

Full 3D Reconstruction of Transparent Objects

BOJIAN WU, SIAT, Shenzhen University, and University of Chinese Academy of Sciences

YANG ZHOU, Shenzhen University and Huazhong University of Science & Technology

YIMING QIAN, University of Alberta

MINGLUN GONG, Memorial University of Newfoundland

HUI HUANG*, Shenzhen University

Numerous techniques have been proposed for reconstructing 3D models for opaque objects in past decades. However, none of them can be directly applied to transparent objects. This paper presents a fully automatic approach for reconstructing complete 3D shapes of transparent objects. Through positioning an object on a turntable, its silhouettes and light refraction paths under different viewing directions are captured. Then, starting from an initial rough model generated from space carving, our algorithm progressively optimizes the model under three constraints: surface and refraction normal consistency, surface projection and silhouette consistency, and surface smoothness. Experimental results on both synthetic and real objects demonstrate that our method can successfully recover the complex shapes of transparent objects and faithfully reproduce their light refraction properties.

CCS Concepts: • Computing methodologies → Computer graphics; Shape modeling; Point-based models;

Additional Key Words and Phrases: 3D reconstruction, transparent Objects

ACM Reference Format:

Bojian Wu, Yang Zhou, Yiming Qian, Minglun Gong, and Hui Huang. 2018. Full 3D Reconstruction of Transparent Objects. *ACM Trans. Graph.* 37, 4, Article 103 (August 2018), 11 pages. <https://doi.org/10.1145/3197517.3201286>

1 INTRODUCTION

Reconstructing 3D models of real objects has been an active research topic in both Computer Vision and Graphics for decades. A variety of approaches have been proposed for different applications, such as autonomous scanning [Wu et al. 2014], multi-view stereo [Galliani et al. 2015], photometric stereo [Chen et al. 2007], etc. While these techniques are able to faithfully capture and reconstruct the shapes of opaque or even translucent objects, none of them can be directly applied on transparent objects. As a result, people often have to paint those transparent objects before capturing their shapes.

On another front, how transparent objects refract lights toward a fixed viewpoint can be accurately acquired using environment matting techniques [Chuang et al. 2000; Qian et al. 2015]. Since

*Corresponding author: Hui Huang (hhzhiyan@gmail.com)

Authors' addresses: Bojian Wu, SIAT, Shenzhen University, University of Chinese Academy of Sciences; Yang Zhou, Shenzhen University, Huazhong University of Science & Technology; Yiming Qian, University of Alberta; Minglun Gong, Memorial University of Newfoundland; Hui Huang, College of Computer Science & Software Engineering, Shenzhen University.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2018 Association for Computing Machinery.

0730-0301/2018/8-ART103 \$15.00

<https://doi.org/10.1145/3197517.3201286>



Fig. 1. A transparent object refracts lights in its environment (left) and hence its shape cannot be reconstructed using conventional techniques. We present a novel method that, for the first time, directly reconstruct the shapes of transparent objects from captured images. This allows photo-realistic rendering of real transparent objects in virtual environments (right). For a clear view of the reconstructed geometry, please see Fig. 17.

light refraction paths are determined by surface normals, one has to wonder whether the shape of the transparent object can be inferred accordingly. This has been demonstrated as a feasible direction in a previous work [Qian et al. 2016]. Through enforcing a position-normal consistency constraint, their approach can generate point clouds on two sides of a given transparent object. Nevertheless, the captured surface shape is incomplete. In addition, this approach is not that easy to apply, involving setting up and calibrating among 4 different camera/monitor configurations.

This paper presents a fully automatic approach for reconstructing complete 3D shapes of transparent objects with known refractive indexes. Through positioning the object on a turntable, its silhouettes and light refraction paths under different viewing directions are captured using two fixed cameras (Section 3). An initial model generated through space carving is then gradually evolved toward the accurate object shape using novel point consolidation formulations that are constrained by captured light refraction paths and silhouettes (Section 4). Results on both synthetic and real objects (Section 5) demonstrate the effectiveness and robustness of our approach; see e.g., Fig. 1.

2 RELATED WORK

Surface reconstruction. The literature on 3D surface reconstruction [Berger et al. 2013, 2014] is vast. Specifically, reconstructing transparent objects is well-known a challenging problem [Ihrke et al. 2010]. Recent developments, such as reconstruction of flames [Ihrke and Magnor 2004; Wu et al. 2015b], mixing fluids [Gregson et al.



Fig. 2. Our data acquisition setup. The object to be captured (the *Monkey* statue in this case) is placed on Turntable #1. A LCD monitor is placed on Turntable #2 and serves as a light source. Camera #1 faces the object and the monitor for capturing silhouettes and ray-pixel correspondences. Camera #2 looks downward to the Turntable #1 for its rotation axis calibration. The bottom right monitor belongs to a PC that controls the data capture and is not used for illuminating the scene.

2012], gas flow [Atcheson et al. 2008; Ji et al. 2013], and cloud [Levis et al. 2015, 2017], aim at dynamic inhomogeneous transparent objects, whereas we focus on static reflective and refractive surfaces with homogeneous materials. Our approach is automatic and non-intrusive, different from existing intrusive acquisition methods [Aberman et al. 2017; Hullin et al. 2008; Trifonov et al. 2006].

Environment matting. To composite transparent objects into novel backgrounds, environment matting is often applied. This problem is introduced by Zongker et al. [1999], wherein environment mattes are estimated by projecting a series of horizontal and vertical color stripes. Chuang et al. [2000] extend the work for locating multiple distinct contributing sources, and also propose a single-image solution for colorless and purely specular objects. Wexler et al. [2002] develop an image-based method, which allows estimating environment mattes under natural scene background but requires a large amount of sample images. The problem can also be solved in the wavelet [Peers and Dutré 2003] and frequency [Qian et al. 2015] domain. Compressive sensing is leveraged to reduce the number of projecting patterns [Duan et al. 2015; Qian et al. 2015].

Shape-from-X. To estimate the geometry of transparent objects, the shape-from-distortion techniques [Ben-Ezra and Nayar 2003; Tanaka et al. 2016; Wetzstein et al. 2011] focus on analyzing known or unknown distorted background patterns. Zuo et al. [2015] incorporate internal occluding contours into traditional shape-from-silhouette methods, and propose a visual hull refinement scheme. It is also possible to reconstruct transparent objects by capturing exterior specular highlights [Morris and Kutulakos 2007; Yeung et al. 2011] known as shape-from-reflectance. However, the acquisition

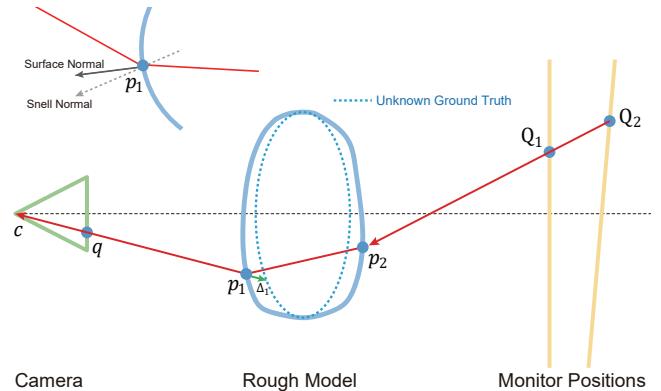


Fig. 3. Estimate light refraction path from ray-ray correspondence. For the exit ray that reaches camera center c through pixel q , its corresponding incident ray can be acquired by connecting the illuminating pixels Q_1 and Q_2 on monitor at different positions. Under the assumption that only two refractions happen, a full light path can be formed by linking points p_1 and p_2 on the rays based on hypothesized surface depths. The true light refraction path is the one that minimizes the differences between the surface normal (estimated from surface shape) and refraction normal (estimated using Snell's law) at both p_1 and p_2 locations; see inset on the top left corner.

approach requires manually moving a spotlight around the hemisphere to illuminate the object and a reference sphere from different directions. Recent shape-from-polarization methods [Cui et al. 2017; Huynh et al. 2010; Miyazaki and Ikeuchi 2005] connect polarization states of light with shape and surface material properties. Here we utilize the shape-from-silhouette to initialize our reconstruction.

Direct ray measurements. Kutulakos and Steger [Kutulakos and Steger 2008] provide theoretical analysis of the reconstruction feasibility using light path triangulation, and categorize the problem with numbers of reflections or refractions involved. Some researchers only focus on one-refraction events [Schwartzburg et al. 2014; Shan et al. 2012; Yue et al. 2014], in particular for fluid surface reconstruction [Morris and Kutulakos 2011; Qian et al. 2017; Zhang et al. 2014]. Tsai et al. [2015] consider two-refraction cases instead. Note that with given incident and exit ray-ray correspondences [Isenberghausen et al. 2017; Ji et al. 2013; Wetzstein et al. 2011], depth-normal ambiguity still exists as they are interrelated with each other. Qian et al. [2016] propose a position-normal consistency constraint for solving the two-refraction reconstruction problem, but they only compute a pair of front-back surface depth maps. Kim et al. [2017] develop a method to reconstruct axially-symmetric objects that could contain more than two refractions, however, cannot be applied to general non-symmetric objects.

Point data consolidation. Point clouds estimated from the ray-ray acquisition are in general highly unorganized, and corrupted heavily with noise, outliers, overlapping or missing regions [Qian et al. 2016]. A straightforward data cleaning step may easily cause the over-smoothing. Inspired by the point projection based data consolidation framework [Huang et al. 2009, 2013; Lipman et al. 2007; Wu et al. 2015a], we define two novel point consolidation formulations that project points sampled from initial rough surfaces toward the

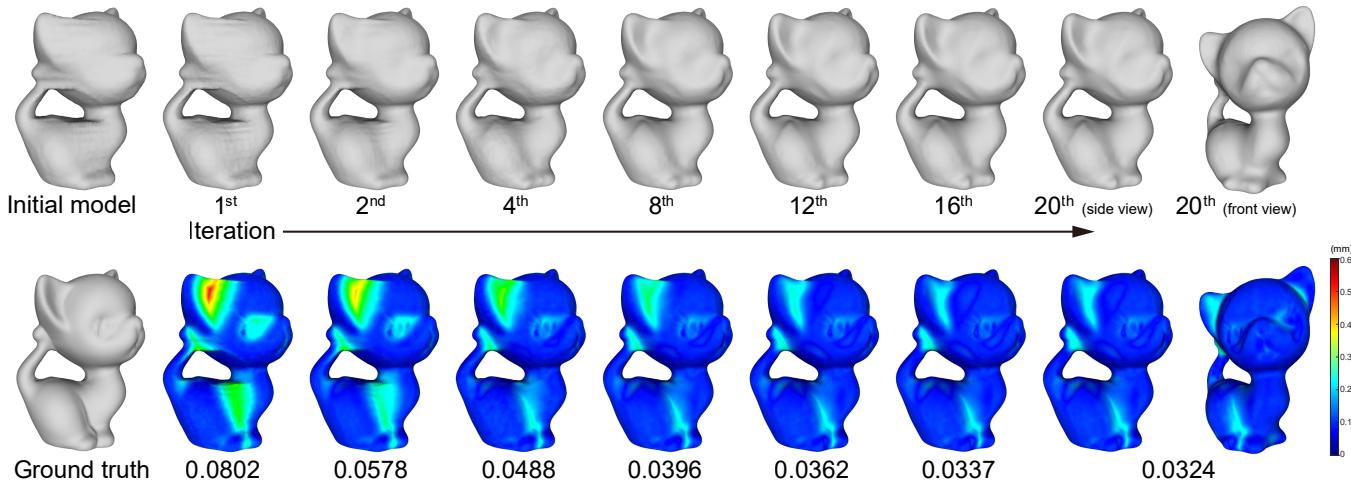


Fig. 4. Progressive reconstruction on a synthetic *Kitten* model with refractive index set to 1.15. Top: starting from an initial rough model obtained by space carving, our approach gradually recovers geometric details in concave areas and converges after 20 iterations. Bottom: the reconstruction error is visualized using the Hausdorff distance (measured per vertex) between the corresponding model and ground truth, forming as an error map. The number below each error map denotes the average Hausdorff distance (in millimeters). Compared to the size of the bounding box ($6.3 \times 9.7 \times 5.7$ mm), the error is quite small.

latent object geometry based on captured light refraction paths and silhouettes, respectively. Applying the two consolidation formulations in alternating manner can effectively guide the reconstructed model toward the true object shape, whereas directly applying existing data consolidation techniques does not yield satisfiable results.

3 CAPTURING SETUP

Here we first explain the setup that we have designed for data acquisition. As shown in Fig. 2, the transparent object to be captured is placed on Turntable #1. Two cameras are used and both are fixed during the capture process. Camera #1 is positioned in front of the transparent object and Camera #2 above it. Both cameras have their intrinsic parameters and relative positions calibrated [Zhang 2000]. In addition, through putting a checkerboard pattern on the turntable, its rotation axis with respect to the two cameras is also calibrated.

Similar to the previous work [Qian et al. 2016], a monitor is used as light source. Nonetheless, instead of manually moving the monitor during acquisition to capture starting locations and orientations of incoming rays, we place the monitor on top of Turntable #2. The monitor’s position can then be precisely and automatically adjusted.

To start the acquisition, we use Turntable #2 to set the monitor at its first position, which is calibrated with the cameras through displaying a checkerboard pattern. At this monitor position, we rotate the transparent target object using Turntable #1 to observe it from a set of (8 by default) directions that evenly sample the 360° viewing angle. At each direction, a series of binary Gray codes are displayed for both silhouette extraction [Zongker et al. 1999] and environment matting. The latter allows us to determine the pixel location on monitor that corresponds to a given ray refracted by the object and observed by Camera #1.

The process is repeated after setting the monitor to its second position. Here the monitor is moved using Turntable #2, but it can also be moved manually. The object is rotated again to perform

environment matting from the exact same set of view directions. Since the monitor is moved, a new illuminating pixel location can be computed for each observed ray. Connecting it with the corresponding one obtained in the previous round thus provides the incoming ray orientation; see 2D illustration in Fig. 3.

The images captured using the aforementioned procedure not only provide ray-ray correspondences, but also allow us to compute object silhouettes at each captured view. Since extracting the silhouettes requires much less computational effort, a higher sampling density is used to capture additional silhouettes. In practice, 72 view directions that evenly sample the 360° horizontal viewing angle are used for all examples presented in this paper.

4 RECONSTRUCTION METHOD

As described previously, the captured views provide us two important data: 1) silhouettes of the object from different views, which define the visual hull of the object, and 2) ray-ray correspondences before and after rays intersecting with the object, which correlate to light refraction paths and surface geometry details.

Our reconstruction starts from gathering all silhouettes to produce an initial rough model by space carving [Kutulakos and Seitz 2000]. The ultimate goal is to optimize this rough model according to the captured ray-ray correspondences while maintaining shape silhouettes. We achieve it through gradually updating the model under three constraints: surface and refraction normal consistency, surface projection and silhouette consistency, and surface smoothness. Fig. 4 shows our progressive results with reconstruction accuracy measurements on a synthetic *Kitten* example. We detail the optimization process in the following three subsections.

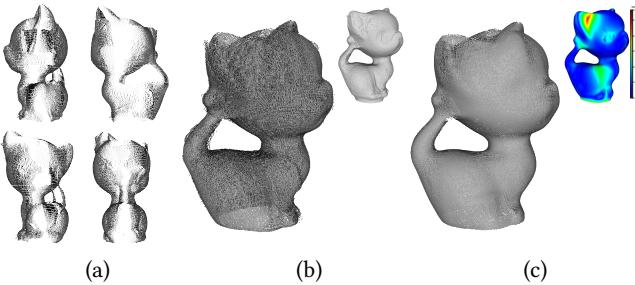


Fig. 5. Model reconstructed using only surface and refraction normal consistency. As in [Qian et al. 2016], each opposite pair of the eight captured views can be used to generate front and back depth maps, resulting four incomplete point clouds (a). Directly merging them together (b) and feeding into Poisson surface reconstruction (top right in (b)) results in a noisy and inaccurate model. Applying point consolidation [Huang et al. 2009] to clean up the data does not fully solve the problem either, as the resulting point cloud (c) is still far away from the ground truth; see the error map color coded on the top right of (c).

4.1 Surface and refraction normal consistency

Given a rough model, we first shoot rays from the camera to find the intersections, and then optimize the depths of these intersections along the corresponding rays according to the captured ray-ray correspondences.

As shown in Fig. 3, each ray-ray correspondence captured at a given view associates an exit ray observed by the camera to two pixels on the monitor, one before and one after the monitor is moved. Connecting the two pixel locations gives us the incident ray parameters. All captured ray-ray correspondences are within the object silhouettes and hence intersect with the object. In Qian et al.’s approach [2016], it is assumed that the light is refracted only twice when traveling between the monitor and the camera. One refraction occurs at the intersection between the exit ray and the object surface (referred to as front intersection), whereas the other occurs at the intersection between the incident rays and the surface (referred to as the back intersection). Under this assumption, directly connecting the front and the back intersections gives us the ray traveling path within the transparent object. The normal needed for achieving the desirable ray refraction effect at each intersection location can also be computed, based on Snell’s law [Born and Wolf 2013].

We adopt the same assumption with certain relaxation. Thanks to our rough model, we are able to trace each individual ray path between the monitor and the camera to filter out the captured ray-ray correspondences that involve more than two intersections. This allows us to reconstruct more complex object shapes (e.g., the *Mouse* statue shown in Fig. 1), which have more than two refractions under some view directions. In addition, rays that are involved in total reflections can also be detected and pruned.

The rough model also provides a good initial solution when optimizing the real surface shape. Similar to Qian et al.’s approach [2016], the surface shape is computed implicitly through optimizing the depth of intersection in captured images. That is:

$$\min_d \sum_{i \in I} \left(\|N(i) - SN(i)\|^2 + \lambda \sum_{i' \in N_i} \|d_i - d_{i'}\|^2 \right), \quad (1)$$

where $N(i)$ denotes the surface normal at the intersection point p_i , and $SN(i)$ denotes the Snell normal [Qian et al. 2016] induced by Snell’s law according to the light path of the refraction happened at the surface point. We have $d = \{d_i \mid i \in I\}$ and d_i is the depth from camera to the intersection point p_i along the ray. The set I only contains valid intersection points by ray-ray correspondences after removing those of involving more than two refractions and total reflections. The set N_i contains the indices of p_i ’s local neighborhood, which is computed using 4-connected neighboring pixels along the corresponding view. A standard 2-norm $\|\cdot\|$ is applied.

The first term minimizes the discrepancy between surface normal (approximated by local PCA analysis) and Snell (refraction) normal [Qian et al. 2016]. The second term penalizes on depth roughness. Empirically, we set the balancing parameter $\lambda = 10/\text{diaglen}$ by default, where diaglen denotes the diagonal length of the object bounding box. That is, the larger the object is, the smaller the value of λ shall be. Optimizing both terms produces a depth map for each captured view. Fig. 5 shows the point cloud generated by registering together four views. The resulting model, even though quite noisy, better captures surface details than the initial rough model, especially in concave regions.

4.2 Surface projection

As pointed out in Qian et al.’s paper [2016], the depth maps obtained using surface and refraction normal consistency are often noisy and incomplete. This is because ray-ray correspondences may not be captured for all pixels within the silhouette and ray refraction is highly sensitive to surface normal. As a result, the point cloud generated from optimizing (1), denoted by $P = \{p_i \mid i \in I\}$, contains heavy noise, outliers, mis-aligned errors, and missing areas; see e.g., Fig. 5 (b). Not only data consolidation is necessary, directly applying state-of-the-art consolidation techniques without any shape priors does not generate satisfiable results either; see e.g., Fig. 5 (c).

To address this challenge, we turn to the initial rough model. By applying Poisson-disk sampling [Corsini et al. 2012] on the rough model, we obtain a set of points (30K by default), $X^0 = \{x_j^0 \mid j \in J\}$, that evenly samples this complete, smooth, yet inaccurate surface. Our strategy is hence to smoothly evolve the point set X to recover more geometric features from the point cloud P while maintaining its completeness and smoothness. Inspired by the point consolidation work [Huang et al. 2009; Wu et al. 2015a], given the current iterate X^k , $k = 0, 1, \dots$, we compute the next iterate $X^{k+1} = \{x_j^{k+1} \mid j \in J\}$ by minimizing:

$$\sum_{j \in J} \left(\sum_{i \in I} \|x_j^{k+1} - p_i\| \theta(\|x_j^k - p_i\|) + \frac{\alpha}{|N_j|} \sum_{j' \in N_j} \|\Delta_j - \Delta_{j'}\|^2 \right), \quad (2)$$

where $\Delta_j = x_j^{k+1} - x_j^k$ is the displacement vector, and $\theta(\|x_j^k - p_i\|) = e^{-\|x_j^k - p_i\|^2/(h_j/4)^2}$ is a fast descending function with an adaptive support radius h_j that defines the size of the influence data neighborhood adaptively with respect to x_j . We set the weighting parameter $\alpha = 7.5$ by default to balance the two terms in (2).

Same as previous approaches [Huang et al. 2009; Lipman et al. 2007], the first term is a local ℓ_1 -median projection, which is known as an effective noise-removal operator for unorganized point clouds

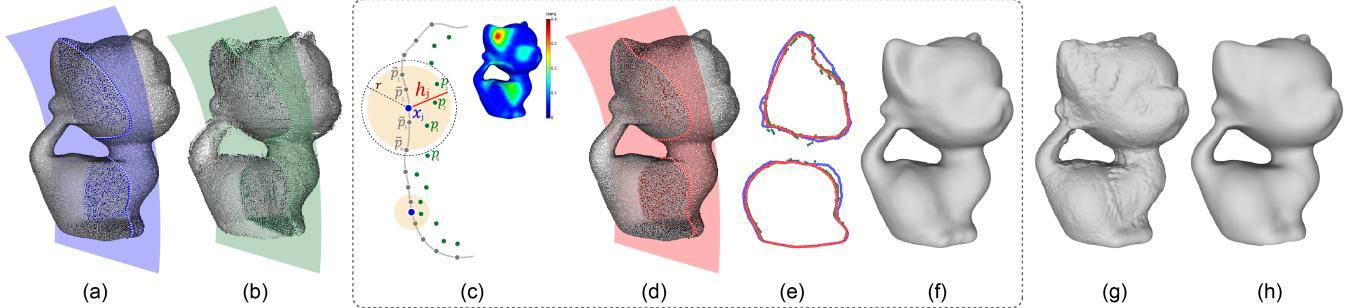


Fig. 6. Importance of adaptive neighborhood size selection for point consolidation. For the point set X (in blue) sampled from the initial rough model (a), our approach adaptively selects the neighborhood size h for each point x based on the average distance between the rough model and the noisy point cloud (in green) estimated using ray-ray correspondences (b). The neighborhood sizes computed for different areas are color coded in (c), which shows that larger neighborhoods are used for areas not well modeled by the rough model, such as the concave ear region, and smaller neighborhoods for convex parts. Projecting sample points using adaptive neighborhoods results in a consolidated point cloud (in red) that better captures the shape of the *Kitten* around its ear and back while maintains to be smooth; see cross-section curves shown in (e). Using the consolidated point cloud shown in (d), we can get better reconstructed shape on these concave areas (f). In comparison, projection using a fixed neighborhood size will either leads to a noisy model when the neighborhood size is small (g), or to an over-smoothed model when the neighborhood is large (h). In both cases, the surface in concave regions are not properly reconstructed.

and is non-sensitive to the presence of outliers. However, unlike previous work that uses a regularization term to enforce sample point distribution, here the second term is defined as the Laplacian on the projection displacements. Such a change is motivated by the fact that the source of initial samples are different. In previous approaches [Huang et al. 2009; Lipman et al. 2007], initial points are sampled from incomplete point cloud and hence are unevenly distributed. In our case, the set X is evenly sampled from the initial complete rough surface. Therefore, we only need to maintain the distribution of samples using a simpler Laplacian regularization.

On the other hand, the previous approaches use a fixed neighborhood size for consolidating all points on the surface [Huang et al. 2009; Lipman et al. 2007]. In our case, the rough model can be very close to real object shape in areas close to the silhouettes, but dramatically different from it in concave regions. Hence, using a small neighborhood size cannot effectively project points in concave regions, whereas using a large neighborhood may blur the geometric details we would like to recover in areas near the silhouettes.

To address this issue, an adaptive neighborhood radius h_j is used for projecting each sample x_j . It is computed based on the average distance between point cloud P and rough model X in the neighborhood of x_j . That is, a smaller neighborhood will be used when the point cloud estimated using ray-ray correspondences agrees with the model generated from silhouettes, whereas a larger neighborhood will be used when the two disagree.

In particular, h_j is defined as the average of the ray shooting distances between p_i and the corresponding \bar{p}_i that hits on the rough model and lies in the local neighborhood N_{x_j} :

$$h_j = \sum_{i \in N_{x_j}} \frac{1}{|N_{x_j}|} \|p_i - \bar{p}_i\|, \quad N_{x_j} = \{i \mid \|x_j - \bar{p}_i\| \leq r\}. \quad (3)$$

The parameter r is computed when we use Poisson-disk sampling to generate X^0 with 30K points by default, i.e., the average distance among initial samples. Hence, only points $\{\bar{p}_i\}$ whose distance to x_j is less than r are used for computing the average value for h_j .

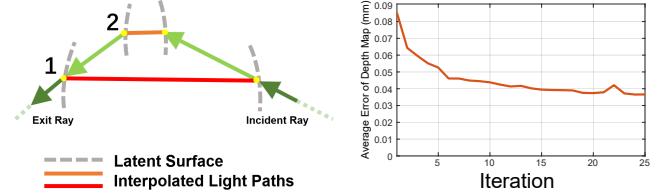


Fig. 7. Limitation on normal consistency constraint. For a pair of ray-ray correspondence, there may exist multiple refraction surfaces that satisfy the surface and refraction normal consistency constraint (left). This introduces ambiguities when estimating surface depth using (1). Feeding in better initial solution can alleviate the problem. As a result, with the rough model getting more accurate during the progressive reconstruction, the errors in depth maps obtained by optimizing (1) also get smaller (right).

The radius h_j is usually large in concave regions and small in areas where the rough model is already approximated well to ground truth. Fig. 6 shows the effectiveness of our adaptive local projection.

4.3 Silhouette consistency

As discussed above, silhouettes and light refraction paths provide independent cues on the shape of real surface. Silhouettes offer accurate shape boundary information under selected viewpoints. The light refraction paths provide surface depth cues for both convex and concave areas, but are prone to noises. More importantly, as shown in Fig. 7(a), the normal consistency constraint can be ambiguous and hence the estimated surface depth may not be accurate. Even though the initial rough model obtained through space carving perfectly matches the silhouettes, after applying the aforementioned consolidation step to satisfy the surface and normal consistency, the resulting model may deviate from the captured silhouettes. It is thus worth to enforce the surface projection and silhouette consistency.

Specifically, we want the projection of the point cloud to fully occupy the captured silhouettes in all views. This is achieved by

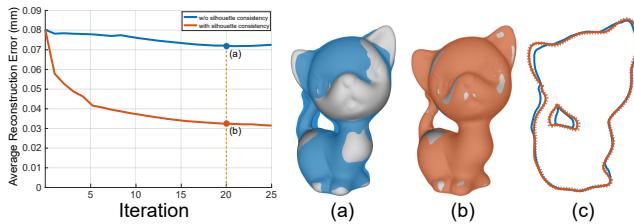


Fig. 8. Impact of constraint on shape silhouette consistency. The comparison shows that, when silhouette consistency (4) is applied, the reconstructed mesh (shown in red in (b)) is much closer to the ground truth (shown in gray) in ears, tail and claws regions than the mesh obtained without using this constraint (blue in (a)). The contours of all three shapes are shown in (c) using the same color coding.

minimizing a data term defined using the distance between the boundary of point cloud projection and the captured silhouettes. Combining such a data term with the same smoothing term as defined in (2) gives us the following objective function:

$$\min_X \sum_{j \in J} \left(\sum_{v=1}^V \delta_j^v D(q_j^v, \partial\Omega_v) + \frac{\beta}{|\mathcal{N}_j|} \sum_{j' \in \mathcal{N}_j} \|\Delta_j - \Delta_{j'}\|^2 \right), \quad (4)$$

where $\Delta_j = x_j - x_j^k$ is the displacement vector as in (2), and V (defaulted to be 72) is the number of captured silhouettes. We have $q_j^v = \mathbf{P}^v \cdot x_j$ to denote the 2D projection of the sample point on view v , where \mathbf{P}^v is the corresponding projection matrix of view v . $\partial\Omega_v$ denotes the boundary of the matting mask Ω_v on view v , i.e., the object silhouette on this view. δ_j^v is a binary indicator function, which equals to 1 if q_j^v lies on the boundary of the projected shape on view v under a threshold, otherwise becomes 0. The function $D(q_j^v, \partial\Omega)$ returns the closest distance from the point q_j^v to $\partial\Omega$ on the projection image plane. We set the parameter $\beta = 50/\text{diaglen}$ by default to balance the fitting and smoothness terms with respect to the sizes of transparent objects.

To compute the binary indicator function δ_j^v in the data term, the following procedure is used. We start with projecting all sample points $\{x_j \mid j \in J\}$ onto the given view v . The areas covered by the latent surface represented by sample points are determined through a flood fill operation, which is performed on a k -NN graph ($k = 6$ by default) build upon the 3D points. The boundary contour of the filled 2D shape is then computed and the value of the indicator function δ_j^v is determined. Please note that, since the projection operation may change the neighborhood structure of the point cloud, the 3D k -NN graph and the 2D filled shapes need to be updated after each optimization iteration.

Solving the above silhouette consistency optimization (4) enforces a smooth point-based shape deformation to ensure that the projections of resulting point cloud match well with silhouettes in all captured views. Fig. 8 quantitatively compares the results generated with and without silhouette consistency optimization. The reconstruction error here is measured using the Hausdorff distance between the reconstructed model and the ground truth. The reconstruction error plotted under the two settings provide convincing evidence on the importance of this constraint.

Table 1. Average computation time per iteration of different processing steps on both synthetic and real examples presented in the paper. Note that space carving is only performed once for each object and the time for normal consistency is the sum of solving 4 front-back view pairs.

Reconstruction steps	Time (mins)
Space carving	15
Normal consistency	15
Surface projection	3
Silhouette consistency	3
Screened Poisson	0.1

4.4 Progressive reconstruction

Once the point cloud sampled from the initial rough model goes through two phases of consolidation using different constraints, a new 3D surface model is generated from the resulting point cloud using screened Poisson surface reconstruction [Kazhdan and Hoppe 2013]. This new model will serve as the rough shape for the next round of sampling, surface depth estimation (Section 4.1), consolidation (Section 4.2 and 4.3), and reconstruction. As the rough model gets more accurate, it can more precisely filter out ray-ray correspondences that involve more than two refractions and total reflections and provide better initial solution for optimizing (1). This helps to alleviate the aforementioned ambiguity problem for normal consistency and leads to better surface depth maps; see Fig. 7. The overall surface model can therefore progressively approach the true object shape; see Figs. 4 and 9.

It is worth noting that using two phases of consolidation to satisfy normal consistency and silhouette consistency constraints is necessary and important. In fact, attempts were made to formulate both consistency constraints into a single objective function. However, since the initial rough model obtained through space carving matches perfectly with all captured silhouettes. It becomes a local optimal solution and hence, the optimization process is often stalled. Alternatively consolidating points under the two constraints with the regularization on surface smoothness well balances the stochasticity that is necessary for searching a global optimal reconstruction.

5 RESULTS AND DISCUSSION

We have implemented our algorithm in C++, with parallelizable parts optimized using OpenMP. On average, each reconstruction iteration takes about 20 minutes on a 24-core PC with 2.30GHz Xeon CPU and 64GB RAM. We solve the objective functions (1) and (4) by L-BFGS-B [Zhu et al. 1997], and adopt the iterative algorithm proposed in [Huang et al. 2009] to optimize (2). Table 1 lists the average computation time per iteration for each step of the reconstruction process. As in general our algorithm converges within 20 iterations, the full reconstruction for an object can be completed within 5~6 hours. In addition, we need roughly another two hours to render or capture all data, and one more hour to compute the alphamatte and ray-ray correspondences for each transparent object.

5.1 Synthetic experiments with evaluation

To evaluate our method, we first run the algorithm on two widely used synthetic models: *Kitten* and *Bunny*. Both models are rendered

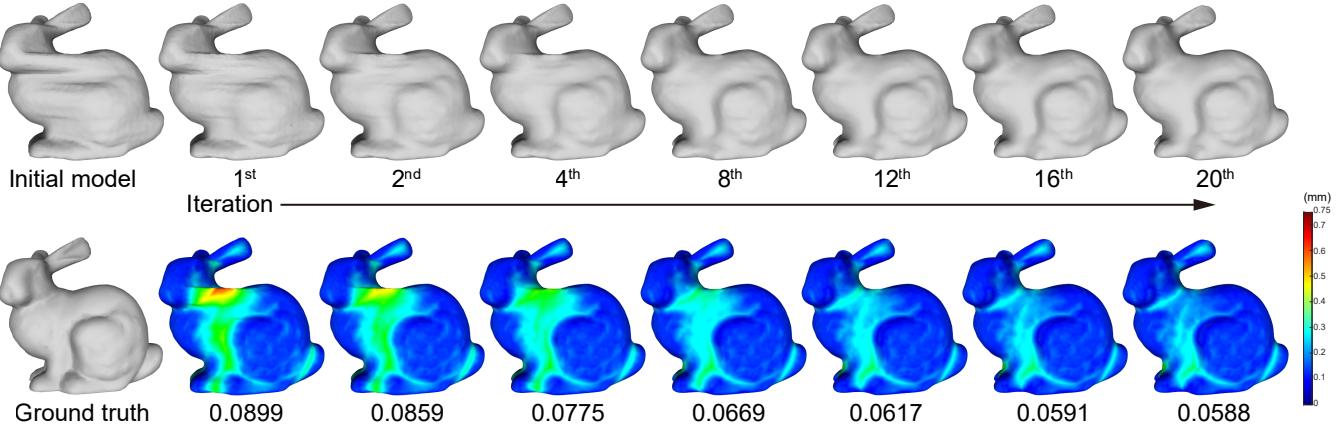


Fig. 9. Results on synthetic *Bunny* with refractive index set to 1.15. The size of the bounding box of *Bunny* is $8.4 \times 8.3 \times 6.5$ mm. The numbers below each error map is the average Hausdorff distance between the reconstructed shape and ground truth.



Fig. 10. Visual comparison on the rendering results of *Kitten*. Each of the three image pairs is generated using the same view angle, where the left one is the rendering of ground truth and the right is our recovered model.



Fig. 11. Visual comparison on the rendering results between ground truth *Bunny* (left) and our recovered model (right).

using POV-Ray* as transparent objects, with the refractive index set to 1.15. The objects, virtual cameras, and virtual monitors are set up the same way as discussed in Section 3. During the rendering, we turn off anti-aliasing to avoid edge blurriness and to better capture ray-ray correspondences. Both two virtual cameras are set to be the pinhole model. The resolution of virtual monitor is 1920×1080 . Thus,

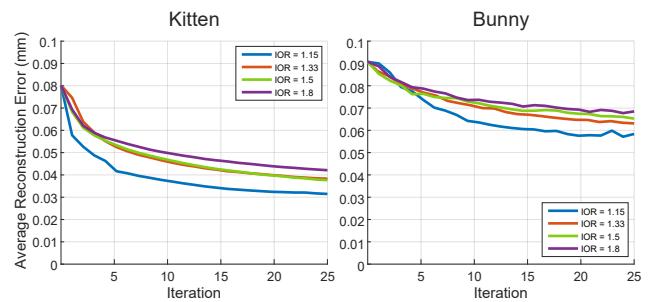


Fig. 12. Synthetic tests with different refractive indexes. The average reconstruction error after each iteration is plotted for *Kitten* and *Bunny*.

for each view of Camera #1, it needs to render 22 Gray encoded images, 11 for rows and 11 for columns.

The models progressively reconstructed from these rendered images can be seen from Figs. 4 and 9. Distances from reconstructed models to the ground truth surface are visualized as error maps for quantitative evaluation. For both models, the average distance decreases with more iterations, which demonstrates the effectiveness of our algorithm. Figs. 10 and 11 visually compare the rendering results of ground truth models and our reconstructions in the same

*Persistence of Vision Raytracer, <http://www.povray.org/>



Fig. 13. Transparent objects used for testing in this paper. The combination of complex object shapes and cluttered environment leads to high frequency signals in the captured image.

environment under different views. They suggest that our approach can nicely reproduce the appearances of transparent objects.

We also test our algorithm on synthetic examples with different refractive indices in Fig. 12. From the error curve we can see that higher IOR correlates to larger residual error. This is mainly due to the fact that the object is only illuminated from the back. Hence, higher IOR leads to fewer captured ray-ray correspondences, resulting in higher reconstruction error.

5.2 Real object experiments with evaluation

Five real transparent objects made from borosilicate 3.3 glass[†] are also used for our experiments; see Fig. 13. A DELL LCD monitor (U2412M) with resolution 1920×1200 is used for displaying Gray code patterns, where 22 images (11 for rows and 11 for columns) are needed under each view of Camera #1. During capturing, the brightness and contrast of this monitor are set to be the highest for sufficient background illumination. The turntable is controlled by a 1.8° 57mm stepper motor with a gear ratio of 180:1, possessing repeatability accuracy of 0.005° . To capture data for real objects, two Point Grey Flea3 color cameras (FL3-U3-13S2C-CS) are used. While Camera #2 uses the default settings with its focus on Turntable #1, we change Camera #1 to be the manual mode and set a small aperture (about f/6.0) in order to mimic the pinhole model. Also, Camera #1 is focused on the object for clear imaging. To optimize the brightness and quality of captured images, the shutter time and gain of the camera are set to 40~50ms and 0dB, respectively.

Fig. 15 shows the images captured for the *Hand* object under one of the views. Using ray-ray correspondences extracted from these images, we are able to recover surface depths under each view through optimizing surface and refraction normal consistency; see Fig. 16. Directly consolidating these depth information does not provide a satisfiable model. Our algorithm, on the other hand, starts from a rough model obtained using space carving and progressively enriches it as shown in Fig. 14 (a). The final converged model is smooth and nicely captures surface details in concave areas.

[†]According to the manufacturer, the glass has a melting point of 820°C and a refractive index of 1.4723

To conduct quantitative evaluation on this real object, we reconstructed its ground truth shape using an intrusive method. As shown in Figs. 14(b-d), we painted it with DPT-5 developer, and scanned it with a high-end industrial level scanner. The reconstruction errors of models generated by our approach are then evaluated through registering them with the ground truth model using ICP and then computing the average distance in-between them. As shown in Fig. 14(e), when dealing with real objects, our progressive reconstruction approach converges equally well as with synthetic data. Even though there is still residual reconstruction error in the end, our result improves significantly (by 26 percent) over the initial model obtained by space carving.

Fig. 17 shows reconstruction results on the transparent *Bunny* and *Mouse* objects. Our final reconstruction provides noticeable improvement on the concave areas (e.g., neck and tummy) over the initial rough model, while the silhouette contours (e.g., leg and back) of the object are well preserved. Fig. 18 shows another two results on the transparent *Monkey* and *Dog*. Our reconstruction successfully recover the eye region for the monkey and the belly shape for the dog. However, the crotch areas of both two models are not well-recovered due to the violation of two-refraction assumption.

5.3 Discussions on reconstruction error

As shown in Figs. 4, 9, 12 and 14, the residual reconstruction errors are less than 1/100 of the object size. Based on our setup in synthetic experiments, each pixel roughly projects to 0.015mm in length on the object's surface. For real data, each pixel from Camera #1 corresponds to 0.18mm on the object. Correspondingly, we can convert the average residual error into about 2 pixels for *Kitten* and *Bunny* synthetic data and 3 pixels for the real *Hand* model.

Our analysis suggests two main sources for the residual error. First, as mentioned in Section 4.3, whether the projected point is on the silhouette boundary is determined at pixel level. Thus, the recovered model after optimization with silhouette consistency might deviate from the ground truth for up to a pixel. The other main error source should be derived from the uncertainties of ray-ray correspondences. Practically, in capturing stage, besides the precision issue of ray-pixel extraction on captured images, the sensitivity of refraction and complexity of the surface geometry could introduce many unreliable correspondences, which would directly affect the reliability of the captured ray-ray correspondences. As Fig. 12 suggests, the more unreliable ray-ray correspondences are, the higher average reconstruction error would be. For the real *Hand* model in Fig. 14, due to the finite thickness of DPT-5 developer layer and/or the possible misalignment during the scanning, extra errors might be introduced into the measurement of reconstruction accuracy.

It is worth noting that, even though our approach still has room for improvement in terms of reconstruction accuracy, it is the first non-intrusive and fully-automatic approach for reconstructing complete 3D shapes of transparent objects. In comparison, Qian et al. [2016]'s method only recovers the incomplete point clouds for front and back surfaces. As shown in Figs. 5 and 16, directly merging the incomplete point clouds under different views does not provide satisfactory results. To recover full 3D models for transparent objects, we formulate a novel silhouette constraint and use it to

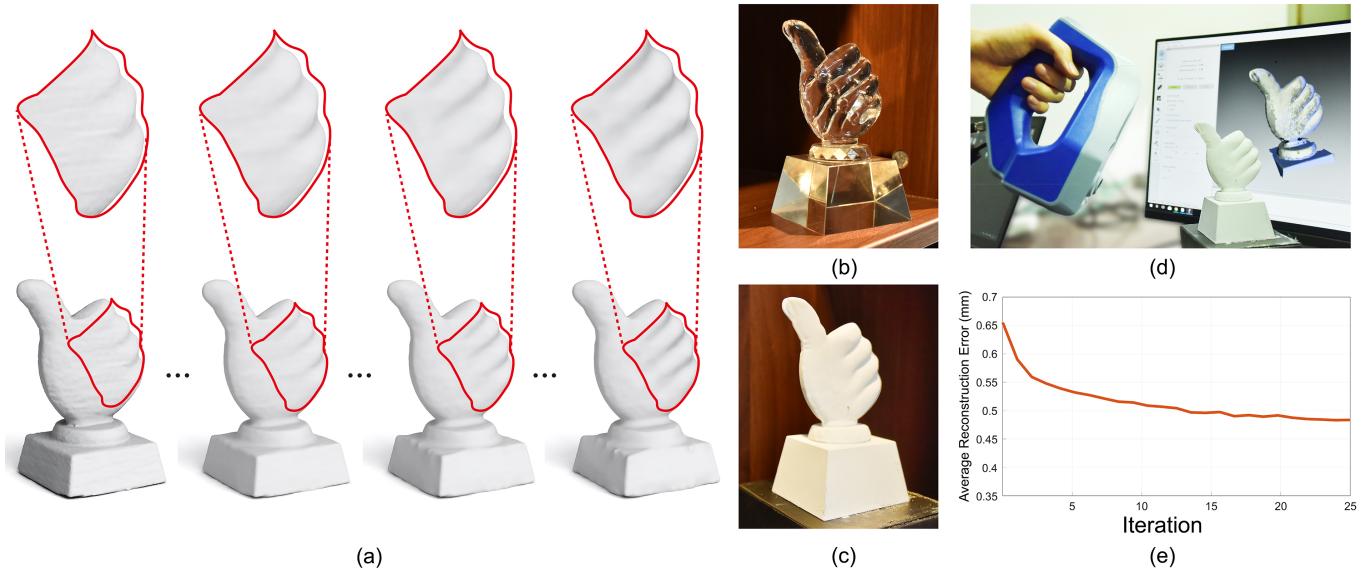


Fig. 14. Progressive reconstruction of the *Hand* object shown in Fig. 15. Our approach gradually recovers surface details that are not available in the initial rough model (a). To conduct quantitative evaluation, we also painted the transparent object (b) with DPT-5 developer (see (c)) and then carefully scanned it using a high-end Artec Space Spider scanner (d). The size of the bounding box of this scanned *Hand* model is $80 \times 119 \times 64$ mm. Using this captured model as ground truth, the average reconstruction error after each iteration is plotted in (e). The result curve shows that our approach can effectively reduce reconstruction errors and converge after 20 iterations.

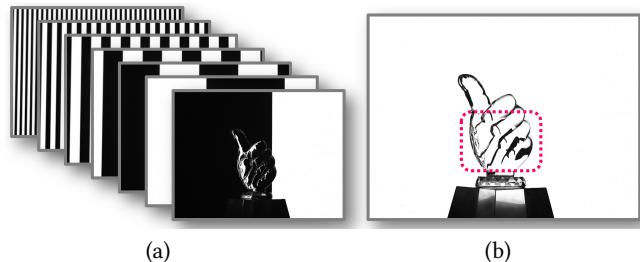


Fig. 15. Examples of captured images of a *Hand* object (a). However, as highlighted in (b), even when all pixels on the monitor are turned on, there are still areas on the object do not get illuminated, as they refract lights from directions not covered by the background monitor. Their ray-ray correspondences, therefore, cannot be captured.

optimize the reconstructed shape by iterating between the normal consistency and silhouette constraint. Since capturing silhouettes is much easier and more reliable than ray-ray correspondences, our reconstruction is more reliable and robust than [Qian et al. 2016].

6 CONCLUSIONS AND FUTURE WORK

This paper presents the first practical method for automatically and directly reconstructing complete 3D models for transparent objects based only on their appearances in a controlled environment. The environment is designed using affordable and off-the-shelf products, which include a LCD monitor, two turntables, and two cameras. This setup can work in fully automatic fashion, removing the needs for manually adjusting object positions and calibrating the cameras.

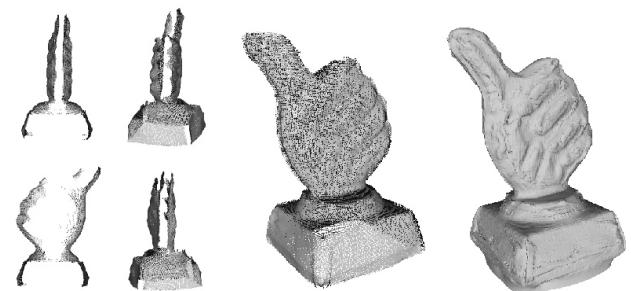


Fig. 16. Using data captured from each front-back view pairs, we can estimate front and back surface depths based on surface and refraction normal consistency, resulting a set of incomplete point clouds as shown in left. Directly merging these point clouds (middle) cannot produce satisfactory surface reconstruction (right).

Two sets of data are captured, one is for shape silhouettes and the other is for ray-ray correspondences before and after light refraction. Our presented algorithm fully utilizes both sets of data and progressively reconstructs the 3D model of a given transparent object using three constraints: surface and refraction normal consistency, surface projection and silhouette consistency, and surface smoothness. Experiments on both synthetic and real objects with quantitative evaluations demonstrate the effectiveness of our algorithm.

Our method still has several limitations, which set up our future work. The first one is about the data capturing process. As shown in Fig. 15, with one LCD monitor serving as light source behind



Fig. 17. Reconstructions of the real transparent *Bunny* and *Mouse*. In each image group, the left image shows the photo-realistic rendering of our recovered model, whereas the middle and right images are used to compare the initial rough model with the final reconstruction. Note how the neck and tummy regions of both shapes are significantly improved, while the silhouettes of the object are well preserved.

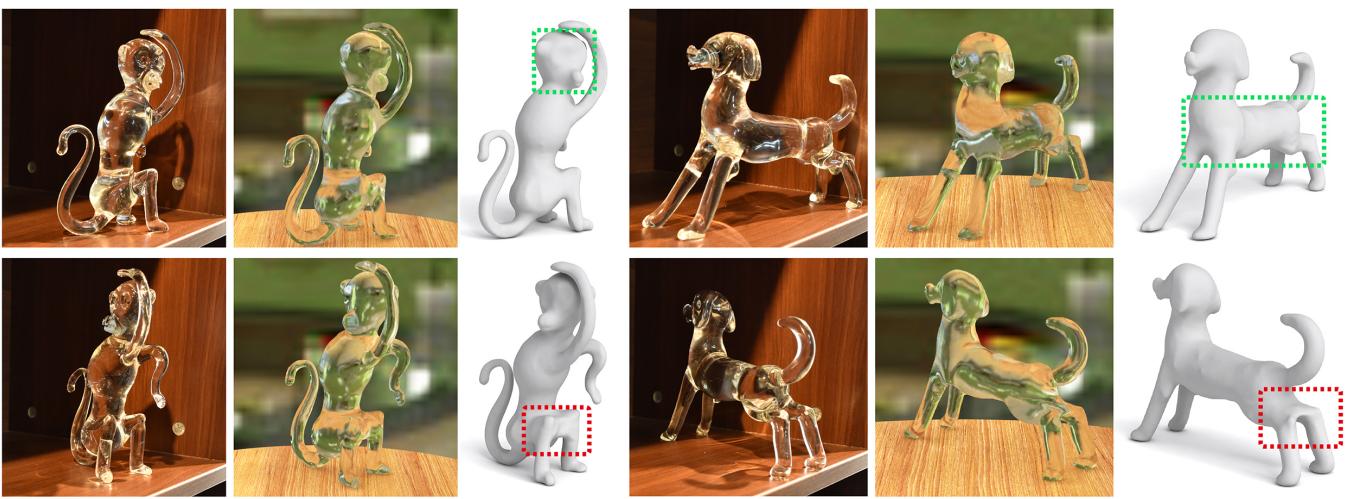


Fig. 18. Reconstruction results for *Monkey* and *Dog* under two view directions. For each object, the reconstruction (right column) successfully captures the featured concave parts (highlighted in green box). Nonetheless, we can still observe areas not well-reconstructed (highlighted in red) due to the violation of two-refraction assumption. Since these reconstruction artifacts only show up in areas involving multiple refractions, they are hardly noticeable when comparing the rendering of our reconstructed model (middle column) with real object photos (left column).

the object, and one single camera in the front, not all ray-ray correspondences information can be captured, resulting missing data in estimated point cloud. If there is an area that is missing in all captured views, its surface can only be inferred based on surface smoothness and hence geometry details can be lost. This limitation can be addressed through adding either additional monitors or additional cameras to cover ray-ray correspondence paths with more oblique angles, but at higher system and computation cost.

Secondly, our approach inherited the assumption from [Qian et al. 2016] that the transparent object is homogeneous and only two refractions occur on each light path. Even though our algorithm can automatically filter out data that violates the two-refraction assumption, and can reconstruct objects that refract lights more than twice in certain directions, it cannot handle multiple refractions directly. This limitation leads to the reconstruction artifacts shown in Fig. 18. In addition, transparent objects that are hollow inside cannot be processed since light is refracted 4 times in all directions. Also, here we assume the refractive index of the transparent object is

known or can be estimated in advance. In the near future, we would like to extend our approach to address this full 3D reconstruction problem of transparent objects without these assumptions.

ACKNOWLEDGMENTS

We thank the anonymous reviewers for their valuable comments. This work was supported in part by NSFC (61522213, 61761146002, 61861130365), 973 Program (2015CB352501), Guangdong Science and Technology Program (2015A030312015), Shenzhen Innovation Program (KQJSCX20170727101233642, JCYJ20151015151249564) and NSERC (293127).

REFERENCES

- Kfir Aberman, Oren Katzir, Qiang Zhou, Zegang Luo, Andrei Sharf, Chen Greif, Baoquan Chen, and Daniel Cohen-Or. 2017. Dip Transform for 3D Shape Reconstruction. *ACM Trans. on Graphics (Proc. of SIGGRAPH)* 36, 4 (2017), 79:1–79:11.
- Bradley Atcheson, Ivo Ihrke, Wolfgang Heidrich, Art Tevs, Derek Bradley, Marcus Magnor, and Hans-Peter Seidel. 2008. Time-resolved 3D Capture of Non-stationary Gas Flows. *ACM Trans. on Graphics (Proc. of SIGGRAPH Asia)* 27, 5 (2008), 132:1–132:9.

- Moshe Ben-Ezra and Shree K. Nayar. 2003. What Does Motion Reveal About Transparency? *Proc. Int. Conf. on Computer Vision* (2003), 1025–1032.
- Matthew Berger, Joshua A. Levine, Luis Gustavo Nonato, Gabriel Taubin, and Claudio T. Silva. 2013. A Benchmark for Surface Reconstruction. *ACM Trans. on Graphics* 32, 2 (2013), 20:1–20:17.
- Matthew Berger, Andrea Tagliasacchi, Lee M. Seversky, Pierre Alliez, Joshua A. Levine, Andrei Sharf, and Claudio Silva. 2014. State of the Art in Surface Reconstruction from Point Clouds. *Eurographics STAR* (2014), 165–185.
- Max Born and Emil Wolf. 2013. *Principles of optics: electromagnetic theory of propagation, interference and diffraction of light*. Elsevier.
- Tongbo Chen, Hendrik PA Lensch, Christian Fuchs, and Hans-Peter Seidel. 2007. Polarization and Phase-Shifting for 3D Scanning of Translucent Objects. *Proc. IEEE Conf. on Computer Vision & Pattern Recognition* (2007), 1–8.
- Yung-Yu Chuang, Douglas E. Zongker, Joel Hindorff, Brian Curless, David H. Salesin, and Richard Szeliski. 2000. Environment Matting Extensions: Towards Higher Accuracy and Real-time Capture. *ACM Trans. on Graphics (Proc. of SIGGRAPH)* (2000), 121–130.
- Massimiliano Corsini, Paolo Cignoni, and Roberto Scopigno. 2012. Efficient and Flexible Sampling with Blue Noise Properties of Triangular Meshes. *IEEE Trans. Visualization & Computer Graphics* 18, 6 (2012), 914–924.
- Zhaopeng Cui, Jinwei Gu, Boxin Shi, Ping Tan, and Jan Kautz. 2017. Polarimetric Multi-View Stereo. *Proc. IEEE Conf. on Computer Vision & Pattern Recognition* (2017), 1558–1567.
- Qi Duan, Jianfei Cai, and Jianmin Zheng. 2015. Compressive Environment Matting. *The Visual Computer* 31, 12 (2015), 1587–1600.
- Silvano Galliani, Katrin Lasinger, and Konrad Schindler. 2015. Massively Parallel Multiview Stereopsis by Surface Normal Diffusion. *Proc. Int. Conf. on Computer Vision* (2015), 873–881.
- James Gregson, Michael Krimerman, Matthias B. Hullin, and Wolfgang Heidrich. 2012. Stochastic Tomography and Its Applications in 3D Imaging of Mixing Fluids. *ACM Trans. on Graphics (Proc. of SIGGRAPH)* 31, 4 (2012), 52:1–52:10.
- Hui Huang, Dan Li, Hao Zhang, Uri Ascher, and Daniel Cohen-Or. 2009. Consolidation of Unorganized Point Clouds for Surface Reconstruction. *ACM Trans. on Graphics (Proc. of SIGGRAPH Asia)* 28, 5 (2009), 176:1–176:7.
- Hui Huang, Shihao Wu, Minglun Gong, Daniel Cohen-Or, Uri Ascher, and Hao Zhang. 2013. Edge-aware Point Set Resampling. *ACM Trans. on Graphics* 32, 1 (2013), 9:1–9:12.
- Matthias B. Hullin, Martin Fuchs, Ivo Ihrke, Hans-Peter Seidel, and Hendrik P. A. Lensch. 2008. Fluorescent Immersion Range Scanning. *ACM Trans. on Graphics (Proc. of SIGGRAPH)* 27, 3 (2008), 87:1–87:10.
- Cong Phuoc Huynh, Antonio Robles-Kelly, and Edwin Hancock. 2010. Shape and refractive index recovery from single-view polarisation images. *Proc. IEEE Conf. on Computer Vision & Pattern Recognition* (2010), 1229–1236.
- Ivo Ihrke, Kiriakos N. Kutulakos, Hendrik Lensch, Marcus Magnor, and Wolfgang Heidrich. 2010. Transparent and specular object reconstruction. *Computer Graphics Forum* 29, 8 (2010), 2400–2426.
- Ivo Ihrke and Marcus Magnor. 2004. Image-based Tomographic Reconstruction of Flames. *Proc. Eurographics Symp. on Computer Animation* (2004), 365–373.
- Julian Isringhausen, Bastian Goldlücke, Nina Pesheva, Stanimir Iliev, Alexander Wender, Martin Fuchs, and Matthias B. Hullin. 2017. 4D Imaging Through Spray-on Optics. *ACM Trans. on Graphics (Proc. of SIGGRAPH)* 36, 4 (2017), 35:1–35:11.
- Yu Ji, Jinwei Ye, and Jingyi Yu. 2013. Reconstructing gas flows using light-path approximation. *Proc. IEEE Conf. on Computer Vision & Pattern Recognition* (2013), 2507–2514.
- Michael Kazhdan and Hugues Hoppe. 2013. Screened Poisson Surface Reconstruction. *ACM Trans. on Graphics* 32, 3 (2013), 29:1–29:13.
- Jaewon Kim, Ilya Reshetouski, and Abhijeet Ghosh. 2017. Acquiring Axially-Symmetric Transparent Objects Using Single-View Transmission Imaging. *Proc. IEEE Conf. on Computer Vision & Pattern Recognition* (2017), 1484–1492.
- Kiriakos N. Kutulakos and Steven M. Seitz. 2000. A Theory of Shape by Space Carving. *Int. J. Computer Vision* 38, 3 (2000), 199–218.
- Kiriakos N. Kutulakos and Eron Steger. 2008. A theory of refractive and specular 3D shape by light-path triangulation. *Int. J. Computer Vision* 76, 1 (2008), 13–29.
- Aviad Levis, Yoav Y Schechner, Amit Aides, and Anthony B Davis. 2015. Airborne three-dimensional cloud tomography. *Proc. Int. Conf. on Computer Vision* (2015), 3379–3387.
- Aviad Levis, Yoav Y Schechner, and Anthony B Davis. 2017. Multiple-scattering microparticle tomography. *Proc. IEEE Conf. on Computer Vision & Pattern Recognition* (2017), 5797–5806.
- Yaron Lipman, Daniel Cohen-Or, David Levin, and Hillel Tal-Ezer. 2007. Parameterization-free Projection for Geometry Reconstruction. *ACM Trans. on Graphics (Proc. of SIGGRAPH)* 26, 3 (2007), 22:1–22:6.
- Daisuke Miyazaki and Katsushi Ikeuchi. 2005. Inverse polarization raytracing: estimating surface shapes of transparent objects. *Proc. IEEE Conf. on Computer Vision & Pattern Recognition* 2 (2005), 910–917.
- Nigel JW Morris and Kiriakos N. Kutulakos. 2007. Reconstructing the surface of inhomogeneous transparent scenes by scatter-trace photography. *Proc. Int. Conf. on Computer Vision* (2007), 1–8.
- Nigel JW Morris and Kiriakos N. Kutulakos. 2011. Dynamic refraction stereo. *IEEE Trans. Pattern Analysis & Machine Intelligence* 33, 8 (2011), 1518–1531.
- Pieter Peers and Philip Dutré. 2003. Wavelet environment matting. *Proc. Eurographics Workshop on Rendering* (2003), 157–166.
- Yiming Qian, Minglun Gong, and Yee Hong Yang. 2016. 3D Reconstruction of Transparent Objects with Position-Normal Consistency. *Proc. IEEE Conf. on Computer Vision & Pattern Recognition* (2016), 4369–4377.
- Yiming Qian, Minglun Gong, and Yee-Hong Yang. 2015. Frequency-based environment matting by compressive sensing. *Proc. Int. Conf. on Computer Vision* (2015), 3532–3540.
- Yiming Qian, Minglun Gong, and Yee-Hong Yang. 2017. Stereo-Based 3D Reconstruction of Dynamic Fluid Surfaces by Global Optimization. *Proc. IEEE Conf. on Computer Vision & Pattern Recognition* (2017), 6650–6659.
- Yuliy Schwartzburg, Romain Testuz, Andrea Tagliasacchi, and Mark Pauly. 2014. High-contrast Computational Caustic Design. *ACM Trans. on Graphics (Proc. of SIGGRAPH)* 33, 4 (2014), 74:1–74:11.
- Qi Shan, Sameer Agarwal, and Brian Curless. 2012. Refractive height fields from single and multiple images. *Proc. IEEE Conf. on Computer Vision & Pattern Recognition* (2012), 286–293.
- Kenichiro Tanaka, Yasuhiro Mukaigawa, Hiroyuki Kubo, Yasuyuki Matsushita, and Yasushi Yagi. 2016. Recovering Transparent Shape from Time-of-Flight Distortion. *Proc. IEEE Conf. on Computer Vision & Pattern Recognition* (2016), 4387–4395.
- Borislav Trifonov, Derek Bradley, and Wolfgang Heidrich. 2006. Tomographic reconstruction of transparent objects. *Proc. Eurographics Conf. on Rendering Techniques* (2006), 51–60.
- Chia-Yin Tsai, Ashok Veeraraghavan, and Aswin C Sankaranarayanan. 2015. What does a single light-ray reveal about a transparent object? *Proc. IEEE Int. Conf. on Image Processing* (2015), 606–610.
- Gordon Wetzstein, David Roodnick, Wolfgang Heidrich, and Ramesh Raskar. 2011. Refractive shape from light field distortion. *Proc. Int. Conf. on Computer Vision* (2011), 1180–1186.
- Yonatan Wexler, Andrew W. Fitzgibbon, and Andrew Zisserman. 2002. Image-based Environment Matting. *Proc. Eurographics Workshop on Rendering* (2002), 279–290.
- Shihao Wu, Hui Huang, Minglun Gong, Matthias Zwicker, and Daniel Cohen-Or. 2015a. Deep Points Consolidation. *ACM Trans. on Graphics (Proc. of SIGGRAPH Asia)* 34, 6 (2015), 176:1–176:13.
- Shihao Wu, Wei Sun, Pinxin Long, Hui Huang, Daniel Cohen-Or, Minglun Gong, Oliver Deussen, and Baoguan Chen. 2014. Quality-driven Poisson-guided Autoscanning. *ACM Trans. on Graphics (Proc. of SIGGRAPH Asia)* 33, 6 (2014), 203:1–203:12.
- Zhaohui Wu, Zhong Zhou, Delei Tian, and Wei Wu. 2015b. Reconstruction of Three-dimensional Flame with Color Temperature. *The Visual Computer* 31, 5 (2015), 613–625.
- Sai-Kit Yeung, Tai-Pang Wu, Chi-Keung Tang, Tony F Chan, and Stanley Osher. 2011. Adequate reconstruction of transparent objects on a shoestring budget. *Proc. IEEE Conf. on Computer Vision & Pattern Recognition* (2011), 2513–2520.
- Yonghao Yue, Kei Iwasaki, Bing-Yu Chen, Yoshinori Dobashi, and Tomoyuki Nishita. 2014. Poisson-Based Continuous Surface Generation for Goal-Based Caustics. *ACM Trans. on Graphics* 33, 3 (2014), 31:1–31:7.
- Mingjie Zhang, Xing Lin, Mohit Gupta, Jinli Suo, and Qionghai Dai. 2014. Recovering Scene Geometry under Wavy Fluid via Distortion and Defocus Analysis. *Proc. Euro. Conf. on Computer Vision* (2014), 234–250.
- Zhengyou Zhang. 2000. A Flexible New Technique for Camera Calibration. *IEEE Trans. Pattern Analysis & Machine Intelligence* 22, 11 (2000), 1330–1334.
- Ciyou Zhu, Richard H. Byrd, Peihuang Lu, and Jorge Nocedal. 1997. Algorithm 778: L-BFGS-B: Fortran Subroutines for Large-scale Bound-constrained Optimization. *ACM Trans. Mathematical Software* 23, 4 (1997), 550–560.
- Douglas E. Zongker, Dawn M. Werner, Brian Curless, and David H. Salesin. 1999. Environment Matting and Compositing. *ACM Trans. on Graphics (Proc. of SIGGRAPH)* (1999), 205–214.
- Xinxin Zuo, Chao Du, Sen Wang, Jiangbin Zheng, and Ruigang Yang. 2015. Interactive visual hull refinement for specular and transparent object surface reconstruction. *Proc. Int. Conf. on Computer Vision* (2015), 2237–2245.