Shape-from-Silhouette/Stereo and Its Application to 3-D Digitizer

Y. Matsumoto, K. Fujimura and T. Kitamura

SANYO Electric Co., Ltd. JAPAN

Abstract. A novel three-dimensional reconstruction algorithm "Shape-from-Silhouette/Stereo (SFS^2)" is presented in this paper. This algorithm combines Shape-from-Silhouette with stereo based on simple voting-localizing operations in a voxel space. In this algorithm, Shape-from-Silhouette roughly estimates the shape of the target object first, and then multi-eye stereo is applied within the estimated area to refine the shape. This algorithm overcomes the shortcomings of each algorithm and offers the following: 1) More precise shape reconstruction than simple Shape-from-Silhouette; 2) Quicker processing than simple stereo-based reconstruction; and 3) Better noise reduction than multi-eye stereo. Our experiments showed that SFS^2 is highly practical for generating 3D models of real objects.

1 Introduction

Recently it has become possible for ordinary personal computer (PC) users to handle three-dimensional (3-D) images and to create their own virtual world. In addition, the use of VRML (Virtual Reality Modeling Language), the standard language for 3-D image communication on the Internet, is spreading widely. On the Internet, ordinary PC users are not only receivers but also providers of information, and may wish to input 3-D images. State-of-the-art techniques for digitizing a real object fall into two categories: contact-based digitizers and non-contact scanners. A typical contact-based digitizer has a stylus, which is used to touch the target object to be measured. On the other hand, most non-contact scanners are based on active methods, using special ray projectors such as a laser[1], sheet-of-light[2,3], or structured-light. However, some of these are very expensive, while others require users to have skill and know-how.

We have been developing an inexpensive, compact and skill-free 3-D scanner based on passive sensing methods. Many studies have been reported in this area and categorized with the "Shape-from-X" framework[4,5,6,7,8]. Shape-from-Silhouette[4,9] is one of the simplest and most powerful techniques, however, it can not reconstruct concave parts of the object. Stereo[6] is another powerful technique but the correspondence search, the main task of stereo, is very time-consuming.

This paper presents a novel algorithm — Shape-from-Silhouette/Stereo (or SFS^2) — which reconstructs the 3-D shape of objects from a series of 2-D images.

G. Bertrand, M. Couprie, L. Perroton (Eds.): DGCI'99, LNCS 1568, pp. 177–188, 1999.

In this algorithm, Shape-from-Silhouette and a stereo technique are combined based on voting-localizing operations in a three-dimensional voxel space. More specifically, Shape-from-Silhouette is applied first to give a rough estimate of the object presence area, and then the stereo process refines the shape.

The combination overcomes the shortcomings of each algorithm and the main advantages are as follows: 1) More precise shape reconstruction than simple Shape-from-Silhouette; and 2) Quicker processing than simple stereo processing. In addition, bounding the probable area by Shape-from-Silhouette leads to: 3) Better noise reduction than conventional multi-eye stereo.

This paper is organized as follows: Section 2 briefly reviews related works. In Section 3, details of the algorithm are described. Section 4 overviews a prototype system for testing the algorithm. Experiments and preliminary evaluations are reported in Section 5, and Section 6 summarizes the paper.

2 Related Works

This section briefly describes typical related works: Shape-from-Silhouette and voxel-based stereo.

2.1 Martin's work

Martin and Aggarwal[10] presented a volumetric description technique suitable for deriving 3-D object representations from two-dimensional images. The technique is based on the idea that by occluding contours from an image sequence with viewpoint specifications, a bounding volume is determined that approximates the object generating the contours.

The technique includes two major stages: initial representation creation and continuing representation refinement. In the initial representation creation, two image planes are selected and the initial *volume segments*, which are a set of line segments with a linked-list structure, are generated. With the continuing representation refinement, the volume segments are projected onto the other image plane, which constrain the extent of the object further so that the volume segments can be modified to represent the object with higher fidelity.

2.2 Zheng's work

Zheng and Kishino[11] reported on an analysis of areas unexposed to contours and a method of detecting them. They focused on the fact that the unexposed area, which might be a concave or planar surface, was numerically related to the nonsmoothness of the contour distribution.

To detect unexposed area, they pointed out the fact that the line of sight will touch two convex tangent points at the same time and then jump from the first mountain to the second along a particular viewing direction. Second-order differential filtering is applied to the trace of the contour point of the epipolar-plane image. The jumping points (or *cusp points*) are extracted as the local maxima in the filtered sequence.

2.3 Seitz's work

Seitz and Dyer[12] proposed a voxel-based shape reconstruction technique which is able to cope with large changes in visibility and its modeling of intrinsic scene color and texture information. They introduced the ordinal visibility constraint and this technique traverses a discretized 3-D space in "depth-order" to identify voxels that have a unique coloring that is constant across all possible interpretations of the scene.

The main advantage of their algorithm is that it has no occluding problem, which afflicts most stereo algorithms. A voxel model of an object is reconstructed by estimating the consistency of each voxel with the likelihood ratio, taking occlusion between voxels into account.

3 Shape from Silhouette/Stereo

Shape from Silhouette/Stereo (SFS^2) is an algorithm for reconstructing the three-dimensional shape of target objects from a series of two-dimensional images. In this scheme, Shape-from-Silhouette[10] and stereo[6] are combined based on voting-localizing operations onto three-dimensional voxel space.

SFS^2 comprises two major stages: 1) Rough shape estimation with voting-based Shape-from-Silhouette, and 2) Precise shape reconstruction based on voxel-based stereo. This combination offers the following advantages:

1. More precise shape reconstruction than simple Shape-from-Silhouette
   A naive Shape-from-Silhouette algorithm reconstructs only convex parts of objects. By applying a stereo technique, concave parts can be reconstructed.
2. Quicker processing than simple stereo-based reconstruction
   Since Shape-from-silhouette provides a rough estimate of the target shape, the succeeding stereo process is applied only to the bounded area, or in other words, search areas for stereo-matching are greatly reduced.
3. Higher noise reduction property than multi-eye stereo
   The preceding Shape-from-Silhouette limits the search area for stereo matching, The risk of wrong matching results and the resultant noise are also reduced. This improves the accuracy.

An additional advantage of SFS^2 is as follows. Assume a very simple 3-D digitizing configuration comprising a monoscopic camera and turntable, where the object is placed on the turntable and all-around views are captured by rotating the table stepwise. With this setup, the captured scene is not stationary; the object moves while the background is stable. Therefore a simple stereo algorithm does not work appropriately with this configuration. Since SFS^2 separates the object area from the background in the input image, it is free from this problem and this leads to the following advantage:

4. Enhanced applicability: Shape-from-Silhouette separates the object and background. Owing to this property, SFS^2 does not constrain the system configuration. SFS^2 is applicable to a multi-eye camera system, moving-camera system as well as monoscopic stationary camera system with a turntable.

Note that SFS^2 differs from the related works mentioned earlier concerning the following points: 1) Voting-based Shape-from-Silhouette is embedded for enhancing robustness, which allows a few silhouette errors, whereas Martin's work and Zheng's work assume errorless silhouettes. 2) Voxel-based stereo is performed with block-matching in order to allow noises in input images although Seitz's work detects color similarity in a pixelwise manner.

3.1 Formalization

Here, we make the following assumptions:

1. A series of images of all-around views of a target object is input.
2. Camera positions (or observation points) for all images are known.
3. Internal camera parameters are known.
4. The probable area of object presence can be defined in the 3-D space.

Hereafter, the description is based on the following notation:

- V : a set of voxels (3-D voxel space).
- V_s : a set of voxels which correspond to the surface of the target object in the voxel space V; $V_s \subseteq V$ (object surface).
- I^i: i-th input image plane. I^i_o and I^i_s are the i-th original input image (gray-scale or color) and its silhouette image (B/W) respectively. Furthermore, I^i_{ss} and I^i_{sb} are the silhouette and background area of I^i_s respectively.
- $p(v, i)$: the projection point of v on I^i.

Original images and silhouette images give important clues and constraints concerning the object presence area in the voxel space. A voxel of V_s is projected on the silhouette part of input images (*silhouette constraint*). On the other hand, the pixels of input images on which a voxel of V_s is projected are expected to have the same color or intensity (*color constant*), assuming 1) homogeneous illumination environment, 2) noise-free image, and 3) Lambertian surface.

Silhouette constraint condition We define the local silhouette constraint for $v \in V_s$ on I^i as

$$Cs(v,i) = \begin{cases} 1 \text{ if } p(v,i) \in I^i_{ss} \\ 0 \text{ otherwise.} \end{cases}$$

The total *Silhouette constraint* is:

$$\prod_i \prod_{v \in V_s} Cs(v,i) = 1. \tag{1}$$

Color constant condition We define the observability of v on I^i as

$$Co(v,i) = \begin{cases} 1 \text{ if } v \text{ is projected on } I^i \text{ without occlusion} \\ 0 \text{ otherwise.} \end{cases}$$

Also, we define the observable image set for v as $O^v = \{i | Co(v,i) = 1\}$.

The *Color constant constraint* is:

$$\sum_{v \in V_s} \sum_{i \in O^v} D(v,i) = 0, \tag{2}$$

where $D(v,i)$ represents an evaluation of color dispersion for v. A typical example for D(v,i) is $\sum_{j=i-m}^{i+m} \sum_{blockarea} \{v(p(v,i)) - v(p(v,j))\}^2$, where $v(p(v,j))$ is the color value at $p(v,j)$.

The target problem is to determine the set V_s using these conditions. The two major stages of SFS^2 determine the localized voxel set V_s from V. The Shape-from-Silhouette stage produces an intermediate set of voxels V_b which gives a rough estimate of the object presence area based on the silhouette constraint condition. The voxel-based stereo further localizes V_s ($\subseteq V_b$) with the color constant condition.

3.2 Stage 1: Shape-from-Silhouette

As mentioned above, a silhouette of a target object gives important constraints of the object shape. More precisely, a point in the 3-D space projected onto the inside of a silhouette may be occupied by the object, whereas a point which is projected onto the outside of the silhouette is definitely free. The 3-D space occupied by the object can be estimated more precisely by using multiple silhouette images taken at various observation points.

For clarity, we first briefly describe simple Shape-from-Silhouette, and then explain voting-based Shape-from-Silhouette which is embedded in SFS^2.

Simple Shape-from-Silhouette In simple Shape-from-Silhouette, the object presence area can be estimated in a manner similar to the process of carving a sculpture from a block of wood, namely, removing the parts which are apparently out of the object area. Assuming a voxel space, this simple algorithm reconstructs the three-dimensional shape of the target object with the steps shown in Fig. 1 (a).

Voting-based Shape-from-Silhouette The simple and conventional "Shape-from-Silhouette" method regards silhouettes as strong constraints. This means the method assumes errorless silhouettes.

In practical applications, however, silhouettes may contain several errors. In this case, the silhouette errors seriously affect the shape reconstruction. If part of the silhouette of the target object is chipped, for example, the resultant shape has considerable errors.

For the above reason, the Silhouette constraint should be relaxed for practical applications. SFS^2 adopts an extended algorithm to handle silhouettes with

```
For all voxels:            — (initialization)
  $flag(v) := ON$.
  $V_b := \phi$.
For all voxels:            — (carving)
  For all silhouettes:
    If $C_s(v, i) = 0$,
    Then $flag(v) := OFF$.
For all voxels:            — (localization)
  If $flag(v) = ON$,
  Then $V_b := V_b \cup \{v\}$.
```

(a)

```
For all voxels:            — (initialization)
  $score(v) := 0$.
  $V_b := \phi$.
For all voxels:            — (voting)
  For all silhouettes:
    If $C_s(v, i) \neq 0$,
    Then $score(v) := score(v) + 1$.
For all voxels:            — (localization)
  If $score(v) \geq Threshold$,
  Then $V_b := V_b \cup \{v\}$.
```

(b)

Fig. 1. Algorithm for simple Shape-from-Silhouette (a) and voting-based Shape-from-Silhouette(b).

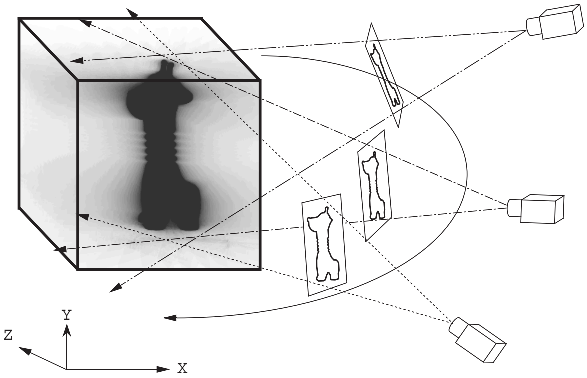

Fig. 2. Voting on the voxel space: the density shows the score of each voxel.

errors for wider applicability, where the silhouettes are treated as weak constraints. In the extended algorithm, the shape estimation is performed with a voting-localizing scheme as shown in Fig. 1 (b).

This extended approach has a great advantage over the simple method; silhouette images including several errors do not seriously affect the object shape obtained. Moreover, the voting-localizing scheme gives a general framework where the conventional approaches can be treated as a special case of voting-based Shape-from-Silhouette: only full-marked voxels are judged to be in the object area.

3.3 Stage 2: Voxel-based Stereo

If the target object contains a concave part, Shape-from-Silhouette does not reconstruct the 3-D shape correctly. Incorporating a stereo technique, SFS^2 detects and recovers the concavity as long as the object surface has sufficient texture information. This stage consists of the following substages: 1) partial surface estimation, and 2) total model generation.

Partial surface estimation The basic idea of this stage is as follows: if a voxel v corresponds to the surface of the object, the color of its projection point on each image plane is similar to each other. Based on this idea, errors of these colors are accumulated on each voxel first (error voting). Then, voxels with the minimum accumulated error are chosen. This set of voxels can be determined for each observation point and is regarded as the partial surfaces for the observation point. Now we refer to the partial surfaces for the i-th observation point as V_{sp}^i.

We assume that the observable image set O^v includes at least n consecutive members for any $v \in V_{sp}^i$ for any observation point i. This means that the occlusion problem can be avoided for n consecutive input images for any voxels of any partial surfaces.

Taking lighting fluctuation and image noises into account, this assumption allows SFS^2 to relax the color constant condition (Eq. 2) as follows:

$$\sum_{v \in V_s} \sum_{i \in O_c^v} D(v,i) \rightarrow min, \tag{3}$$

where O_c^v is n consecutive members of O^v.

Here, let V_{rv}^i be a set of voxels along a ray from the i-th observation point through v. For the i-th observation point, the partial surface V_{sp}^i is identified with the procedure shown in Fig. 3 (a). The steps are repeatedly applied to all observation points. Eventually as many partial surface sets are obtained as the original images.

```
For all voxels $v \in V_b$:  — (initialization)
    $score(v) := 0.$
    $V_{sp}^i := \phi.$

For all voxels $v \in V_b$:  — (voting)
    $score(v) := score(v) + D(v,i).$

For all voxels $v \in V_b$:  — (localization)
    If $score(v) = MIN_{V_r^i}(score(v)),$
    Then $V_{sp}^i := V_{sp}^i \cup \{v\}.$
```

(a)

```
For all voxels $v \in V_b$:      — (initialization)
    $score(v) := 0.$
    $V_c := \phi.$
For all voxels $v \in V_b$: :    — (voting)
    For all observation points:
        If $L(v,i) \geq L(v',i), [v \in V_r^i, v' \in V_r^i \cap V_{sp}^i],$
        Then $score(v) := score(v) + 1.$
For all voxels $v \in V_b$:      — (localization)
    If $score(v) \geq Threshold,$
    Then $V_c := V_c \cup \{v\}.$
```

(b)

Fig. 3. Algorithm for partial surface estimation (a) and total model generation (b).

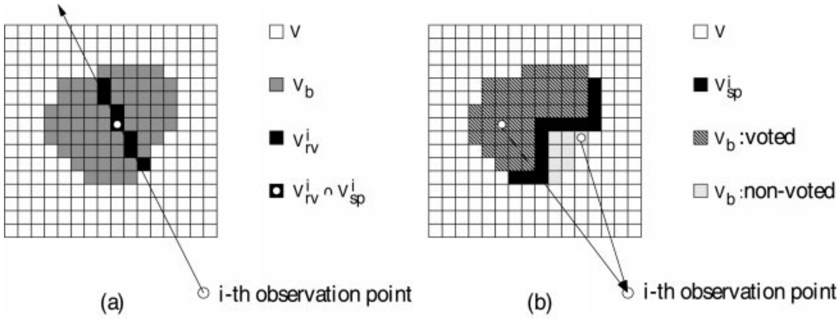

Fig. 4. (a) Partial surface estimation (partial surface localization on a ray): the voxel with the minimum score on a ray is chosen as the partial surface. (b) Total model generation (voting step at the i-th observation point): only voxels behind the partial surface are voted.

Total model generation This substage generates a total 3-D model from partial surfaces. Although a partial surface is calculated at each observation point, each may have errors. Integrating the partial surfaces with the voting-localizing scheme, the errors are rounded to obtain the total model of the object with higher accuracy.

Let $L(v, i)$ be the Euclidean distance between v and the i-th observation point. V_c is a set of voxels which gives the volumetric representation of the object. The integration process is shown in Fig. 3 (b). After the process, V_s is easily obtained by detecting the boundary of V_c. This procedure does not provide the exact solution which satisfies the relaxed color constant condition but gives a good approximation.

4 Application of SFS^2 to 3-D Digitizer

We have built a 3-D digitizer to test the applicability of the SFS^2 algorithm to 3-D modeling. The digitizer comprises a turntable, a monocular camera and a personal computer (Fig. 5). This digitizer is capable of automatically generating the 3-D texture-mapped model of the target object.

The 3-D digitizing process consists of five major steps: 1) calibration, 2) image capturing, 3) silhouette extraction, 4) shape modeling, and 5) texture acquisition. SFS^2 is embedded in the shape modeling process. The other processes are briefly shown below[1].

4.1 Calibration

In this process, camera position relative to the turntable is automatically calculated using an object of known shape placed on the turntable. By rotating the turntable, the object is captured from various observation points (original

[1] For the details, refer to [9,13].

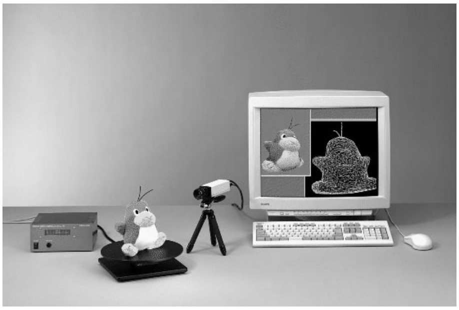

Fig. 5. System overview

images). In the current implementation, a calibration panel with circle patterns is used. After detecting each circle position, the Hough transform[13] is applied to calculate camera position for enhancing robustness.

4.2 Image capturing

In this step, a series of images is captured to cover all views of the object. In addition, the background image is captured as well. The background image is almost the same as the original images except that it excludes the target object, and is used for extracting silhouettes.

4.3 Silhouette extraction

A silhouette image corresponding to each object image is generated based on subtraction operations. In this step, naive silhouette extraction, or pixel-level extraction, is performed first. Then, supplemental silhouette extraction, i.e., region-level extraction, is applied, which improves the accuracy of the silhouette obtained. In the region-level extraction, the average of absolute subtraction at each pixel in a region is calculated to judge if the region corresponds to the object or not.

4.4 Shape modeling

The shape modeling step consists of two processes: volumetric modeling and surface representation conversion. SFS^2 is applied to volumetric modeling. In the surface representation conversion, many initial polygons are generated first by connecting adjacent surface polygons. Then, a polygon reduction operation merges polygons to reduce the total number of polygons.

4.5 Texture acquisition

The texture of a patch can be derived from multiple original images because a patch can be seen from various viewpoints. Therefore, an original image should be identified for each polygon to derive its texture.

In order to determine the source image appropriately, the following points should be considered. 1) A larger amount of texture information in the source image leads to a better derivation of texture. 2) Smoothness of texture on the boundary of patches is required for concealing the patch edges. To meet both requirements, we solve the texture acquisition problem based on an energy minimization method[9].

5 Experiments and Preliminary Evaluations

We tested the SFS^2 algorithm with ceramic toys (a miniature castle and squirrel). In the experiments, the step of table rotation was ten degrees so 36 original images were taken. The image size was 640×480. The resolution of the voxel space was $320 \times 320 \times 240$. The color constant condition was evaluated by block-matching with a 15×15 mask. The number of consecutive member O_c^v was assumed to be five.

Fig. 6 and Fig. 7 show examples of original images of each object, wireframe 3-D models obtained and resultant texture-mapped 3-D images. The modeling results show that the SFS^2 reconstructs the shape with practical accuracy.

The computational time of each stage is shown in Table 1. The data were measured on Dell's Optiplex GX1-400MT personal computer (CPU: Pentium II 400MHz, Memory: 128MB). The measurements show that computation time for voting-stereo is still dominant. However, we consider that a further speed increase will be achieved with a multi-resolution technique, where stereo processing with lower resolution bounds the area of voxels for stereo processing with higher resolution.

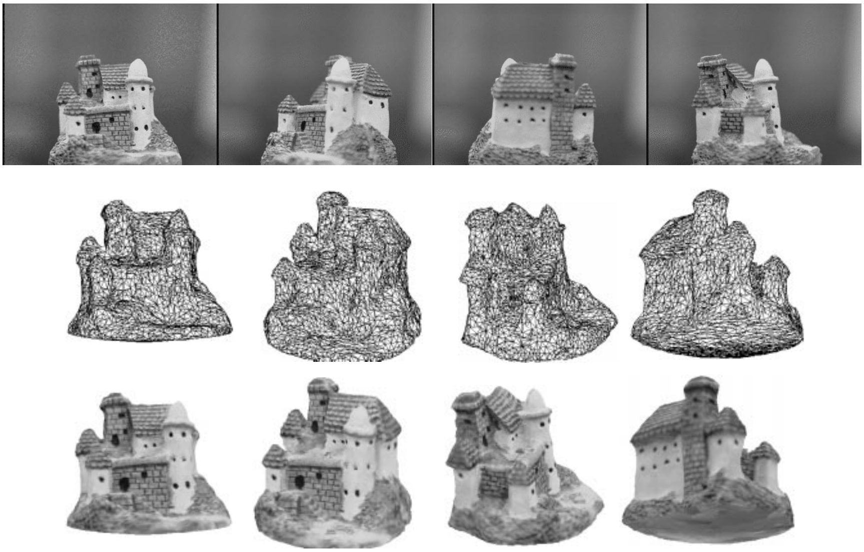

Fig. 6. Examples of original images and modeling results (castle)

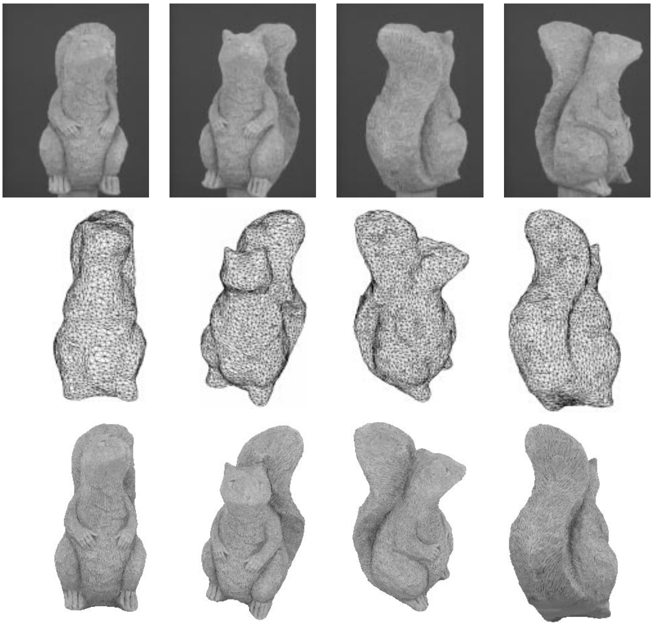

Fig. 7. Examples of original images and modeling results (squirrel)

Table 1. Analysis of processing time

Step		Processing time (sec)	
		Castle	Squirrel
Silhouette extraction		53	31
Shape modeling	Shape-from-Silhouette	93	36
	Voting-based stereo	1,230	1,870
	Surface representation	106	38
Texture acquisition		43	43

6 Summary

A novel three-dimensional reconstruction algorithm "Shape-from-Silhouette/ Stereo (SFS^2)" has been presented in this paper. This algorithm combines the Shape-from-Silhouette with Shape-from-Stereo based on a voting-localizing scheme with the voxel space.

The main advantages of this algorithm are as follows: 1) Quicker processing than a simple stereo-based reconstruction, 2) More precise shape reconstruction than a simple Shape-from-Silhouette, and 3) Better noise reduction than multi-eye stereo. An additional advantage is that this algorithm enables the stereo technique to be applied to the digitizer based on the monoscopic camera system with a turntable, where the object moves against the stationary background.

We tested this algorithm with toys to evaluate the applicability. The experiment showed that SFS^2 is highly practical for generating 3-D models of real objects.

One of the current problems of the algorithm is that sometimes the surface of a thin part may be missing due to errors in partial surfaces. This error can be detected by checking consistency between two silhouettes: one generated from the final model and the other from the probable object area estimated by Shape-from-Silhouette. After detecting the inconsistency, adaptive thresholds for localizing the final model should be applied. Computational time should be also reduced. This will be achieved by applying a multi-resolution technique.

References

1. R B Lewis and A R Johnston. A scanning laser range finder for a robotic vision. *Int. Joint Conf. Artifical Intelligence*, pages 962–968, 1972.
2. G T Agin and T O Binford. Computer description of curved objects. *IEEE Trans. Comput.*, C-25(4):439–449, 1976.
3. M Idesawa et al. Scanning moire method and automatic measurement of 3-d shapes. *Appl. Opt.*, pages 2151–2162, 1977.
4. P Giblin and R Weiss. Reconstruction of surfaces from profiles. *Int. Conf. Comput. Vision*, pages 136–144, 1987.
5. B Horn. Obtaining shape from shading information. *Psychology of Computer Vision (P.H.Winston ed.), McGraw-Hill*, pages 115–155, 1975.
6. D Marr. *Vision*. W. H. Freeman and Co., 1982.
7. S Ullman. The interpretation of structure from motion. *Proc. the Royal Society of London*, pages 405–426, 1979.
8. A Witkin. Recovering surface shape and orientation from texture. *Artificial Intelligence*, 17:17–45, 1981.
9. Y Matsumoto et al. A portable three-dimensional digitizer. *Int. Conf. Recent Advances in 3-D Digital Imaging and Modeling*, pages 197–204, 1997.
10. W N Martin and J K Aggarwal. Volumetric descriptions of objects from multiple views. *IEEE Trans. PAMI*, 5(2):150–158, 1983.
11. J Y Zheng and F Kishino. 3d models from contours: Further identification of unexposed areas. *Int. Conf. Pattern Recognition*, pages 349–353, 1992.
12. S Seitz and C Dyer. Photorealistic scene reconstruction by voxel coloring. *CVPR '97*, pages 1067–1073, 1997.
13. D Ritter and Y Matsumoto. 3d-surface reconstruction with a hand held video camera. *Image Multidimensional Digital Signal Processing '98 (IMDSP'98)*, pages 5–8, 1998.