapes. Researchers introduce ray-ray correspondence between camera rays and the rays from background pattern as the light path’s constraint as a cue for detailed shape reconstruction [14], [15], [16], [17], [18]. However, ray-ray correspondence generally requires strict calibration and precise control.

As a passive imaging principle with weak assumptions of lighting conditions, polarimetric information performs well on many tasks [19], [20], [21] since polarimetric information provides ray’s information from a new dimension in addition to intensity. The polarization state of the reflected light from the object’s surface encodes the azimuth and zenith angle of surface normal. In the context of this paper, the polarimetric cues refer to inferring important information about the object’s surface from the polarization properties of rays. This paper utilizes the polarimetric cues for detailed transparent shapes reconstruction.

However, there is a technical difficulty in applying polarization theory to transparent objects. Due to the complex light interactions of transparent objects, useful polarization information of the directly reflected light, encoding the normal vector of surface, is frequently disturbed by the transmitted light. Using interfered polarimetric cues in optimization will lead to wrong shapes. To make polarimetric cues technically feasible in...
Our method can produce detailed reconstruction results of different transparent objects with complex and irregular shapes and outperform other state-of-the-art methods.

II. RELATED WORK

A. Transparent Shapes Reconstruction

The main challenge of transparent shapes reconstruction is that their surface observation will be interfered by the transmission from background [13]. This feature is exploited by researchers to place known patterns on back of transparent objects to obtain the correspondence between emitted rays from transparent objects and the rays from background patterns. With the rays correspondence, depth or surface normals can be estimated by using the refraction theorem or path triangulation [14], [16], [17], [18]. In contrast to the methods that require strict control of settings to get accurate rays correspondence, our method only needs a dark background and uniform light intensity in directions facing the transparent object.

Recently, data-driven approaches have shown their potential in transparent shapes reconstruction. Several datasets and methods for transparent objects have been proposed [13], [22], [23], [24], [25]. Li et al. [13] propose a neural 3 d reconstruction framework for transparent objects, which simulates light transport within transparent objects through a physically-based rendering layer and uses a pre-trained network for point cloud reconstruction. However, the gap between synthetic and real-world dataset and insufficient constraint of RGB information lead to the over-smooth phenomenon of the reconstructed shapes [25]. Additional information is needed to enhance the constraint.

Polarization information is an important cue for transparent shapes estimating [26], [27] because the polarization state of light reflected from an object’s surface encodes the information of the surface normal [28]. However, the methods purely rely on polarization information usually suffer from large errors. The reflected light, which encodes the surface normal vector, is generally covered by other light components. Hence, different from these methods purely relying on polarization information, our method serves polarization information as cues for multi-view reconstruction to provide auxiliary constraints.

B. Neural Inverse Rendering

Neural inverse rendering learns 3D shapes (usually implicitly represented by neural networks) from 2D images by leveraging differentiable rendering systems [29]. Neural implicit representation adopts neural networks to represent 3D objects or scenes, which is more flexible to represent arbitrary surfaces and theoretically enables infinite resolution rendering. Recently, some works based on neural radiation fields have attempted to use neural representations to reconstruct and render transparent media such as glass and liquids, and they have achieved good rendering results [30], [31], [32], [33]. However, these methods focus on rendering images from novel views rather than detailed reconstruction of objects. While some other works have achieved outperforming results in multi-view reconstruction by leveraging neural implicit representation [10], [11], [12], [34], [35],

1 Our codes are available at https://github.com/Shaomq2187/TransPIR.
Our goal is to recover transparent shapes from multi-view 2D images by exploiting polarimetric cues and the pipeline of our method is shown in Fig. 2. Instead of starting with the space carving method [47], we adopt the neural implicit representation in IDR [11] to produce a smooth initial shape. The polarimetric render will render the polarization maps of different views and then the rendered polarization maps will be compared with the captured polarization maps to get polarization loss. Since only the polarization information of points with a high proportion of reflection component is reliable, the ray tracer will trace reflection percentage of each point to weight the polarization loss. Afterward, the weighted polarization loss guides the optimization to produce final detailed shape. In the following subsections, we will demonstrate our method in detail. In this paper, we use following two assumptions: Transparent objects are solid inside and material is uniform; Diffuse reflection component from transparent surface can be ignored.

### III. Method

Our goal is to recover transparent shapes from multi-view 2D images by exploiting polarimetric cues and the pipeline of our method is shown in Fig. 2. Instead of starting with the space carving method [47], we adopt the neural implicit representation in IDR [11] to produce a smooth initial shape. The polarimetric render will render the polarization maps of different views and then the rendered polarization maps will be compared with the captured polarization maps to get polarization loss. Since only the polarization information of points with a high proportion of reflection component is reliable, the ray tracer will trace reflection percentage of each point to weight the polarization loss. Afterward, the weighted polarization loss guides the optimization to produce final detailed shape. In the following subsections, we will demonstrate our method in detail. In this paper, we use following two assumptions: Transparent objects are solid inside and material is uniform; Diffuse reflection component from transparent surface can be ignored.
The distance \( d \) is negative when \( x \) is inside the boundary, positive outside, and zero when \( x \) is on the boundary. Similar to IDR [11], we represent the transparent objects’ geometry as a multi-layer perceptron network (MLP) [48] \( f_\theta \) with learnable parameters \( \theta \) and optimize \( f_\theta(x) \) to the object’s ground-truth SDF \( f(x) \):

\[
\text{opt} : f_\theta(x) \rightarrow f(x)
\]  

(2)

The normal \( \hat{n}_\theta(x) \) of the surface represented by MLP-based SDF \( f_\theta(x) \) can be expressed as follows:

\[
\hat{n}_\theta(x) = \nabla_x f_\theta(x)/\|\nabla_x f_\theta(x)\|_2
\]  

(3)

The derivative of \( f_\theta(x) \) can be easily acquired from the automatic differentiation mechanism.

Both the polarimetric render and ray tracer in our method require the intersections of rays and geometry, especially the ray-tracer requires all the interactions of rays including the internal refracted rays. The ray marching method, which is often used with the SDF representation, can not meet the requirements since it can only be used on one side of the SDF function. To address this issue, we adopt the secant algorithm to approach the intersections of rays and geometry at inside and outside.

As shown in Fig. 5, we denote the ray as \( x = x_0 + tv, t \geq 0 \), the intersection point of geometry as \( x_+ = x_0 + t_+v \) and the intersection point of the unit sphere as \( x_\ast = x_0 + t_\ast v \), where \( x_0 \) is the start point of the ray and \( v \) is the unit direction vector. Similar to [49], we sample 100 equal steps between 0 and \( t_\ast \), that is, \( 0 < t_1 < \ldots < t_{99} < t_\ast \). Then we find the first \( t_i \) and \( t_{i+1} \) where \( \text{sgn}(f_\theta(x_0 + t_i v)) \neq \text{sgn}(f_\theta(x_0 + t_{i+1} v)) \), the transition of signs of SDF values represents the ray crossing of the surface. The secant algorithm is used for approximation in the interval \( (t_i, t_{i+1}) \) and the nearest values to \( t_\ast \) from both sides are recorded as \( t_+ \) and \( t_\ast \), where \( \text{sgn}(f_\theta(x_0 + t_+ v)) = \text{sgn}(f_\theta(x_0 + t_\ast v)) \) and \( \text{sgn}(f_\theta(x_0 + t_- v)) \neq \text{sgn}(f_\theta(x_0 + t_- v)) \). Both the \( t_+ \) and \( t_\ast \) can be used as an approximation of \( t_\ast \), the specific selection will depend on whether the next ray is refracted or reflected. This algorithm has a slower computation efficiency, but it enables the ray-tracing technique to be applied to SDF representation. The reason for calculating two values of \( t_\ast \) is that the sign of start point will affect the correctness of intersection result. For example, when \( x_+ = x_0 + t_\ast v \) is start point of a refracted ray and its SDF value \( f_\theta(x_+^+) > 0 \), the obtained initial interval must be \( (t_0, t_\ast) \) since the refracted direction is toward inside of the shape, resulting in a wrong intersection. The secant method with two interactions provides basis for future multi-bounce (>2) ray tracing in SDF, although only 2-bounce ray tracing is used in this paper.

C. Polarimetric Rendering

In this paper, we use the Mueller calculus [50] to calculate polarization state of the specular reflection on the transparent surface. In Muller calculus, the full polarization state of light is represented by Stokes vector \( s = [s_0, s_1, s_2, s_3]^T \), where \( s_0 \) represents the light intensity, \( s_1 \) and \( s_2 \) denote the linear polarization components of the \( x \)-axis and \( 45^\circ \) directions, and \( s_3 \) represents the right circular polarization component. The three polarimetric components intensity \( I \), degree of linear polarization DoLP \( \rho \), angle of linear polarization AoLP \( \psi \) can be parameterized.
by the stokes vector:

\[ I = s_0 \]

\[ \rho = \frac{\sqrt{s_1^2 + s_2^2}}{s_0} \]

\[ \psi = \frac{1}{2} \tan^{-1} \left( \frac{s_2}{s_1} \right) \]

We only consider specular reflection on transparent surface, disregarding diffuse reflection component due to the smoothness of transparent surface. Consequently, the polarization state of transmission component, which carries little normal information, has not been rendered. When a ray reflects off object’s surface, its polarization state will change according to Fresnel specular reflection model [50]. As shown in Fig. 6, when the stokes vectors \( s_i, s_r \) of the incident ray and the reflected ray are in the same coordinate system, the transformation between \( s_i, s_r \) can be represented by a mueller matrix \( M_r \), that is,

\[ s_r = M_r s_i \]

\[ M_r = \frac{1}{2} \begin{bmatrix} (r_s^2 + r_p^2) & (r_s^2 - r_p^2) & 0 & 0 \\ (r_s^2 - r_p^2) & (r_s^2 + r_p^2) & 0 & 0 \\ 0 & 0 & 2r_s r_p & 0 \\ 0 & 0 & 0 & 2r_s r_p \end{bmatrix} \]

where \( r_s, r_p \) are the amplitude reflection coefficients perpendicular to the parallel and incident plane, respectively. Their expressions are provided here:

\[ r_s = \frac{\eta_i \cos \chi_i - \eta_\text{c} \cos \chi_\text{c}}{\eta_i \cos \chi_i + \eta_\text{c} \cos \chi_\text{c}} = -\frac{\sin(\chi_i - \chi_\text{c})}{\sin(\chi_i + \chi_\text{c})} \]

\[ r_p = \frac{\eta_\text{c} \cos \chi_\text{c} - \eta_i \cos \chi_i}{\eta_\text{c} \cos \chi_\text{c} + \eta_i \cos \chi_i} = -\frac{\tan(\chi_i - \chi_\text{c})}{\tan(\chi_i + \chi_\text{c})} \]

Eqs. (7), (8), (9) and (10) form the Fresnel specular reflection model. By implementing this model, our polar render is able to simulate the variation of polarization state when the rays reflect on object’s surface.

Since the Stokes vectors are defined in different reference frames, the coordinate frame transformations are required. Fig. 7 shows the schematic of the polarimetric rendering in our method. We adopts the transformations of frames similar to Mitsuba2 [51], and detailed expressions of \( R_i(\mathbf{v}_i; \mathbf{n}_\theta), R_o(\mathbf{v}_o; \mathbf{n}_\theta), R_c(\mathbf{v}_o) \) are presented in supplementary material. With these transformations, the stokes vector \( s_c = [s_{c_0}, s_{c_1}, s_{c_2}, s_{c_3}]^T \) captured by the polarization camera can be written as the linear operations:

\[ s_c = R_c(\mathbf{v}_o) R_o(\mathbf{v}_o; \mathbf{n}_\theta) M_r R_i(\mathbf{v}_i; \mathbf{n}_\theta) s_0 \]

Then, from the (6), the rendered AoLP \( \hat{\psi} \) is,

\[ \hat{\psi} = \frac{1}{2} \tan^{-1} \left( \frac{s_{c_2}}{s_{c_1}} \right) \]

The polarimetric render in Fig. 2 renders the AoLP maps through (11), (12), then they will be compared with the real-world captured AoLP maps to calculate polarization loss, which will be described in detail in later subsection Optimization.

D. Reflection Percentage Estimation

The polar render employs Fresnel specular reflection model to render the AoLP maps from given shape. However, in some areas, the rendered AoLP map has a large error compared with the AoLP map captured in real world as shown in Fig. 9. This error is mainly caused by the high transmission rather than shape.
The observed light from transparent surface consists of two components: the directly reflected light on surface (the specular reflection component) and the transmitted light from inside (the transmission component). The AoLP of the specular component is only related to the normal of intersection point on surface, while the AoLP of transmission component is related to normals of all intersection points in its transmission process. Therefore, the higher proportion of specular reflection, the more reliable of captured AoLP. To quantify and reduce the resulting error, we use ray tracing to calculate reflection percentage of each point on the transparent surface.

Due to high computational cost of ray-tracing based on the SDF representation and the fact that multiple ray tracing does not bring much benefit, we just implement a 2-bounce ray tracer to simplify calculation of reflection proportion. Fig. 8 illustrates the ray-tracing procedure. The rays in Fig. 8 flow from the light source to the camera, which is inverse in our actual implementation. Given the shape, illumination configuration, and the camera viewing direction $-\mathbf{v}_0$, the directions of other rays can be derived from Fresnel’s law of refraction. In each interaction, the energy of incident ray is distributed according to Fresnel term $F$ [52].

$$F_{\eta_i,\eta_t}^{v_i,v_t,n} = \frac{1}{2} \left( \frac{\eta_i v_i \cdot n - \eta_t v_t \cdot n}{\eta_i v_i \cdot n + \eta_t v_t \cdot n} \right)^2$$

$$I_r = F_{\eta_i,\eta_t}^{v_i,v_t,n}I_t$$

$$I_t = (1 - F_{\eta_i,\eta_t}^{v_i,v_t,n})I_t$$

where $I_i, I_r, I_t$ are the intensities of incident, reflected and refracted rays, $v_i, v_r, v_t$ represent their directions, respectively. $n$ is the normal vector.

Using (13), (14), (15), we can calculate the intensities of reflected ray and transmitted ray $I_{t0}, I_{r0}$ observed by the camera in Fig. 8:

$$I_{t1} = (1 - F_{\eta_{air},\eta_{glass}}^{v_t2,v_1,n})I_{t2}$$

$$I_{t0} = (1 - F_{\eta_{air},\eta_{glass}}^{v_t1,v_0,n})I_{t1}$$

$$I_{r0} = F_{\eta_{air},\eta_{glass}}^{v_r1,v_0,n}I_{r1}$$

where the values of $I_{t2}, I_{r1}$ are sampled from the illumination configuration, which is related to our dataset acquisition setup. Due to the total internal reflection, the values of $I_{t1}$ of some rays are unable to calculate from the 2-bounce ray tracer, in this case, we set $I_{t1}$ to a constant to avoid 100% reflection percentage since it is impossible in the real world.

To determine the value of $I_{t2}, I_{r1}$, we model the illumination configuration as shown in Fig. 10 according to our dataset collection setup, where $\mathbf{v}_c$ is the view direction and $\mathbf{v}_s$ is direction of the sampling ray. Our illumination sampling function can be written as follows:

$$\alpha = \arccos \left( \frac{-\mathbf{v}_c \cdot \mathbf{v}_s}{\|\mathbf{v}_c\|\|\mathbf{v}_s\|} \right)$$

$$I_{sample} = \begin{cases} 1.0, & 0 \leq \alpha < \frac{\pi}{2} - \delta \\ 0.1, & \text{otherwise} \end{cases}$$

The relative position between the light source and polarization camera is fixed and hence only the angle $\alpha$ between the direction
of sampled ray $v_s$ and view direction $v_v$ is required to determine whether the sampled ray intersects the light source. We limit the angular range that can directly sample the light source to $[0, \frac{\pi}{2} - \delta]$ since the bottom of the light source cannot fit object surface and some rays within a range $\delta$ cannot sample the light source directly. The default value of $\delta$ is $\frac{\pi}{10}$. We set the intensity of the light source to 1.0, and the intensity of other areas to 0.1 since the environment has weak illumination from diffuse reflection and others.

Finally, the reflection percentage $w$ is defined as the ratio of reflection intensity to total intensity:

$$w = \frac{I_r}{I_r + I_d}$$

(21)

As shown in Fig. 9, the real-world captured AoLP map (Fig. 9(b)) has obvious demarcation in some areas compared to the rendered AoLP map (Fig. 9(a)) since the proportion of reflection component changes. These changes are consistent with the reflection percentage map (Fig. 9(c)) rendered by the ray tracer, which further illustrates the importance of the reflection percentage. Otherwise, the error caused by the smaller reflection proportion will guide the optimization to a wrong shape.

### E. Optimization

We minimize the following loss function to optimize the MLP-based SDF $f_\theta(x)$ to the object’s ground-truth SDF $f(x)$:

$$L_{net} = L_{sil} + \lambda_{sdf} L_{sdf} + \lambda_{pol} L_{pol}$$

(22)

where $L_{sil}$, $L_{sdf}$, $L_{pol}$ represent the silhouette loss, sdf regularization loss term, and weighted polarization loss term. The default values of $\lambda_{sdf}$ and $\lambda_{pol}$ are 0.1 and 0.4, respectively.

1) **Silhouette Loss**: We adopt the same silhouette loss as in IDR [11] for shape optimization supervision, which plays an important role in initial shape reconstruction. Let $O_p \in \{0, 1\}$ be the mask values, the silhouette loss is:

$$L_{sil} = \frac{1}{\alpha \|P\|_1} \sum_{p \in P^o} \text{CE}(O_p, \text{sigmoid}(-\alpha \min_{t > 0} f_\theta(r)))$$

(23)

where $P$ represents the set of intersection points of sampled rays in the mini-batch with the surface. $P_o$ denotes the points in the mini-batch for which no ray-geometry intersection and $r = cr + tv$ represents the ray. $\alpha > 0$ is a parameter and its default value is 50.0.

2) **SDF Regularization Loss**: To encourage the $f_\theta(x)$ approximations a signed distance function, we add the regularization loss, i.e., the Eikonal regularization [53]:

$$L_{sdf} = \frac{1}{\|P\|_1} \sum_{p \in P} (\|\nabla f_\theta(p)\|_2 - 1)^2$$

(24)

3) **Weighted Polarization Loss**: We use polarimetric cues to calculate polarization loss to guide the optimization. The weighted polarization loss for each sampled ray is defined as follows:

$$L_{pol}^p = w_p \|\hat{\psi}_p - \psi_p\|_1, \quad p \in P$$

(25)

where $w_p$ is the reflection percentage of the point $p$. $\hat{\psi}_p$ represents the rendered AoLP, and $\psi_p$ denotes the real-world captured AoLP. As mentioned before, the reflection percentage is related to the reliability of the $\psi_p$, hence we employ it as the weight for the error $\|\hat{\psi}_p - \psi_p\|_1$.

We assume that the difference of the normal between the initial shape and ground-truth shape is smaller than $\varepsilon$ since the supervision of $L_{sil}$ can produce a good initial shape. With this assumption, we can clip the excessive polarization loss to avoid guiding the optimization to wrong shape:

$$L_{pol}' = \begin{cases} L_{pol}^p, & \|\hat{\psi}_p - \psi_p\|_1 \leq \varepsilon \\ 0, & \text{otherwise} \end{cases}$$

(26)

$$L_{pol} = \frac{1}{\|P\|_1} \sum_{p \in P} L_{pol}'$$

(27)

where the default value of $\varepsilon$ in this paper is $\pi/6$.

### IV. EXPERIMENTS

We implement the $f_\theta(x)$ as an MLP with 8 layers and 512 units per layer in PyTorch [54], and a skip connection from the input layer to the middle layer is added as in IDR [11]. The model is trained on an RTX 3090 GPU (24 GB). We use the Adam optimizer [55] with a learning rate of $1e-4$ to optimize the network. 20480 rays per iteration are sampled and the secant algorithm is employed to calculate the intersection points of the sampled rays to the shape. Each object is trained for 1000 epochs, the first 100 epochs are the initial shape reconstruction stage, and $\lambda_{pol}$ is set to zero. After 100 epochs, $\lambda_{pol}$ is set to 0.4(0.2 for object ELEPHANT) to introduce polarimetric cues.

For convenience, we normalize the camera’s poses to ensure that the reconstructed models are all within a unit sphere and all evaluations are performed on this basis.

We adopt the chamfer distance(CD) and chamfer normal angle(CDN) as the metrics and the metrics are calculated by uniformly sampling 10000 points from the ground-truth and reconstructed shape. The chamfer distance and chamfer normal angle are defined as follows:

$$CD = \frac{1}{2} \left( \sum_{p \in P} \min_{p' \in P^*} \|p - p'\|_1 + \sum_{p' \in P^*} \min_{p \in P} \|p - p'\|_1 \right)$$

(28)

$$Acc = \sum_{p \in P^o} \left( 1 - \frac{\hat{\mathbf{n}}_p \cdot \mathbf{n}_p}{\|\hat{\mathbf{n}}_p\|_2 \|\mathbf{n}_p\|_2} \right),$$

$$p^* = \arg \min_{p' \in P^*} \|p - p'\|_1$$

(29)

$$Com = \sum_{p \in P^o} \left( 1 - \frac{\hat{\mathbf{n}}_p \cdot \mathbf{n}_p}{\|\hat{\mathbf{n}}_p\|_2 \|\mathbf{n}_p\|_2} \right),$$

$$p = \arg \min_{p' \in P} \|p - p'\|_1$$

(30)
Fig. 11. Visualization of the reconstruction results compared with baselines. Benefiting from the polarization cues, our method can recover more details of transparent objects than other methods.

\[ CDN = \frac{Acc + Com}{2} \]  

(31)

where \( P \) and \( P^* \) are the sets of points sampled from the reconstructed and the ground-truth shape. \( \hat{n}_p \) and \( n_{p^*} \) represent the normal vector at point \( p \) and \( p^* \), respectively.

A. Comparisons With Baselines

In this paper, we adopt PMVIR [20], IDR [11], visual hull (VH), and TransShape [13] as our baselines. PMVIR is a polarimetric inverse rendering method designed for opaque objects reconstruction. IDR and VH are general 3D reconstruction methods, while TransShape is one of the state-of-the-art methods for transparent object reconstruction.

**PMVIR:** PMVIR (polarimetric multi-view inverse rendering) is a polarimetric inverse rendering method that aims to achieve high-quality reconstruction results using polarimetric cues. PMVIR utilizes structure-from-motion and multi-view stereo to get the initial model for optimization, which these techniques will fail in transparent objects. Therefore, we adopt the initial model obtained from our framework, and we use the PMVIR’s loss function to guide the shape optimization. All the hyper-parameters in PMVIR’s loss function are the same as PMVIR’s reported.

**Visual Hull (VH):** Visual hull or space carving is a traditional algorithm for 3D reconstruction and is usually used as the initial shape reconstruction. We utilize the code from Li et al. [13] to compute the visual hull. Since our camera poses are already normalized, we limit the visual hull to \([-1, 1]^3\) and set the spatial resolution to 256.

**IDR:** IDR is one of the state-of-the-art methods for inverse rendering reconstruction using MLP-based implicit representation. The renderer in IDR will diverge when applied to transparent objects since it is designed for opaque objects. Hence, we remove the RGB loss term in IDR, only silhouette loss and regularization loss are used. All the other super parameters are the same as in the original IDR.

**TransShape:** TransShape is the state-of-the-art method for transparent objects from multi-view RGB images. Different from our approach, TransShape samples rays from a known fixed environment map. Hence, we re-implemented the light sampling module in TransShape to adapt our dataset.

Fig. 11 is the visualization of the comparisons and it shows that due to the limitation of resolution, the visual hull can only reconstruct rough outline of the object, and its reconstructed surfaces are rough and lack details. TransShape further optimizes the model based on the visual hull, TransShape’s results are smooth but due to the insufficient information from RGB images, it fails to recover objects’ details. IDR exhibits the advantage of using MLP-based implicit representation, which can produce water-tight surfaces. However, the reconstruction results of IDR also lack details about objects with only silhouette supervision. PMVIR utilizes the additional information from polarimetric cues, while the loss function which works well on opaque objects can not apply to transparent objects since PMVIR assumes that changes in the polarization state are solely due to surface interactions, which is not applicable to transparent objects. Directly using interfered polarization information is even worse than not using polarization information. Benefiting from polarimetric cues and our adaptation to transparent shapes, our
The best results listed in the table are marked as bold.

### TABLE I

| Method   | CAT  | FROG | ELEPHANT | SQUIRREL |
|----------|------|------|----------|----------|
|          | CD   | CDN  | CD       | CDN      | CD       | CDN      | CD       | CDN      |
| PMVIR    | 13.5625 | 2375.44 | 53.93   | 3035.79  | 20.28    | 3326.02  | 31.9     | 4078.58  |
| VH       | 18.96 | 3143.56 | 34.06   | 2752.86  | 24.26    | 3810.97  | 14.76    | 3081.47  |
| IDR      | 9.82  | 978.79 | 18.99   | 1152.16  | 12.52    | 1579.04  | 13.94    | 1691.07  |
| TransShape | 13.32 | 1014.11 | 33.59   | 882.59   | 18.81    | 1815.98  | 14.63    | 1865.12  |
| Ours     | 8.97  | 744.05 | 13.35   | 1088.37  | 11.31    | 1510.27  | 9.55     | 1476.51  |

### B. Ablation Studies

We conduct ablation experiments on the important parts of our method, including loss terms $L_{pol}$, $L_{sdf}$, and the reflection percentage $w$. In the ablation studies, only the weight of the studied module is set to zero, other parameters are kept the same. The object CAT is selected for ablation studies. The results of other objects in our dataset are provided in the supplementary material. The ablation results of $L_{pol}$ and $L_{sdf}$ have been shown in Fig. 11 and Table I in the previous subsection, i.e., the comparison of ours and IDR, the loss term $L_{pol}$ improves the reconstruction quality by supplementing information of shape’s details. Fig. 12 shows the results of the ablation study on the SDF loss term $L_{sdf}$. After removing $L_{sdf}$, obvious contour lines and hollows appear on the reconstructed surface. $L_{sdf}$ constraints surface normal of the shape that is implicitly represented by an MLP to approach the unit vector, which ensures the reconstructed shape is smooth and realistic. Therefore, when the SDF loss term omits, the MLP will have large gradients at some areas, resulting in holes in the reconstructed shape.

Fig. 13 presents the results with and without reflection percentage $w$ and it shows that polarimetric cues will misguide the shape reconstruction, especially the folded areas. The high transmission component proportion in these areas leads to the coupling of the observed polarization state with all the interaction points in the transmitted light path. Calculating the loss directly with the rendered polarization images will lead to an erroneous shape. The reflection percentage to weight polarization loss can effectively reduce the error. Intuitively, the reflection percentage decreases the utilization of polarimetric cues, but no significant degradation of reconstruction details is observed in the experiment. This can be attributed to the redundancy of polarization information across multi-views, where areas experiencing severe transmission interference in one view exhibit higher reliability in other views.

### TABLE II

| Metric | W/o $L_{pol}$ | W/o $L_{sdf}$ | W/o $w$ | Full |
|--------|---------------|---------------|---------|------|
| CD     | 9.82          | 26.81         | 9.51    | 8.97 |
| CDN    | 978.79        | 1788.01       | 1017.79 | 744.05 |

The best results listed in the table are marked as bold.

### TABLE III

| Method | 10 Views | 20 Views | 34 Views |
|--------|----------|----------|----------|
|        | CD       | CDN      | CD       | CDN      |
| IDR    | 11.75    | 1259.79  | 11.61    | 1371.06  | 9.82    | 978.79  |
| Ours   | 9.89     | 1009.11  | 9.29     | 934.67   | 8.97    | 744.05  |

The best results listed in the table are marked as bold.
Fig. 14. Reconstruction results under the different number of views. The shapes listed in the two rows are the results without/with polarimetric cues, respectively.

10 views. This is because the new images in 20 views contain more head and back areas, resulting in more silhouette loss and over-optimization in these areas.

V. DISCUSSION

As studied by prior works, polarimetric cues indeed can effectively complement the details of reconstruction results. However, due to existence of the severe transmission interference, it is crucial to apply appropriate filtering techniques to ensure correct polarization information is used. In this paper, with the goal of introducing polarimetric cues to transparent shapes reconstruction, we propose a polarimetric inverse rendering framework for detailed transparent shapes reconstruction from multi-view polarization images. Specifically, we employ the implicit neural representation for object’s geometry, then it is rendered by the polarimetric render and compared to the real-world captured polarization images. To make the polarimetric cues technically feasible on transparent shapes reconstruction, we propose the reflection percentage to calculate the weighed polarization loss. In addition, we construct a polarization dataset for multi-view transparent shapes reconstruction and our method is verified on this dataset. Experimental results show that our method is capable of recovering detailed shapes and improving reconstruction quality of transparent objects, and our method provides a technically feasible solution for the application of polarization information in transparent objects.

Limitations and future work: Only the polarimetric cues of the areas with high reflection percentage are effectively utilized. Hence, using the polarization ray tracing technique to render more realistic polarization images of transparent objects will be our future work. In addition, the quality of our method heavily depends on the quality of initial shape, how to reconstruct outperforming shapes based on poor initial shapes is also one of our future directions.

REFERENCES

[1] M. Hendrikx, S. Meijer, J. Van Der Velden, and A. Iosup, “Procedural content generation for games: A survey,” ACM Trans. Multimedia Comput. Commun. Appl., vol. 9, no. 1, pp. 1–22, 2013.

[2] A. Kargas, G. Loumos, and D. Varoutas, “Using different ways of 3D reconstruction of historical cities for gaming purposes: The case study of napflio,” Heritage, vol. 2, no. 3, pp. 1799–1811, 2019.

[3] F. Bruno et al., “From 3D reconstruction to virtual reality: A complete methodology for digital archaeological exhibition,” J. Cultural Heritage, vol. 11, no. 1, pp. 42–49, 2010.

[4] K. Bahirat, C. Lai, R. P. Mcmahon, and B. Prabhakaran, “Designing and evaluating a mesh simplification algorithm for virtual reality,” ACM Trans. Multimedia Comput. Commun. Appl., vol. 14, no. 3s, pp. 1–26, 2018.

[5] J. Lou et al., “Realistic facial expression reconstruction for VR HMD users,” IEEE Trans. Multimedia, vol. 22, pp. 730–743, 2019.

[6] J. Hafez, S. Lee, S. Kwon, and A. Hanacher, “Image based 3D reconstruction of texture-less objects for VR contents,” Int. J. Adv. Smart Convergence, vol. 6, no. 1, pp. 9–17, 2017.

[7] S.-Y. Chen, Y.-K. Lai, S. Xia, P. L. Rosin, and L. Gao, “3D face reconstruction and gaze tracking in the HMD for virtual interaction,” IEEE Trans. Multimedia, vol. 25, pp. 3166–3179, 2023.

[8] C. Yan et al., “3D room layout estimation from a single RGB image,” IEEE Trans. Multimedia, vol. 22, no. 11, pp. 3014–3024, Nov, 2020.

[9] M. Michalkiewicz, J. K. Pontes, D. Jack, M. Baktashmotlagh, and A. Eriksson, “Implicit surface representations as layers in neural networks,” in Proc. IEEE/CVF Int. Conf. Comput. Vis., 2019, pp. 4743–4752.

[10] M. Oechsle, S. Peng, and A. Geiger, “UNISURF: Unifying neural implicit surfaces and radiance fields for multi-view reconstruction,” in in Proc. IEEE/CVF Int. Conf. Comput. Vis., 2021, pp. 5589–5599.

[11] L. Yariv et al., “Multiview neural surface reconstruction by disentangling geometry and appearance,” in Proc. Adv. Neural Inf. Process. Syst., 2020, pp. 2492–2502.

[12] L. Yariv, J. Gu, Y. Kasten, and Y. Lipman, “Volume rendering of neural implicit surfaces,” in Proc. Adv. Neural Inf. Process. Syst., 2021, pp. 4805–4815.

[13] Z. Li, Y.-Y. Yeh, and M. Chandraker, “Through the looking glass: Neural 3D reconstruction of transparent shapes,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2020, pp. 1262–1271.

[14] K. N. Kutulakos and E. Stiegler, “A theory of refractive and specular 3D shape by light-path triangulation,” Int. J. Comput. Vis., vol. 76, no. 1, pp. 13–29, 2008.

[15] C.-Y. Tsai, A. Veeraraghavan, and A. C. Sankaranarayanan, “What does a single light-ray reveal about a transparent object?,” in Proc. IEEE Int. Conf. Image Process., 2015, pp. 606–610.

[16] Y. Qian, M. Gong, and Y. H. Yang, “3D reconstruction of transparent objects with position-normal consistency,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2016, pp. 4369–4377.

[17] B. Wu, Y. Zhou, Y. Qian, M. Cong, and H. Huang, “Full 3D reconstruction of transparent objects,” ACM Trans. Graph., vol. 37, no. 4, pp. 1–11, 2018.

[18] J. Lyu, B. Wu, D. Lischinski, D. Cohen-Or, and H. Huang, “Differentiable refraction-tracing for mesh reconstruction of transparent objects,” ACM Trans. Graph., vol. 39, no. 6, pp. 1–13, 2020.

[19] Y. Ba et al., “Deep shape from polarization,” in Proc. Euro. Conf. Comput. Vis., 2020, pp. 554–571.

[20] J. Zhao, Y. Monno, and M. Okutomi, “Polarimetric multi-view inverse rendering,” in Proc. Eur. Conf. Comput. Vis., 2020, pp. 85–102.

[21] V. Deschaintre, Y. Lin, and A. Ghosh, “Deep polarization imaging for 3D shape and SVBRDF acquisition,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2021, pp. 15567–15576.

[22] S. Sajjan et al., “Clear Grasp: 3D shape estimation of transparent objects for manipulation,” in Proc. IEEE Int. Conf. Robot. Automat., 2020, pp. 3634–3642.

[23] H. Fang, H.-S. Fang, S. Xu, and C. Lu, “TransCG: A large-scale real-world dataset for transparent object depth completion and a grasping baseline,” IEEE Robot. Automat. Lett., vol. 7, no. 3, pp. 7383–7390, Jul. 2022.

[24] J. Ichnowsky, Y. Avigal, J. Kerr, and K. Goldberg, “Dex-NeRF: Using a neural radiance field to grasp transparent objects,” in Proc. 5th Conf. Robot. Learn., Nov. 8–11, 2022, pp. 526–536.

[25] J. Xu, Z. Zhu, H. Bao, and W. Xu, “Hybrid mesh-neural representation for 3D transparent object reconstruction,” 2022, arXiv:2203.12613.

[26] D. Miyazaki, M. Kagesawa, and K. Ikeuchi, “Transparent surface modeling from a pair of polarization images,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 26, no. 1, pp. 73–82, Jan. 2004.

[27] S. Mingqi, X. Chongkun, Y. Zhendong, H. Junnan, and W. Xueqian, “Transparent shape from a single view polarization image,” IEEE/CVF Int. Conf. Comput. Vis., 2023, pp. 9277–9286.

[28] G. A. Atkinson and E. R. Hancock, “Recovery of surface orientation from diffuse polarization,” IEEE Trans. Image Process., vol. 15, no. 6, pp. 1653–1664, Jun. 2006.

[29] A. Tewari et al., “Advances in neural rendering,” Comput. Graph. Forum, vol. 41, no. 2, pp. 703–735, 2022.
[30] Y.-C. Guo, D. Kang, L. Bao, Y. He, and S.-H. Zhang, “NeRFReN: Neural radiance fields with reflections,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2022, pp. 18409–18418.

[31] C. Zhu, R. Wan, and B. Shi, “Neural transmitted radiance fields,” in Proc. Adv. Neural Inf. Process. Syst., 2022, pp. 38994–39006.

[32] D. Levy et al., “SeaThru-NeRF: Neural radiance fields in scattering media,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2023, pp. 56–65.

[33] Y. Zhan, S. Nobuhara, K. Nishino, and Y. Zheng, “NeRFrac: Neural radiance fields through refractive surface,” in Proc. IEEE/CVF Int. Conf. Comput. Vis., 2023, pp. 2015–2023.

[34] J. J. Park et al., “DeepSDF: Learning continuous signed distance functions for shape representation,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2019, pp. 165–174.

[35] P. Wang et al., “NeuS: Learning neural implicit surfaces by volume rendering for multi-view reconstruction,” in Proc. Adv. Neural Inf. Process. Syst., M. Ranzato, A. Beygelzimer, Y. Dauphin, P. S. Liang, and J. W. Vaughan, 2021, vol. 34, pp. 27171–27183.

[36] X. Wang, Y. Guo, Z. Yang, and J. Zhang, “Prior-guided multi-view 3D head reconstruction,” IEEE Trans. Multimedia, vol. 24, pp. 4028–4040, 2022.

[37] D. Miyazaki and K. Ikeuchi, “Inverse polarization raytracing: Estimating surface shapes of transparent objects,” in Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., 2005, pp. 910–917.

[38] C. Lei et al., “Polarization-aided neural decomposition of radiance,” in Proc. 17th Eur. Conf. Comput. Vis., Tel Aviv, Israel, Springer, 2022, pp. 538–556.

[39] A. Paszke et al., “PyTorch: An imperative style, high-performance deep learning library,” in Proc. Adv. Neural Inf. Process. Syst., 2019, Art. no. 721.

[40] D. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in Proc. Int. Conf. Learn. Representations, 2015, vol. 5.