Through the Looking Glass: Neural 3D Reconstruction of Transparent Shapes

Zhengqin Li* Yu-Ying Yeh* Manmohan Chandraker
University of California, San Diego
{zhl378, yuyeh, mkchandraker}@eng.ucsd.edu

Abstract

Recovering the 3D shape of transparent objects using a small number of unconstrained natural images is an ill-posed problem. Complex light paths induced by refraction and reflection have prevented both traditional and deep multiview stereo from solving this challenge. We propose a physically-based network to recover 3D shape of transparent objects using a few images acquired with a mobile phone camera, under a known but arbitrary environment map. Our novel contributions include a normal representation that enables the network to model complex light transport through local computation, a rendering layer that models refractions and reflections, a cost volume specifically designed for normal refinement of transparent shapes and a feature mapping based on predicted normals for 3D point cloud reconstruction. We render a synthetic dataset to encourage the model to learn refractive light transport across different views. Our experiments show successful recovery of high-quality 3D geometry for complex transparent shapes using as few as 5-12 natural images. Code and data will be publicly released.

1. Introduction

Transparent objects abound in real-world environments, thus, their reconstruction from images has several applications such as 3D modeling and augmented reality. However, their visual appearance is far more complex than that of opaque objects, due to complex light paths with both refractions and reflections. This makes image-based reconstruction of transparent objects extremely ill-posed, since only highly convoluted intensities of an environment map are observed. In this paper, we propose that data-driven priors learned by a deep network that models the physical basis of image formation can solve the problem of transparent shape reconstruction using a few natural images acquired with a commodity mobile phone camera.

While physically-based networks have been proposed to solve inverse problems for opaque objects [25], the complexity of light paths is higher for transparent shapes and small changes in shape can manifest as severely non-local changes in appearance. However, the physical basis of image formation for transparent objects is well-known – refraction at the interface is governed by Snell's law, the relative fraction of reflection is determined by Fresnel's equations and total internal reflection occurs when the angle of incidence at the interface to a medium with lower refractive index is below critical angle. These properties have been used to delineate theoretical conditions on reconstruction of transparent shapes [23], as well as acquire high-quality shapes under controlled settings [46, 52]. In contrast, we propose to leverage this knowledge of image formation within a deep network to reconstruct transparent shapes using relatively unconstrained images under arbitrary environment maps.

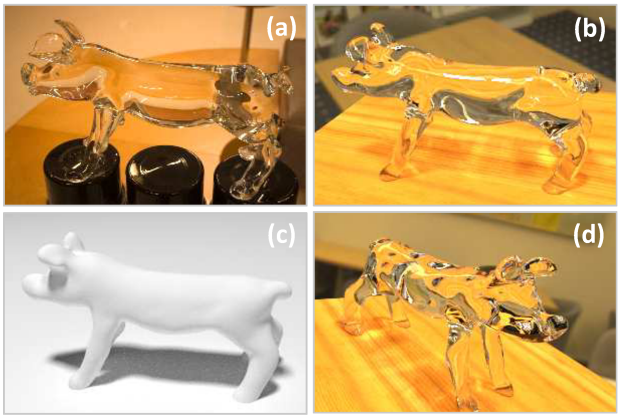

Figure 1. We present a novel physically-based deep network for image-based reconstruction of transparent objects with a small number of views. (a) An input photograph of a real transparent object captured under unconstrained conditions (1 of 10 images). (b) and (c): The reconstructed shape rendered under the same view with transparent and white diffuse material. (d) The reconstructed shape rendered under a novel view and environment map.

Specifically, we use a small number of views of a glass object with known refractive index, observed under a known but arbitrary environment map, using a mobile phone camera. Note that this is a significantly less restricted setting compared to most prior works that require dark room environments, projector-camera setups or controlled acquisition of a large number of images. Starting with a visual hull construction, we propose a novel in-network differentiable rendering layer that models refractive light paths up to two bounces to refine surface normals corresponding to a back-projected ray at both the front and back of the object, along with a mask to identify regions where total internal reflection

*These two authors contributed equally

Figure 2. Reconstruction using 10 images of synthetic *kitten* model. The left image is rendered with the reconstructed shape while the right image is rendered with the ground-truth shape.

occurs. Next, we propose a novel cost volume to further leverage correspondence between the input image and environment map, but with special considerations since the two sets of normal maps span a four-dimensional space, which makes conventional cost volumes from multiview stereo intractable. Using our differentiable rendering layer, we perform a novel optimization in latent space to regularize our reconstructed normals to be consistent with the manifold of natural shapes. To reconstruct the full 3D shape, we use PointNet++ [32] with novel mechanisms to map normal features to a consistent 3D space, new loss functions for training and architectural changes that exploit surface normals for better recovery of 3D shape.

Since acquisition of transparent shapes is a laborious process, it is extremely difficult to obtain large-scale training data with ground truth [40]. Thus, we render a synthetic dataset, using a custom GPU-accelerated ray tracer. To avoid category-specific priors, we render images of random shapes under a wide variety of natural environment maps. On both synthetic and real data, the benefits of our physically-based network design are clearly observed. Indeed, we posit that such physical modeling eases the learning for a challenging problem and improves generalization to real images. Figures 1 and 2 show example outputs on real and synthetic data. All code and data will be publicly released.

To summarize, we propose the following contributions that solve the problem of transparent shape reconstruction with a limited number of unconstrained views:

- A physically-based network for surface normal reconstruction with a novel differentiable rendering layer and cost volume that imbibe insights from image formation.
- A physically-based 3D point cloud reconstruction that leverages the above surface normals and rendering layer.
- Strong experimental demonstration using a photorealistically rendered large-scale dataset for training and a small number of mobile phone photographs for evaluation.

2. Related Work

Multiview stereo Traditional approaches [37] and deep networks [49] for multiview stereo have achieved impressive results. A full review is out of our scope, but we note that they assume photoconsistency for opaque objects and cannot handle complex light paths of transparent shapes.

Theoretical studies In seminal work, Kutulakos and Steger [23] characterize the extent to which shape may be recovered given the number of bounces in refractive (and specular) light paths. Chari and Sturm [5] further constrain the system of equations using radiometric cues. Other works study motion cues [3, 29] or parametric priors [43]. We derive inspiration from such works to incorporate physical properties of image formation, by accounting for refractions, reflections and total internal reflections in our network design.

Controlled acquisition Special setups have been used in prior work, such as light field probes [44], polarimetry [9, 15, 28], transmission imaging [20], scatter-trace photography [30], time-of-flight imaging [41] or tomography [42]. An external liquid medium [14] or moving spotlights in video [50] have been used too. Wu et al. [46] also start from a visual hull like us, to estimate normals and depths from multiple views acquired using a turntable-based setup with two cameras that image projected stripe patterns in a controlled environment. A projector-camera setup is also used by [35]. In contrast to all of the above works, we only require unconstrained natural images, even obtainable with a mobile phone camera, to reconstruct transparent shapes.

Environment matting Environment matting uses a projector-camera setup to capture a composable map [56, 8]. Subsequent works have extended to mutliple cameras [27], natural images [45], frequency [55] or wavelet domains [31], with user-assistance [52] or compressive sensing to reduce the number of images [10, 33]. In contrast, we use a small number of unconstrained images acquired with a mobile phone camera in arbitrary scenes, to produce full 3D shape.

Reconstruction from natural images Stets et al. [39] propose a black-box network to reconstruct depth and normals from a single image. Shan et al. [38] recover height fields in controlled settings, while Yeung et al. [51] have user inputs to recover normals. In contrast, we recover high-quality full 3D shapes and normals using only a few images of transparent objects, by modeling the physical basis of image formation in a deep network.

Refractive materials besides glass Polarization [7], differentiable rendering [6] and neural volumes [26] have been used for translucent objects, while specular objects have been considered under similar frameworks as transparent ones [16, 57]. Gas flows [2, 18], flames [17, 47] and fluids [13, 34, 54] have been recovered, often in controlled setups. Our experiments are focused on glass, but similar ideas might be applicable for other refractive media too.

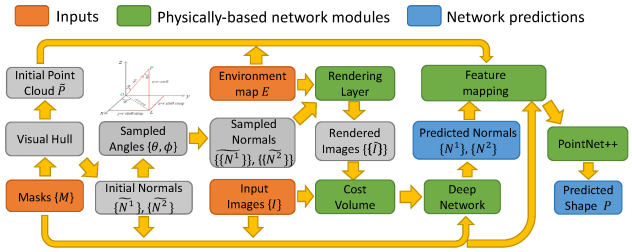

Figure 3. Our framework for transparent shape reconstruction.

3. Method

Setup and assumptions Our inputs are V images $\{I_v\}_{v=1}^V$ of a transparent object with known refractive index (IoR), along with segmentation masks $\{M_v\}_{v=1}^V$. We assume a known and distant, but otherwise arbitrary, environment map E. The output is a point cloud reconstruction $\mathcal{P}$ of the transparent shape. Note that our model is different from (3-2-2) triangulation [22] that requires two reference points on each ray for reconstruction, leading to a significant relaxation over prior works [46, 52] that need active lighting, carefully calibrated devices and controlled environments. We tackle this severely ill-posed problem through a novel physically-based network that models the image formation in transparent objects over three sub-tasks: shape initialization, cost volume for normal estimation and shape reconstruction.

To simplify the problem and due to GPU memory limits, we consider light paths with only up to two bounces, that is, either the light ray gets reflected by the object once before hitting the environment map or it gets refracted by it twice before hitting the environment map. This is not a severe limitation – more complex regions stemming from total internal reflection or light paths with more than two bounces are masked out in one view, but potentially estimated in other views. The overall framework is summarized in Figure 3.

Shape initialization We initialize the transparent shape with a visual hull [21]. While a visual hull method cannot reconstruct some concave or self-occluded regions, it suffices as initialization for our network. We build a 3D volume of size 128^3 and project segmentation masks from V views to it. Then we use marching cubes to reconstruct the hull and loop L3 subdivision to obtain smooth surfaces.

3.1. Normal Reconstruction

A visual hull reconstruction from limited views might be inaccurate, besides missed concavities. We propose to reconstruct high quality normals by estimating correspondences between the input image and the environment map. This is a very difficult problem, since different configurations of transparent shapes may lead to the same appearance. Moreover, small perturbations of normal directions can cause pixel intensities to be completely different. Thus, strong shape priors are necessary for a high quality reconstruction, which we propose to learn with a physically-inspired deep network.

Basic network Our basic network architecture for normal estimation is shown in Figure 4. The basic network structure consists of one encoder and one decoder. The outputs of our network are two normal maps N^1 and N^2, which are the normals at the first and second hit points P^1 and P^2 for a ray backprojected from camera passing through the transparent shape, as illustrated in Figure 5(a). The benefit of modeling the estimation through N^1 and N^2 is that we can easily use a network to represent complex light transport effects without resorting to ray-tracing, which is time-consuming and difficult to treat differentiably. In other words, given N^1 and N^2, we can directly compute outgoing ray directions after passage through the transparent object. The inputs to our network are the image I, the image with background masked out $I \odot M$ and the $\tilde{N}^1$ and $\tilde{N}^2$ of the visual hull (computed off-line by ray tracing). We also compute $\hat{N}^1$ and $\hat{N}^2$ of the ground-truth shape for supervision. The definition of $\tilde{N}^1$, $\tilde{N}^2$ and $\hat{N}^1$, $\hat{N}^2$ are visualized in Figure 5(b). The basic network estimates:

$$N^1, N^2 = \mathbf{NNet}(I, I \odot M, \tilde{N}^1, \tilde{N}^2) \tag{1}$$

The loss function is simply the L_2 loss for N^1 and N^2.

$$\mathcal{L}_N = ||N^1 - \hat{N}^1||_2^2 + ||N^2 - \hat{N}^2||_2^2 \tag{2}$$

Rendering layer Given the environment map E, we can easily compute the incoming radiance through direction l using bilinear sampling. This allows us to build a differentiable rendering layer to model the image formation process of refraction and reflection through simple local computation. As illustrated in Figure 5(a), for every pixel in the image, the incident ray direction l^i through that pixel can be obtained by camera calibration. The reflected and refracted rays l^r and l^t can be computed using N^1 and N^2, following Snell's law. Our rendering layer implements the full physics of an intersection, including the intensity changes caused by the Fresnel term $\mathcal{F}$ of the refractive material. More formally, with some abuse of notation, let L^i, L^r and L^t be the radiance of incoming, reflected and refracted rays. We have

$$\mathcal{F} = \frac{1}{2}\left(\frac{l^i \cdot N - \eta l^t \cdot N}{l^i \cdot N + \eta l^t \cdot N}\right)^2 + \frac{1}{2}\left(\frac{\eta l^i \cdot N - l^t \cdot N}{\eta l^i \cdot N + l^t \cdot N}\right)^2.$$
$$L^r = \mathcal{F} \cdot L^i, \quad L^t = (1 - \mathcal{F}) \cdot L^i$$

Due to total internal reflection, some rays entering the object may not be able to hit the environment map after one more bounce, for which our rendering layer returns a binary mask, M^{tr}. With I^r and I^t representing radiance along the directions l^r and l^t, the rendering layer models the image formation process for transparent shapes through reflection, refraction and total internal reflection:

$$I^r, I^t, M^{tr} = \mathbf{RenderLayer}(E, N^1, N^2). \tag{3}$$

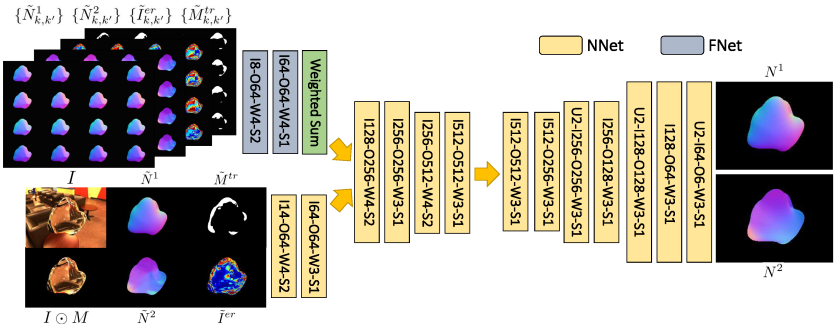

Figure 4. The network architecture for normal reconstruction. Yellow blocks represent **NNet** and blue blocks represent **FNet**. IX_1-OX_2-WX_3-SX_4 represents a convolutional layer with input channel X_1, output channel X_2, kernel size X_3 and stride X_4. UX_5 represents bilinear upsampling layer with scale factor X_5.

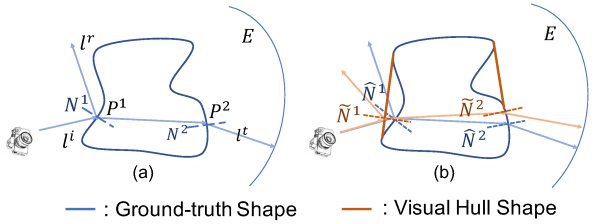

Figure 5. (a) Illustration of the first and second normal (N^1 and N^2), the first and second hit points (P^1 and P^2), and the reflection and refraction modeled by our deep network. (b) Illustration of visual hull ($\tilde{N}^1$, $\tilde{N}^2$) and ground-truth normals ($\hat{N}^1$, $\hat{N}^2$).

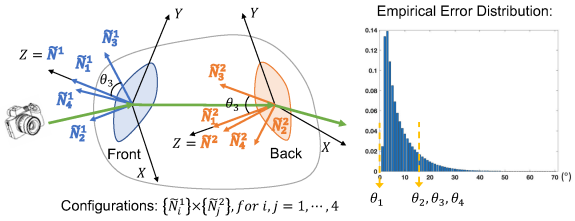

Figure 6. We build an efficient cost volume by sampling directions around visual hull normals according to their error distributions.

Our in-network rendering layer is differentiable and end-to-end trainable. But instead of just using the rendering loss as an extra supervision, we compute an error map based on rendering with the visual hull normals:

$$\tilde{I}^r, \tilde{I}^t, \tilde{M}^{tr} = \mathbf{RenderLayer}(E, \tilde{N}^1, \tilde{N}^2), \quad (4)$$

$$\tilde{I}^{er} = |I - (\tilde{I}^r + \tilde{I}^t)| \odot M. \quad (5)$$

This error map is used as an additional input to our normal reconstruction network, to help it better learn regions where the visual hull normals $\tilde{N}^1$ and $\tilde{N}^2$ may not be accurate:

$$N^1, N^2 = \mathbf{NNet}(I, I \odot M, \tilde{N}^1, \tilde{N}^2, \tilde{I}^{er}, \tilde{M}^{tr}) \quad (6)$$

Cost volume We now propose a cost volume to leverage the correspondence between the environment map and the input image. While cost volumes in deep networks have led to great success for multiview depth reconstruction of opaque objects, extension to normal reconstruction for transparent objects is non-trivial. The brute-force approach would be to uniformly sample the 4-dimensional hemisphere of $N^1 \times N^2$, then compute the error map for each sampled normal. However, this will lead to much higher GPU memory consumption compared to depth reconstruction due to higher dimensionality of the sampled space. To limit memory consumption, we sample N^1 and N^2 in smaller regions around the initial visual hull normals $\tilde{N}^1$ and $\tilde{N}^2$, as shown in Figure 6. Formally, let U be the up vector in bottom-to-top direction of the image plane. We first build a local coordinate system with respect to $\tilde{N}^1$ and $\tilde{N}^2$:

$$Z = \tilde{N}^i, \; Y = U - (U^T \cdot \tilde{N}^i)\tilde{N}^i, \; X = \texttt{cross}(Y, Z), \quad (7)$$

where Y is normalized and $i = 1, 2$. Let $\{\theta_k\}_{k=1}^K$, $\{\phi_k\}_{k=1}^K$ be the sampled angles. Then, the sampled normals are:

$$\tilde{N}_k^i = X \cos\phi_k \sin\theta_k + Y \sin\phi_k \sin\theta_k + Z \cos\theta_k. \quad (8)$$

We sample the angles $\{\theta_k\}_{k=1}^K$, $\{\phi_k\}_{k=1}^K$ according to the error distribution of visual hull normals. The angles and distributions are shown in the supplementary material. Since we reconstruct N^1 and N^2 simultaneously, the total number of configurations of sampled normals is $K \times K$. Directly using the K^2 sampled normals to build a cost volume is too expensive, so we use a learnable pooling layer to aggregate the features from each sampled normal configuration in an early stage. For each pair of $\tilde{N}_k^1$ and $\tilde{N}_{k'}^2$, we compute their total reflection mask $\tilde{M}_{k,k'}^{tr}$ and error map $\tilde{I}_{k,k'}^{er}$ using (4) and (5), then perform a feature extraction:

$$F(k, k') = \mathbf{FNet}(\tilde{N}_k^1, \tilde{N}_{k'}^2, \tilde{I}_{k,k'}^{er}, \tilde{M}_{k,k'}^{tr}). \quad (9)$$

We then compute the weighted sum of feature vectors $F(k, k')$ and concatenate them with the feature extracted from the encoder of **NNet** for normal reconstruction:

$$F = \sum_k^K \sum_{k'}^K \omega(k, k') F(k, k'), \quad (10)$$

where $\omega(k, k')$ are positive coefficients with sum equal to 1, that are also learned during the training process. The detailed network structure is shown in Figure 4.

Post processing The network above already yields reasonable normal reconstruction. It can be further improved by optimizing the latent vector from the encoder to minimize the rendering error using the predicted normal N^1 and N^2:

$$\mathcal{L}_N^{\text{Opt}} = ||(I - (I^r + I^t)) \odot M^{tr}||_2^2, \quad (11)$$

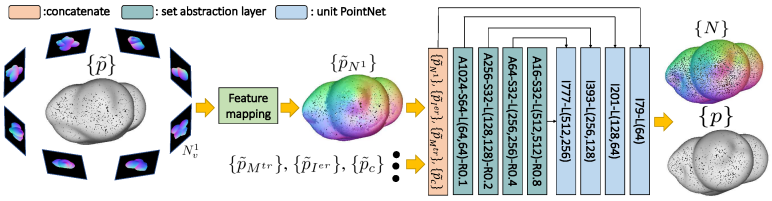

Figure 7. Our method for point cloud reconstruction. AX_1-SX_2-$L(X_3, X_4)$-RX_5 represents a set abstraction layer with X_1 anchor points, X_2 sampled points, 2 fully connected layers with X_3, X_4 feature channels and sampling radius X_5. IY_1-$L(Y_2, Y_3)$ represents a unit PointNet with Y_1 input channels and 2 fully connected layers with Y_2, Y_3 feature channels.

where I_t, I_t, M^{tr} are obtained from the rendering layer (3). For this optimization, we keep the network parameters unchanged and only update the latent vector. Note that directly optimizing the predicted normal N^1 and N^2 without the deep network does not yield comparable improvements. This is due to our decoder acting as a regularization that prevents the reconstructed normal from deviating from the manifold of natural shapes during the optimization. Similar ideas have been used for BRDF reconstruction [11].

3.2. Point Cloud Reconstruction

We now reconstruct the transparent shape based on the predictions of **NNet**, that is, the normals, total reflection mask and rendering error. Our idea is to map the predictions from different views to the visual hull geometry. These predictions are used as input features for a point cloud reconstruction to obtain a full 3D shape. Our point cloud reconstruction pipeline is illustrated in Figure 7.

Feature mapping We propose three options to map predictions from different views to the visual hull geometry. Let $\{\tilde{p}\}$ be the point cloud uniformly sampled from visual hull surfaces and $\mathcal{S}_v(\tilde{p}, h)$ be a function that projects the 3D point $\tilde{p}$ to the 2D image plane of view v and then fetches the value of a function h defined on image coordinates using bilinear sampling. Let $\mathcal{V}_v(\tilde{p})$ be a binary function that verifies if point $\tilde{p}$ can be observed from view v and $\mathcal{T}_v(\tilde{p})$ be a transformation that maps a 3D point or normal direction in view v to world coordinates. Let $\mathcal{C}_v(\tilde{p})$ be the cosine of the angle between the ray passing through $\tilde{p}$ and camera center.

The first option is a feature f that averages observations from different views. For every view v that can see the point $\tilde{p}$, we project its features to the point and compute a mean:

$$\tilde{p}_{N^1} = \frac{\sum_v \mathcal{T}_v(\mathcal{S}_v(\tilde{p}, N_v^1))\mathcal{V}_v(\tilde{p})}{\sum_v \mathcal{V}_v(\tilde{p})}, \quad \tilde{p}_{I^{er}} = \frac{\sum_v \mathcal{S}_v(\tilde{p}, I_v^{er})\mathcal{V}_v(\tilde{p})}{\sum_v \mathcal{V}_v(\tilde{p})},$$
$$\tilde{p}_{M^{tr}} = \frac{\sum_v \mathcal{S}_v(\tilde{p}, M_v^{tr})\mathcal{V}_v(\tilde{p})}{\sum_v \mathcal{V}_v(\tilde{p})}, \quad \tilde{p}_c = \frac{\sum_v \mathcal{C}_v(\tilde{p})\mathcal{V}_v(\tilde{p})}{\sum_v \mathcal{V}_v(\tilde{p})}.$$

We concatenate to get: $f = [\tilde{p}_{N^1}, \tilde{p}_{I^{er}}, \tilde{p}_{M^{tr}}, \tilde{p}_c]$.

Another option is to select a view v^* with potentially the most accurate predictions and compute f using the features from only that view. We consider two view-selection strategies. The first is nearest view selection, in which we simply select v^* with the largest $\mathcal{C}_v(\tilde{p})$. The other is to choose the view with the lowest rendering error and no total reflection, with the algorithm detailed in supplementary material. Note that although we do not directly map N^2 to the visual hull geometry, it is necessary for computing the rendering error and thus, needed for our shape reconstruction.

Point cloud refinement We build a network following PointNet++ [32] to reconstruct the point cloud of the transparent object. The input to the network is the visual hull point cloud $\{\tilde{p}\}$ and the feature vectors $\{f\}$. The outputs are the normals $\{N\}$ and the offset of visual hull points $\{\delta\tilde{p}\}$, with the final vertex position is computed as $p = \tilde{p} + \delta\tilde{p}$:

$$\{\delta\tilde{p}\}, \{N\} = \mathbf{PNet}(\{\tilde{p}\}, \{f\}). \tag{12}$$

We tried three loss functions to train our PointNet++. The first loss function is the nearest L_2 loss $\mathcal{L}_P^{\text{nearest}}$. Let $\hat{p}$ be the nearest point to $\tilde{p}$ on the surface of ground-truth geometry and $\hat{N}$ be its normal. We compute the weighted sum of L_2 distance between our predictions p, N and ground truth:

$$\mathcal{L}_P^{\text{nearest}} = \sum_{\{p\},\{N\}} \lambda_1 ||p - \hat{p}||_2^2 + \lambda_2 ||N - \hat{N}||_2^2. \tag{13}$$

The second loss function is a view-dependent L_2 loss $\mathcal{L}_P^{\text{view}}$. Instead of choosing the nearest point from ground-truth geometry for supervision, we choose the point from the best view v^* by projecting its geometry into world coordinates:

$$\hat{p}_{v^*}, \hat{N}_{v^*} = \begin{cases} \mathcal{T}_{v^*}(\mathcal{S}_{v^*}(\tilde{p}, \hat{P}_{v^*}^1)), \mathcal{T}_{v^*}(\mathcal{S}_{v^*}(\tilde{p}, \hat{N}_{v^*}^1)), & v^* \neq 0 \\ \hat{p}, \hat{N}, & v^* = 0. \end{cases}$$

Then we have

$$\mathcal{L}_P^{\text{view}} = \sum_{\{p\},\{N\}} \lambda_1 ||p - \hat{p}_{v^*}||_2^2 + \lambda_2 ||N - \hat{N}_{v^*}||_2^2. \tag{14}$$

The intuition is that since both the feature and ground-truth geometry are selected from the same view, the network can potentially learn their correlation more easily. The last loss function, $\mathcal{L}_P^{\text{CD}}$, is based on the Chamfer distance. Let $\{q\}$ be the set of points uniformly sampled from the ground-truth geometry, with normals N_q. Let $\mathcal{G}(p, \{q\})$ be a function which finds the nearest point of p in the point set $\{q\}$ and function $\mathcal{G}_n(p, \{q\})$ return the normal of the nearest point. The Chamfer distance loss is defined as

$$\mathcal{L}_P^{\text{CD}} = \sum_{\{p\},\{N\}} \frac{\lambda_1}{2} ||p - \mathcal{G}(p, \{q\})|| + \frac{\lambda_2}{2} ||N - \mathcal{G}_n(p, \{q\})|| +$$
$$\sum_{\{q\},\{N_q\}} \frac{\lambda_1}{2} ||q - \mathcal{G}(q, \{p\})|| + \frac{\lambda_2}{2} ||N_q - \mathcal{G}_n(q, \{p\})||. \tag{15}$$

Figure 8 is a demonstration of the three loss functions. In all our experiments, we set $\lambda_1 = 200$ and $\lambda_2 = 5$.

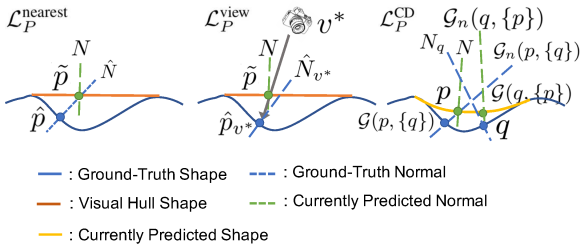

Figure 8. A visualization of the loss functions for point cloud reconstruction. From left to the right are nearest L_2 loss $\mathcal{L}_P^{\text{nearest}}$, view-dependent L_2 loss $\mathcal{L}_P^{view}$ and chamfer distance loss $\mathcal{L}_P^{\text{CD}}$.

Our network, shown in Figure 7, makes several improvements over standard PointNet++. First, we replace max-pooling with average-pooling to favor smooth results. Second, we concatenate normals $\{N\}$ to all skip connections to learn details. Third, we augment the input feature of set abstraction layer with the difference of normal directions between the current and center points. Section 4 and supplementary material show the impact of our design choices.

4. Experiments

Dataset We procedurally generate random scenes following [25, 48] rather than use shape repositories [4], to let the model be category-independent. To remove inner structures caused by shape intersections and prevent false refractions, we render 75 depth maps and use PSR [19] to fuse them into a mesh, with L3 loop subdivision to smooth the surface. We implement a physically-based GPU renderer using NVIDIA OptiX [1]. With 1499 HDR environment maps of [12] for training and 424 for testing, we render 3000 random scenes for training and 600 for testing. The IoR of all shapes is set to 1.4723, to match our real objects.

Implementation Details When building the cost volume for normal reconstruction, we set the number of sampled angles K to be 4. Increasing the number of sampled angles will drastically increase the memory consumption and does not improve the normal accuracy. We sample ϕ uniformly from 0 to 2π and sample θ according to the visual hull normal error. The details are included in the supplementary material. We use Adam optimizer to train all our networks. The initial learning rate is set to be 10^{-4} and we halve the learning rate every 2 epochs. All networks are trained over 10 epochs.

4.1. Ablation Studies on Synthetic Data

Normal reconstruction The quantitative comparisons of 10 views normal reconstruction are summarized in Table 1. We report 5 metrics: the median and mean angles of the first and the second normals (N^1, N^2), and the mean rendering error (I^{er}). We first compare the normal reconstruction of the basic encoder-decoder structure with (`wr`) and without rendering error and total reflection mask as input (`basic`). While both networks greatly improve the normal accuracy compared to visual hull normals (`vh10`), adding rendering error and total reflection mask as inputs can help achieve overall slightly better performances. Next we test the effectiveness of the cost volume (`wr+cv`). Quantitative numbers show that adding cost volume achieves better results, which coincides with our intuition that finding the correspondences between input image and the environment map can help our normal prediction. Finally we optimize the latent vector from the encoder by minimizing the rendering error (`wr+cv+op`). It significantly reduces the rendering error and also improves the normal accuracy. Such improvements cannot be achieved by directly optimizing the normal predictions N^1 and N^2 in the pixel space. Figure 9 presents normal reconstruction results from our synthetic dataset. Our normal reconstruction pipeline obtains results of much higher quality compared with visual hull method. Ablation studies of 5 views and 20 views normal reconstruction and the optimization of latent vector are included in the supplementary material.

	`vh10`	`basic`	`wr`	`wr+cv`	`wr+cv +op`	`wr+cv var. IoR`
N^1 median (°)	5.5	3.5	3.5	3.4	**3.4**	3.6
N^1 mean (°)	7.5	4.9	5.0	4.8	**4.7**	5.0
N^2 median (°)	9.2	6.9	6.8	6.6	**6.2**	7.3
N^2 mean (°)	11.6	8.8	8.7	8.4	**8.1**	9.1
Render Err.(10^{-2})	6.0	4.7	4.6	4.4	**2.9**	5.5

Table 1. Quantitative comparisons of normal estimation from 10 views. `vh10` represents the initial normals reconstructed from 10 views visual hull. `wr` and `basic` are our basic encoder-decoder network with and without rendering error map (I^{er}) and total reflection mask (M^{tr}) as inputs. `wr+cv` represents our network with cost volume. `wr+cv+op` represents the predictions after optimizing the latent vector to minimize the rendering error. `wr+cv var. IoR` represents sensitivity analysis for IoR, explained in text.

	CD(10^{-4})	CDN-mean(°)	CDN-med(°)	Metro(10^{-3})
`vh10`	5.14	7.19	4.90	15.2
RE-$\mathcal{L}_P^{\text{nearest}}$	2.17	6.23	4.50	7.07
RE-$\mathcal{L}_P^{\text{view}}$	2.15	6.51	4.76	6.79
RE-$\mathcal{L}_P^{\text{CD}}$	**2.00**	**6.02**	**4.38**	**5.98**
NE-$\mathcal{L}_P^{\text{CD}}$	2.04	6.10	4.46	6.02
AV-$\mathcal{L}_P^{\text{CD}}$	2.03	6.08	4.46	6.09
RE-$\mathcal{L}_P^{\text{CD}}$, `var. IoR`	2.13	6.24	4.56	6.11
PSR	5.13	6.94	4.75	14.7

Table 2. Quantitative comparisons of point cloud reconstruction from 10 views. `RE`, `NE` and `AV` represent feature mapping methods: rendering error based view selection, nearest view selection and average fusion, respectively. $\mathcal{L}_P^{\text{nearest}}$, $\mathcal{L}_P^{\text{view}}$ and $\mathcal{L}_P^{\text{CD}}$ are the loss functions defined in Sec. 3. RE-$\mathcal{L}_P^{\text{CD}}$, `var. IoR` represents sensitivity analysis for IoR, as described in text. PSR represents optimization [19] to refine the point cloud based on predicted normals.

Point cloud reconstruction Quantitative comparisons of the 10-view point cloud reconstruction network are summa-

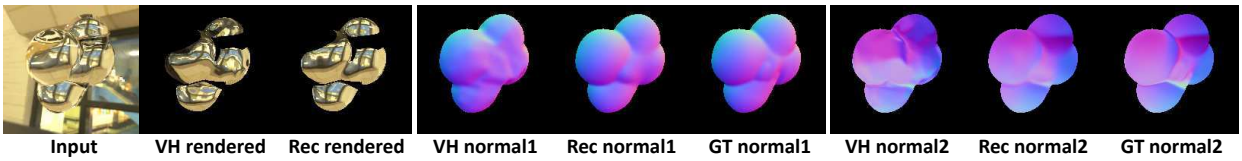

Figure 9. An example of 10 views normal reconstruction from our synthetic dataset. The region of total reflection has been masked out in the rendered images.

Figure 10. Our transparent shape reconstruction results from 5 views, 10 views and 20 views from our synthetic dataset. The images rendered with our reconstructed shapes are much closer to the ground-truth compared with images rendered with the visual hull shapes. The inset normals are rendered from the reconstructed shapes.

	CD(10^{-4})	CDN-mean(°)	CDN-med(°)	Metro(10^{-3})
`vh5`	31.7	13.1	10.3	66.6
`Rec5`	**6.30**	**11.0**	**8.7**	**15.2**
`vh20`	2.23	4.59	**2.71**	6.83
`Rec20`	**1.20**	**4.04**	2.73	**4.18**

Table 3. Quantitative comparisons of point cloud reconstruction from 5 views and 20 views. In both cases, our pipeline significantly improves the transparent shape reconstruction accuracy compared with classical visual hull method.

rized in Table 2. After obtaining the point and normal predictions $\{p\}$ and $\{N\}$, we reconstruct 3D meshes as described above. We compute the Chamfer distance (CD), Chamfer normal median angle (CDN-med), Chamfer normal mean angle (CDN-mean) and Metro distance by uniformly sampling 20000 points on the ground-truth and reconstructed meshes. We first compare the effectiveness of different loss functions. We observe that while all the three loss functions can greatly improve the reconstruction accuracy compared with the initial 10-view visual hull, the Chamfer distance loss (RE-$\mathcal{L}_P^{\text{CD}}$) performs significantly better than view-dependent loss (RE-$\mathcal{L}_P^{\text{view}}$) and nearest L_2 loss (RE-$\mathcal{L}_P^{\text{nearest}}$). Next, we test different feature mapping strategies, where the rendering error based view selection method (RE-$\mathcal{L}_P^{\text{CD}}$) performs consistently better than the other two methods. This is because our rendering error can be used as a meaningful metric to predict normal reconstruction accuracy, which leads to better point cloud reconstruction. Ablation studies for the modified PointNet++ are included in supplementary material.

The last row of Table 2 shows that an optimization-based method like PSR [19] to refine shape from predicted normals does not lead to much improvement, possibly since visual hull shapes are still significantly far from ground truth. In contrast, our network allows large improvements.

Different number of views We also test the entire reconstruction pipeline for 5 and 20 views, with results summarized in Table 3. We use the setting that leads to the best performance for 10 views, that is, `wr` + `cv` + `op` for normal reconstruction and RE-$\mathcal{L}_P^{\text{CD}}$ for point cloud reconstruction, achieving significantly lower errors than the visual hull method. Figure 10 shows an example from the synthetic test set for reconstructions with different number of views. Further results and comparisons are in supplementary material.

Sensitivity analysis for IoR We also evaluate the model on another test set with the same geometries, but unknown IoRs sampled uniformly from the range $[1.3, 1.7]$. As shown in Tables 1 and 2, errors increase slightly but stay reasonable, showing that our model can tolerate inaccurate IoRs to some extent. Detailed analysis is in the supplementary material.

4.2. Results on Real Transparent Objects

We acquire RGB images using a mobile phone. To capture the environment map, we take several images of a mirror sphere at the same location as the transparent shape. We use COLMAP [36] to obtain the camera poses and manually create the segmentation masks.

Normal reconstruction We first demonstrate the normal reconstruction results on real transparent objects in Figure 11. Our model significantly improves visual hull normal quality. The images rendered from our predicted normals are much more similar to the input RGB images compared to those rendered from visual hull normals.

3D shape reconstruction In Figure 12, we demonstrate our 3D shape reconstruction results on real world transparent objects under natural environment map. The dog shape in the first row only takes 5 views and the mouse shape in the second row takes 10 views. We first demonstrate the reconstructed shape from the same view as the input images by rendering them under different lighting and materials. Even with very limited inputs, our reconstructed shapes are still of high quality. To test the generalizability of our predicted shapes, we render them from novel views that have not been used as inputs and the results are still reasonable. Figure 13 compares our reconstruction results with the visual

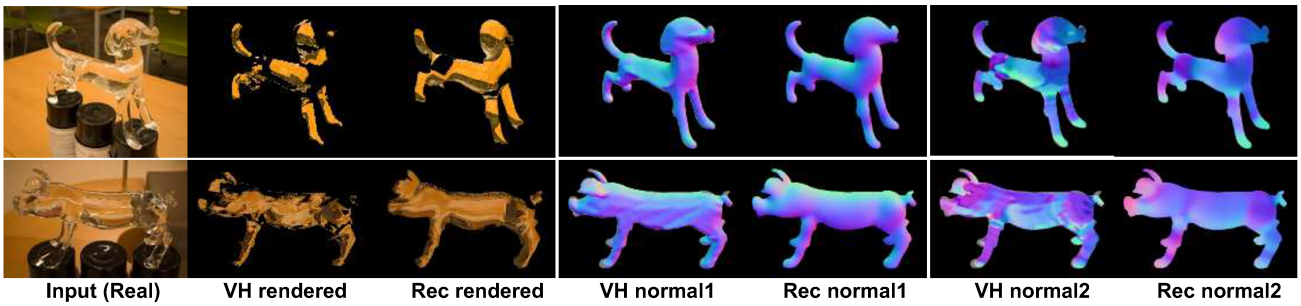

Figure 11. Normal reconstruction of real transparent objects and the rendered images. The initial visual hull normals are built from 10 views. The region of total reflection has been masked out in the rendered images.

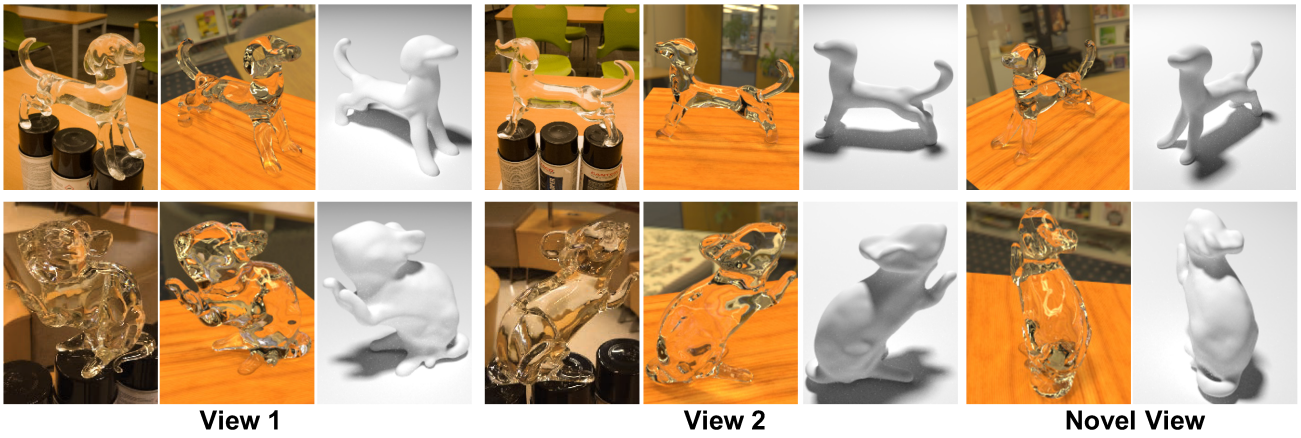

Figure 12. 3D shape reconstruction on real data. Columns 1-6 show reconstruction results from 2 known view directions. For each view, we show the input image and the reconstructed shape rendered from the same view under different lighting and materials. Columns 7-8 render the reconstructed shape from a novel view direction that has not been used to build the visual hull. The first shape is reconstructed using only 5 views (top row) while the second uses 10 views (bottom row). Also see comparisons to ground truth scans in supplementary material.

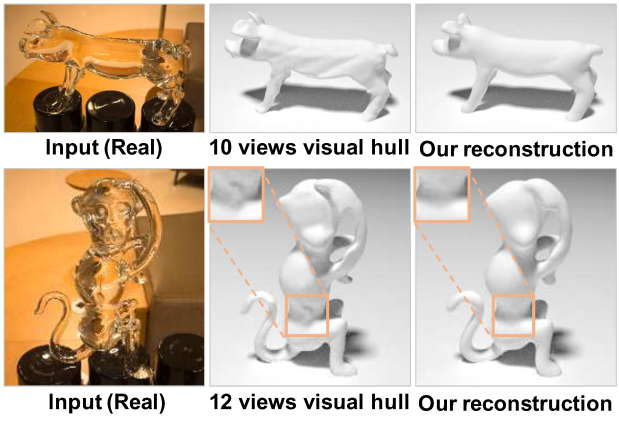

Figure 13. Comparison between visual hull initialization and our shape reconstruction on real objects. Our method recovers more details, especially for concave regions.

hull initialization. We observe that our method performs much better, especially for concave regions. Comparisons with scanned ground truth are in supplementary material.

Runtime Our method requires around 46s to reconstruct a transparent shape from 10 views on a 2080 Ti, compared to 5-6 hours for previous optimization-based methods [46].

5. Discussion

We present the first physically-based deep network to reconstruct transparent shapes from a small number of views captured under arbitrary environment maps. Our network models the properties of refractions and reflections through a physically-based rendering layer and cost volume, to estimate surface normals at both the front and back of the object, which are used to guide a point cloud reconstruction. Extensive experiments on real and synthetic data demonstrate that our method can recover high quality 3D shapes.

Limitations and future work Our limitations suggest interesting future avenues of research. A learnable multiview fusion might replace the visual hull initialization. We believe more complex light paths of length greater than 3 may be handled by differentiable path tracing along the lines of differentiable rendering [24, 53]. While we assume a known refractive index, it may be jointly regressed. Finally, since we reconstruct N^2, future works may also estimate the back surface to achieve single-view 3D reconstruction.

Acknowledgments This work is supported by NSF CAREER Award 1751365 and a Google Research Award. We also thank Adobe and Cognex for generous support.

References

[1] Nvidia OptiX. https://developer.nvidia.com/optix. 6

[2] Bradley Atcheson, Ivo Ihrke, Wolfgang Heidrich, Art Tevs, Derek Bradley, Marcus Magnor, and Hans-Peter Seidel. Time-resolved 3D capture of non-stationary gas flows. *ACM ToG*, 27(5):132:1–132:9, Dec. 2008. 2

[3] Ben-Ezra and Nayar. What does motion reveal about transparency? In *ICCV*, pages 1025–1032 vol.2, 2003. 2

[4] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015. 6

[5] Visesh Chari and Peter Sturm. A theory of refractive photo-light-path triangulation. In *CVPR*, pages 1438–1445, Washington, DC, USA, 2013. IEEE Computer Society. 2

[6] Chengqian Che, Fujun Luan, Shuang Zhao, Kavita Bala, and Ioannis Gkioulekas. Inverse transport networks. *CoRR*, abs/1809.10820, 2018. 2

[7] T. Chen, H. P. A. Lensch, C. Fuchs, and H. Seidel. Polarization and phase-shifting for 3D scanning of translucent objects. In *CVPR*, pages 1–8, June 2007. 2

[8] Yung-Yu Chuang, Douglas E. Zongker, Joel Hindorff, Brian Curless, David H. Salesin, and Richard Szeliski. Environment matting extensions: Towards higher accuracy and real-time capture. In *SIGGRAPH*, pages 121–130, 2000. 2

[9] Z. Cui, J. Gu, B. Shi, P. Tan, and J. Kautz. Polarimetric multi-view stereo. In *CVPR*, pages 369–378, July 2017. 2

[10] Qi Duan, Jianfei Cai, and Jianmin Zheng. Compressive environment matting. *Vis. Comput.*, 31(12):1587–1600, Dec. 2015. 2

[11] Duan Gao, Xiao Li, Yue Dong, Pieter Peers, Kun Xu, and Xin Tong. Deep inverse rendering for high-resolution svbrdf estimation from an arbitrary number of images. *ACM Transactions on Graphics (TOG)*, 38(4):134, 2019. 5

[12] Marc-André Gardner, Kalyan Sunkavalli, Ersin Yumer, Xiaohui Shen, Emiliano Gambaretto, Christian Gagné, and Jean-François Lalonde. Learning to predict indoor illumination from a single image. *arXiv preprint arXiv:1704.00090*, 2017. 6

[13] James Gregson, Michael Krimerman, Matthias B. Hullin, and Wolfgang Heidrich. Stochastic tomography and its applications in 3D imaging of mixing fluids. *ACM ToG*, 31(4):52:1–52:10, July 2012. 2

[14] Kai Han, Kwan-Yee K. Wong, and Miaomiao Liu. Dense reconstruction of transparent objects by altering incident light paths through refraction. *Int. J. Comput. Vision*, 126(5):460–475, May 2018. 2

[15] C. P. Huynh, A. Robles-Kelly, and E. Hancock. Shape and refractive index recovery from single-view polarisation images. In *CVPR*, pages 1229–1236, June 2010. 2

[16] Ivo Ihrke, Kiriakos Kutulakos, Hendrik Lensch, Marcus Magnor, and Wolfgang Heidrich. Transparent and specular object reconstruction. *Comput. Graph. Forum*, 29:2400–2426, 12 2010. 2

[17] Ivo Ihrke and Marcus Magnor. Image-based tomographic reconstruction of flames. In *Proceedings of the 2004 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '04, pages 365–373, Goslar Germany, Germany, 2004. Eurographics Association. 2

[18] Y. Ji, J. Ye, and J. Yu. Reconstructing gas flows using light-path approximation. In *CVPR*, pages 2507–2514, June 2013. 2

[19] Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe. Poisson surface reconstruction. In *Proceedings of the fourth Eurographics symposium on Geometry processing*, volume 7, 2006. 6, 7

[20] J. Kim, I. Reshetouski, and A. Ghosh. Acquiring axially-symmetric transparent objects using single-view transmission imaging. In *CVPR*, pages 1484–1492, July 2017. 2

[21] Kiriakos N. Kutulakos and Steven M. Seitz. A theory of shape by space carving. *International Journal of Computer Vision*, 38:199–218, 2000. 3

[22] K. N. Kutulakos and E. Steger. A theory of refractive and specular 3D shape by light-path triangulation. In *ICCV*, volume 2, pages 1448–1455 Vol. 2, Oct 2005. 3

[23] Kiriakos N. Kutulakos and Eron Steger. A theory of refractive and specular 3D shape by light-path triangulation. *IJCV*, 76(1):13–29, 2008. 1, 2

[24] Tzu-Mao Li, Miika Aittala, Frédo Durand, and Jaakko Lehtinen. Differentiable Monte Carlo ray tracing through edge sampling. *ACM ToG (SIGGRAPH Asia)*, 37(6):222:1 – 222:11, 2018. 8

[25] Zhengqin Li, Zexiang Xu, Ravi Ramamoorthi, Kalyan Sunkavalli, and Manmohan Chandraker. Learning to reconstruct shape and spatially-varying reflectance from a single image. *ACM ToG (SIGGRAPH Asia)*, 37(6):269:1 – 269:11, 2018. 1, 6

[26] Stephen Lombardi, Tomas Simon, Jason Saragih, Gabriel Schwartz, Andreas Lehrmann, and Yaser Sheikh. Neural volumes: Learning dynamic renderable volumes from images. *ACM ToG (SIGGRAPH Asia)*, 38(4):65:1–65:14, 2019. 2

[27] Wojciech Matusik, Hanspeter Pfister, Remo Ziegler, Addy Ngan, and Leonard McMillan. Acquisition and rendering of transparent and refractive objects. In *Eurographics Workshop on Rendering*, EGRW '02, pages 267–278, Aire-la-Ville, Switzerland, Switzerland, 2002. Eurographics Association. 2

[28] D. Miyazaki and K. Ikeuchi. Inverse polarization raytracing: estimating surface shapes of transparent objects. In *CVPR*, volume 2, pages 910–917 vol. 2, June 2005. 2

[29] N. J. W. Morris and K. N. Kutulakos. Dynamic refraction stereo. In *ICCV*, volume 2, pages 1573–1580 Vol. 2, Oct 2005. 2

[30] N. J. W. Morris and K. N. Kutulakos. Reconstructing the surface of inhomogeneous transparent scenes by scatter-trace photography. In *ICCV*, pages 1–8, Oct 2007. 2

[31] Pieter Peers and Philip Dutré. Wavelet environment matting. In *Proceedings of the 14th Eurographics Workshop on Rendering*, EGRW '03, pages 157–166, Aire-la-Ville, Switzerland, Switzerland, 2003. Eurographics Association. 2

[32] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in neural information processing systems*, pages 5099–5108, 2017. 2, 5

[33] Y. Qian, M. Gong, and Y. Yang. Frequency-based environment matting by compressive sensing. In *ICCV*, pages 3532–3540, Dec 2015. 2

[34] Y. Qian, M. Gong, and Y. Yang. Stereo-based 3d reconstruction of dynamic fluid surfaces by global optimization. In *CVPR*, pages 6650–6659, July 2017. 2

[35] Yiming Qian, Minglun Gong, and Yee-Hong Yang. 3d reconstruction of transparent objects with position-normal consistency. In *CVPR*, pages 4369–4377, 06 2016. 2

[36] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 7

[37] S. M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *CVPR*, volume 1, pages 519–528, June 2006. 2

[38] Qi Shan, Sameer Agarwal, and Brian Curless. Refractive height fields from single and multiple images. In *CVPR*, pages 286–293, 06 2012. 2

[39] Jonathan Stets, Zhengqin Li, Jeppe Revall Frisvad, and Manmohan Chandraker. Single-shot analysis of refractive shape using convolutional neural networks. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 995–1003. IEEE, 2019. 2

[40] Jonathan Dyssel Stets, Alessandro Dal Corso, Jannik Boll Nielsen, Rasmus Ahrenkiel Lyngby, Sebastian Hoppe Nesgaard Jensen, Jakob Wilm, Mads Brix Doest, Carsten Gundlach, Eythor Runar Eiriksson, Knut Conradsen, Anders Bjorholm Dahl, Jakob Andreas Bærentzen, Jeppe Revall Frisvad, and Henrik Aanæs. Scene reassembly after multimodal digitization and pipeline evaluation using photorealistic rendering. *Appl. Optics*, 56(27):7679–7690, 2017. 2

[41] K. Tanaka, Y. Mukaigawa, H. Kubo, Y. Matsushita, and Y. Yagi. Recovering transparent shape from time-of-flight distortion. In *CVPR*, pages 4387–4395, June 2016. 2

[42] Borislav Trifonov, Derek Bradley, and Wolfgang Heidrich. Tomographic reconstruction of transparent objects. In *ACM SIGGRAPH 2006 Sketches*, SIGGRAPH, New York, NY, USA, 2006. ACM. 2

[43] C. Tsai, A. Veeraraghavan, and A. C. Sankaranarayanan. What does a single light-ray reveal about a transparent object? In *ICIP*, pages 606–610, Sep. 2015. 2

[44] G. Wetzstein, D. Roodnick, W. Heidrich, and R. Raskar. Refractive shape from light field distortion. In *ICCV*, pages 1180–1186, Nov 2011. 2

[45] Yonatan Wexler, Andrew Fitzgibbon, and Andrew Zisserman. Image-based environment matting. In *CVPR*, pages 279–290, 01 2002. 2

[46] Bojian Wu, Yang Zhou, Yiming Qian, Minglun Cong, and Hui Huang. Full 3d reconstruction of transparent objects. *ACM ToG*, 37(4):103:1–103:11, July 2018. 1, 2, 3, 8

[47] Zhaohui Wu, Zhong Zhou, Delei Tian, and Wei Wu. Reconstruction of three-dimensional flame with color temperature. *Vis. Comput.*, 31(5):613–625, May 2015. 2

[48] Zexiang Xu, Kalyan Sunkavalli, Sunil Hadap, and Ravi Ramamoorthi. Deep image-based relighting from optimal sparse samples. *ACM Transactions on Graphics (TOG)*, 37(4):126, 2018. 6

[49] Yao Yao, Zixin Luo, Shiwei Li, Tianwei Shen, Tian Fang, and Long Quan. Recurrent MVSNet for high-resolution multi-view stereo depth inference. In *CVPR*, June 2019. 2

[50] S. Yeung, T. Wu, C. Tang, T. F. Chan, and S. Osher. Adequate reconstruction of transparent objects on a shoestring budget. In *CVPR*, pages 2513–2520, June 2011. 2

[51] S. Yeung, T. Wu, C. Tang, T. F. Chan, and S. J. Osher. Normal estimation of a transparent object using a video. *PAMI*, 37(4):890–897, April 2015. 2

[52] Sai-Kit Yeung, Chi-Keung Tang, Michael S. Brown, and Sing Bing Kang. Matting and compositing of transparent and refractive objects. *ACM ToG (SIGGRAPH)*, 30(1):2:1–2:13, 2011. 1, 2, 3

[53] Cheng Zhang, Lifan Wu, Changxi Zheng, Ioannis Gkioulekas, Ravi Ramamoorthi, and Shuang Zhao. A differential theory of radiative transfer. *ACM Trans. Graph.*, 38(6), 2019. 8

[54] Mingjie Zhang, Xing Lin, Mohit Gupta, Jinli Suo, and Qionghai Dai. Recovering scene geometry under wavy fluid via distortion and defocus analysis. In *ECCV*, volume 8693, pages 234–250, 09 2014. 2

[55] Jiayuan Zhu and Yee-Hong Yang. Frequency-based environment matting. In *Pacific Graphics*, pages 402–410, 2004. 2

[56] Douglas E. Zongker, Dawn M. Werner, Brian Curless, and David H. Salesin. Environment matting and compositing. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '99, pages 205–214, New York, NY, USA, 1999. ACM Press/Addison-Wesley Publishing Co. 2

[57] X. Zuo, C. Du, S. Wang, J. Zheng, and R. Yang. Interactive visual hull refinement for specular and transparent object surface reconstruction. In *ICCV*, pages 2237–2245, Dec 2015. 2