

# What Does Motion Reveal About Transparency? \*

Moshe Ben-Ezra and Shree K. Nayar  
Computer Science Department, Columbia University  
New York, NY, USA  
E-mail: {moshe, nayar}@cs.columbia.edu

## Abstract

The perception of transparent objects from images is known to be a very hard problem in vision. Given a single image, it is difficult to even detect the presence of transparent objects in the scene. In this paper, we explore what can be said about transparent objects by a moving observer. We show how features that are imaged through a transparent object behave differently from those that are rigidly attached to the scene. We present a novel model-based approach to recover the shapes and the poses of transparent objects from known motion. The objects can be complex in that they may be composed of multiple layers with different refractive indices. We have conducted numerous simulations to verify the practical feasibility of our algorithm. We have applied it to real scenes that include transparent objects and recovered the shapes of the objects with high accuracy.

## 1 Why is Transparency Hard?

The perception of transparency is a hard and important vision problem that has not received much attention. Transparent objects violate most of the fundamental assumptions made by vision algorithms. For instance, they cause the projection of a three-dimensional scene to the image plane to not be perspective. Furthermore, this projection can vary from one viewpoint to the next.

As seen from Figure 1(a), a single view does not even determine with certainty whether the scene includes transparent objects within it, leave alone reveal the geometric properties of the transparent objects. The circular shape in Figure 1(a) may be a transparent sphere located in front of a painting, or it may be a part of the painting itself. This ambiguity arises from the fact that a transparent object does not have features of its own. It simply maps features that exist in its environment onto the image plane. This mapping is complex to say the least; even for a simple shape the mapping of a real feature to the image through a transparent object involves complex interactions between rays from the real feature, the surfaces of the transparent object, and the viewpoint of the observer.

Several researchers have addressed issues related to transparency in vision and graphics. Zongker et al. [14] and Chuang et al. [3] presented a method called *environment matting* for capturing the optical behavior of transparent objects from known and controlled backgrounds for rendering

\*This work was supported by an NSF ITR Award IIS-00-85864 "Interacting with the Visual World: Capturing, Understanding, and Predicting Appearance".



**Figure 1:** (a) A single view of a scene does not determine with certainty whether the scene includes transparent objects. The circular region in this image could be a transparent sphere placed in front of the painting or it could be a part of the painting itself. (b) This image is also of the scene in (a) but taken from a different viewpoint. The way in which the background changes (within the circular region as well as outside it) between (a) and (b) reveals that there is indeed a transparent sphere placed in front of the painting

and compositing purposes. Wexler et al. [13] have extended this idea to obtain the environment matting model from uncontrolled backgrounds. Matusik et al. [8] use environment mattes obtained from multiple viewpoints to create novel views by interpolation. Note that these methods do not address the problem of explicit shape and pose estimation; in fact, they have been designed to avoid this problem. An interesting approach was proposed by Murase [9], where the shape of the surface of pool of water is recovered from the way it refracts the texture of the bottom of the pool. More recently, Saito et al. [12] proposed a very novel method for recovering the surface of a transparent object using polarized light. Hata et al. [5] used structured light and genetic algorithms to find the shapes of transparent objects such as paste drops. Other controlled methods, such as ones that use hologram laser lighting, have also been suggested [11, 2]. A similar problem, addressed in photogrammetry, arises when measuring objects through transparent multimedia, usually objects immersed in liquids [7].

In this paper, we explore what can be said about transparent objects by a moving observer. Figure 1(b) is another image of the scene in Figure 1(a), taken from a different viewpoint. The manner in which the background changes (within the circular region as well as outside it) with viewpoint confirms that there is indeed a transparent sphere placed in front of the painting. We show that it is possible to estimate the shapes of transparent objects immersed in an environment of unknown structure from a sequence of images taken during known camera motion. Our algorithm is a model-based

one in that it assumes a parametric form for the shapes of the transparent objects, and estimates the shape parameters from the motion of features within the image of the object. Since the parametric model used is not restricted to any particular form, the algorithm can be used for a wide class of shapes.

We have used simulations to extensively test the performance of algorithm. In addition, we have conducted several experiments with real objects. The objects we have used in our simulations and experiments have simple shapes. It is important to note, however, that the even for a very simple shape the underlying problem is hard as it involves highly non-linear interactions between light rays and the object surfaces. The most difficult case we have tried is the estimation of the shape parameters and the pose of a pipe filled with water; this is a very complex case as each light ray is subjected to four refractions through three media with different refractive indices. We view the results of this paper as an important initial step towards developing a general framework for the perception of transparency.

## 2 The Physics of Transparency

Figure 2 shows a transparent object with refractive index  $\mu_2$  immersed in a transparent medium with refractive index  $\mu_1$ , where  $\mu_1 < \mu_2$ . A ray of light that passes through the transparent object interacts with it at two interfaces, upon entrance and exit. At the first interface, a fraction of the ray is reflected and another fraction is refracted<sup>1</sup>. The angles  $(\theta_1, \theta_2)$  between the incident and the reflected rays relative to the normal  $N_1$  are equal, and the angles  $(\theta_1, \theta_3)$  of the incident and refracted rays relative to  $N_1$  are given by Snell's law:

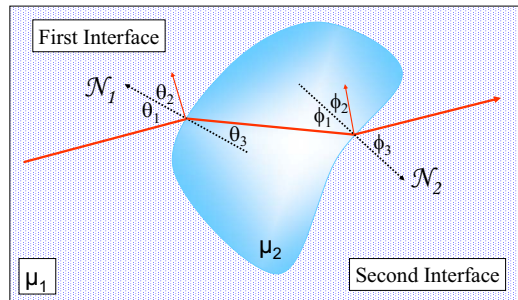
$$\mu_1 \sin \theta_1 = \mu_2 \sin \theta_3. \quad (1)$$

The same relations hold for the second interface as well. For  $\mu_1 < \mu_2$  there exists a value  $\alpha = \sin^{-1} \frac{\mu_1}{\mu_2}$  which is called the *critical angle*. When  $\phi_1 > \alpha$  a complete reflection of the ray occurs at the second interface (the ray does not pass through the interface), a condition known as *total internal reflection*. It is worth noting that the complete path of light from the source to the observer (including the incident ray, refracted ray at the first interface and the refracted ray at the second interface) is not restricted to lie on a plane.

## 3 Real and Virtual Features

Classical vision algorithms, such as stereo and structure from motion, are based on the assumption that image features correspond to scene features that are rigidly attached

<sup>1</sup>As the ray passes through the object, some fraction of its energy may be absorbed as well. The exact fractions that are reflected, refracted and absorbed depend on the Fresnel coefficients [1] and the absorption coefficient of the object. These fractions are not important as long as the ray has sufficient energy left in it when it leaves the object at the second interface. By sufficient energy we mean that the scene point it represents produces a detectable brightness in the image. All we require in our approach is that at least some of features in the background can be detected in the image after they pass through the object.



**Figure 2:** A ray of light interacts with a transparent object at two interfaces. The first interaction occurs when the ray passes from the surrounding medium having a refractive index of  $\mu_1$  into the transparent object having a refractive index  $\mu_2$ . The second interaction occurs when the ray passes from the transparent object back into the surrounding medium. At each interface a portion of the ray is reflected and another portion is refracted.

to the scene. Figure 3(a) shows an example of such a “real” feature located on the surface of a diffuse object. In this case, we know that it is possible to determine the actual position of the feature from multiple views. In the real world, the assumption of real features does not always hold. For example, objects with highly specular surfaces such as the one shown in Figure 3(b) reflect features that are located elsewhere in the scene. Such reflected features, which are referred to as *virtual features* [10], are not faithful to the reflecting surface; their apparent location on the reflecting surface varies with the viewpoint of the observer/camera<sup>2</sup>.

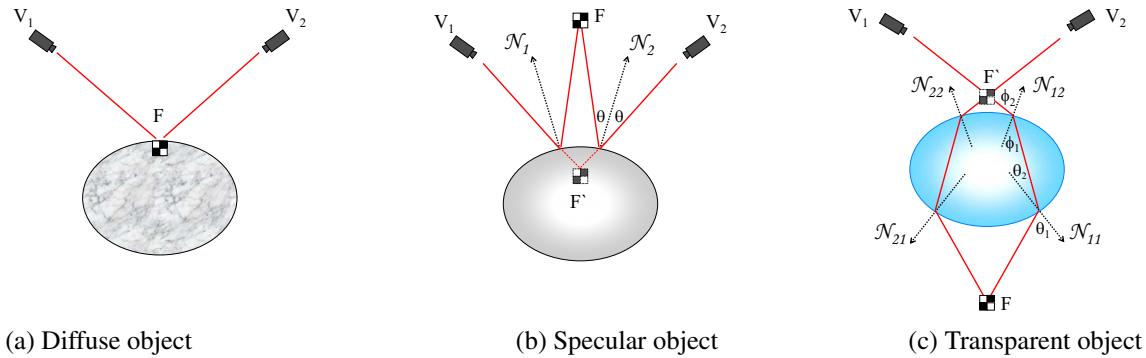
Note that the creation of virtual features by a single specular object involves only one reflecting surface. Furthermore, the incident ray, the reflected ray and the surface normal at the reflecting point are always coplanar. It is well known that specular surfaces pose a very hard problem for structure recovery algorithms [10]. It is worth noting that transparent objects are even more confounding for vision algorithms as the physical phenomena involved are more complex. This is illustrated in Figure 3(c), where we see that two surfaces (interfaces) participate in formation of a virtual feature<sup>3</sup>. The angles of the incident and refracted rays in this case are determined by the non-linear Snell's law (Eq. 1). In practice, handling transparency gets even more challenging due to the simultaneous occurrence of reflection and refraction, total internal reflection, and wavelength related effects (chromatic aberrations) caused by the refractive index of the object.

## 4 Transparency from Motion

Our goal is to develop an algorithm for recovering the shapes and poses of transparent objects from motion. The known

<sup>2</sup>In the case of a specular surface, if one applies stereo to consecutive views in a motion sequence to compute the location of the feature, one obtains a locus of feature locations referred to as a *catacaustic* [4].

<sup>3</sup>In the case of transparency, the locus of the locations of the virtual feature that results from the motion of the observer is called a *diacaustic* [4].



**Figure 3:** (a) Features that are located on the face of a diffuse object (such as the feature  $F$ ) are *real features*. Real features are inherent attributes of the object itself and remain stationary with respect to the object while the viewpoint of the observer changes. Therefore, the location in the scene of such a feature can be determined from two views of the object. In contrast, features that are reflected from the surface of a specular object such as the one in (b), or viewed through a transparent object as in (c), are *virtual features*. They are not faithful to the objects that produce them in that their apparent locations can vary with the viewpoint of the observer. While specular reflection involves a single surface, transparency involves at least two interfaces. Hence, the virtual features created by a transparent object have more complex properties than those produced by a specular object.

and unknown factors in our setting can be summarized as follows (see Figure 4):

1. The shape as well as the pose (position and orientation) of the transparent objects are unknown.
2. The structure of the background is unknown, however it is assumed that the background is far away (relative to the sizes of the transparent objects and their distances from the observer).
3. Since the background is far, a feature (such as  $F$  in Figure 4) is seen in the same direction  $\mathbf{d}$  by all points on the surface of the transparent object. The direction  $\mathbf{d}$  itself is unknown.
4. The general parametric form (model) of the transparent object is known, as well as its refractive index. Note that the refractive indices of materials like glass and plastic are very similar and one can use reasonable estimates using existing tables [6].
5. The intrinsic parameters and the motion of the camera are known<sup>4</sup>. Hence, the location and direction in 3D of each imaged ray can be determined.
6. All the imaged rays associated with any given feature are determined by tracking features in the acquired image sequence.

Our recovery problem can be stated more formally as:

1. Let  $S$  be a set of  $k$  true features  $F_1 \dots F_k$ , where  $F_i$  is associated with an unknown direction  $\mathbf{d}_i$ .
2. Let  $O = O(\xi)$  be a transparent object parameterized by the shape parameter vector  $\xi$ . Its orientation and position are represented using the rotation matrix  $\mathbf{R}$  and the translation vector  $\mathbf{T}$ .
3. Let  $V$  be the set of  $n$  rays that are captured by the camera. This set includes subsets of rays, each associated with a single feature.

<sup>4</sup>In principle, the motion of the camera can be computed by applying an egomotion algorithm to the “background” features. Here, we will simply assume that it is known.

4. Given  $V$ , find the transparent object’s shape  $\xi$ , its pose  $\mathbf{R}, \mathbf{T}$  and the feature directions  $\mathbf{d}_i$ .

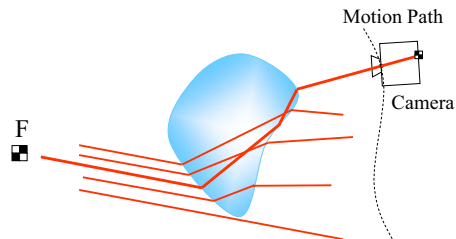
## 5 Finding Shape and Pose

From Figure 4 we see that, since we assume that all features are distant, all rays that are associated with the same feature are parallel to each other before they are scattered by the transparent object. Conversely, if the optical system we are dealing with were fully known (which of course is not the case), the rays measured by the camera, if traced back through the transparent object, would produce a set of parallel rays. It is important to note that these reversed rays will be parallel regardless of the complexity of the optical system; that is, the system could include any number of refracting layers with different refractive indices. Also, the reversed rays will be parallel irrespective of their individual paths; they may undergo complex total internal reflections or, for that matter, not even pass through the transparent object. It is worth noting that the reversal of rays requires the explicit use of the physics of transparency presented in section 2 as well as the object’s shape and pose.

The above observation gives us a powerful means for formulating our problem; *although the interaction between transparency and motion is very complex, the set of reversed rays associated with each tracked feature must be parallel*. Given such a set of reversed rays (corresponding to a single feature), its degree of parallelism can be measured by computing the variance of the angles between the rays and the average direction of the set of rays. An objective function that measures the goodness of an object’s shape and pose, given a set of tracked features, can be written as:

$$f(\xi, \mathbf{R}, \mathbf{T}) = \sum_{i=1}^k \text{var}(o_i(\xi, \mathbf{R}, \mathbf{T})), \quad (2)$$

where  $o_i$  is the set of the directions of the reversed rays associated with the feature  $F_i$ . The shape and pose of the object



**Figure 4:** The set of parallel incident rays from a distant feature  $F$  are scattered by a transparent object in various directions. Some of these rays are captured by a moving camera, when they pass through the optical center of the camera. Given the intrinsic parameters and motion of the camera, we can compute the location (in 3D) of each captured ray in space. Note that if we knew the shape and pose of the object, these rays could be traced backwards to form a set of (outgoing) parallel rays. This holds true regardless of the complexity of the transparent object. It also remains valid for rays that go through total internal reflections (like the one imaged by the camera) as well as rays that do not pass through the transparent object (such as the bottom most one).

are then determined as the ones that minimize the above objective function:

$$\arg \min_{\xi, \mathbf{R}, \mathbf{T}} f(\xi, \mathbf{R}, \mathbf{T}). \quad (3)$$

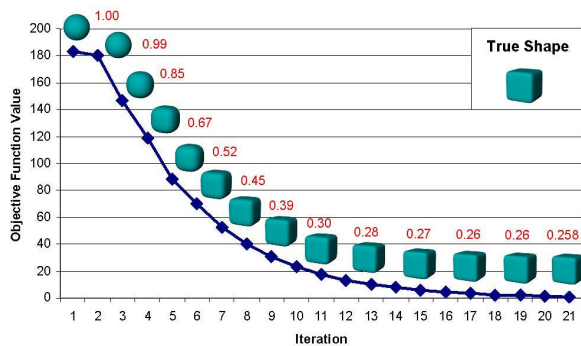
Note that, in general, this is a multi-parameter, non-linear optimization problem. However, it can be efficiently solved when a reasonable initial guess  $(\xi_0, \mathbf{R}_0, \mathbf{T}_0)$  for the unknown parameters is provided. While a variety of non-linear optimization algorithms can be used, in this paper we have implemented a very simple steepest descent approach.

Once a solution is found, the true directions  $\mathbf{d}_i$  of the features are also determined - they are simply the directions of the reversed rays. In our discussion we have assumed that the features are distant, thereby eliminating the need to account for the exact depths of the features. It is worth noting that our approach to the problem is general and that this distant feature assumption can be replaced by a variety of other constraints based on the peculiarities of the application to which the method is applied. For instance, in some settings all the features may be forced to lie on a plane or a sphere<sup>5</sup>. In each of these special settings, the objective function should be appropriately modified to fully exploit the constraints of the setting.

## 6 Validation using Simulation

In order to verify our approach, we have conducted extensive simulation tests with a variety of shape models. A few of our simulation results are reported in this section. In each of our tests, we have used only 4 features and 21 captured rays for each feature (i.e. an image sequence with 21 frames).

<sup>5</sup>Such assumptions are particularly easy to satisfy in the case of structured environments, where the system can include known background surfaces that are placed around the transparent object of interest.



**Figure 5:** Simulation test that shows the recovery of the parameters of a transparent superquadric. The value of the objective function is plotted as a function of the iteration number. The numbers next to the shapes are the shape parameter values where 1.0 is a sphere and 0.25 is a cube with rounded corners, which is the ground truth in this case.

In several of our simulations, we have used superquadrics to model the shape of the transparent object. Superquadrics are popular as they nicely subsume a wide variety of simple shapes. The superquadric ellipsoid is given by:

$$s_{e,n}(x, y, z) = \left( |x|^{\frac{e}{n}} + |y|^{\frac{e}{n}} \right)^{\frac{n}{e}} + |z|^{\frac{2}{n}} = 1, \quad (4)$$

where  $e$  and  $n$  are the parameters that control the shape. In the case of a symmetric superquadric,  $n = e$ . While superquadrics are relatively simple shapes, it is important to recognize how complex the underlying recovery problem is. The physics of transparency turns even this seemingly easy setting into a difficult non-linear problem; the rays used to compute the shape can go through multiple refractions, total internal reflection, or not pass through the transparent object at all.

Figure 5 illustrates how our simulations are conducted. In this case, the ground truth shape is a cube with slightly rounded corners ( $n = 0.25$ ) and the initial guess is a sphere ( $n = 1$ ), which is a natural initial guess for the class of superquadrics. We see that the algorithm is successful in converging from the initial guess to the desired final shape in less than 20 gradient-descent iterations.

In our simulations, for each object, we have assumed different settings of the recovery problem; known shape and unknown pose, unknown shape and known pose, and unknown shape and pose. Our results for a few different objects are shown in Figure 6. We see that accurate values for shape and/or pose were obtained in all cases except for the case of the bi-convex lens. In this case, there is a slight error in the computed pose which results from the fact that horizontal translation and vertical rotation produce similar motion fields. When the parameters are not tightly coupled, the convergence is fast. In particular, finding the shape for a known pose is very efficient in all cases.

The example shown in the fourth row of Figure 6 is particularly challenging. Here, the model is that of a circular glass pipe filled with water. In this case, each ray must pass through four interfaces and three media with different re-



fractive indices (including air outside the pipe). In this test, the model used is also that of a circular pipe, except that the parameters of model (diameters of the water column and the pipe) are unknown. This is a really difficult setting and yet the algorithm is able to recover all the unknown parameters accurately.

## 7 Real Experiments

We have conducted several experiments with transparent objects in complex scenes. In each case, a digital camera was used to take 9 views of the scene. While the camera motion and the parametric model of the transparent object were known, the location and shape parameters of the object as well as the structure of the scene were not known. After the images were taken, about ten features in each image were manually tagged<sup>6</sup>. It is important to note that the features we not restricted to lie within the transparent object; features within and outside the transparent object were included. The rays in 3D space that are associated with each feature for each camera position were computed using the known camera parameters. Finally, the shape and pose of the objects were recovered using our method with very rough initial guesses. While we could measure the physical dimensions of the objects to obtain ground truth shape parameters, there was no easy way to obtain the actual pose parameters.

The results of the tests are shown in Figure 7 through Figure 10 and Table 1. In the case of the plano-convex lens (Figure 8), we see noticeable errors in the shape results. This is probably because of strong coupling between the pose and shape parameters in this case. In all other cases, fairly precise estimates of shape were obtained. In the difficult case of the water-filled pipe (Figure 9), the results were very accurate - the errors are less than 1mm for both the diameter of the pipe and its thickness. We believe that this is because the image of the water-filled pipe was larger and therefore the feature positions were more accurate.

**Table 1: Summary of experimental results for sphere, plano-convex lens, pipe filled with water and superquadric. Computed shape and pose parameters as well as ground truth shape parameters are included.**

<b>Sphere</b>	True value	Initial	Final	Error
<b>Diam. (in)</b>	3.0	8	3.18	0.18
<b>Pos. (mm)</b>	Unknown	(0, 0, 0)	(66, 0.8, -0.7)	N/A
<b>PCX lens</b>	True value	Initial	Final	Error
<b>F. Len. (mm)</b>	500	150	371	129
<b>Pos. (mm)</b>	Unknown	(0, 0, 0)	(95, 9.5, 37.9)	N/A
<b>Pipe</b>	True value	Initial	Final	Error
<b>Diam. (mm)</b>	117	200	116.1	0.9
<b>Thick. (mm)</b>	3	20	2.3	0.7
<b>Pos. (mm)</b>	Unknown	(0, 0, 0)	(43, -3.0, 0)	N/A
<b>S. Quad</b>	True value	Initial	Final	Error
<b>n</b>	Unknown	1.0	0.18	N/A
<b>Pos. (mm)</b>	Unknown	(0, 0, 0)	(8, 0.1, 0.4)	N/A











































<sup>6</sup>Since we do not require all the features to lie inside the transparent object, the tracked features could well have been found using any one of the popular tracking algorithms. We did the manual selection to ensure that we had precise feature locations

## 8 Conclusion

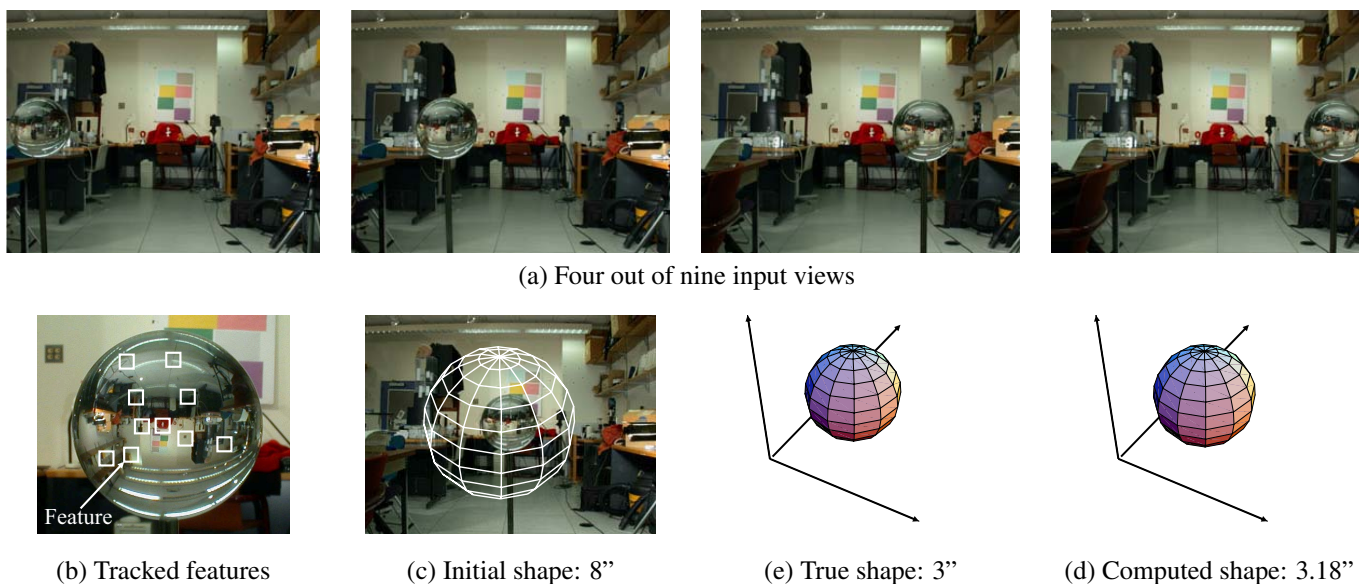
The perception of transparency is a fundamental and difficult problem in computer vision. We have shown that a vision system can use motion to recover the properties of transparent objects in a scene. We developed a model-based algorithm for the recovery of a object shape and pose from motion. We have extensively tested our algorithm via simulations as well as real experiments. These tests indicate that, at least for relatively simple geometries, motion can be used to robustly recover the properties of the object. We also applied our approach to a more complex case (pipe filled with water) which involves four material interfaces and three refractive indices. While we have discussed the problem of transparency in a general setting, our results can be immediately exploited in several applications domains such as graphics rendering and visual inspection.

## References

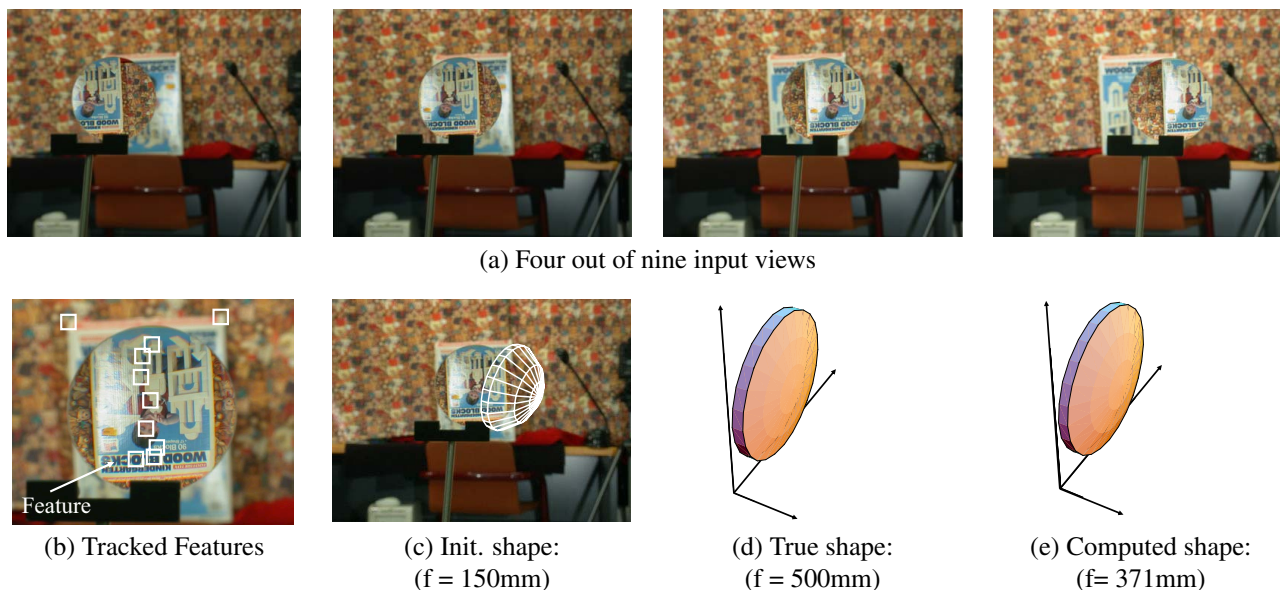
- [1] Max Born and Emil Wolf. *Principles of Optics*. Cambridge University Press, 7th edition, 1999.
- [2] L.A. Borynyak, O.N. Zakharov, A.V. Loginov, and P.M. Mednis. Determination of the surface shape of deformed bodies using holographic interferometers with an adaptive base. *Avtometriya*, page 48, 1997.
- [3] Yung-Yu Chang, D.E. Zongker, J. Hindorff, B. Curless, D.H. Salesin, and R. Szeliski. Environment matting extensions: towards higher accuracy and real-time capture. In *Proceedings of SIGGRAPH'00*, page 121, 2000.
- [4] S. Cornbleet. *Microwave and Geometric Optics*. Academic Press, London, 1994.
- [5] S. Hata, Y. Saitoh, S. Kumamura, and K. Kaida. Shape extraction of transparent object using genetic algorithm. *Proceedings of ICPR'96*, page 684, 1996.
- [6] David R. Lide. *Handbook of Chemistry and Physics*. CRC Press, 2002.
- [7] H.G. Maas. New developments in multimedia photogrammetry. *Optical 3-D Measurement Techniques III (Eds.: A. Grün, H. Kahmen)*, Wichmann Verlag, Karlsruhe, 1995.
- [8] W. Matusik, H. Pfister, R. Ziegler, A. Ngan, and L. McMillan. Acquisition and rendering of transparent and refractive objects. In *13th Eurographics Workshop on Rendering*, page 267, 2002.
- [9] H. Murase. Surface shape reconstruction of a nonrigid transparent object using refraction and motion. *PAMI*, 14(10):1045–1052, October 1992.
- [10] M. Oren and S.K. Nayar. A theory of specular surface geometry. *IJCV*, 24(2):105–124, September 1997.
- [11] Young Kee Ryu, Choonsuk Oh, and Jong-Seul Lim. Development of a noncontact optical sensor for measuring the shape of a surface and thickness of transparent objects. *Optical Engineering*, 40:500, 2001.
- [12] M. Saito, Yoichi Sato, Katsushi Ikeuchi, and H. Kashiwagi. Measurement of surface orientations of transparent objects using polarization in highlight. In *Proceedings of CVPR'99*, volume 1, pages 381 – 386, June 1999.
- [13] Y. Wexler, W. Fitzgibbon, and A. Zisserman. Image-based environment matting. In *13th Eurographics Workshop on Rendering*, page 279, 2002.
- [14] D.E. Zongker, D.M. Werner, B. Curless, and D.H. Salesin. Environment matting and compositing. In *Proceedings of SIGGRAPH 99*, page 205, 1999.

SPHERE		Shape Test (Iter: < 20)			Pose Test (Iter: < 20)			Shape+Pose Test (Iter: < 40)		
Param.	True Val.	Initial	Final	Error	Initial	Final	Error	Initial	Final	Error
Diam. (mm)	160	80	159.914	-0.086	NA	NA	NA	80	159.931	-0.069
T.Opt (mm)	-40	NA	NA	NA	0	-40.065	-0.065	0	-40.0107	-0.0107
T.Par (mm)	20	NA	NA	NA	0	19.986	-0.014	0	19.9313	-0.0687
Top View										
Note the very fast convergence and accuracy of the recovered pose and shape.										
BI-CONVEX LENS		Shape Test (Iter: < 20)			Pose Test (Iter: 100+)			Shape+Pose Test (Iter: 500+)		
Param.	True Val.	Initial	Final	Error	Initial	Final	Error	Initial	Final	Error
F.Len. (mm)	80	160	80.0095	0.0095	NA	NA	NA	160	80.0304	0.0304
T.Opt (mm)	20	NA	NA	NA	0	19.0768	-0.9232	0	19.48	-0.52
T.Par (mm)	10	NA	NA	NA	0	12.321	2.321*	0	12.47	2.47*
R.Ver (deg)	5	NA	NA	NA	0	5.628	0.628*	0	5.29	0.29*
Top View										
* Pose (only) error due to the ambiguity between horizontal translation and vertical rotation.										
BI-CONCAVE LENS		Shape Test (Iter: < 10)			Pose Test (Iter: < 40)			Shape+Pose Test (Iter: 500+)		
Param.	True Val.	Initial	Final	Error	Initial	Final	Error	Initial	Final	Error
F.Len. (mm)	80	160	80.024	0.024	NA	NA	NA	160	79.6464	-0.3536
T.Opt (mm)	5	NA	NA	NA	0	5.010	0.010	0	4.9	-0.1
T.Par (mm)	5	NA	NA	NA	0	4.764	-0.236	0	4.33	-0.67
R.Ver (deg)	2	NA	NA	NA	0	2.102	0.102	0	1.279	-0.721
Top View										
WATER-FILLED PIPE*		Shape Test (Iter: < 20)			Pose Test (Iter: < 20)			Shape+Pose Test (Iter: 500+)		
Param.	True Val.	Initial	Final	Error	Initial	Final	Error	Initial	Final	Error
Diam. (mm)	80	90	80.028	0.028	NA	NA	NA	82*	80.023	0.023
Thick. (mm)	15	20	15.0048	0.0048	NA	NA	NA	17	15.013	0.013
T.Opt (mm)	10	NA	NA	NA	0	9.9338	-0.0662	0	9.464	-0.536
T.Par (mm)	5	NA	NA	NA	0	5.0173	0.0173	0	5.019	0.019
Top View										
* Note that this test involves 4 interfaces and 3 different refractive indices. ** Close initial guess was required in this case due to ambiguity between Diam. and T.Opt.										
SUP.QUAD. ELLIPSE		Shape Test (Iter: < 20)			Pose Test (Iter: < 40)			Shape+Pose Test (Iter: < 100*)		
Param.	True Val.	Initial	Final	Error	Initial	Final	Error	Initial	Final	Error
n	0.25	1.0	0.2521	0.0021	NA	NA	NA	1.0	0.2499	0.0001
T Opt (mm)	10	NA	NA	NA	0	9.952	0.048	0	9.957	0.043
T.Par (mm)	5	NA	NA	NA	0	4.991	0.009	0	4.997	0.003
R Ver (deg)	5	NA	NA	NA	0	5.018	0.018	0	5.014	0.014
Top View										
* Note the fast convergence in this case, similar to the sphere test.										
PIECEWISE SUP.QUAD *		Shape Test (Iter: < 40)			Pose Test (Iter: < 40)			Shape+Pose Test (Iter: < 100)		
Param.	True Val.	Initial	Final	Error	Initial	Final	Error	Initial	Final	Error
n - First	0.6666	1.0	0.6656	0.0008	NA	NA	NA	1.0	0.6628	-0.0038
n - Second	0.3333	1.0	0.3339	0.0006	NA	NA	NA	1.0	0.3372	0.0039
T Opt (mm)	10	NA	NA	NA	0	9.954	-0.046	0	10.3304	0.3304
T.Par (mm)	5	NA	NA	NA	0	4.987	-0.013	0	5.0253	0.0253
R Ver (deg)	5	NA	NA	NA	0	5.010	0.01	0	4.9868	-0.0132
Top View										
* Note: Asymmetric shape with different parameters for first and second interfaces.										

**Figure 6: Simulation test results for sphere, bi-convex lens, bi-concave lens, circular pipe filled with water (four interfaces), and superquadrics. *T.Opt* is translation along the optical axis, *T.par* is horizontal translation parallel to the image plane, *R.Ver* is rotation about the vertical axis (intentionally selected to test coupling with *T.Par*), *Diam.* and *Thick.* are the diameter and the thickness of the pipe, and *n* is the superquadric shape parameter. In each case, the algorithm successfully estimates the shape and/or pose parameters.**



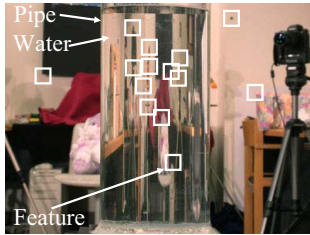
**Figure 7: Sphere:** (a) Some of the views used as input. (b) The features used by the algorithm. (c) The initial guess overlaid on one of the images. (d) The true shape and (e) the recovered shape shown at the same scale. The recovered sphere is slightly larger than the actual one. This experiment may appear to be a relatively trivial one as the initial shape is the same as the actual one. It was done as an initial test as even the estimation of the position and diameter of the sphere turns out to be a highly non-linear problem.



**Figure 8: Plano-Convex Lens:** (a) Few of the views used as input. (b) The features used by the algorithm. Note that some of the features lie outside the lens. (c) The initial guess overlaid on one of the images. (d) The true shape and (e) the recovered shape. The initial guess in this case is quite far from the actual one – the curvatures of the surfaces of the lens are very different. Although the lens has a simple geometry, there is strong coupling between pose and shape parameters in this case. This results in noticeable errors in the computed focal length (even though the computed and actual shapes are quite close).



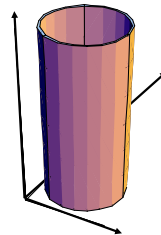
(a) Four out of nine input views



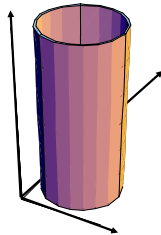
(b) Tracked features



(c) Init. shape:  
(D=200, T=20)



(d) True shape:  
(D=117, T=3)

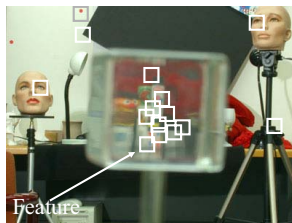


(e) Computed Shape:  
(D=116.1, T=2.3)

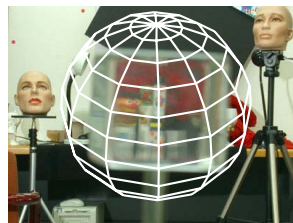
**Figure 9: Acrylic Pipe filled with water (4 interfaces):** (a) Few of the views used as input. (b) The features used by the algorithm. (c) The initial guess overlaid on one of the images. (d) The true shape and (e) the recovered shape. Even though this is a very complex case, the results are very accurate. “D” and “T” denote the outer diameter and thickness, respectively, of the acrylic pipe. The refractive indices of the acrylic and the water were taken to be 1.49 and 1.33, respectively.



(a) Four out of nine input views.



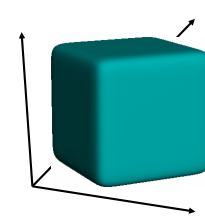
(b) Tracked features



(c) Initial guess:  
(n = 1.0)



(d) True shape:  
(n=Unknown)



(e) Computed shape:  
(n=0.18)

**Figure 10: Superquadric (cube with rounded edges):** (a) Few of the views used as input. (b) The features used by the algorithm. (c) The initial guess overlaid on one of the images. (d) The true shape and (e) the recovered shape. In this case, not only are the physical interactions non-linear but also the shape itself. The algorithm successfully recovers the cube even though the initial guess is a sphere.