Computer-Aided Design and Applications
© 2009 CAD Solutions, LLC
http://www.cadanda.com

3D Shape Reconstruction from 2D Images

Daisuke Hirano[1], Yusuke Funayama[2] and Takashi Maekawa[3]

[1]DISCO, dhirano@disco.co.jp
[2]RICOH, tecnocharge@ybb.ne.jp
[3]Yokohama National University, maekawa@ynu.ac.jp

ABSTRACT

We present a method of reconstructing a 3D model from several 2D images of an object taken from different views. The data acquisition system consists of a camera on a tripod, and a computer-controlled turntable. We employ the Shape-From-Silhouette (SFS) method, which constructs a voxel-based 3D model from silhouette images. The concave shapes are further carved by using the space carving technique. In order to make the resulting model compatible with the CAD/CAM system, the voxel model is converted into a triangular mesh using the Marching Cubes algorithm. Since the mesh generated from the voxel model by using the Marching Cubes algorithm inherits the staircase effect, the mesh is adjusted to recover the object precisely by using silhouette images. Finally, we evaluate the accuracy of the proposed method. The reconstructed models of complex objects show the effectiveness of the proposed inexpensive 3D shape reconstruction system.

Keywords: shape reconstruction, Shape-From-Silhouette, Marching Cubes
DOI: 10.3722/cadaps.2009.701-710

1. INTRODUCTION

3D laser scanners have become increasingly popular in the field of reverse engineering for reconstructing 3D models because of their accuracy [2]. However, these scanners are expensive and are not portable. Moreover, they encounter some difficulties in digitizing highly reflective, transparent, translucent, or dark-colored objects.

In this paper, we introduce a method of reconstructing a 3D model from several 2D images of an object taken from different views. The data acquisition system consists of a camera on a tripod, and a computer-controlled turntable, which is considerably more cost-effective than a laser scanner. We employ the Shape-From-Silhouette (SFS) method [13], which constructs a voxel-based 3D model from silhouette images. However, the SFS method cannot remove the voxels in the concave regions that do not appear in the silhouette images. We remove these non-object voxels by using the Space Carving technique [5],[12].

Although the representation of 3D objects by using voxels is conceptually simple and easy to implement, the staircase effect of the surface voxel exists at a low resolution, and in the case of high resolution, the algorithm slows down. Furthermore, it is not compatible with the commercial CAD/CAM software. In order to remedy such problems, we employ the Marching Cubes algorithm [7] in converting the surface voxels to a triangular mesh. Finally, the triangular mesh is adjusted for recovering the object precisely by using silhouette images.

Fig. 1 outlines the flow of the proposed automated reverse engineering system. We first calibrate the camera to relate between the 2D image coordinates and the 3D world coordinates. Then, we capture the images of the entire 3D object by rotating the turntable. Silhouettes are extracted from all the captured images, and the voxel model is reconstructed from these silhouettes by using the SFS method. Then, the voxels are transformed into triangular meshes, followed by the smoothing. In the last step, we examine the accuracy of our reconstructed model.

The paper is organized as follows: In Section 2, we begin our discussion with an SFS-based data acquisition system. In Section 3, we transform the voxel model into a triangular mesh model by using the Marching Cubes algorithm. In Section 4, we introduce a novel method to smooth out the staircase effect and recover the accurate boundary of the method by using the silhouette information. In Section 5, we evaluate the accuracy of the proposed acquisition system by comparing it with the laser-scan-based method. In Section 6, we discuss the advantages of the proposed method over the laser-scan-based method. Finally, Section 7 concludes the paper.

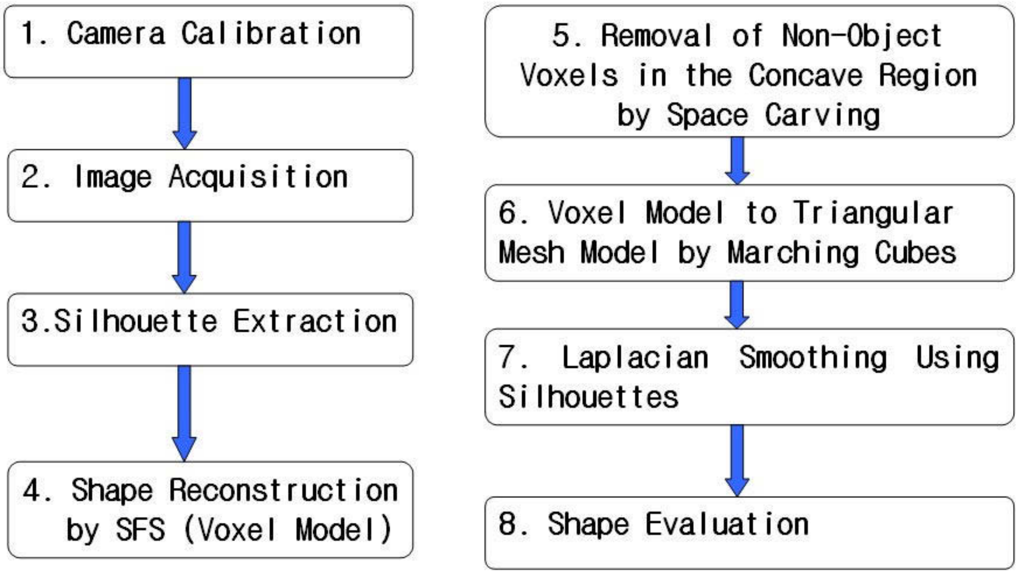

1. Camera Calibration
2. Image Acquisition
3.Silhouette Extraction
4. Shape Reconstruction by SFS (Voxel Model)
5. Removal of Non-Object Voxels in the Concave Region by Space Carving
6. Voxel Model to Triangular Mesh Model by Marching Cubes
7. Laplacian Smoothing Using Silhouettes
8. Shape Evaluation

Fig. 1: Flowchart of the proposed automated reverse engineering system.

2. SFS-BASED DATA ACQUISITION SYSTEM

In this section, we will briefly explain the fundamental techniques used in this study.

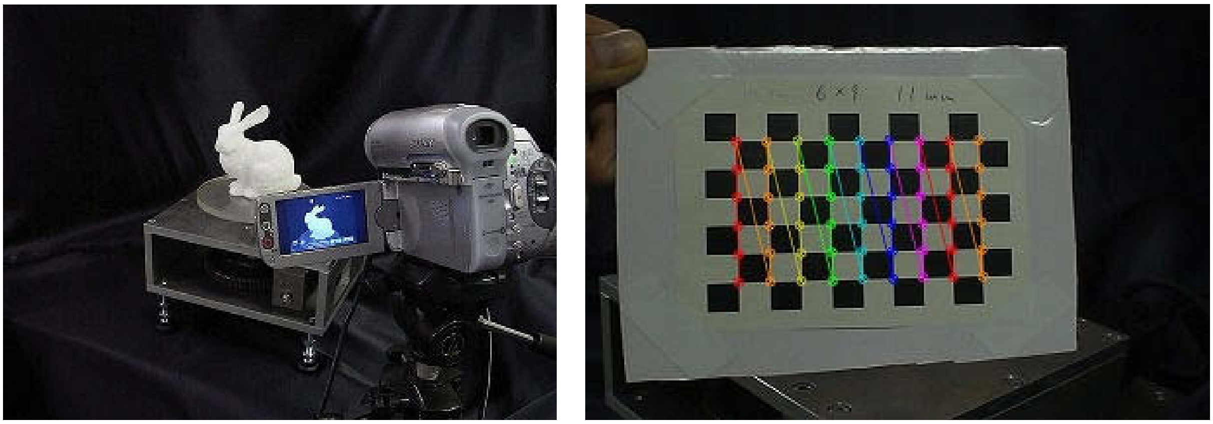
2.1 Data Acquisition System

The data acquisition system (see Fig. 2 (a)) consists of a SONY HANDYCAM DCR-HC90 camera, which captures still images at a resolution of 720×480, and a turntable rotated by a computer-controlled stepping motor. Video images are transferred to a PC with Core 2 Duo E6750 2.66 GHz with 2GB RAM via IEEE1394 interface. The OpenCV library is used for camera calibration, which is based on the method described in [14], in order to determine the intrinsic and extrinsic parameters of the camera. The camera is calibrated using the planar chessboard pattern shown in Fig. 2 (b) [14].

After every turntable rotation of 20 ° , an image of an object is acquired by the camera on a tripod with fixed angle. The space of interest is typically a 15 cm ×15 cm ×15 cm cube that is divided into 256×256×256 voxels with a resolution of approximately 0.586 mm for each voxel. The images are captured at a resolution of 720×480 pixels.

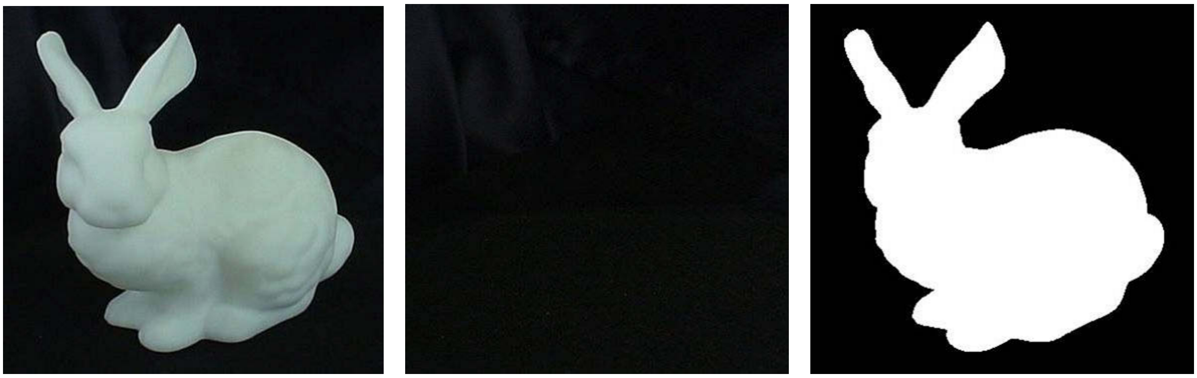
2.2 SFS Method

Shape From Silhouette (SFS) is a popular method for reconstructing 3D objects from multiple silhouette images [6],[8],[13]. Here, we have briefly explained the concept of SFS. As the first step, we need to extract silhouettes from all the images. It is crucial to extract silhouettes as accurately as possible. We adopt the background subtraction algorithm developed by [1], which removes shadows effectively by using color information to distinguish shadow and non-shadow pixels (see Fig. 3).



(a) (b)

Fig. 2: (a) Data acquisition system, (b) Calibration chessboard.



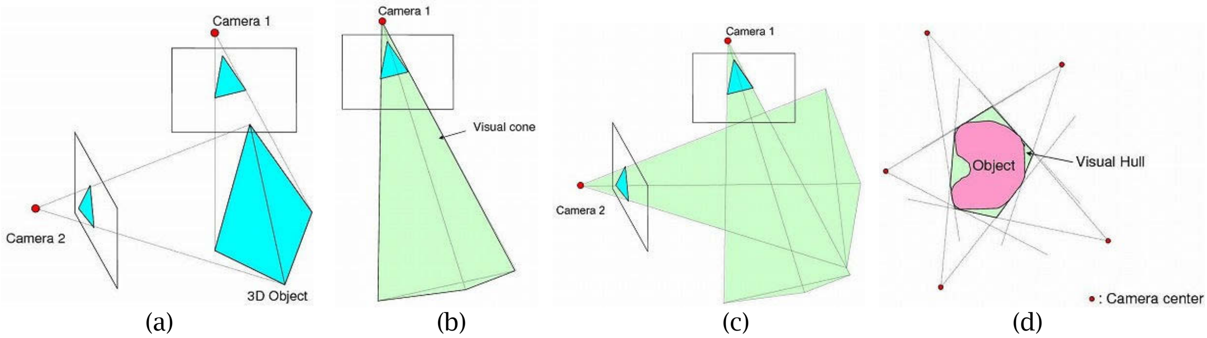
(a) (b) (c)

Fig. 3: Silhouette extraction: (a) An acquired image, (b) Background image, (c) Extracted foreground silhouette.

A tetrahedron casts silhouettes on two cameras as shown in Fig. 4 (a). A visual cone, which is a bounding volume of the object, is constructed by projecting the 2D silhouette image into the 3D space through the center of the camera, as illustrated in Fig. 4 (b). If we intersect all the visual cones from each silhouette, we obtain an approximated object called *visual hull* [6]. Laurentini [6] describes the visual hull as the maximal object that gives the same silhouette of the 3D object from any possible viewpoint.

In order to represent the visual hull by voxels, the space of interest is subdivided into $N_1 \times N_2 \times N_3$ voxels, and then each voxel is projected onto all the silhouette images. A voxel is classified as *inside*, if its projection onto each and every one of the silhouette images is inside or partially overlaps with the corresponding silhouette images. If the projection of the voxel is completely outside the silhouette even for a single image, it is classified as *outside*. After all the voxels are classified as either inside or outside, we can construct the approximation of the visual hull by taking the union of all the inside voxels.

Szeliski [13] employs octree voxels where *inside* voxels are further classified as completely *inside* or partially overlapping *ambiguous* voxels. The ambiguous voxels are recursively subdivided into eight subcubes until no ambiguous voxels exist or the size of the subdivided voxel reaches the minimum size criterion. The size of the smallest voxel determines the resolution of the visual hull.



(a) (b) (c) (d)

Fig. 4: Shape-from-Silhouette method: (a) A tetrahedron casts silhouettes on two cameras. (b) A visual cone is formed by the silhouette image and the center of camera 1. (c) The shape of the object is estimated by intersecting two visual cones. (d) The 2D visual hull shows the concave regions that do not appear in the silhouettes and therefore cannot be represented by using SFS.

2.3 Space Carving

A drawback of the SFS method is that the concave shapes of the object, which do not appear explicitly in the silhouettes, are not recovered. Accordingly, after the voxel model is built using SFS, we need to remove non-object voxels by using the Space Carving method [5],[12]. Space Carving progressively carves all non-photo-consistent voxels of the shape reconstruction space. At each iteration of the Space Carving algorithm, the voxels are removed until the resulting shape is consistent with the input images. A shape is said to be photo-consistent if all points within the volume are consistent with the input images. During this carving process, colors are assigned to the photo-consistent voxels such that the photo integrity with a set of input images is maximized.

3. VOXEL TO TRIANGULAR MESH

As the voxel representation is not smooth (see Fig. 5 (a)) and not compatible with the commercial CAD/CAM software, The Marching Cubes algorithm [7] is often used for converting the voxel model to the triangular mesh model [8],[9],[11]. Marching Cubes [7] is an algorithm used for producing a triangular mesh on an isosurface of a scalar field sampled on a cubical 3D grid. Each cube has eight vertices, and there are two states, inside and outside. Therefore, the isosurface can intersect the cube in $2^8 = 256$ ways. Since there are two different symmetries of the cube, the 256 cases reduce to only 15 patterns. The 14 basic patterns consist of at least one and as many as four triangles per cube.

Iwanari et al. [3] reconstructed a triangular mesh from the vertices of the surface voxels by using the Hoppe method [2], which utilizes a variation of the Marching Cubes algorithm. Milne et al. [9] improved the positional accuracy of each vertex of the visual hull by applying a binary search. For placing triangles while using the Marching Cubes algorithm for the *ambiguous voxels*, Mercier and Meneveaux [8] used pixel rays that originated from the camera center and passed through a pixel. They computed the intersection points between the pixel rays and voxel faces and determined the vertices of the Marching cubes along the cube edges. In our application, we first refine octree voxels into plain voxels having the minimum size of the octree voxels (see Fig. 5 (a)). Then, we assign scalar values to grid points as follows: 0 to outside grid points, 1 to surface grid points, and 2 to inside grid points. If we set the isovalue to be 0.99, then the isosurface intersects the cube edges where one vertex has value 0 and the other has value 1. A triangular mesh that is slightly larger than the real object is generated as shown in Fig. 5 (b). In general, the mesh generated from the voxel model by using the Marching Cubes algorithm is not smooth, as it inherits the staircase effect (see Fig. 5 (b)).

4. SILHOUETTE-BASED LAPLACIAN SMOOTHIHG METHOD

In this section, we introduce a novel method for smoothing out the staircase effect without losing the accuracy by using silhouette information.

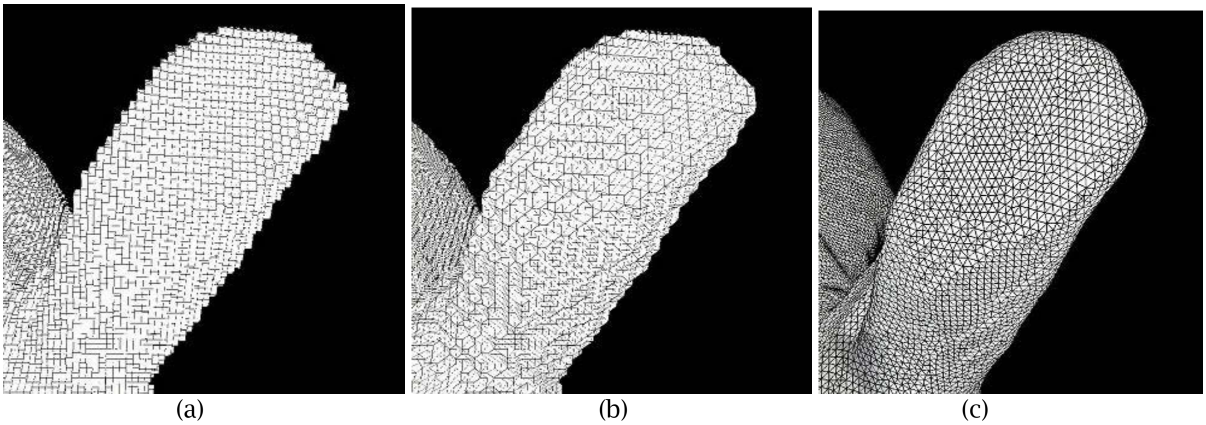


Fig. 5: Close-up views of the right ear of the Standford bunny: (a) Voxel representation, (b) Mesh generated by the Marching Cubes algorithm, (c) Mesh after silhouette-based Laplacian smoothing.

4.1 Laplacian Smoothing

Laplacian smoothing is a technique used for smoothing a polygonal mesh without changing the topology of the mesh. Each vertex V_i^j , (*i*=1, . . . ,*N*) of the polygonal mesh is moved to a new position $V_i^{j+1}$by taking the average of the positions of the 1-ring neighbor of vertices v_i^j of the vertex V_i^j and itself as follows:

$$V_i^{j+1} = \frac{V_i^j + \sum_{k=1}^{M} v_k^j}{M+1} \tag{4.1}$$

where the superscript *j* denotes the *j-th* iteration, and *M* is the number of vertices of the 1-ring neighbor of V_i. Accordingly, the amount of movement of V_i^j is given by

$$\Delta V_i^j = V_i^{j+1} - V_i^j = \frac{V_i^j + \sum_{k=1}^{M} v_k^j}{M+1} - V_i^j = \frac{\sum_{k=1}^{M} (v_k^j - V_i^j)}{M+1} \tag{4.2}$$

Fig. 6 shows a 2D version of Laplacian smoothing applied to a square with 32 vertices. As the number of applications of the Laplacian smoothing increases, the square slowly becomes circular and gradually shrinks to a small circle. Therefore, we cannot directly apply the Laplacian smoothing to our triangular mesh.

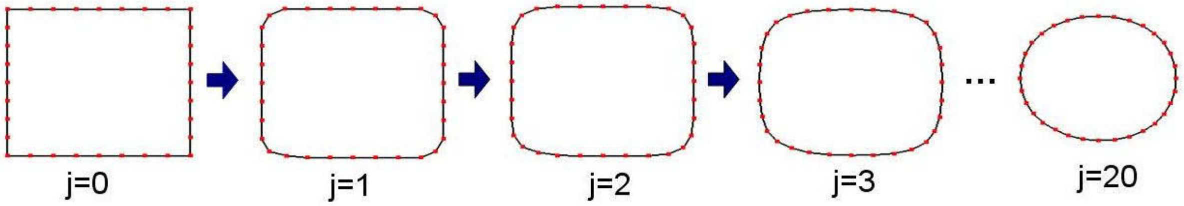


Fig. 6: Laplacian smoothing method.

4.2 Silhouette-based Laplacian Smoothing Method

As shown in Fig. 6, the application of Laplacian smoothing shrinks the model. Fortunately, the voxel model constructed by using the SFS method is generally slightly larger than the real object, and the conversion process from the voxel model to the polygonal model using the Marching Cubes algorithm further fattens the model. Consequently, we can apply the Laplacian smoothing cleverly to the fattened model and generate an accurate smoothed model. In order to accomplish this, we multiply a weight $(0 \le w_i^j \le 1)$ to Eqn. (4.2) as follows:

$$V_i^{j+1} = V_i^j + w_i^j \Delta V_i^j = V_i^j + w_i^j \frac{\sum_{k=1}^{M} (v_k^j - V_i^j)}{M+1} \tag{4.3}$$

Next, we describe how to adjust the weight w_i^j.We denote the number of images captured by the camera by *L*. Let us introduce a new term called *SilhouetteNumber* Ns_i^j, which is defined as the number of silhouettes that contain V_i^j when projected to each image. First, we acquire Silhouette Numbers Ns_i^j, Ns_i^{j+1} for V_i^j,and V_i^{j+1}, respectively. Then, we determine w_i^j on the basis of the following conditions:

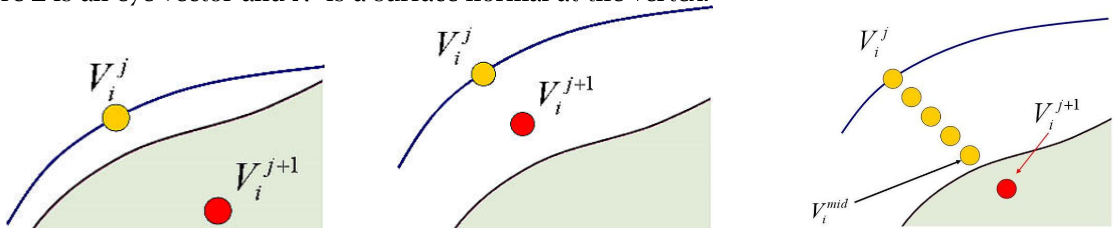
1. If $Ns_i^j = L$, then $w_i^j = 0.0$ (see Fig. 7 (a))
2. If $Ns_i^j \neq L$ and $Ns_i^j = Ns_i^{j+1}$, then $w_i^j = 1.0$ (see Fig. 7 (b))
3. If $Ns_i^j \neq L$ and $Ns_i^j \neq Ns_i^{j+1}$, then $w_i^j = \dfrac{V_{mid}^j - V_i^j}{V_i^{j+1} - V_i^j}$ (see Fig. 7 (c))

Where V_{mid}^j is a point between V_i^j and V_i^{j+1} such that the Silhouette Number Ns_i^j coincides with Ns_i^{j+1} for the first time. Once the vertex is moved on the basis of case 2, the weight is set to zero $(w_i^j = 0.0)$ after the rest of the iterations. We repeat the Silhouette-based Laplacian smoothing until most of the vertices, say 80%, fall into case 1. Fig. 3 (c) shows the effectiveness of our Silhouette-based Laplacian smoothing method. The method by Mercier and Meneveaux [8], which has a similar approach to our method, computes the intersection points between the pixel-rays and voxel faces and determines the vertices of the Marching Cubes along the cube edges. However, the mesh generated by their method does not provide smooth surfaces.

Although we have already assigned RGB colors to each voxel, it is not easy to transfer this information to the triangular mesh. Therefore, once the smoothing is completed, we project each vertex of the triangle mesh onto the images that satisfy the condition

$$\mathbf{E} \bullet \mathbf{N} \leq 0 \tag{4.4}$$

where **E** is an eye vector and **N** is a surface normal at the vertex.



(a) (b) (c)

Fig. 7: Silhouette-based Laplacian smoothing: (a) $Ns_i^j = L$, (b) $Ns_i^j \neq L$ and $Ns_i^j = Ns_i^{j+1}$, (c) $Ns_i^j \neq L$ and $Ns_i^j \neq Ns_i^{j+1}$.

5. ACCURACY ESTIMATION

In this section, we examine the accuracy of the SFS-based model by comparing this method with the laser-scan-based model. We use a 3D laser scanner for inspecting the accuracy of our SFS-based 3D shape reconstruction system. The object is placed on a turntable of the 3D laser scanner and is scanned by a laser scanning head in the vertical plane. A spot beam is projected onto the object, and its reflection on the sensor plane is measured. The range data of the object are obtained by triangulation with the projected beam direction and the position of the incident beam on the sensor plane. The 3D laser scanner can scan an object measuring up to $203mm$ (diameter)$\times 304mm$ (height) with a repeat accuracy of 0.1 mm. The scanner cannot scan around the top of the object where the laser beam either hits at a considerably shallow angle or cannot hit the object at all.

Since the laser-scan-based model *A* and the SFS-based model *B* are generated in different coordinate systems, the rigid-body translations and rotations must be applied iteratively to the SFS-based model in order to bring the vertices of the model into closest correspondence with those of the laser-scan-based model. This iterative procedure is called *localization* [4], and during each iteration the vertices of the SFS-based reconstructed model $\mathbf{R}_i$ are translated by the vector $\mathbf{t} = (t_x, t_y, t_z)^T$ and are rotated by the angles ψ, θ, and ϕ in order to yield the positions $\mathbf{r}_i$

$$\mathbf{r}_\mathrm{i} = \mathbf{C}\mathbf{R}_\mathrm{i} + \mathbf{t}, \tag{4.5}$$

where [**C**] is the rotation matrix given by

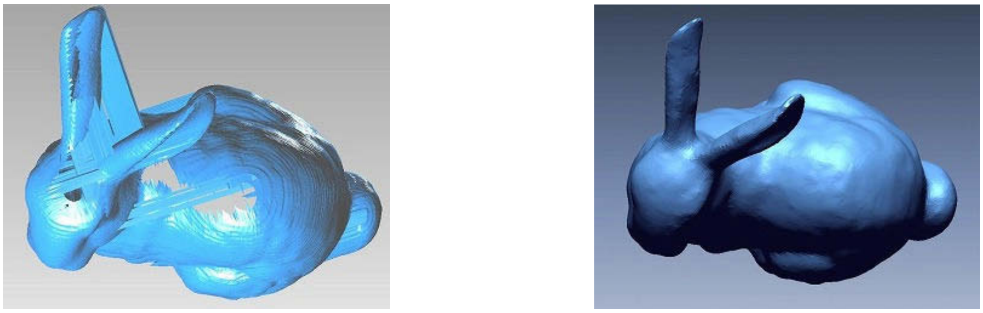
$$\mathbf{C} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\psi & -\sin\psi \\ 0 & \sin\psi & \cos\psi \end{pmatrix} \begin{pmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{pmatrix} \begin{pmatrix} \cos\phi & -\sin\phi & 0 \\ \sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{4.6}$$

The localization procedure minimizes the sum of the squared distances $d_\mathrm{i}^{\,2} = \left|\mathbf{r}_\mathrm{i} - \mathbf{Q}_\mathrm{i}\right|^2$, where $\mathbf{Q}_\mathrm{i}$ is the closest point on *A* from $\mathbf{r}_\mathrm{i}$. Hence, the objective function for the unconstrained minimization is

$$f\left(\psi,\theta,\phi,t_x,t_y,t_z\right) = \sum_{i=1}^{p} d_i^2 = \left|\mathbf{C}\mathbf{R}_\mathrm{i} + \mathbf{t} - \mathbf{Q}_\mathrm{i}\right|^2 \tag{4.7}$$

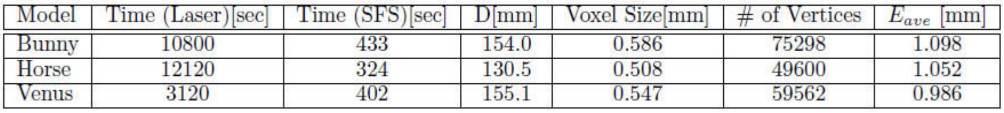
We employ the modified Powell's method [10], which does not involve the explicit computation of the function's gradient, for the minimization. Once the two models are localized, for each vertex of model *B*, we evaluate the closest point on model *A* in order to calculate the signed distance $E_i (i = 1,\ldots n_B)$ between the two models. We define the average error $E_{ave} = \frac{1}{n_B D}\left(\sum\left|E_i\right|\right)$, where *D* is the bounding box diagonal of the model.

We fabricated the Stanford bunny with ABS plastic by using a Fused Deposition Modeling (FDM) machine. An STL file, converted from a triangular mesh model in the OFF format with 60000 vertices, is sliced into 388 layers and input to the FDM machine. As shown in Fig. 8(a) the original laser-scaned mesh model has many defects, and has to be repaired by a software. Fig. 8(b) shows the repaired model. We list the reconstruction time for the 3D laser scanner, reconstruction time for SFS (excluding calibration time), bounding box diagonal of the model D, voxel size, number of vertices of the reconstructed model, and the average error in Tab. 1. The computation is performed by using a PC with a Core 2 Duo E6750 2.66 GHz with 2GB RAM.



(a) (b)

Fig. 8: Laser-scaned model: (a) Before repair. (b) After repair.



Model	Time (Laser)[sec]	Time (SFS)[sec]	D[mm]	Voxel Size[mm]	# of Vertices	E_{ave} [mm]
Bunny	10800	433	154.0	0.586	75298	1.098
Horse	12120	324	130.5	0.508	49600	1.052
Venus	3120	402	155.1	0.547	59562	0.986

Tab. 1: Performance of the proposed method.

Fig. 9 (a)~(d) show the Stanford bunny model made of ABS plastic viewed from different angles, while (e)~(h) show the SFS-based reconstructed polygonal model shown in angles corresponding to (a)~(d).Fig. 10 shows the color-coded errors of the SFS-based model *B* against those of the laser-scan-based model *A*. We assigned R (red), G (green), and B (blue) to the maximum (positive), zero, and minimum (negative) signed distance values.

6. ADVANTAGES OF THE SFS-BASED METHOD

With the current 720×480 camera resolution, the SFS-based model is not as accurate as the laser-scan-based model. However, we believe that the SFS-based method can reach to the level of accuracy of the laser-scan-based method, provided we use a camera with a higher resolution. In this section, we demonstrate that the SFS-based method can handle a highly reflective object and can reconstruct models using colors.

6.1 Highly Reflective Object

The ABS horse model (OFF file with 48483 vertices and 180 layers) is painted in gold to test if the proposed method can handle highly reflective objects. Fig. 11 (a) shows the real model, and (b) shows the model reconstructed by the laser scanner, while (c) shows the polygonal model reconstructed by using the proposed method. The advantages of the SFS-based method over the laser-scan based method are apparent from the figures.

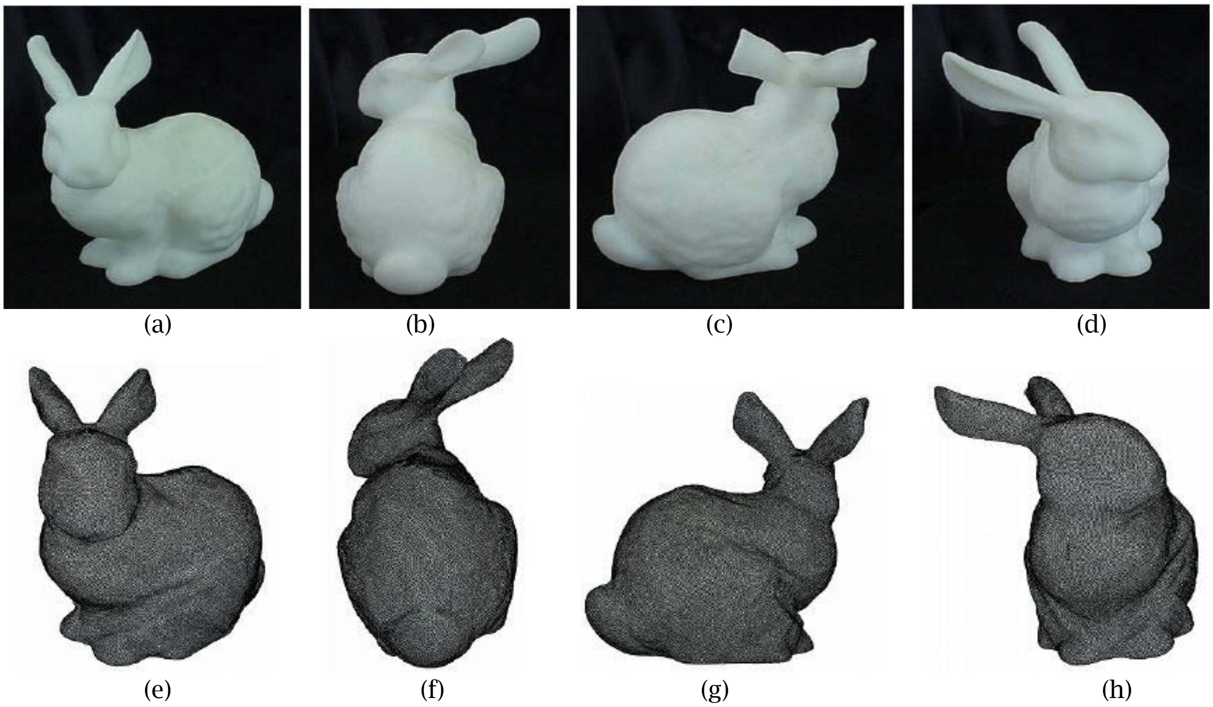


Fig. 9: Stanford bunny: (a)~(d) Real model made of ABS plastic viewed from four different angles, (e)~(h) Reconstructed polygonal model shown in angles corresponding to (a)~(d).

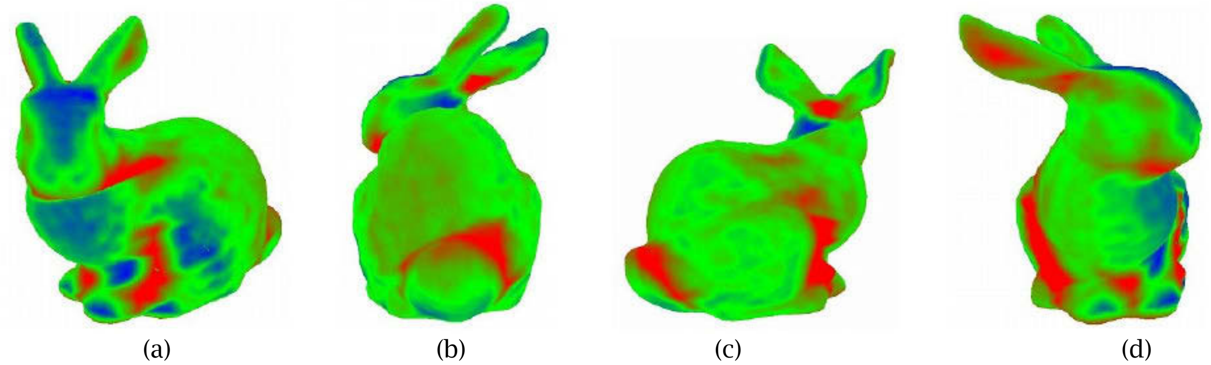


Fig. 10: Color-coded error map: (a)~(d) Color-coded error map of reconstructed polygonal model shown in angles corresponding to Fig. 9 (a)~(d).

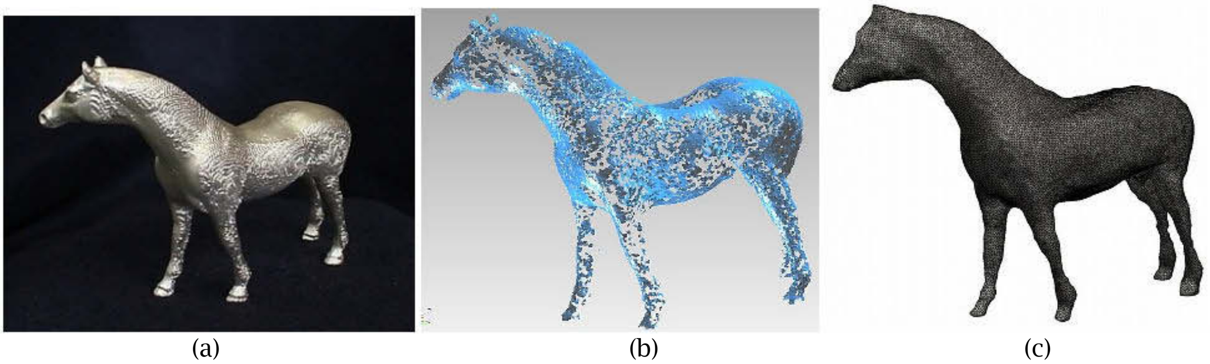


Fig. 11: Highly reflective object: (a) ABS horse model painted in gold, (b) Laser-scan-based model, (c) SFS-based polygonal model.

6.2 Mesh Coloring

The Venus of Milo model made of bronze is used for demonstrating that the SFS-based method can reconstruct a model with colors. Fig. 12 (a)~(d) show the Venus of Milo made of bronze with colors, while (e)~(h) show the corresponding reconstructed model in color.

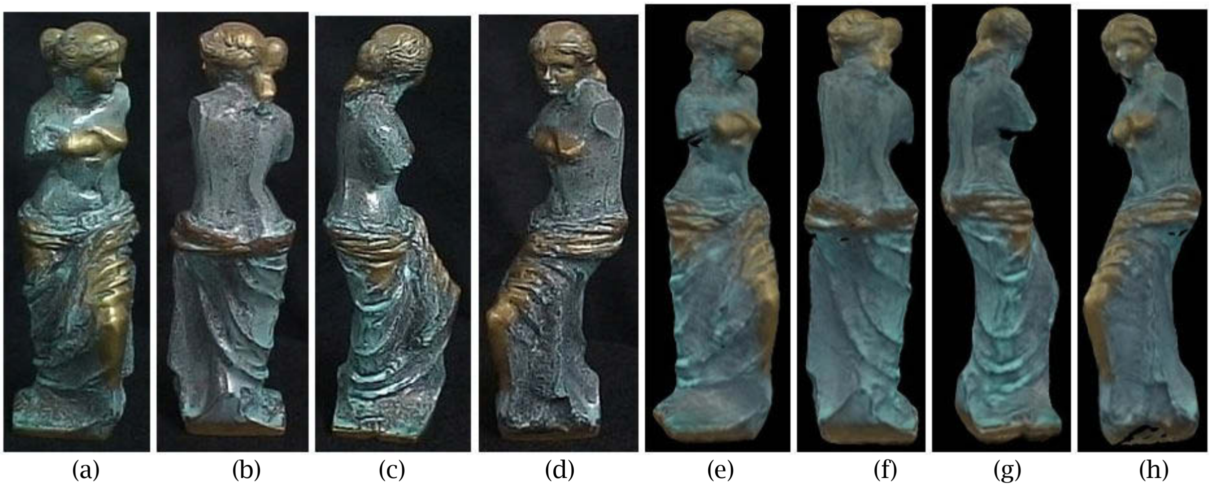


Fig. 12: Mesh coloring: (a)~(d) Real model made of bronze viewed from four different angles, (e)~(h) Reconstructed model shown in angles corresponding to (a)~(d).

7. CONCLUSION

We have developed an inexpensive automatic system, which consists of a camera on a tripod, and a computer controlled turntable, in order to reconstruct a 3D model in a triangular mesh from several 2D images of the object taken from different views.

The advantages of the proposed system are as follows:

- It costs only about 1/4 of the price of a laser scanner.
- It does not require special illumination.
- It can quickly reconstruct a polygonal model, typically within several minutes.
- It can handle objects that are highly reflective.
- It can reconstruct models with colors.

On the other hand, the drawbacks of our current system are as follows:

- It is not as accurate as a laser scanner.
- It cannot effectively delete voxels using the Space Carving algorithm in concave regions that do not appear in the silhouettes, when the object has a single color.

There are several possibilities for the improvement of our system; we have listed a few of these together with a future research as follows:

- We can improve the accuracy by employing a higher-resolution camera.
- We can acquire the images of the upper and the lower parts of the object separately using a zoom lens. After reconstructing the upper and lower parts, we can match them on the basis of the iterative
  closest point (ICP) algorithm.
- We can illuminate the object with gradation in the concave regions in order to enhance the Space Carving for single-color objects.
- As a future research, we want to compare our results with the data acquired by a handheld laser scanner.

8. ACKNOWWLEDGEMENT

This work was supported by the Ministry of Education, Culture, Sports, Science and Technology of Japan, City Area Program 2007, Yokohama Inland Area. The bunny model is the courtesy of the Stanford University Computer Graphics Laboratory, and the horse model is the courtesy of INRIA.

9. REFERENCES

[1] Cheung, G. K. M.: Visual hull construction, alignment and refinement across time, Technical Report CMURI- TR-02-05, Carnegie Mellon University, 2002.

[2] Hoppe, H.; DeRose, T.; Duchamp, T.; McDonald, J.; Stuetzle, W.: Surface reconstruction from unorganized points, Computer Graphics, 26(2), 1992, 71–78.

[3] Iwanari, E.; Sakamoto, N.; Okada, Y.: 3D shape reconstruction from multiple any views by octree representation, In MIRU2000, 2000, 161–166.

[4] Jinkerson, R. A.; Abrams, S. L.; Bardis, L.; Chryssostomidis, C.; Clement, A.; Patrikalakis, N. M.; Wolter.; F.-E.: Inspection and feature extraction of marine propellers, Journal of Ship Production, 9(2), 1993, 88–106.

[5] Kutulakos, K. N.; Seitz, S. M.: A theory of shape by space carving.,Technical Report TR692, 1998.

[6] Laurentini, A.: The visual hull concept for silhouette-based image understanding, IEEE Transactions on Pattern Analysis and Machine Intelligence, 16(2), 1994, 150–162.

[7] Lorensen, W. E.; Cline, H. E.: Marching cubes: A high resolution 3-D surface construction algorithm, In SIGGRAPH ’87: Proceedings of the 14th annual conference on Computer graphics and interactive techniques, volume 21, pages 163–169, New York, NY, USA, July 1987.

[8] Mercier, B.; Meneveaux, D.: Shape from silhouette: Image pixels for marching cubes. Journal of WSCG’2005, 2005, 13:112–118.

[9] Milne, P.; Nicolls, F.; de Jager, G.: Visual hull surface estimation, In PRASA2004, 13–18, Grabouw, Cape Town, 2004.

[10] Press, W. H.; Teukolsky, S. A.; Vetterling, W. T.; Flannery, B. P.: Numerical Recipes in C, Cambridge University Press, 1988.

[11] Sawa, H.; Sakamoto, N.; Iwanari, E.; Okada, Y.; Kanade, T.: Reconstruction of 3D objects by octree representation from multiple any views, Joho Shori Gakkai Kenkyu Hokoku, 99(29), 1999, 33–40.

[12] Seitz, S. M.; Dyer, C. R.: Photorealistic scene reconstruction by voxel coloring. In CVPR ’97: Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition, 1067-1073, Washington, DC, USA, 1997, IEEE Computer Society.

[13] Szeliski, R.; Rapid octree construction from image sequences. CVGIP: Image Understanding, 58(1), 1993, 23–32.

[14] Zhang, Z.: A flexible new technique for camera calibration. IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(11), 2000, 1330–1334.