Simultaneously Reconstructing Transparent and Opaque Surfaces from Texture Images

Mohamad Ivan Fanany and Itsuo Kumazawa

Imaging Science and Engineering Labs., Tokyo Institute of Technology
fanany.m.aa@m.titech.ac.jp

Abstract. This paper addresses the problem of reconstructing non-overlapping transparent and opaque surfaces from multiple view images. The reconstruction is attained through progressive refinement of an initial 3D shape by minimizing the error between the images of the object and the initial 3D shape. The challenge is to simultaneously reconstruct both the transparent and opaque surfaces given only a limited number of images. Any refinement methods can theoretically be applied if analytic relation between pixel value in the training images and vertices position of the initial 3D shape is known. This paper investigates such analytic relations for reconstructing opaque and transparent surfaces. The analytic relation for opaque surface follows diffuse reflection model, whereas for transparent surface follows ray tracing model. However, both relations can be converged for reconstruction both surfaces into texture mapping model. To improve the reconstruction results several strategies including regularization, hierarchical learning, and simulated annealing are investigated.

1 Introduction

Current advances in computer graphics and imaging techniques enables a very realistic visualization of a transparent object. Visualization of the transparent object in the absence of a 3D shape model, however, generally needs a very large number of view images (up to several tenth thousands images, see [1]). Such requirement certainly demands a complicated image acquisition system and high expertise in photography. Even a notably technique to accurately acquire light reflection and refraction under a fixed single viewpoint without a 3D model also demands a large number of backdrop and sidedrop images [2]. In either case, the parameters to render the interpolated object's images from arbitrary viewpoint are either hard to compute or unavailable. Beyond the task to only visualize such transparent object, we pursue a shape refinement system that will also recover its three-dimensional shape. The object may contain non-overlapping transparent and opaque surfaces. The challenge is to recover these surfaces simultaneously given only a limited number of views.

Proceedings of the 5th International Conference on Computer Vision Systems (ICVS 2007)
Published in 2007 by Applied Computer Science Group, Bielefeld University, Germany, ISBN 978-3-00-020933-8

2 Previous Work

Many methods acquire high quality 3D shape of opaque object with a diffuse surface [3], but still not many methods acquire 3D shape of transparent object. This is because the perception of transparent surface is a hard vision problem. Transparent surface lacks of body and surface reflections, is suffered much from inter-reflection [4], and lacks of naturally-occurring shape. The only potential sources of surface information are specular highlights, environmental reflections, and refractive distortion, whereas depth information is almost completely unavailable [5]. Only recently, some prospective techniques for modeling transparent surface have emerged. We categorize these methods into two groups as follows.

The first group elaborates as much the surface related features as possible to explicitly define the surface's shape. It includes a method to recover the shape of water surface [6], and a transparent surface, projected by a light stripe, using genetic algorithm [7]. More recently, polarization gains more popularity for recovering transparent surface [8], but faces two difficult problems, i.e., lack of surface reflection and ambiguity problems. The first problem is addressed by introducing a whole-surface lighting (photometric sampler) for small objects [4] or edge lighting (multiple parallel linear lighting) [9] for rotationally symmetric objects. The second problem is addressed by introducing additional information such as thermal radiation [10], new view image [11], trinocular stereo [12], or object's symmetry [9].

The second group elaborates as much ways as possible to synthesize a realistic image of transparent object without using any 3D shape information. It includes a method called environment matting for capturing the optical behavior of transparent surface from known and controlled background for rendering and compositing purposes [2, 13]. This method is extended to obtain the environment matting from uncontrolled backgrounds [14]. The Environment matte can also be obtained from multiple viewpoints to create novel views by interpolation [1]. Other method separates overlapped image of glass plates into two images, one is the reflected images, and the other is the transmitted images [15].

The first group relies heavily on real images and especially aimed for accurate 3D shape reconstruction. Whereas the second group relies heavily on synthesized graphical images and especially aimed for realistic 2D visualization. We believe that the ability to represent realistic synthetic images is beneficial, not only visually, but also for understanding the 3D shape. So for example, in medical radiation therapy for control or cure of cancer, the physician can easily locate cancerous tissue from normal tissue by modeling transparent 3D distribution of radiation dose to avoid complications [5].

In this paper, we pursue an integrated framework that enables the use of both synthesized graphical images and real images to infer the 3D shape of transparent object containing non-overlapping opaque surfaces. It is a neural network (NN) that minimize the error between the synthesized projection and the teacher images observed in multiple views to approximate the true object's

Proceedings of the 5th International Conference on Computer Vision Systems (ICVS 2007)
Published in 2007 by Applied Computer Science Group, Bielefeld University, Germany, ISBN 978-3-00-020933-8
This document and other contributions archived and available at: http://biecoll.ub.uni-bielefeld.de

shape. It analytically refines the vertices position of the initial 3D model using error back-propagation learning.

The main contribution of this paper is the analytic relations between the vertices position and the pixel value inside projection images of this 3D model for rendering and learning both transparent and opaque surfaces. Without such relations we have to heuristically establish a number of trial (candidate) vertex positions and choosing the positions that will maximize some objective functions [16, 17]. The problem with such techniques is the appropriate trial number and positions are hard to determine, hence some additional restrictions such as texture correlation, smoothness, and silhouette restrictions are needed [18].

3 Problem Formulation

In this section, we set a relation between 3D vertices of a triangle $\mathbf{V}_k(k = 0, 1, 2)$ and the pixel value $f(x, y)$ inside projection image of this triangle. In computer graphics, this relation is called as rendering problem. But here, our genuine interest is not only to render the triangle but to actually 'learn' (modify) the triangle's vertices based on the pixel value error of its projection image compared to a given teacher image. For that purpose, we devise an analytic relation between these two variables so that the vertices position could be learned through error back-propagation learning. In our framework, the rendering problem is actually a forward mapping process that should be followed by back-propagation learning.

3.1 Learning Opaque Surface

If the triangle is opaque, the changes in vertices position give rise to different surface's normal $\mathbf{N}$, which in turn give rise to different pixel value $F(x, y)$ for a given light source pointing to $\mathbf{L}$ direction and ambient/diffuse light A spreading inside the scene. We may write this relation as:

$$\{\mathbf{V}_0, \mathbf{V}_1, \mathbf{V}_2\} \Longrightarrow \mathbf{N} \Longrightarrow \rho\lambda(\mathbf{N} \cdot \mathbf{L}) + A \Longrightarrow F(x, y). \tag{1}$$

where ρ is surface reflectance and λ is intensity of the illuminant. The $A \Longrightarrow B$ is read as changes in A give change to B. In the forward mapping process, first we give the triangle vertices position $\mathbf{V}_k$ into our NN. Then this NN uses three sigmoid gates which mimic AND gate functions to specify whether the pixel under observation is inside the triangle. If it is inside then the NN assigns a value of another sigmoid unit placed at its output, i.e., $f(x, y)$, as the value of that pixel. If the sigmoid gain is set sufficiently high, it produces near flat intensity surface, except at area closed to triangle edges. The $f(x, y)$ is then superimposed by $\rho\lambda(\mathbf{N} \cdot \mathbf{L}) + A$ to give $F(x, y)$. A smooth shaded representation (Gouraud shading) of $F(x, y)$, i.e., $S(x, y)$, is added to give more flexibility and stability during learning. Instead of explicitly compute $\rho\lambda(\mathbf{N} \cdot \mathbf{L}) + A$, we use implicit lighting, i.e., we take the average pixel values of the teacher images at corresponding projection area of the triangle. It is aimed to implicitly capture the lighting effects instead of explicitly searching the true lighting

Proceedings of the 5th International Conference on Computer Vision Systems (ICVS 2007)
Published in 2007 by Applied Computer Science Group, Bielefeld University, Germany, ISBN 978-3-00-020933-8

which is complicated. In the backward learning process, we measure the error $E = \|F(x,y) - G(x,y)\|^2 + \|(S(x,y) - G(x,y)\|^2$, where $G(x,y)$ is the pixel value of teacher image, to be back propagated for updating $\mathbf{V}_k$ as

$$\mathbf{V}_k^m = \boldsymbol{V}_k^{m-1} - \varsigma \frac{\partial E}{\partial \mathbf{V}_k} + \mu \triangle \mathbf{V}_k^{m-1} \qquad (k = 0, 1, 3), \tag{2}$$

where ς is learning rate and μ is momentum constant.Where $\partial E \,/\, \partial\, \boldsymbol{V}_k$ is derived using a chain rule:

$$\frac{\partial E}{\partial\, \boldsymbol{V}_k} = \frac{\partial E}{\partial F} \frac{\partial F}{\partial \boldsymbol{N}} \frac{\partial \boldsymbol{N}}{\partial \mathbf{v}_k} \frac{\partial \mathbf{v}_k}{\partial \mathbf{V}_k}. \tag{3}$$

3.2 Learning Transparent Surface

If the triangle is transparent, the changes in vertices position also give rise to different surface normal $\mathbf{N}$, which in turn gives rise to different pixel value $I(x,y)$ due to reflection $\mathbf{R}$ and transmission $\mathbf{T}$ of the light ray in that pixel. We may write this relation as:

$$\{\mathbf{V}_0, \mathbf{V}_1, \mathbf{V}_2\} \Longrightarrow \mathbf{N}_b \Longrightarrow \mathbf{R} + \mathbf{T} \Longrightarrow I(x,y). \tag{4}$$

$$\mathbf{R} = \mathbf{u} - (2\mathbf{u} \cdot \mathbf{N}_b)\mathbf{N}_b \tag{5}$$

$$\mathbf{T} = \frac{\eta_i}{\eta_r}\mathbf{u} - (\cos\,\theta_r - \frac{\eta_i}{\eta_r}\cos\,\theta_i)\mathbf{N}_b \tag{6}$$

$$\cos\,\theta_r = \sqrt{1 - \frac{\eta_i}{\eta_r}^2 (1 - \cos^2\theta_i)} \tag{7}$$

where $\mathbf{u}$ is incoming ray direction as viewed from the center of camera, $\theta_i = -\hat{\mathbf{u}} \cdot \hat{\mathbf{N}}_b$ and $\theta_r = -\hat{\mathbf{T}} \cdot \hat{\mathbf{N}}_b$, and η_i and η_r are respectively the refraction index of incident and refracting materials [21]. Here we use $\mathbf{N}_b$, i.e., interpolated barycentric normal, instead of $\mathbf{N}$. Without this barycentric normal, we will have a faceted look as shown in Figure 1 (b), which we think is not appropriate for learning smooth surface. The barycentric normal $\mathbf{N}_b$ is computed as:

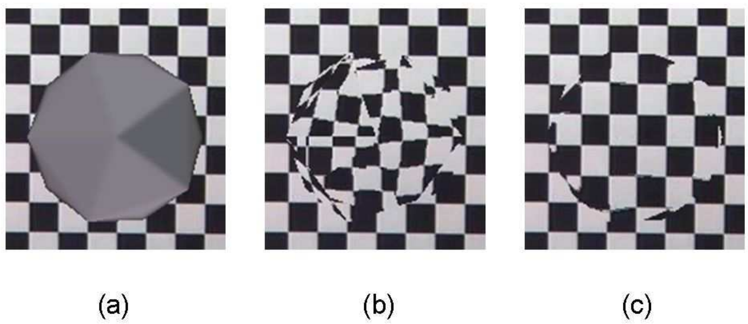


Fig. 1. Icosahedron model (a), ray traced without barycentric normals (b), and ray traced with barycentric normals.

$$\boldsymbol{N}_b = w_0\boldsymbol{N}_0 + w_1\boldsymbol{N}_1 + w_2\boldsymbol{N}_2 \tag{8}$$

where $\boldsymbol{N}_k$, $(k = 0, 1, 2)$ are the vertices normal, $w_0 = 1 - u - v$, $w_1 = u$, and $w_2 = v$, whereas u and v are computed as:

$$\begin{aligned} v = &-(x_1y_0 + x_0y - y_1x_0 - x_1y - y_0x + y_1x)/ \\ &(x_1y_2 - x_1y_0 - x_0y_2 - y_1x_2 + y_1x_0 + y_0x_2), \end{aligned} \tag{9}$$

$$\begin{aligned} u = &(x_0y - x_0y_2 - yx_2 + xy_2 + y_yx_2 - y_0x)/ \\ &(x_1y_2 - x_1y_0 - x_0y_2 - y_1x_2 + y_1x_0 + y_0x_2). \end{aligned} \tag{10}$$

Hence the relation in Equation 4 can also be written as:

$$\{\mathbf{V}_0, \mathbf{V}_1, \mathbf{V}_2\} \Longrightarrow \{w_0, w_1, w_2\} \Longrightarrow \mathbf{N} \Longrightarrow \mathbf{R} + \mathbf{T} \Longrightarrow I(x, y). \tag{11}$$

The Equations (5 - 10) are analytic continuous functions. As in opaque surface reconstruction, we measure the error $E = \|I(\mathbf{R}(x, y), \mathbf{T}(x, y)) - G(x, y)\|^2$ to be back propagated for updating $\boldsymbol{V}_k$ using Equation 2. Whereas $\partial E \;/\; \partial \boldsymbol{V}_k$ is computed by simple chain rule as:

$$\frac{\partial E}{\partial \boldsymbol{V}_k} = \frac{\partial E}{\partial I}\,\frac{\partial I}{\partial(\boldsymbol{R} + \boldsymbol{T})}\,\frac{(\boldsymbol{R} + \boldsymbol{T})}{\partial \boldsymbol{N}_b}\,\frac{\partial \boldsymbol{N}_b}{\partial \boldsymbol{w}}\,\frac{\partial \boldsymbol{w}}{\partial \mathbf{v}_k}\,\frac{\partial \mathbf{v}_k}{\partial \mathbf{V}_k}. \tag{12}$$

3.3 Learning Opaque and Transparent Surfaces

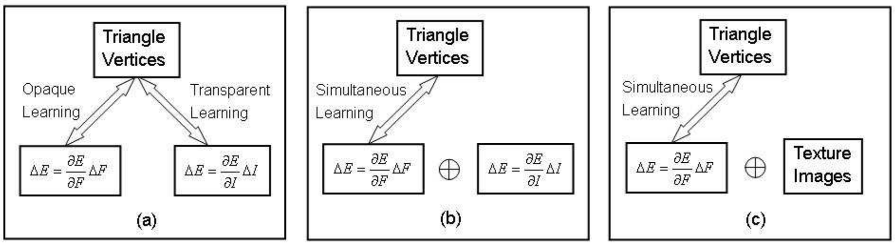


Fig. 2. Two-ways learning (a). Simultaneous learning using ray tracing (b) and texture mapping (c)

Now we have two separate ways to render and to learn the opaque and transparent surfaces as shown in Figure 2 (a). The learning of transparent surface using ray tracing framework is aimed for accurate rendering by simulating a wide range of physical phenomena including refraction, reflection, shadows, and polarization of light ray [22]. Basic ray tracing algorithm, however, is limited to sharp refraction and sharp shadows. Beside its time-consuming computation and the necessity to also compute the barycentric normal, ray tracing using pinhole camera model is inaccurate for the points far from the contours, i.e., the locus of all points where the surface normal is orthogonal to the viewing direction

Proceedings of the 5th International Conference on Computer Vision Systems (ICVS 2007)
Published in 2007 by Applied Computer Science Group, Bielefeld University, Germany, ISBN 978-3-00-020933-8
This document and other contributions archived and available at: http://biecoll.ub.uni-bielefeld.de

(Figure 3 (e)). Even though such inaccuracy can be minimized by reducing the perspective effect using longer focal distance, such limitation can not be applied when we intended to capture the detail shape of the surface. Facing with these difficulties, we prefer not to take the two-ways learning, but instead embedding the transparent surface information into the opaque surface learning as shown in Figure 2 (b).

Even though originally intended more as a means of improving rendering efficiency than as a device for improving the comprehensibility of surface shape, texture mapping is evidently serves both purposes well [5]. In this study, we implement the ray tracing formulation by blending the flat shaded, the smooth shaded, and the texture mapped images. We view the flat shaded (Figure 3 (c)), the smooth shaded (Figure 3 (d)), and the texture mapped (Figure 3 (f)) outputs respectively preserves the silhouette, the contour, and the transparency information. Since the texture actually applicable to both opaque and transparent surfaces, still we can simultaneously reconstruct these surfaces in a single learning shown in Figure 2 (c). Whereas ray traced model images can only be accurately created by explicitly specifying lighting information, the texture mapped, flat, and smooth shaded model images implicitly store the lighting information. Hence, the texture mapped, flat, and smooth shaded model images can be safely added up to represent the object.

Following this reasons, in the forward mapping process, the process until $f(x, y)$ is produced is the same as in the opaque surface mapping. However, currently we also map the texture in addition to $(\mathbf{N}\cdot\mathbf{L})+A$ to give $F(x, y)$. In the backward learning process, again we measure the error $E = \|T(x, y) - G(x, y)\|^2$, where $G(x, y)$ is the pixel value of teacher image, and $T(x, y) = ((G(x, y) + F(x, y) + S(x, y))/3$ is the pixel value of the blended texture image to be back propagated for updating $\mathbf{V}_k$ using Equation 2.

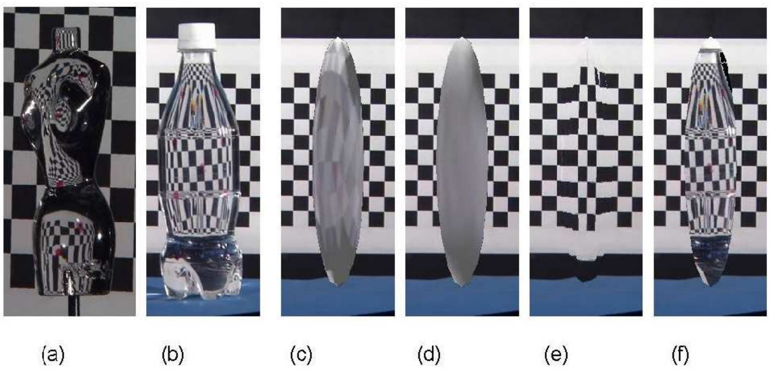


Fig. 3. The objects to be reconstructed (a, b). The projection images of the initial 3D polyhedron model rendered as flat shaded (c), smooth shaded d), ray traced (e), and texture mapped (f) images.

Proceedings of the 5th International Conference on Computer Vision Systems (ICVS 2007)
Published in 2007 by Applied Computer Science Group, Bielefeld University, Germany, ISBN 978-3-00-020933-8
This document and other contributions archived and available at: http://biecoll.ub.uni-bielefeld.de

4 Images Acquisition

In this study, we want to reconstruct two transparent objects as shown in Figure 3 (a) and (b): one is purely transparent (woman torso model) and the other contains opaque surface (coca-cola bottle with its opaque cap). We analyze the construction of a regular pattern such as checkerboard pattern put behind that object. The refractive index of woman torso is 1.5 (acrylic) and bottle is 1.3 (plastic filled with water). For each view point, we take an image of the object with the background and also an image of the background only. We acquired eight images for woman torso and six images for bottle. The objects are put on a turn table and captured using a high resolution (HDTV) camera.In illuminating the object, we used two point light sources at the left and the right of the camera and heuristically tried to reduce shadows and specular reflections. The focus of camera was set to be in the middle of the object and the background, so we obtain a just focus for both. Such settings were aimed to view the background through the bottle as clearly as possible.

For our NN learning, we have to pull the object image from its background by subtracting the image from blue screen matte background [19]. Some color spill, i.e., reflection of back-light on the foreground object at grazing angles due to Fresnel effect, are simply removed manually. In addition, many holes in alpha mattes occurred in the area where the light is not sufficiently refracted. We remove the holes using greedy algorithm to fill them in eight directions. By setting the threshold to 16, we can completely remove all holes after 3 iterations.

5 Learning Strategy

Our problem to reconstruct the object from its images is surely ill-posed [20] because we rely on limited number of view images. To deal with such problem, we refer to regularization techniques that have been shown powerful in making the solution well-posed. The regularization problem is usually formulated as:

$$\varepsilon(f) = c(f) + \beta s(f). \tag{13}$$

The solution is the one that minimizes the criterion functional $\varepsilon(f)$. The criterion functional $\varepsilon(f)$ consists of the cost functional $c(f)$ which corresponds to the mean squared error (MSE) at each data point and the stabilizing functional $s(f)$ which specifies the smoothness constraint of the surface. β is a non-negative parameter which should be appropriately chosen to adjust the weighting between the two functionals. We can directly apply this regularization technique to our NN learning by defining $c(f) = \|T(x,y) - G(x,y)\|^2$, i.e., the MSE between the blended texture and the teacher images and $s(f) = \|F(x,y) - S(x,y)\|^2$, i.e., the MSE between the flat and the smooth shaded images. We believe that it is desirable for the flat shaded image to be as closed as possible to the smooth shaded image. Currently, we choose β parameter empirically.

No matter how large the number of pixels, if the number of vertices to represent the object is too small we will not obtain an adequate representation and

Proceedings of the 5th International Conference on Computer Vision Systems (ICVS 2007)
Published in 2007 by Applied Computer Science Group, Bielefeld University, Germany, ISBN 978-3-00-020933-8
This document and other contributions archived and available at: http://biecoll.ub.uni-bielefeld.de

vice versa. Hence, the image's size and the number of vertices should be balanced. The exact relation between the two variables is, however, difficult to determine. In order to achieve a well balance between them in each reconstruction stage, we perform a hierarchical reconstruction. In addition, our system is a complex system with many degrees of freedom. Its complexity sharply increases as we add the number of vertices to be trained. It is possible to get stuck in local minima or meta-stable results along the training process and to also destruct a near optimal state that has been learned in previous steps of the training process. To deal with these problems, we refer to the SA optimization method.

6 Experiments

In this paper, we performed three experiments. These experiments are respectively aimed to observe the 3D reconstruction results during learning, the influence of regularization, and the influence of simulated annealing (SA) optimization. The experiments was run on a client Pentium(R) D CPU 3.00GHz PC, with 2.00 GB RAM.

The first experiment was performed at first at level-0 (200x533 pixels teacher images and 162 vertices and 320 faces initial icosahedron model) and then performed at level-1 (300x800 pixels teacher images, and 642 vertices, 1280 faces refined icosahedron model) of hierarchical learning. We set the regularization parameter $\beta = 1.0$ but no SA-optimization. At level-0, we set the learning rate $\eta = 3.0\text{E} - 9$, whereas at level-1 we set the learning rate $\eta = 5.0\text{E} - 11$. At the two levels we performed 1000 iterations. The frontal and bottom views of reconstruction results at level-0 was shown in Figure 4, while the results at level-1 was shown in Figure 5.

The second experiment was performed at level-0. We compared the error profiles of the experiments without and with regularization. In the two experiments we set the learning rate $\eta = 3.0\text{E} - 9$. When using regularization, we set the regularization parameter $\beta = 1.0$. We normalized the error profile with regularization by dividing it by $(1+\beta)$. The results of this experiment was shown in Figure 6(a). The regularization was faster and gave lower error, but slightly more unstable.

	NoReg-Plain	Reg-Plain	Reg-Plain+SA
Lowest relative error	0.01164	0.00319	0.00315
Comp. time (1000 iterations)	34.47 mins	36.96 mins	37.13 mins.

Table 1. Results summary.

The third experiment was also performed at level-0. We compared the error profiles of experiment with and without SA-optimization. In the two experiments we use regularization ($\beta = 1.0$). We set the initial temperature $T = 3.0E11$, the cooling rate $\zeta = 0.99$, and the learning rate $\eta = 3.0\text{E} - 9$. The results of this experiment was shown in Figure 6(b). The SA-optimization make the learning much faster to converge. Even though at the beginning it was very unstable, it

Proceedings of the 5th International Conference on Computer Vision Systems (ICVS 2007)
Published in 2007 by Applied Computer Science Group, Bielefeld University, Germany, ISBN 978-3-00-020933-8
This document and other contributions archived and available at: http://biecoll.ub.uni-bielefeld.de

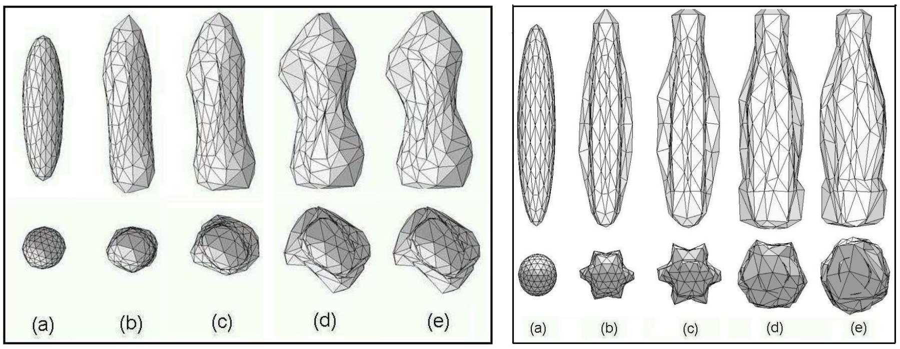


Fig. 4. The first experiment results for the woman torso (left) and cola bottle (right) at level-0 after 0 (a), 10 (b), 50 (c), 100 (d), and 1000 (e) iterations.

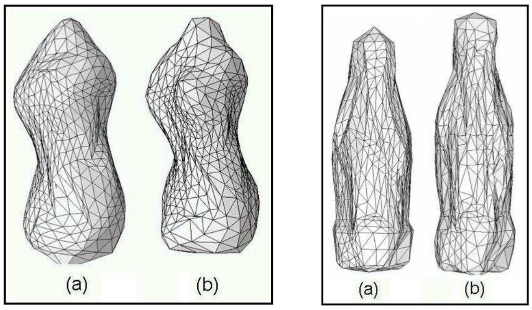
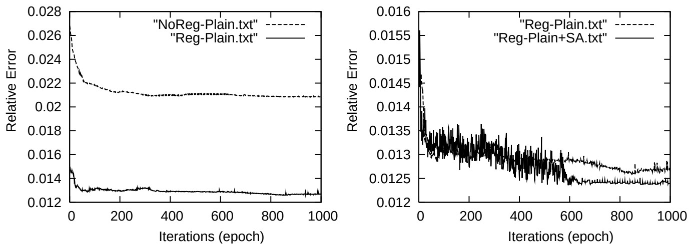


Fig. 5. The first experiment results for the woman torso (left) and cola bottle (right) at level-1 after 0 (a) and 1000 (b) iterations.



(a) (b)

Fig. 6. The second (a) and the third (b) experiments result.

Proceedings of the 5th International Conference on Computer Vision Systems (ICVS 2007)
Published in 2007 by Applied Computer Science Group, Bielefeld University, Germany, ISBN 978-3-00-020933-8
This document and other contributions archived and available at: http://biecoll.ub.uni-bielefeld.de

reached a stable state at after 600 iterations and even lower relative error. We summarized the results obtained from each experiment in Table 1.

7 Conclusion

In this paper, we presented an integrated framework to simultaneously reconstruct opaque and transparent surfaces from a limited number of views. We formulated a shape learning method based on analytic functions that relates the pixel value inside training images and the vertices of an initial 3D shape. Such functions provide a way to directly refine the vertices based on images difference, instead of heuristically establish some trial vertices positions. We incorporated ray tracing formulation to ensure its generality and future use, and implemented this formulation as texture mapping to ensure its efficiency and practicality. To improve the reconstruction results we implemented some strategies including regularization, hierarchical learning and SA-optimization. We believe our method will further open ways for practical integration of computer vision and computer graphics through neural network learning.

References

1. Matusik, M., Hanspeter, P., Zieglar, R., Ngan, N., McMillan, L.: Acquisition and Rendering of Transparent and Refractive Objects, Rendering Techniques, 267-278, (2002)
2. Zongker, D.E., Werner, D.M., Curless, B., Salesin, D.: Environment Matting and Compositing, SIGGRAPH, 205-214, (1999)
3. Bolle, R.M., Vemuri, B.C.: On three-dimensional surface reconstruction methods. IEEE Trans. on PAMI-13, 1, 1– 13, (Jan. 1991)
4. Saito, M., Kashiwagi, H., Sato, Y., Ikeuchi, K.: Measurement of Surface Orientations of Transparent Objects Using Polarization in Highlight, CVPR, 1381-, (1999)
5. Interrante, V., Fuchs, H., Pizer, S.M.: Conveying the 3D Shape of Smoothly Curving Transparent Surfaces via Texture, IEEE Trans. on VCG, Vol. 3, 2, (1997)
6. Murase, H.:Surface Shape Reconstruction of a Nonrigid Transport Object Using Refraction and Motion, IEEE Trans. on PAMI, Vol. 14, 10, 1045–1052, (1992)
7. Hata, S., Saitoh, Y., Kumamura, S., Kaida, K.: Shape Extraction of Transparent Object Using Genetic Algorithm, ICPR96, D93.6, (1996)
8. Wolff, L.B.: Shape from Polarization Images, CVWS97, 79-85, (1987)
9. Fanany, M.I., Kobayashi, K., Kumazawa, I.: A Combinatorial Transparent Surface Modeling from Polarization Images, IWCIA, 65-76, (2004)
10. Miyazaki, D., Saito, M., Sato, Y., Ikeuchi, K.: Determining surface orientations of transparent objects based on polarization degrees in visible and infrared wavelengths, JOSA-A, Vol.19, 4, (2002)
11. Miyazaki D., Kagesawa, M., Ikeuchi, K.: Polarization-based Transparent Surface Modeling from Two Views, ICCV, 1381-1386, (2003)
12. Rahman, S., Centerakis, N.: Reconstruction of Specular Surfaces using Polarization Imaging, CVPR01, I:149-155, (2001)
13. Chuang, Y., Zongker, E., Hindorff, J., Curless, B., Salesin, D., Szeliski, R.: Environment matting extensions: towards higher accuracy, SIGGRAPH, 121-130, (2000)

Proceedings of the 5th International Conference on Computer Vision Systems (ICVS 2007)
Published in 2007 by Applied Computer Science Group, Bielefeld University, Germany, ISBN 978-3-00-020933-8

14. Wexler, Y., Fitzgibbon, A.W., Zisserman, A.: Image-based Environment matting, Rendering Techniques, 279-290, (2002)
15. Szeliski, R., Avidan, Shai., Anandan, P.: Layer Extraction from Multiple Images Containing Reflections and Transparency, CVPR, 1246-, (2000)
16. Eckert, G., Wingbermuhle, J., Niem, W.: Mesh Based Shape Refinement for Reconstructing 3D-Objects from Multiple Images, CVMP04, 103-110, (2004)
17. Yaguchi, S., Saito, H.: Mesh Based 3D Shape Deformation for Image Based Rendering from Uncalibrated Multiple Views, ICAT05, (2005)
18. Nobuhara, S., Matsuyama, T.: Dynamic 3D Shape from Multi-Viewpoint Images using Deformable Mesh Models, Proc. of 3rd Int. Symposium on Image and Signal Processing and Analysis, 192-197, (2003)
19. Smith, A.R., Blinn, J.F.: Blue Screen Matting, SIGGRAPH, 259-268, (1996)
20. Chen, Z., Haykin, S.: On Different Facets of Regularization Theory, Neural Computation, Vol. 14, 12, 2791-2846, (2002)
21. Hearn, D., Baker, M.P.: Computer Graphics: C Version, Prentice Hall, Upper Saddle River, (1998)
22. Wolff, B., Kurlander, D.J.: Ray Tracing with Polarization Parameters, IEEE Computer Graphics Appl., Vol. 10, 6, 44-55, (1990)
23. Romeo, F., Vincentelli, A. S., A theoretical framework for Simulated Annealing . *Algorithmica* 6, 302-345, (1991).

Proceedings of the 5th International Conference on Computer Vision Systems (ICVS 2007)
Published in 2007 by Applied Computer Science Group, Bielefeld University, Germany, ISBN 978-3-00-020933-8