# Three-Dimensional Shape Reconstruction from Images by Shape-from-Silhouette Technique and Iterative Triangulation

**Jung-Ho Cho, Samuel Moon-Ho Song***
*Seoul National University, School of Mechanical and Aerospace Engineering, Machine Vision and Visualization Systems Lab, Kwanak-gu Shillim-dong San 56-1 Seoul 151-742, Korea*

We propose an image-based three-dimensional shape determination system. The shape, and thus the three-dimensional coordinate information of the 3-D object, is determined solely from captured images of the 3-D object from a prescribed set of viewpoints. The approach is based on the shape-from-silhouette (SFS) technique, and the efficacy of the SFS method is tested using a sample data set. The extracted three-dimensional shape is modeled with polygons generated by a new iterative triangulation algorithm, and the polygon model can be exported to commercial software. The proposed system may be used to visualize the 3-D object efficiently, or to quickly generate initial CAD data for reverse engineering purposes, including three dimensional design applications such as 3-D animation and 3-D games.

**Key Words :** Shape-from-Silhouette, Triangulation, Cloud of Points, Shape Reconstruction

## 1. Introduction

Recent advances in computer and related technologies have enabled ordinary users to experience virtual three-dimensional space just with a conventional desktop computer. What used to be only available to professionals is now reaching the general public. For instance, three-dimensional design software tools such as 3D-MAX or Pro-E have advanced to provide incredible levels of sophistication. Parallel to these software development efforts, 3-D shape detection and content generation systems have also been keeping pace to satisfy modern user's appetite for 3-D content.

The three-dimensional shape detection systems can be categorized broadly into two groups. The first is based on laser-scanning (or halogen lamp) and the second is based on captured images. The laser-based technique consists of a laser that illuminates a known pattern (such as a grid) on the surface of the object and an optical camera that detects the reflected pattern. The acquired images from the optical camera called the "range images" (Curless and Levoy, 1996 ; Stamos and Allen, 2000 ; Karbacher et al., 2001 ; Sappa, 2000 ; Eggert et al., 1998) are analyzed to determine the 3-D shape of the object. This technique has been used successfully in the "Digital Michelangelo Project" (Levoy et al., 2000) and many similar commercial systems are now available in the market.

On the other hand, the image-based technique is still in the development stage, showing much promise. This technique determines the 3-D shape solely based on captured images of the object from a prescribe set of viewpoints. Currently, there are various research groups working towards improving this technique for their own special needs (Dyer, 2001). In this paper, for our eventual goal of efficient visualization of 3-D objects, we present our own design of an image-

* Corresponding Author,
**E-mail** : smsong@snu.ac.kr
**TEL** : +82-2-880-1700; **FAX** : +82-2-883-1513
Seoul National University, School of Mechanical and Aerospace Engineering, Machine Vision and Visualization Systems Lab, Kwanak-gu Shillim-dong San 56-1 Seoul 151-742, Korea. (Manuscript **Received** January 27, 2003; **Revised** August 12, 2002)

based 3-D shape determination system. The 3-D shape is "carved" based on the silhouette data determined from captured images. The shape-from-silhouette (SFS) technique is followed by a new iterative triangulation algorithm to generate the final mesh model of the 3-D object.

Our new contribution towards 3-D shape determination research can be summarized as: (1) the new voxel-based shape-from-shading algorithm (2) proposal of a fast iterative triangulation algorithm with new computationally efficient approximate-tangent-plane (ATP) determination procedure. The new ATP determination procedure is summarized by Eqs. (7) through (10) in Section 4.

This paper is organized as follows. Section 2 presents the image acquisition system with isotropic image sampling strategy. The shape-from-silhouette technique to extract three-dimensional information including its error estimation is discussed in Section 3. Our new iterative triangulation, from cloud of points, is presented in Section 4. Several exemplary 3-D mesh models of real objects are shown in Section 5, and finally we conclude our work in Section 6.

## 2. Image Acquisition System

### 2.1 System description

Most previous image acquisition systems used for three-dimensional extraction (Matsumoto et al., 1997) are composed of two main parts as shown in Figure 1. The turntable turns the 3-D object while the fixed camera captures images. These systems can successfully generate 3-D models of cylindrically shaped objects but will fail for arbitrarily shaped objects.

The proposed 3-D image acquisition system captures images of 3-D objects from all possible viewpoints. Thus, the camera rotates around the object on a surface enclosing the object and in addition, points towards the object for capturing such images. Figure 2 shows our 3-D shape generation system, which captures images of 3-D objects from all possible view angles.

The object is placed on the turntable and is imaged by the camera on the carriage. The carriage, powered by a stepper-motor equipped with an encoder for position feedback, moves along the vertical C-arm by means of a timing belt (difficult to see in the photo) attached to it. The turntable is powered by a second independent stepper-motor, also is equipped with an encoder for position feedback. Thus, the camera is able to capture images of the 3-D object at all possible view angles on the upper hemisphere for generation of the 3-D shape. If a view from the lower hemisphere is desired, the object is simply placed on the turntable upside down for imaging. The captured images from the lower hemisphere are then merged with the previously obtained upper hemisphere images to generate the 3-D shape comprising of images captured from all view angles.

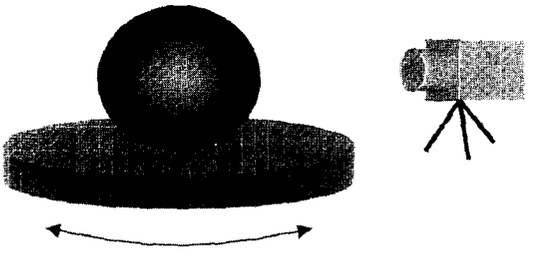


**Fig. 1** Previous image acquisition system composed of two main parts: a turntable and a fixed digital camera

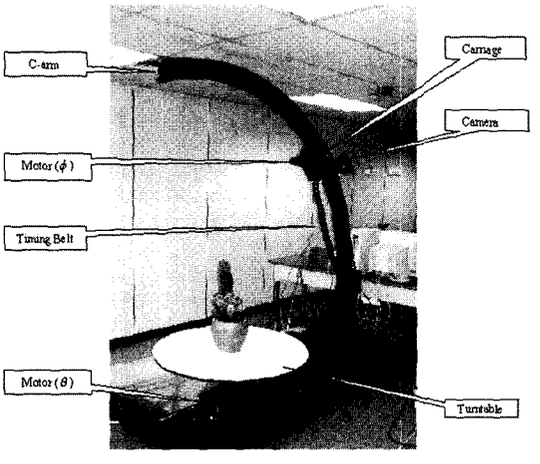



**Fig. 2** The 3-D content generation system for capturing images of objects from all possible view angles

Two motors controlling $\theta$ (turntable) and $\phi$ (camera carriage) are pre-programmed from the operator console (PC) to run in a prescribed manner. The shutters are released at the predetermined position of $(\theta, \phi)$ upon consulting feedback values of the two encoders on both motors. As the expected and the actual view angles $(\theta, \phi)$ may differ, the system records the encoder readings of all image capture positions and tags each captured images with the actual $\theta$ and $\phi$ values.

### 2.2 Viewpoint sampling strategies

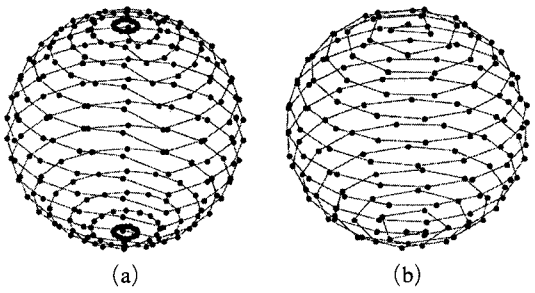
The aforementioned view angles $(\theta, \phi)$ are precisely the position of the camera, which is ultimately controlled by the two stepper-motors. We parameterize the two angles as $\theta(t)$ and $\phi(t)$, $0 \le t \le T$ where $T$ is the total capture time. These angles are the discretized as $\theta_k$ and $\phi_k$, $k=0, 1, \cdots, N-1$, so that the total number of captured images become $N$. The two angle sequences, $\theta_k$ and $\phi_k$, establish the camera path uniquely and must be designed efficiently to keep the imaging time short. As the location of the camera is fixed at $R$ (radius of the C-arm) from the origin, the camera motion must be designed to cover the surface of the sphere of radius $R$ in a single sweep to keep the imaging time short. We present two such designs. More details can be found in (Song, 2001).

#### 2.2.1 Slice sampling strategy

For slice sampling, where $N_\theta$ images are captured every slice, the turntable $(\theta)$ is rotated at a constant angular speed but the carriage $(\phi)$ must be rotated in discrete steps of $\pi/N_\phi$, where $N_\phi$ is the number of slices. Thus, as $T$ is the total time required for capturing all $N$ frames, the two motor are controlled according to

$$\begin{aligned} \theta(t) &= \theta_o + \frac{2\pi t}{T_\theta}, \\ \phi(t) &= \phi_o + \frac{\pi[t/T_\theta]}{N_\phi}, \end{aligned} \qquad 0 \le t \le T \qquad (1)$$

where the operator $[\cdot]$ denotes the floor operation, which takes the greatest integer less than or equal to the argument and $T_\theta$ is the time taken for the camera to rotate one revolution in $\theta$ direction.



(a) (b)

**Fig. 3** Image capture locations for the slice sampling (a) and the isotropic spiral sampling strategies (b). The locations are marked with dots on the sphere according to the appropriate capture times. Notice the crowing of capture points near the two poles for the slice sampling case

Thus, if the two motors are controlled according to the above, and the image acquisition times are taken as: $t = kT_\theta/N_\theta$, $k=0, 1, 2, \cdots, N-1$, the image capture locations are given by:

$$\begin{aligned} \theta_k &= \theta_o + \frac{2\pi k}{N_\theta}, \\ \phi_k &= \phi_o + \frac{\pi[k/N_\theta]}{N_\phi}, \end{aligned} \qquad k=0, 1, \cdots, N-1 \qquad (2)$$

Figure 3(a) shows the capture positions and the dots indicate the actual image capture locations of the camera. There are two undesired properties of this strategy: (i) the capture points are unnecessarily closer together near the two poles and (ii) the requirement of moving $\phi$ in discrete steps delays the capture times as the system must wait for the camera to settle (without vibration) into its newly moved $\phi$. We next describe the isotropic spiral sampling strategy, which alleviates these problems.

#### 2.2.2 Isotropic spiral sampling strategy

The efficient design of image capture locations must achieve two goals: (i) the two motors rotate at constant speeds and (ii) the capture positions must be isotropically distributed on the surface of the sphere of radius $R$. What this means is that the capture times must be adjusted to capture less number of images near the poles so that the

viewing resolution will be independent of the camera position.

The first requirement establishes the following motor control:

$$\theta(t)=\theta_o+\frac{2\pi t}{T_\theta},\quad \phi(t)=\phi_o+\frac{\pi t}{T},\quad 0\le t\le T \tag{3}$$

where as before, $T$ is the total time required to capture a total of $N$ images and $T_\theta$ is the time taken for the camera to rotate one revolution in the direction $\theta$. It can be shown (Song, 2001) that using

$$t_k=\frac{T}{\pi}\left(\cos^{-1}\left(\cos(\phi_o)\left(1-\frac{2k}{N-1}\right)\right)-\phi_o\right),\quad k=0,\ 1,\ \cdots,\ N-1 \tag{4}$$

as the $k^{\text{th}}$ capture time, in conjunction with the motor control of Eqn. (3) results in the isotropic spiral sampling. Figure 3(b) shows the resulting image capture locations. Note that the capture positions are equally spaced, which in effect offers isotropic viewing resolution. The disadvantage of the previous slice sampling, where more images are unnecessarily captured towards the two poles, has been eliminated.

## 3. Extraction of Three-Dimensional Shape

### 3.1 Image segmentation

Before the 3-D shape can be determined, individually captured images must be segmented. That is, the object must be separated from its background. This segmentation is a difficult process at best in a general setting. In other words, we cannot determine the object from the background as in general, pictures are combinations of the many different objects. However, if the image is captured with a monotone background (which can be controlled when imaging the object), the desired object region can be segmented with a relative ease through a simple thresh holding and a connectivity test. Thus, all objects have been imaged with a monotone (usually white to avoid glares) background. Therefore, we can easily determine the object region and save the segmented as binary images as seen in Fig. 4.

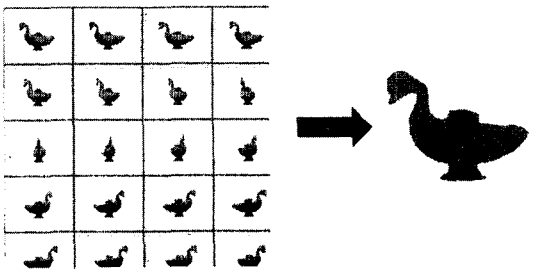


**Fig. 4** All images in the image set of an object must be segmented to differentiate object and background. The segmented results are saved as binary images

### 3.2 Shape-from-silhouette

Martin and Aggawal (Martin and Aggawal, 1983) presents a three-dimensional shape determination technique using orthogonal projected silhouettes. Others have followed refining this shape-from-silhouette strategy technique to date (Tosovic and Sablating, 2001 ; Weik, 2000). We present our own SFS implementation.

Our carving process of the shape-from-silhouette technique proceeds as follows. First, a three-dimensional array is allocated in memory at a desired resolution. Second, the discrete points or voxels in space are projected to two-dimensional image plane and marked whether the points belong to object region or to background region. This process repeats for all the captured and segmented images. When this simple calculation is performed for all the pre-segmented binary images, all points marked as object region describe the volume of the 3-D object. There have been efforts to reduce the memory requirements for 3-D data points such as octree structure (Chien and Aggawal, 1986 ; Potmesil, 1987 ; Ahjua and Veenstra, 1989 ; Srivastava and Ahjua, 1990) that describes volume as sub-divided cubes. This strategy can reduce the size of memory but at increased complexity. Therefore, we adopt computationally efficient three-dimensional $M_x\times M_y\times M_z$ array to describe the shape.

Our proposed technique delineates the boundary of the 3-D object and uses the SFS algorithm as follows. Figure 5 defines the coordinate

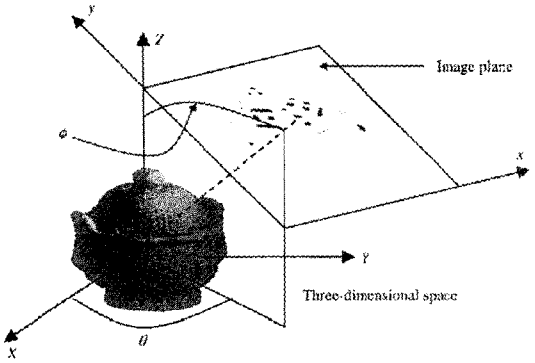



**Fig. 5** Coordinates set for orthogonal projection

system used in the following description. On the image plane $[x, y]^T$, let $R_k(x, y)=1$ denote the object and $R_k(x, y)=0$ denote background regions, respectively. Here, the index $k$ represents the image number and $(\phi_k, \theta_k)$, $k=0, 1, \cdots N-1$, denote corresponding view angles, where $N$ is the total number of acquired images. The aforementioned segmentation step determines $R_k(\circ, \circ)$. Based on this segmented object data, the binary volumetric data $V(X, Y, Z)$ is generated. The volume data $V(X, Y, Z)$ is set to, "1" if that voxel lies within the object, or to a, "0" if not. This determination requires the transformation matrix $M(\phi_k, \theta_k)$ that takes the 3-D point $[X, Y, Z]^T$ to the 2-D image point $[x, y]^T$ according to the view angle $(\phi_k, \theta_k)$. This transformation matrix $M(\phi_k, \theta_k)$ is defined below:

$$M(\phi_k, \theta_k) = \begin{bmatrix} \cos\theta_k & -\sin\theta_k\cos\phi_k & -\sin\theta_k\sin\phi_k \\ 0 & \sin\phi_k & -\cos\phi_k \end{bmatrix} \quad (5)$$

Thus, the transform described above can be expressed as:

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \cos\theta_k & -\sin\theta_k\cos\phi_k & -\sin\theta_k\sin\phi_k \\ 0 & \sin\phi_k & -\cos\phi_k \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad (6)$$

## 4. Three-Dimensional Shape Reconstruction

### 4.1 Point sampling

The volume data with voxels marked as, "1" after the carving process completely describe the shape of the 3-D object. It would be most accurate to provide this data for the shape description. However, due to the sheer size reaching millions of voxels, we subsample the voxels on the surface of the object to generate triangles that best describe the 3-D object.

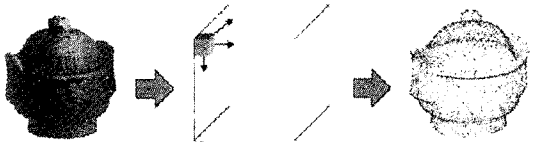


**Fig. 6** *Marching cube-like* point sampling. The left is the shape rendered with all voxels on the surface. The center represents the sampling process and sampled surface points are shown on right

The generation of *cloud of points* in which the triangulation procedure must be proceeded by (1) detection of surface points and (2) subsampling of the surface points. The detection of surface points is based on the following idea. If all the six points (voxels) surrounding the center point are marked as 1, then the center point is marked, "inside". If at least one neighboring point is marked 0, then the center point is marked, "surface".

There are several ways to subsample the surface points. Intuitively, points must be sampled according to the curvature of the each region because flat regions do not need many points but curved regions require more points to express the curved surface. However, this is unnecessarily too complex for our visualization purposes and we implement our own simple sampling strategy. We call it *marching cube-like* sampling algorithm and it samples one point on surface at each step. The selected point is the surface point closest to the center of the cube. The size, i.e., dimension of this small cube, determines the spatial sampling rate. In addition, this algorithm guarantees the state of isotropic *cloud of points* as shown in Figure 6.

### 4.2 Polygon generation: triangulation

There exists many algorithms to perform the task of triangulation from cloud of points. The

most popularly used Delaunay-Vornoi triangulation algorithm (Fortune, 1992) guarantees optimum polygon state. However, its complexity of order $N^4$ for triangles in 3-D is costly. In addition, it is known to generate superfluous triangles in concave regions, and thus the process referred to as *crust algorithm* (Amenta et al., 1998) is needed. Other approaches include *marching cubes* (Lin et al., 2001), *ball pivot algorithm* (BPA) (Bernardini et al., 1999), and a method using linear interpolation between polygon slices (Barequet and Sharir, 1996) also have been proposed.

Among many known triangulation, we applied the $k$-neighborhood points (Piegel and Tiller, 2002) and their tangent plane based triangulation algorithm (Hoppe et al., 1992; Oblonšěk and Guid, 1998). Its main idea is as follows. When one pair of points, i.e., one edge, is selected, the approximate tangent plane (ATP) for their $k$-neighbor points is determined first. Then, neighboring points are projected to the tangent plane and the triangulation is performed in this plane but triangles are saved in three-dimensional space. In other words, a new point for one edge to make a triangle is selected in the approximate tangent plane but the composed triangle is projected back to the three-dimensional space.

#### 4.2.1 Determination of approximate tangent plane

When $k$-neighbor points for an edge are selected, parameters of the ATP, i.e., the normal vector of the plane, must be determined. Previous work for this calculation was generally performed by using least square method which finds the normal vector minimizing the function as the following

$$J=\sum_{i=0}^{k-1}\{(\mathbf{r}_i-\mathbf{r}_c)\cdot\mathbf{n}\}^2 \tag{7}$$

where $\mathbf{r}_i$ is the each neighbor point and $\mathbf{r}_c$ is the centroid of neighbor points set and $\mathbf{n}$ is the normal vector of the plane.

Upon differentiation of $J$ with respect to each members of $\mathbf{n}$, this minimization problem leads to the following system of equations that $\mathbf{n}$ is required to satisfy.

$$\begin{bmatrix} \sum_i (r_{i,x}-r_{c,x})^2 & \sum_i (r_{i,x}-r_{c,x})(r_{i,y}-r_{c,y}) & \sum_i (r_{i,x}-r_{c,x})(r_{i,z}-r_{c,z}) \\ \sum_i (r_{i,y}-r_{c,y})(r_{i,x}-r_{c,x}) & \sum_i (r_{i,y}-r_{c,y})^2 & \sum_i (r_{i,y}-r_{c,y})(r_{i,z}-r_{c,z}) \\ \sum_i (r_{i,z}-r_{c,z})(r_{i,x}-r_{c,x}) & \sum_i (r_{i,z}-r_{c,z})(r_{i,y}-r_{c,y}) & \sum_i (r_{i,z}-r_{c,z})^2 \end{bmatrix} \begin{bmatrix} n_x \\ n_y \\ n_z \end{bmatrix} \tag{8}$$

where $\mathbf{r}_{i,x}$, $\mathbf{r}_{i,y}$ and $\mathbf{r}_{i,z}$ denote $x$, $y$ and $z$ components of $\mathbf{r}_i$, respectively.

But this approach can be costly as there are thousands of points. Thus, we propose a simpler algorithm to determine the normal vector by using only vector cross product calculation. We simply set the normal vector $\mathbf{n}$ as the average of all cross products of the vectors $(\mathbf{r}_i-\mathbf{r}_c)$, $i=0, 1, \cdots, k-1$ as shown below.

$$\mathbf{n}=\frac{\left\{\sum_{i=0}^{k-1}\sum_{j=i+1}^{k-1}\frac{(\mathbf{r}_i-\mathbf{r}_c)\times(\mathbf{r}_j-\mathbf{r}_c)}{\|(\mathbf{r}_i-\mathbf{r}_c)\times(\mathbf{r}_j-\mathbf{r}_c)\|_2}\right\}}{\left\|\left\{\sum_{i=0}^{k-1}\sum_{j=i+1}^{k-1}\frac{(\mathbf{r}_i-\mathbf{r}_c)\times(\mathbf{r}_j-\mathbf{r}_c)}{\|(\mathbf{r}_i-\mathbf{r}_c)\times(\mathbf{r}_j-\mathbf{r}_c)\|_2}\right\}\right\|_2} \tag{9}$$

The denominator above is to normalize $\mathbf{n}$. Next, the $k$-neighbor points are projected onto the ATP as follows:

$$\mathbf{r}_{i,new}=\mathbf{n}\mathbf{n}^{\mathrm{T}}(\mathbf{r}_i-\mathbf{r}_c)+\mathbf{r}_i,\ i=0, 1, \cdots, k-1 \tag{10}$$

Note that $\mathbf{r}_i$ is the $i^{\text{th}}$ original point, $\mathbf{r}_{i,new}$ is the $i^{\text{th}}$ projected point in the ATP.

#### 4.2.2 Choice of new point for an edge in ATP and iterative generation

When one edge and its neighboring points are selected with their ATP determined, one proper point must be selected to form a triangle. The

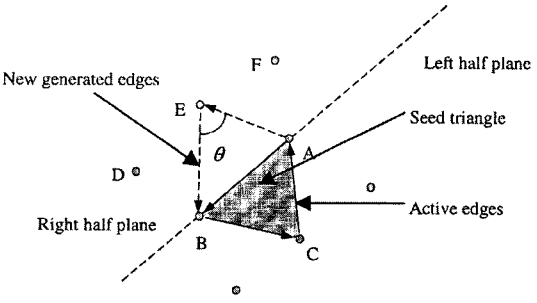
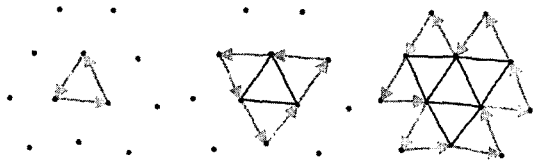



**Fig. 7** Make a new triangle in ATP. New point E is selected to compose a second triangle because E has the maximum opposite angle for an active edge AB

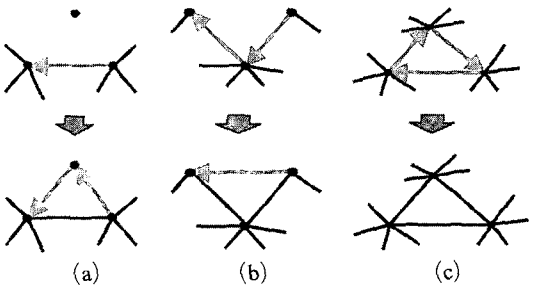
criterion for one cycle of triangle generation process, the ATP is divided into two half planes, the right and the left half plane. Between these two planes, we use the right half plane only. As seen in Figure 7, among the three points on the right half plane, point E makes the masimum opposite angle $\theta$ and thus is selected to form the new triangle (A, E, B).



(a) (b) (c)

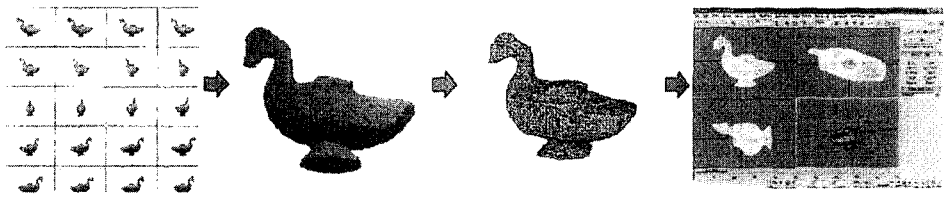
**Fig. 8** Iterative polygon generation process. (a) Seed triangle with three active edges. (b) Previous three active edges became fixed and three triangles are added after one iteration. (c) Like previous iteration six triangles are added after the second iteration

Figure 8 and Figure 9 show a summary of our proposed triangulation procedure. Iterative generation procedure is shown in Figure 8, where new triangles are formed at each iterative step.

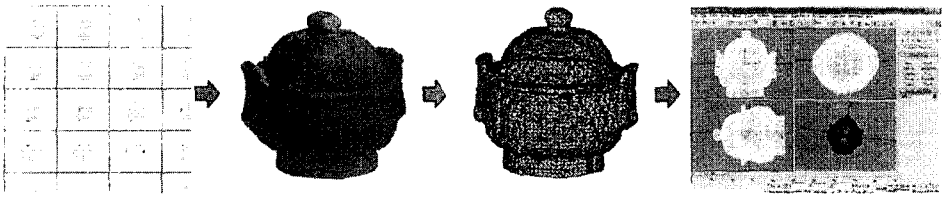


(a) (b) (c)

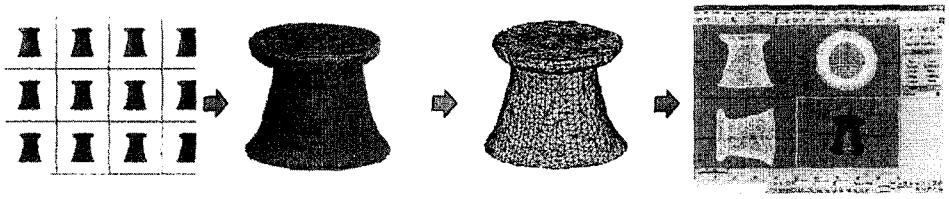
**Fig. 9** Triangle adding patterns. (a) Two new active edges are formed and one triangle is generated from one active edge. (b) One new active edge and one triangle is generated from two active edges. (c) No active edge is formed and one triangle is generated from three three active edges



(a) Korean traditional art, "Kaya duck-shaped crockery"



(b) Korean traditional art, "Baekja incense burner"



(c) Korean traditional art, "Kaya-kidae"

**Fig. 10** Three-dimensional shape reconstruction results of archaelolgical heritages. The left images are the thnmbnails of achieved original scenes from proposed image acquisition system. The two images of the middle shows the carved shapes and the generated polygon models in turn, and the images of right side shows the successful transplantation to the representative 3-D commercial software 3D-MAX

Figure 9 shows three different ways of formation of new triangles ; where (a) shows the case where one active edge is frozen while two new active edges are generated ; (b) shows two active edges becoming one ; and finally (c) shows three active edges forming a new triangle which terminates the iteration. Thus, the iteration begins by forming the, "seed" triangle with three active edges, and new triangles are formed from these active edges. The triangulation terminates when there are no more active edges as shown in Fig. 12(c).

## 5. Results

The object regions of all images are segmented from their backgrounds, and shape-from-silhouette method is performed with their view-angle information. Three-dimensional shapes rendered with the small cubes and their images are shown in Figure 10. The images on the left side show thumbnails of images acquired from the proposed image acquisition system. The carved shapes of objects are shown on the right side, that are rendered with the small cubes to express the volume. In Addition, Korean traditional heritage arts in this Figure are also available on the Internet.

Polygon models of these shapes generated by the triangulation algorithm mentioned above, can also be manipulated in commercial software as shown in Figure 10. Generated mesh models with thousands of triangles are exported via the stereo-lithography file format, i.e., *.STL. Models are successfully exported to the representative commercial 3-D design software 3D-MAX.

## 6. Conclusions

We proposed three-dimensional shape reconstruction system with promising results. The advanced iterative triangulation algorithm efficiently generated the polygon (mesh) models, and these models are successfully exported to commercial three-dimensional tools such as 3D-MAX by stereo-lithography file format. We expect that these results of the work can be adapted in providing initial CAD model or in 3D game applications or in reverse engineering after more progressions.

The advantages of the proposed system based on shape-from-shading technique stem from (1) that it is inexpensive and (2) that state of the object is not disturbed. Henceforth, reverse engineering technique generally used direct approaches that needed to be touched (cf. Cho et al., 2000). However, the proposed image-based shape reconstruction system does not require the object to be touched, that may actually damage the object. Even laser-based techniques such as the one used for the, "Digital Michelangelo Project" (Levoy et al., 2000), may also damage certain light-sensitive objects although highly accurate (and expensive). Our proposed system can be considered as an alternative to the more expensive laser-based systems as it works only with images captured under a normal lighting condition.

## Acknowledgment

This research was supported in part by a grant from the Korea Ministry of Culture and in part by the BK 21 program. We would also like to thank the anonymous reviewers for their detailed review of this manuscript.

## References

Ahuja, N. and Veenstra, J., 1989, "Generating Octrees from Objects Silhouetts in Orthographic View," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 11, No. 2, pp. 137~149.

Amenta, N., Bern, M. and Kamvysselis, M., 1998, "A New Vornoi-Based Surface Reconstruction Algorithm," in *Proc. SIGGRAPH 98*, pp. 415~422.

Barequet, G. and Sharir, M., 1996, "Piecewise-Linear Interpolation Between Polygonal Slices," *Computer Vision and Image Understanding*, Vol. 63, pp. 251~272.

Bernardini, F., Mittleman, J., Rushmeier, H., Silva C. and Taubin, G., 1999, "The Ball-Pivoting Algorithm for Surface Reconstruction," *IEEE Transactions on Visualization and Computer Graphics*, Vol. 5, No. 4, pp. 349~359.

Chien, C. H. and Aggarwal, J. K., 1986, "Volume/Surface Octrees for the Representation of Three-Dimensional Objects," *Computer Vision, Graphics, and Image Processing*, Vol. 36, pp. 100~113.

Cho, M. W., Seo, T. I., Kim J. D. and Kwon, O. Y., 2000, "Reverse Engineering of Compound Surfaces Using Boundary Detection Method," *KSME International Journal*, Vol. 14, No. 10, pp. 1104~1113.

Curless, B. and Levoy, M., 1996, "A Volumetric Method for Building Complex Models from Range Images," *Computer Graphics*, Vol. 30 (Annual Conference Series), pp. 303~312.

Dyer, C. R., 2001, "Volumetric Scene Reconstruction from Multiple Views," *Foundations of Image Understanding*, L. S. Davis, (ed.), Kluwer, Boston, pp. 369~489.

Eggert, D. W., Fitzgibbon, A. W. and Fisher, R. B., 1998, "Simultaneous Registration of Multiple Range Views for Use in Reverse Engineering of CAD Models," *Computer Vision and Image Understanding*, Vol. 69, No. 3, pp. 253~272.

Fortune, S., 1992, "Vornoi Diagrams and Delaunay Triangulations," *Computing in Euclidean Geometry*, Ding-Zh Du and Frank Hwang (ed.), World Scientific, Singapore, pp. 225~265.

Hoppe, H., DeRose, T., Duchamp, T., McDonald, J. and Stuetzle, W., 1992, "Surface Reconstruction from Unorganized Points," in *Proc. of ACM SIGGRAPH*, pp. 71~78.

Karbacher, S., Laboureux, X., Schon, N. and Hausler, G., 2001, "Processing Range Data for Reverse Engineering and Virtual Reality," in *Proce. Third International Conference on 3-D Digital Imaging and Modeling*, pp. 314~321.

Levoy, M., Rusinkiewicz, S., Ginzton, M., Ginsberg, J., Pulli, K., Koller, D., Anderson, S., Shade, J., Curless, B., Pereira, L., Davis, J. and Fulk, D., 2000, "The Digital Michelangelo Project: 3D Scanning of Large Statues," in *SIGGRAPH 2000 Proc. Computer Graphics*, pp. 131~144.

Lin, C., Yang, D. and Chung, Y., 2001, "A Marching Voxels Method for Surface Rendering of Volume Data," in *International 2001. Proc. Computer Graphics*, pp. 306~313.

Martin, W. N. and Aggawal, J. K., 1983, "Volumetric Description of Objects from Multiple Views," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-5, No. 2, pp. 150~158.

Matsumoto, Y., Terasaki, H., Sugimoto, K. and Arakawa, T., 1997, "A Portable Three-Dimensional Digitizer," in *Proc. International Conference on Recent Advances in 3-D Digital Imaging and Modeling*, pp. 197~204.

Oblonšek Č. and Guid, N., 1998, "A Fast Surface-Based Procedure for Object Reconstruction from 3D Scattered Points," *Computer Vision and Image Understanding*, Vol. 69, No. 2, pp. 185~195.

Piegel, L. A. and Tiller, W., 2002, "Algorithm for Finding All Nearest Neighbors," *Computer-Aided Design*, Vol. 34, pp. 167~172.

Potmesil, M., 1987, "Generating Octree Models of 3D objects from Their Silhouette in a Sequence of Images," *Computer Vision, Graphics, and Image Processing*, Vol. 40, pp. 1~29.

Sappa, A. D., 2000, "Incremental Multiview Integration of Range Images," in *Proc. 15th International Conference on Pattern Recognition*, Vol. 1, pp. 546~549.

Song, S. M., 2001, "Method and Apparatus for Visualization and Manipulation of Real 3-D Objects in Networked Environments," *U.S. patent pending*.

Srivastava, S. K. and Ahuja, N., 1990, "Octree Generation from Object Silhouette in Perspective View," *Computer Vision, Graphics, and Image Processing*, Vol. 49, pp. 68~84.

Stamos, I. and Allen, P. A., 2000, "3-D Model Construction Using Range and Image Data," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, Vol. 1, pp. 531~536.

Tosovic, S. and Sablating, R., 2001, "3D Modeling of Archaeological Vessels Using Shape from Silhouette," in *Proc. Third International Conference on 3-D Digital Imaging and Modeling*, pp. 51~58.

Weik, S., 2000, "A Passive Full Body Scanner Using shape from Silhouettes," in *Proc. 15th International Conference on Pattern Recognition*, Vol. 1 : 2000, pp. 750~753.