Photorealistic Object Model Generation from Observation for Virtual Reality Applications

Yoichi Sato
The Robotics Institute
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213, U.S.A.
tel: +1-412-268-6551
fax: +1-412-241-8932
ysato@ri.cmu.edu
http://www.cs.cmu.edu/afs/cs/usr/ysato/www/home-ysato.html

Katsushi Ikeuchi
Institute of Industrial Science
University of Tokyo
7-22-1 Roppongi, Minato-ku
Tokyo 106, Japan
tel: 03-3402-6231 ext. 2324
fax: 03-3401-1433
ki@iis.u-tokyo.ac.jp

Abstract

This paper describes automatic generation of photorealistic object models from observation. An object model for computer graphics applications should contain two aspects of information: shape and reflectance properties of the object. The object surface shape is recovered as a triangular mesh from multiple range images of the object. By using the reconstructed object shape and a sequence of color images of the object, parameters of the reflection model are estimated in a robust manner. The recovered shape and reflectance parameters are then used for synthesizing object images under arbitrary illumination conditions.

Keywords: color image analysis, shape reconstruction, reflectance measurement, object model generation

1 Introduction

As a result of significant advancement of graphics hardware and image rendering algorithms, the 3D computer graphics capability has become available even on low-end computers. In addition, the rapid spread of the internet technology has caused a significant increase in the demand for 3D computer graphics. For instance, a new format for 3D computer graphics on the internet, called VRML, is becoming an industrial standard format, and the number or applications using the format is increasing quickly.

However, it is often the case that 3D object models are created manually by users. That input process is normally time-consuming and can be a bottleneck for realistic image synthesis. Therefore, techniques to obtain object model data automatically by observing real objects could have great significance in practical applications.

In this paper, we study the automatic reconstruction of 3D object models from observation of real objects. For generating realistic images of a three dimensional object, two aspects of infor-

mation are necessary: the object's shape and reflectance properties such as color and specularity. For robust shape reconstruction, we use the volumetric representation for merging multiple range images of the object. Then, using an isosurface extraction algorithm, the object surface shape is generated as a triangular mesh from the volumetric data. For estimating reflectance properties of the object, our method is based on separation of the diffuse and specular reflection components from a color image sequence. The separation of the reflection components enables us to estimate reflection model parameters for each of the two components separately. The recovered object shape and reflectance properties are used to synthesize highly realistic object images.

This paper is organized as follows: Section 2 describes our image acquisition system for obtaining a sequence of range and color images of the object. Section 3 explains our method for reconstructing the object surface shape from the range image sequence. Section 4 describes our method for estimating reflectance properties of the object using the reconstructed object shape and the color image sequence. Object images synthesized using the recovered object shape and reflectance properties are shown in Section 5. Concluding remarks are presented in Section 6.

2 Image Acquisition System

The experimental setup for the image acquisition system used in our experiments is illustrated in Fig. 1. The object whose shape and reflectance information is to be recovered is mounted to the end of a PUMA 560 manipulator. The object used in our experiment is a ceramic mug whose height is about $100mm$. Using the system, a sequence of range and color images of the object is obtained as the object is rotated at a fixed angle step. Twelve range images and 120 color images were used in our experiment shown in this paper.

A range image is obtained using a light-stripe range finder with a liquid crystal shutter and a color CCD video camera [5]. We can compute 3D point locations corresponding to image pixels based on optical triangulation. Each of the range-image pixels represents an (X, Y, Z) location of a corresponding point on an object surface. The same color camera is used for digitizing color images. Therefore, pixels of the range images and the color images directly correspond.

The range finder is calibrated to produce a 3×4 projection matrix Π which represents the projection transformation between the world coordinate system and the image coordinate system. The location of the PUMA 560 manipulator with respect to the world coordinate system is also given by calibration. Therefore, the object location is given as a 4×4 transformation matrix T for each digitized image.

A single incandescent lamp is used as a point light source. In our experiments, the light source direction and the light source color are measured by calibration.

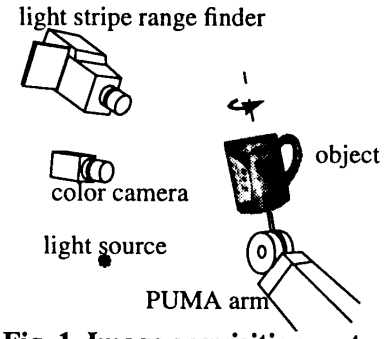


Fig. 1 Image acquisition system

3 Shape Information

A sequence of range images of the object is used to reconstruct the object shape as a triangular mesh. Then, the number of triangles used for the object shape model is reduced by simplifying the object shape without losing its details for efficient storage and rendering of the object.

One disadvantage of using the simplified object model is that a surface normal computed from the simplified model does not approximate a real surface normal accurately even though the object shape is preserved reasonably well. Thus, we compute surface normals at dense grid points within each triangle of the object surface mesh by directly using the input range images.

In Section 3.1, we describe our method for building a triangular mesh model of the object from a sequence of range images. Estimation of dense surface normals is explained in Section 3.2.

3.1 Shape Modeling from Range Image Merging

Our method for building a triangular mesh model from multiple range images consists of the following five steps, each of which is briefly described in this section. Interested readers are directed to the paper [10] for details.

1. Surface acquisition from each range image

The range finder in our image acquisition system cannot measure the object surface itself. In other words, the range finder can produce only images of 3D points on the object surface. Because of this limitation, we need to somehow convert the measured 3D points into a triangular mesh which represents the object surface shape. This is done by connecting two neighboring range image pixels based on the assumption that those points are connected by a locally smooth surface. If those two points are closer in a 3D distance than some threshold, then we consider them to be connected on the object surface.

In Fig. 2, 4 out of 12 input range images of the mug are shown as triangular meshes.

2. Alignment of all range images

All of the range images are measured in the coordinate system fixed with respect to the range finder system, and they are not aligned to each other initially. Therefore, after we obtain the triangular surface meshes from the range images, we need to transform all of the meshes into a unique object coordinate system.

For aligning all of the range images, we use a transformation matrix T which represents an object location for each range image (Section 2). Suppose we select one of the range images as a key range image whose coordinate system is used as the world coordinate system. We refer to the transformation matrix for the key range image as T_{merge}. Then, all other range images can be transformed into the key range image's coordinate system by transforming all 3D points $P = (X, Y, Z, 1)$ as $P' = T_{merge} T_f^{-1} P$ where $f = 1 \ldots n$ is a range image frame number.

3. Merging based on a volumetric representation

After all of the range images are converted into triangular patches and aligned to a unique coordinate system, we merge them using a volumetric representation. First, we consider imaginary 3D volume grids around the aligned triangular patches. (The volume grid is usually called *voxel* in the computer graphics field.) Then, in each voxel, we store the value, $f(x)$, of the signed distance from the center point of the voxel, x, to the closest point on the object surface. The sign indicates whether the point is outside, $f(x) > 0$, or inside, $f(x) < 0$, the object surface, while $f(x) = 0$ indicates that x lies on the surface of the object.

Fig. 3 (a) shows two cross sections of the volumetric data constructed from the input range images of the mug. A darker color represents a shorter distance to the object surface, and a brighter color represents a longer distance.

4. Surface shape extraction from volumetric data

The volumetric data is then used to construct the object surface as a triangular mesh. The marching cubes algorithm [3] constructs a triangular mesh by examining voxels which are zero crossings of the implicit surface $f(x) = 0$. In our method, the marching cube algorithm was modified so that it handles holes and missing data correctly.

Fig. 3 (b) shows the result of triangular mesh reconstruction. In this example, 3782 triangles are generated from the volumetric data.

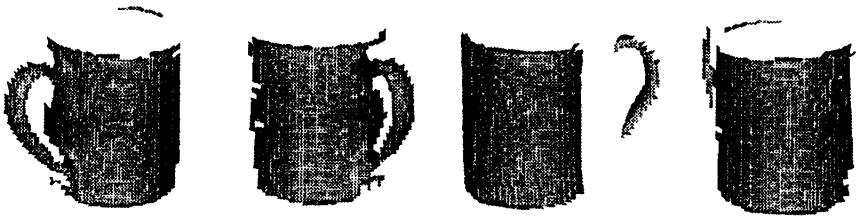
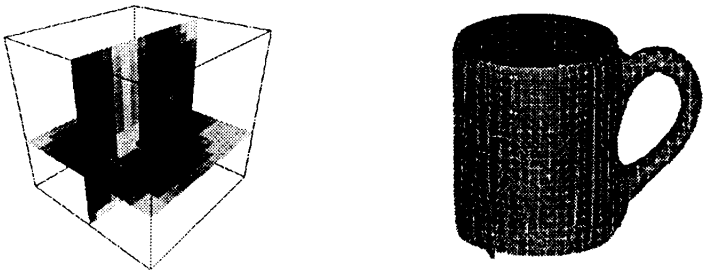


Fig. 2 Input surface meshes
4 out of 12 surface meshes are shown here.



two cross sections of volumetric data
(a)

3782 triangles
(b)

Fig. 3 (a) volumetric data of merged range images and (b) shape reconstruction result
Darker color represents closer distance to the estimated surface.

5. Mesh simplification

The marching cube algorithm generally produces a large number of triangles whose sizes vary significantly. Therefore, it is desirable to simplify the reconstructed object surface shape by reducing the number of triangles. We used the Hoppe's mesh simplification method [2] for this purpose. In our experiment, the total number of triangles was reduced from 3782 to 488 (Fig. 4).

488 triangles

Fig. 4 Simplified shape model

3.2 Surface Normal Estimation

Surface normals which are computed from the simplified triangular mesh do not approximate true surface normals accurately enough. This is because the true object surface is approximated less accurately with larger triangular patches. As a result, small highlights observed within each triangle cannot be analyzed correctly, and they cannot be reproduced in synthesized images. For this reason, we compute surface normals at regular grid points (20×20 points in our experiment) within each triangle using the 3D points from the range images.

The surface normal at a grid point P_g is determined from a least squares best fitting plane to all neighboring 3D points whose distances to the point P_g are shorter than some threshold [1]. The surface normal is computed as an eigen vector of the covariance matrix of the neighboring 3D points; specifically, the eigen vector associated with the eigenvalue of smallest magnitude.

The covariance matrix of n 3D points $[X_i, Y_i, Z_i]^T$, with centroid $[\bar{X}, \bar{Y}, \bar{Z}]^T$, is defined as:

$$C = \sum_{i=1}^{n} \begin{bmatrix} (X_i - \bar{X}) \\ (Y_i - \bar{Y}) \\ (Z_i - \bar{Z}) \end{bmatrix} \begin{bmatrix} (X_i - \bar{X}) & (Y_i - \bar{Y}) & (Z_i - \bar{Z}) \end{bmatrix} \quad \text{(EQ1)}$$

The surface normals computed at regular grid points within each triangle are then stored as a three-band surface normal image which is later used for mapping dense surface normals to the triangular mesh of the object shape. The mapped surface normals are used both for reflectance parameter estimation and for rendering color images of the object.

4 Reflectance Information

After the object shape is reconstructed, we measure reflectance properties of the object surface using the reconstructed shape and the input color images. First, the two fundamental reflection components (i.e., the diffuse and specular reflection components) are separated from the input color images. Then, the parameters for the two reflection components are estimated separately. That contributes to a reliable estimation of the specular reflection parameters. Also, the specular reflection component (i.e., highlight) in the color images does not affect estimated diffuse reflection parameters of the object surface.

In Section 4.1 we introduce the reflection model used in this analysis. Then, in Section 4.2 we describe how to determine an observed color sequence for a 3D point on the object surface from the input color images. Using the observed color sequence, the algorithm for separating the diffuse and specular reflection components is explained in Section 4.3. We explain the measurement of the diffuse reflection parameters in Section 4.4. Finally, we describe the estimation of the specular reflection parameters in Section 4.5.

4.1 Reflection Model

A general reflection model is described in terms of three reflection components, namely the diffuse lobe, the specular lobe, and the specular spike [4]. In many computer vision and computer graphics applications, reflection models are represented by linear combinations of two reflection components: the diffuse lobe component and the specular lobe component. The specular spike component can be observed only from mirror-like smooth surfaces where reflected light rays of the specular spike component are concentrated in a specular direction. It is thus difficult to observe the specular spike component from a coarsely sampled set of view-

ing directions.

The diffuse lobe component and the specular lobe component are normally called the diffuse reflection component and the specular reflection component, respectively. This reflection model was formally introduced by Shafer as the dichromatic reflection model [8].

In our analysis, the Torrance-Sparrow model [9] is used for representing the diffuse and specular reflection components. As Fig. 1 illustrates, the illumination and viewing directions are fixed with respect to the world coordinate system. The reflection model used in our analysis is given as

$$I_m = K_{D,m}\cos\theta_i + K_{S,m}\frac{1}{\cos\theta_r}e^{-\alpha^2/2\sigma^2} \qquad m = R, G, B \tag{EQ2}$$

where θ_i is the angle between the surface normal $\vec{n}$ and the light source direction, θ_r is the angle between the surface normal and the viewing direction, α is the angle between the surface normal and the bisector $\vec{n}'$ of the light source direction and the viewing direction, $K_{D,m}$ and $K_{S,m}$ are constants for the diffuse and specular reflection components, and σ is the standard deviation of a facet slope of the Torrance-Sparrow model.

This reflection model represents reflections which bounce only once from the light source. Therefore, the reflection model is valid only for convex objects, and it cannot represent interreflections on concave object surfaces. However, we empirically determined that interreflection did not affect our analysis significantly.

In this paper, we refer to $K_{D,R}$, $K_{D,G}$, and $K_{D,B}$ as the diffuse reflection parameters, and $K_{S,R}$, $K_{S,G}$, $K_{S,B}$, and σ as the specular reflection parameters.

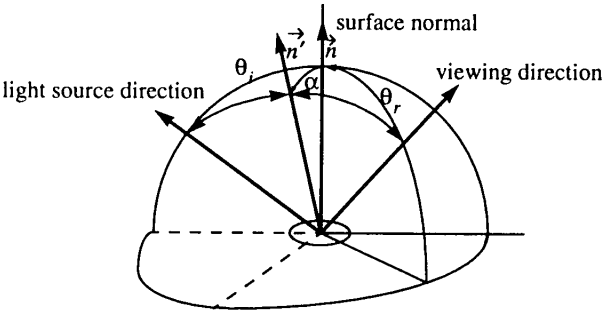


Fig. 5 Geometry for the Torrance-Sparrow model [9]

4.2 Observed Color Sequence for Each Point on Object Surface

For separating the diffuse and specular reflection components and for estimating parameters of each reflection component, we need to know a sequence of observed colors at each point on the object surface as the object is rotated. In this section, we describe how to obtain an observed color sequence of a surface point (X,Y,Z) from the input color image sequence.

We represent world coordinates and image coordinates using homogeneous coordinates. A point on the object surface with Euclidean coordinates (X, Y, Z) is expressed by a column vector $P = [X, Y, Z, 1]^T$. An image pixel location (x, y) is represented by $p = [x, y, 1]^T$. As described in Section 2, the camera projection transformation is represented by a 3×4 matrix Π, and the object location is given by a 4×4 object transformation matrix T. We denote the object transformation matrix for the input color image frame f by T_f $(f = 1 \ldots n)$.

Thus, using the projection matrix Π and the transformation matrix T_{merge} for the key range image (Section 3.1), the projection of a 3D point on the object surface in the color image frame f is given as

$$p_f = \Pi T_f T_{merge}^{-1} P \quad (f = 1 \ldots n) \tag{EQ3}$$

where the last component of p_f has to be normalized to give the projected image location (x, y) .

The observed color of the 3D point in the color image frame f is given as the (R, G, B) color intensity at the pixel location (x, y) . If the 3D point is not visible in the color image (i.e., the point is facing away from the camera, or it is occluded), the observed color for the 3D point is set to $(R, G, B) = (0, 0, 0)$. By repeating this procedure for all frames of the input color image sequence, we get an observed color sequence for the 3D point on the object surface.

4.3 Reflection Component Separation from Color Image Sequence

The algorithm for separating the two reflection components is described here. This separation algorithm was originally introduced for the case of a moving light source by Sato and Ikeuchi [6]. In this paper, a similar algorithm is applied for the case of a moving object.

Using three color bands; red, green, and blue, the coefficients $K_{D,m}$ and $K_{S,m}$, in (EQ2), become two linearly independent vectors, $\underset{\sim}{K}_D$ and $\underset{\sim}{K}_S$, unless the colors of the two reflection components are accidentally the same

$$\underset{\sim}{K}_D = \left[K_{D,R}\ K_{D,G}\ K_{D,B}\right]^T \qquad \underset{\sim}{K}_S = \left[K_{S,R}\ K_{S,G}\ K_{S,B}\right]^T \tag{EQ4}$$

First, the color intensities in the R, G, and B channels from n input images of the object are measured for each point on the object surface as described in Section 4.2. The three sequences of intensity values are stored in the columns of an $n \times 3$ matrix M. Considering the reflectance model (EQ2) and two color vectors in (EQ4), the intensity values in the R, G, and B channels can be represented as

$$\begin{aligned} M &= \left[\underset{\sim}{M}_R\ \underset{\sim}{M}_G\ \underset{\sim}{M}_B\right] \\ &= \begin{bmatrix} \cos\theta_{i1}\, E(\theta_{r1}, \alpha_1) \\ \cos\theta_{i2}\, E(\theta_{r2}, \alpha_2) \\ \vdots \\ \cos\theta_{in}\, E(\theta_{rn}, \alpha_n) \end{bmatrix} \begin{bmatrix} K_{D,R} & K_{D,G} & K_{D,B} \\ K_{S,R} & K_{S,G} & K_{S,B} \end{bmatrix} \\ &= \left[\underset{\sim}{G}_D\ \underset{\sim}{G}_S\right] \begin{bmatrix} \underset{\sim}{K}_D^T \\ \underset{\sim}{K}_S^T \end{bmatrix} \\ &\equiv GK \end{aligned} \tag{EQ5}$$

where $E(\theta_r, \alpha) = \exp(-\alpha^2/2\sigma^2)/\cos\theta_r$, and the two vectors $\underset{\sim}{G}_D$ and $\underset{\sim}{G}_S$ represent the intensity values of the diffuse and specular reflection components with respect to the illuminating/

viewing directions θ_i, θ_r, and α. The vectors $\underset{\sim}{K}_D$ and $\underset{\sim}{K}_S$ represent the diffuse and the specular reflection color vectors, respectively.

Suppose we have an estimate of the matrix K. Then, the two reflection components represented by the matrix G are obtained by projecting the observed reflection stored in M onto the two color vectors $\underset{\sim}{K}_D$ and $\underset{\sim}{K}_S$.

$$G = MK^{+} \qquad \textbf{(EQ6)}$$

where K^{+} is the 3×2 pseudoinverse matrix of the color matrix K.

The derivation shown above is based on the assumption that the matrix K is known. In our experiments, the specular reflection color vector $\underset{\sim}{K}_S$ is directly measured by a calibration procedure. Therefore, only the diffuse color vector $\underset{\sim}{K}_D$ is unknown and needs to be determined.

From (EQ2), it can be seen that the distribution of the specular reflection component is limited to a fixed angle, depending on σ. Thus, if the angle α is sufficiently large at a point on the object surface, an observed color at the point should represent the color of the diffuse reflection component. The angle α, as well as the angle θ_i and the θ_r, is computed using the object transformation matrix T_f $(f = 1 \ldots n)$ and the camera projection matrix Π as follows. The light source location is given from calibration, and the camera projection center can be computed from the projection matrix Π. Also, the surface normal $\vec{n}$ at the point for the color image frame f can be given by rotating the surface normal at the point of the object model by the object transformation matrix T_f. Using the light source direction, the viewing direction and the surface normal, α, θ_i, and θ_r are computed.

Once we get the matrix K, the matrix G can be calculated from (EQ6). Each of the diffuse and specular reflection components is then given as

$$M_D = \underset{\sim}{G}_D \underset{\sim}{K}_D^T \qquad M_S = \underset{\sim}{G}_S \underset{\sim}{K}_S^T \qquad \textbf{(EQ7)}$$

Fig. 6 (a) illustrates a typical observed color sequence with specularity. The separation algorithm was applied to the observed color sequence, and the separation result is shown in Fig. 6 (b).

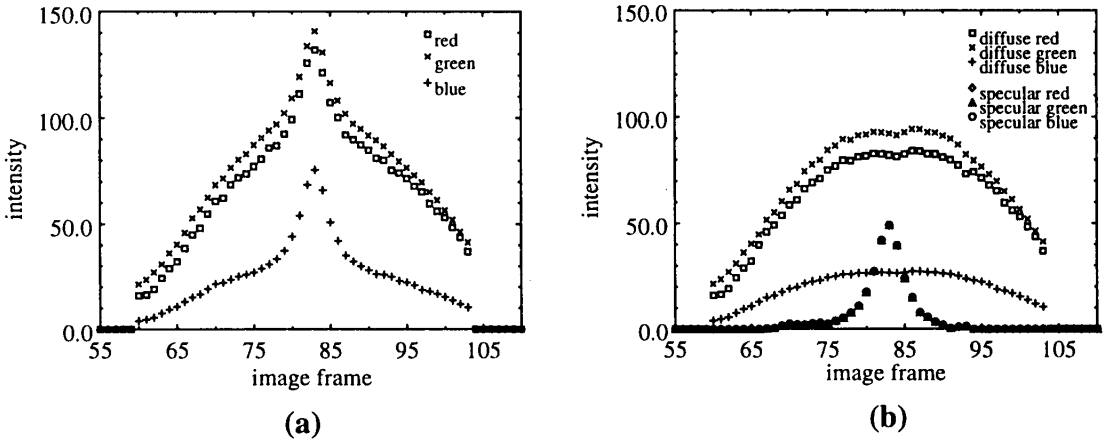


Fig. 6 (a) observed color sequence and (b) separation result

4.4 Diffuse Reflection Parameter Estimation

Using the diffuse reflection component separated from the observed color sequence, we now can estimate the diffuse reflection parameters ($K_{D,R}$, $K_{D,G}$, and $K_{D,B}$) without undesirable effects from the specular reflection component (i.e., highlights). Using the angle θ_i computed as stated in the previous section, the diffuse reflection parameters are estimated by fitting the reflection model, which is shown as the first term of (EQ2), to the separated diffuse reflection component.

The diffuse reflection parameters are estimated at regular grid points within each triangle just as the surface normals in Section 3.2 are estimated. The resolution of the regular grid points is 80×80 in our experiment. Fig. 7 shows the result of the diffuse reflection parameter estimation where the estimated parameters are visualized as surface texture on the mug.

Fig. 7 Estimated diffuse reflection parameters*

4.5 Specular Reflection Parameter Estimation

Like the diffuse reflection parameter estimation, the specular reflection parameters ($K_{S,R}$, $K_{S,G}$, $K_{S,B}$, and σ) are computed using the angle θ_r and the angle α. However, there is a significant difference between estimation of the diffuse and specular reflection parameters. The diffuse reflection parameters can be estimated as long as the object surface is illuminated and viewed from the camera. On the other hand, the specular reflection component can be observed only from a limited range of viewing directions. Therefore, the specular reflection component can be observed only at a small portion of the object surface in the input color image sequence. That is, we cannot estimate the specular reflection parameters for the rest of the object surface. Even if the specular reflection component is observed, the parameter estimation can become unreliable if the specular reflection component is not sufficiently strong, or if the separation of the two reflection components is not performed well.

For the above reasons, we decided to use a slightly different strategy for estimating the specular reflection parameters. Unlike the diffuse reflection parameter estimation, we estimate the specular parameters only at points on the object surface where the parameters can be computed reliably. Then we interpolate the estimated specular reflection parameters on the object surface to assign parameters to the rest of the object surface.

In Section 4.5.1, we describe how to select object surface points which are suitable for estimating the specular reflection parameters. Then, interpolation of the estimated specular reflection parameters on the object surface is explained in Section 4.5.2.

4.5.1 Selection of Surface Points for Parameter Estimation

For the specular refection parameters to be estimated reliably, the following three conditions are necessary at a point on the object surface:

---

* Fig. 7, Fig. 10, and Fig. 11 are available in color at the web site http://www.cs.cmu.edu/afs/cs/usr/ysato/www/research4.html.

1. The two reflection components are separated reliably. Because the diffuse and specular reflection components are separated using the difference of the colors of the two components (Section 4.3), those color vectors should differ as much as possible. This can be examined by saturation of the diffuse color. Since the light source color is generally close to white (saturation = 0), if the diffuse color has a high saturation value, the diffuse and specular reflection colors are different.
2. The magnitude of the specular reflection component is as large as possible.
3. The magnitude of the diffuse reflection component is as large as possible. Although this condition might seem to be unnecessary, we empirically found that the specular reflection parameters can be obtained more reliably if this condition is satisfied.

In our experiment, we used the vertices of the triangular surface mesh as candidates for parameter estimation. Taking these three conditions into account, 100 vertices with the largest values: v = diffuse saturation × max specular intensity × max diffuse intensity were chosen. Fig. 8 illustrates 100 selected vertices for specular parameter estimation out of 266 vertices.

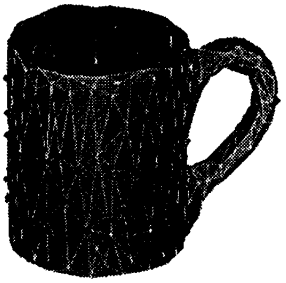
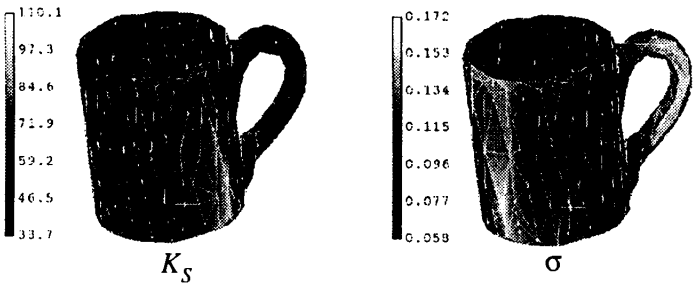


Fig. 8 100 selected vertices for specular parameter estimation

4.5.2 Interpolation of Estimated Specular Parameters

In our experiment, the camera output is calibrated so that the specular reflection color (i.e., the light source color) has the same value from the three color channels. For instance, the separated specular reflection component shown in Fig. 6 (b) has more or less the same output from the three color channels. Therefore, only one color band was used to estimate K_S and σ in our experiment.

After the specular reflection parameters K_S and σ were estimated at the 100 selected vertices, the estimated values were linearly interpolated based on a distance on the object surface, so that the specular reflection parameters were obtained at regular grid points (20×20 points in our experiment) within each triangle of the object surface mesh. The result of the interpolation is shown in Fig. 9. The obtained specular reflection parameters were then stored as two specular reflection parameter images (a K_S image and a σ image) in the same manner as was the surface normal image.



K_S σ

Fig. 9 Interpolated K_S and σ

5 Image Synthesis

Using the reconstructed object shape (Section 3.1), the surface normal image (Section 3.2), the diffuse reflection parameter image (Section 4.4), the specular reflection parameter image (Section 4.5), and the reflection model (EQ2), we can synthesize color object images under arbitrary illumination/viewing conditions.

Fig. 10 shows a synthesized image of the object with two point light sources. Note that the image represents highlights on the object surface naturally. For comparing synthesized images with the input color images of the object, the object model was rendered using the same illumination and viewing directions as some of the input color images. Fig. 11 shows two frames of the input color image sequence as well as two synthesized images that were generated using the same illuminating/viewing condition as the input color images. It can be seen that the synthesized images closely resemble the corresponding real images. In particular, highlights, which generally are a very important cue of surface material, appear on the side and the handle of the mug naturally in the synthesized images. Details of the mug's surface texture were also reproduced well.

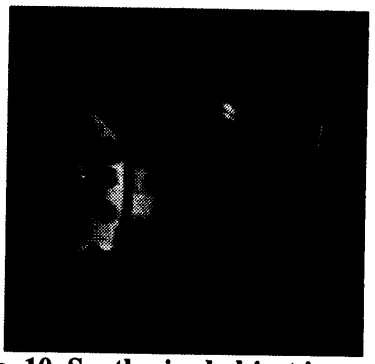


Fig. 10 Synthesized object image*

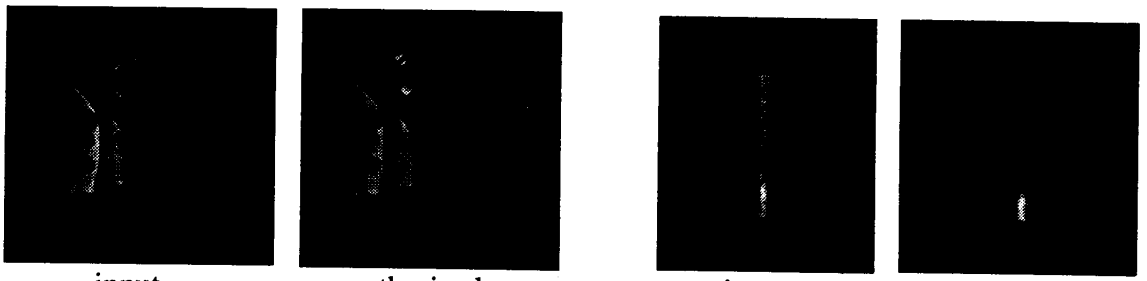


Fig. 11 Input color images and synthesized images*

6 Conclusion

We have studied automatic generation of photorealistic object models from observation. An object model for computer graphics applications should contain two aspects of information: shape and reflectance properties of the object. For robust shape reconstruction, we used the volumetric representation for merging multiple range images of the object. Then, using a modified marching cube algorithm, the object surface shape was generated as a triangular mesh. For estimating reflectance properties of the object robustly, our method is based on separation

* Again, these figures are available in color at the we site http://www.cs.cmu.edu/afs/cs/usr/ysato/www/research4.html.

of the diffuse and specular reflection components from a color image sequence. Using separated reflection components, reflection model parameters for each of the two components were estimated separately. In particular, the specular reflection parameters were successfully obtained by identifying suitable surface points for estimation and by interpolating estimated parameters over the object surface. Finally, highly realistic object images were synthesized using the recovered shape and reflectance information to demonstrate the feasibility of our method.

7 Acknowledgment

The authors thank Marie Elm for comments on the draft of the paper. The part of the work reported in Section 3.1 was done in cooperation with Mark D. Wheeler. This research was sponsored in part by the Advanced Research Projects Agency under the Department of the Army, Army Research Office under grant number DAAH04-94-G-006, and partially by NSF under Contract IRI-9224521.

8 References

[1] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle, "Surface reconstruction from unorganized points," *Computer Graphics (SIGGRAPH '92 Proceedings)*, pp. 71-78, 1992.

[2] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle, "Mesh Optimization," *Computer Graphics (SIGGRAPH '93 Proceedings)*, pp. 19-26, 1993.

[3] W. E. Lorensen and H. E. Cline, "Marching cubes: a high resolution 3D surface construction algorithm," *Computer Graphics (SIGGRAPH '87 Proceedings)*, vol. 21, no. 4, pp. 163-169, 1987.

[4] S. Nayar, K. Ikeuchi, and T. Kanade, "Surface reflection: physical and geometrical perspectives," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 13, no. 7, pp. 611-634, 1991.

[5] K. Sato, H. Yamamoto, and S. Inokuchi, "Range imaging system utilizing nematic liquid crystal mask," *Proceedings of International Conference on Computer Vision*, pp. 657-661, 1987.

[6] Y. Sato and K. Ikeuchi, "Temporal-color space analysis of reflection," *Journal of Optical Society of America A*, vol. 11, no. 11, pp. 2990-3002, November 1994.

[7] Y. Sato and K. Ikeuchi, "Reflectance analysis for 3D computer graphics model generation," to appear in *Graphical Models and Image Processing*, 1996.

[8] S. Shafer, "Using color to separate reflection components," *COLOR Research and Application*, vol. 10, no. 4, pp. 210-218, 1985.

[9] K. E. Torrance and E. M. Sparrow, "Theory for off-specular reflection from roughened surface," *Journal of Optical Society of America,* vol. 57, pp. 1105-1114, 1967.

[10] M. D. Wheeler, Y. Sato, and K. Ikeuchi, "Consensus surfaces for modeling 3D objects from multiple range images," Technical Report, School of Computer Science, Carnegie Mellon University, CMU-CS-96-185, 1996.