〔非公開〕

ＴＲ－Ｃ－００８７

Ａｃｑｕｉｒｉｎｇ　３Ｄ　Ｍｏｄｅｌｓ　ｆｒｏｍ Ｓｅｑｕｅｎｃｅｓ　ｏｆ　Ｃｏｎｔｏｕｒｓ

鄭　　絳宇
Jiang Yu Zheng

１９９３．３．２５

ＡＴＲ通信システム研究所

Acquiring 3D Models from Sequences of Contours

Jiang Yu Zheng

Key Words: 3D graphics model, rotation, shape from contour, spatio-temporal analysis, silhouette unexposed region, smoothness, surface categorization, projections, face modeling.

Abstract

This report explores shape from contour for acquiring 3D graphics models. In this method, a continuous sequence of images is taken as an object rotates. A smooth convex shape can be estimated instantaneously from its contour and by the first derivative of contour movement (trace of contour, or contour distribution with time). We also analyze shapes that do not satisfy the conditions of smoothness and visibility, which are indispensable for modeling an object. A region that does not expose as contour yields a nonsmoothness in the tracked contour movement. We can, thus, detect such a region by contour distribution filtering, and extract its accurate location by computing the left and right derivatives of the distribution. This has not been studied previously. These unknown regions are obtained for further investigation using other visual cues. A general approach for building a geometrical object model using contours is then described. The entire process from silhouettes to a 3D model is based on local computation; this is promising for producing shapes in real time. Our direct goal is to establish 3D graphics models of human faces for the growing needs of visual communications. We have obtained some good results.

References

[1] P. Giblin, R. Weiss, "Reconstruction of surface from profiles", Proc. 1st ICCV, pp. 136-144, 1987.

[2] H. Baker, and R. Bolles, "Generalizing epipolar-plane image analysis on the spatiotemporal surface", Proc. CVPR-88, pp.2-9, 1988.

[3] M. Kass, and A. Witkin, "Constraints on deformable models: Recovering 3D shape and nonrigid motion", Artificial Intelligence 36, pp. 91-123, 1988.

[4] A. Blake, G. Brelstaff, "Geometry from specularities", 2nd ICCV, pp.394-403, 1988.

[5] J. J. Koenderink, "Solid shape", The MIT Press, 1990.

[6] J. Y. Zheng, and S. Tsuji, "From Anothorscopic Perception to dynamic vision", Proc. 1990 IEEE Conf. Robotics and Automation, Vol.2, pp.1154-1160, 1990.

[7] J.Y. Zheng, and S. Tsuji, "Panoramic representation of scenes for route understanding", Proc. 10th ICPR, Vol.1, pp.161-167, 1990.

[8] A. Pantland, "Automatic extraction of deformable part models", IJCV, Vol.4, no.2, pp. 107-126, 1990.

[9] L. Williams, "Performance-driven facial animation", SIGGRAPH90, pp.235-242, 1990.

[10] C. Tomasi and T. Kanade, "Shape without depth", 3rd ICCV, pp.91-95, Dec. 1990.

[11] R. Cipolla, and A. Black, "The dynamic analysis of apparent contours", Proc. 3th ICCV, pp. 616-623, Dec. 1990.

[12] D. Terzopoulos, and K. Waters, "Analysis of facial images using physical and anotomical models", Proc. 3rd ICCV, pp. 727-732, Dec. 1990.

[13] H. Harajima and F. Kishino, "Intelligent image coding and communications with realistic sensations", IEICE Transactions, Vol.E 74, No. 6, pp. 1582-1592, June, 1991.

[14] J.Y. Zheng, M. Nagashima, and F. Kishino, "Construct a 3D face model from rotational images", EIC Technique Report, IE91-87, pp.45-50, Nov., 1991.

[15] R. Vaillant and O. D. Faugeras, "Using extremal boundaries for 3D object modeling", PAMI, Vol. 14, No. 2, pp. 157-173. Feb. 1992.

[16] J.Y. Zheng, and F.Kishino, "Verifying and combining different visual cues into a complete 3D model", Proc. CVPR92, pp. 777-780, June, 1992.

[17] J.Y. Zheng, and F. Kishino, "3D model from contours: further identification of unexposed areas", Proc. 11th ICPR, Vol.1, pp. 349-353, Sept., 1992.

[18] J.Y. Zheng and S. Tsuji, "Panoramic Representation of scenes for route recognition by a mobile robot", IJCV, Vol.9, no.1, 55-77, 1992.

[19] S. Petitjean, J. Ponce, and D. J. Kriegman, "Computing exact aspect graphs of curved objects: Algebraic surfaces", IJCV, Vol.10, no.3, pp.231-255, Dec. 1992.

[20] R. Cipolla, and A. Blake, "Surface shape from the deformation of apparent contours" IJCV, Vol.9, no.2, pp.83-112, 1992.

[21] M. P. Carmo, Differential geometry of curves and surfaces, Prentice-Hall, Inc., New Jersey 1976.

[22] R. Wang, and H. Freeman, "Object recognition based on characteristic view classes", 10th ICPR, 1990.

[23] K. Ikeuchi, and T. Kanade, "Automatic generation of object recognition problems", Proc. IEEE 76(8): 1016-1035.

1. Introduction

In recent years, there have been increasing demands for developments in model-based coding, stereoscopic display, and graphics-generated virtual environments which require three-dimensional modeling of the human face as well as other objects. Application areas for them include visual communications, education and entertainment. We are attempting to generate 3D views in a virtual space for teleconferencing and video phones.

Laser stripes and other coded light projecting systems have been developed for acquiring accurate 3D shapes, but they still have the following drawbacks: (1) they receive unsatisfactory data from areas where the surface reflectance is low (such as black hair), (2) they usually require some special illumination (except when a laser is used) in a dark room or when a subject exists at a close distance, in order for the projected stripes to be extracted in the images, (3) they all fail for surfaces with strong specula reflectance, (4) they are difficult to calibrate and inaccurate for distant subjects (because a triangular measure is used), and (5) people do not feel comfortable when laser beam continually scans across their faces. Based on these facts, we plan to select a method capable of working in an ordinary environment with simple devices. We will focus on such capabilities as the sensing of distant subjects and sensing without special illumination, because these capabilities are promising for future on-site real-time modeling. As the first step, we explore a vision method using a single camera.

Shape from rotation is a natural way to observe an object. We look at an object while rotating it by hand, or, alternatively, we move around the object. Information on rotational movement and the axis of rotation is particularly easy to obtain from control of apparatuses such as turntables and swivel-chairs, magnetic position sensors attached on the subjects, and from input image sequences.

In the image sequence of a rotating object, there are different kinds of information ---- information such as contour, flow, stereo disparity and shading. Among them, the occluding contour (extreme boundary, extreme contour) in a 2D image directly reflects the 3D shape. Continuous contours can yield surfaces, more than only the edges and points generated by shape from motion [10]. They are less influenced by specula reflectance which is an annoying problem in shape from motion and stereo schemes [4]. At a smooth surface point, the surface normal is perpendicular to both the line of sight and its projected contour in the image. As the view point moves, the moving contour in the image corresponds to a 3D curve called a rim that shifts on the surface.

The efforts made on this topic so far have focused on local shape estimation, mainly concerning differentiable shapes (curves or surfaces) whose shapes and curvatures are computable. The scheme to recover shape from contour has been formalized in many ways under basically the same principle. Giblin et al. described a scheme to recover the shape of a

planar curve from continuous contour points [1]. Cipolla et al. generalized the relative motion between camera and object from pure rotation to arbitrary translation and rotation, and applied their algorithm to linear camera motion which recovers simple surfaces [11].

Another problem studied by Vaillant et al. [15] as well as in [11] is how to distinguish an extreme contour from a fixed surface feature (surface mark, or object corner with a continuous surface but a discontinuous normal) among a set of edges extracted from images. Based on their results, one can further fuse multiple measurements of a fixed feature on a surface or compute the differential properties of a shape from contours. The idea is nothing more than checking whether the 3D positions of an edge estimated from several views are coincident, when at least three views are available.

A remaining problem of shape from contour is, however, how to deal with concave shapes which cannot be recovered from contours. Analyses so far have not considered the visibility at concave regions (when viewing silhouettes in images) nor how to detect them.

We use occluding contours to recover a shape because they offer good detectability in front of a homogeneous background. We further explore how to model a whole object instead of only local shapes from silhouettes [17]. The relative rotation, which is considered as the intrinsic motion for the shape from contour method, reveals more of the surface to the viewer. Finally, the model is represented by connected patches to be displayed using graphics tools [14].

The model recovery process should include both a qualitative aspect that determines where an unknown region is, or where a region with a single or multiple measure is, and a quantitative aspect that determines what the shape is inside these regions. We detect regions that are invisible when viewing only silhouettes. We have found that an unexposed area, possibly a concave, parabolic or planar surface, has its size numerically related to a nonsmoothness factor in the contour distribution (or a discontinuity factor in the image velocity of the contour). We detect these regions by filtering the contour sequence with time (or traces of contour points) according to the required model resolution. The extraction of unknown regions allows us to plan further acquisition of information inside the regions ---- that is, it allows us to look for other information such as fixed features, self-occluding edges, shading, and etc [16]. We believe this as an important step toward automatic exploration of a model by an active vision approach.

A general approach for dealing with silhouettes is given in the following. Based on a surface categorization by their appearances in image sequences, we segment the silhouette distribution into sub-distributions where each sub-distribution corresponds to either a surface with good attributes such as smoothness and visibility or a 3D edge. The shape is represented by triangulated patches connecting 3D points estimated from continuous silhouettes. The entire process (from images to 3D points) applies local computation to the image data so that neither

a big memory nor a general parameter from the whole sequence is required. It is possible to achieve the successive processing of image data in real-time if a pipe-line system is available.

According to the geometric properties of different shapes and their detectabilities in the images, we categorizes edges extractable in the image (see Fig. 1). In the following, an occluding contour viewed in front of a background (with different albedo) is referred to as a silhouette (or contour) and an occluding contour inside the object occupied area in an image is referred to as a self-occluding contour, although in the principle of differential geometry they have the same behavior in the space. (In many cases, a self-occluding contour here corresponds to, in a line drawing, a curve accompanied by some 'cusp point' (see Koenderink [5])). We deal with them separately for the reason that they have different detectabilities in sensing. The silhouette is guaranteed to be extracted in every image with some effort such as adjusting the threshold or focusing the camera. The self-occluding contour, however, may appear near a concave region on a surface homogeneously with the same albedo; This yields weak discontinuity in the image intensity and can be extracted only at some particular viewing angles.

We use continuous image sequences because they provide more evidence as to the computation of surface attributes such as shape and differential properties than a few discrete images. In our case, shape is estimated from contours without computing epipolar planes. The way we represent surface data (or parameterize surfaces) by continuous contours simplifies the surface estimation and makes, in the practical sense, the acquisition of a whole object model possible.

An epipolar plane constraint has been chosen for tracking contours in shape estimation [11,15,20]. This constraint was originally designed to match fixed surface points in different images. For a smooth surface, however, it is unnecessary since silhouettes in successive images basically correspond to different rims on the surface. Epipolar parameterization may be feasible for orthogonal projection or linear motion under perspective projection. However, if a relative rotation is involved, an epipolar plane is only determinable between two successive frames and, hence, must be updated at each instance. It turns out that the epipolar plane is less valuable for surface parameterization when continuous contours come from motion with rotation.

We track contours along image pixel lines parallel to the rotational plane. The shape is estimated on two series of surface curves, namely S curves and Θ curves. Each S curve is projected as a contour in the image while each Θ curve is projected on the same pixel line of different images. The surface normal is also easy to determine.

In the second section, we will start from orthogonal projection and explain how to qualitatively and quantitatively detect regions that do not produce any contour. Section three considers a general approach of modeling an object from a sequence of contours. In the fourth

section, we will generalize the analysis to perspective projection. Experiments carried out on simple objects and complicated sculptures, and a system built to model real people will be shown in the fifth section.

2. Recovery of 3D Model from Contours

2.1 Previous Work

For simplicity, we first consider orthogonal projection. With orthogonal projection, an extreme contour is invariant when observed from the same direction. It has no relation with the camera distance, so we can just discuss the case of relative rotation between an object and the camera. The object rotates around an axis that is stationary to the camera. We assume that the camera is set so that the y axis of the image frame is parallel to the axis of rotation; otherwise, a simple rotation can transform the image to satisfy this condition. The 3D surface of the object can therefore be decomposed into many 2D curves on the cross sections of the surface with connected rotational planes.

Figure 2 shows one of such planar curves. The rotation axis is at the position O, the rotation angle of the object is θ, and the contour point on the planar curve viewed at the angle is denoted as P(X,Z). In the image frame, the projection of this point is at $p(\theta)$ and its image distance from the projection of the rotation axis x_0 is $\omega(\theta)$. We thus have the directions of vectors $\mathbf{x_0p}$ and $\mathbf{pP}$ as $(\cos\theta, \sin\theta)$ and $(-\sin\theta, \cos\theta)$, respectively, in the object centered coordinates system. The point P hence should be on the line of sight

$$L(\theta): \quad (X, Z)\bullet(\cos\theta, \sin\theta) = \omega(\theta) \tag{1}$$

Note that the point P shifts on the surface when the viewing direction changes. The shape on the slice is the envelope of lines of sight through the contour points $p(\theta)$ that are changing in the image sequence. The normal of the planar curve (X', Z') is perpendicular to the line of sight. We have

$$(\frac{\partial X}{\partial \theta}, \frac{\partial Z}{\partial \theta}) \cdot (\cos\theta, \sin\theta) = 0 \tag{2}$$

for a smooth planar curve where the first derivative exists. Taking the first derivative of Eq. (1), we obtain

$$(\frac{\partial X}{\partial \theta}, \frac{\partial Z}{\partial \theta}) \cdot (\cos\theta, \sin\theta) + (X, Z) \cdot (-\sin\theta, \cos\theta) = \frac{\partial \omega(\theta)}{\partial \theta} \tag{3}$$

Solving for X and Z by Eqs. (1-3), the coordinates of P(X,Z) in the object centered coordinates system can be given as

$$\begin{bmatrix} X \\ Z \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} \omega \\ \partial\omega/\partial\theta \end{bmatrix} \tag{4}$$

The depth of P(X,Z) from the image frame is $\omega'(\theta)$ and the radius of curvature of the planar curve is $\omega+\omega''$ [1]. In practice, ω can be located in the image to the precision of one pixel,

whereas $\omega'(\theta)$ is less precise because it is determined from digitized values. The method is therefore good at estimating the distance of 3D points from the axis or is accurate along the normal direction of the planar curve (determined by ω in Eq. (4)), but contains errors in the angle with respect to the axis or is inaccurate along the ray direction (determined by ω' in Eq. (4)).

2.2 Shape Unexposed to Contour

A good vision algorithm should be able to locate where an unknown part is in its output so that the area can be further explored by other methods or active viewing procedures. In the shape from contour method, a nonconvex shape on the moving plane, as well as other occluded parts, are unknown. Let us look at the behavior of the line of sight that views an occluded planar curve on a rotational plane (Fig. 3).

A concave or a linear segment of an object surface on a rotational plane exist between two convex segments and is occluded by them on both sides. As the surface rotates, the line of sight that is tangent to the convex segment shifts its touching point on the surface. Along a particular viewing direction θ_0, the line of sight will touch two convex tangent points at the same time and then jump from the first mountain to the second. At angle θ_0, the trace of image point $p(\theta)$, or the change of $\omega(\theta)$, is not smooth, although it remains continuous. The $\omega(\theta)$ is not differentiable at θ_0.

Figure 4 shows three objects set on a turntable for the purpose of modelling: a sphere, a cylinder and a box. A horizontal x-θ slice image in the spatio-temporal volume (equal to the epipolar plane image in the orthogonal projection case) located at the center of the sphere is shown in Fig. 5. The outlines of the object occupied part represent the traces of the left and right contours. We can intuitively see the corner points (or tangent discontinuities) on the traces of the left and right contours; they are due to the existence of occluded areas between the sphere and cylinder.

This phenomenon can also be interpreted in terms of Giblin's pedal curve of a surface [1]. If a planar curve containing a concave segment is pursued continuously in turns of convex, concave and convex, its trace (pedal curve) will have three segments connected at two cusp points and crossed by the first and third ones. The cross point is simply the corner point (tangent discontinuity) above. For an opaque surface, however, the trace segment between two cusp points is never visible, and segments from the cross point to the cusp points only appear partially as traces of self-occluding contours.

Denoting the left and right derivatives of $\omega(\theta)$ as $\omega'-$ and $\omega'+$, and denoting the tangent points sharing the same line of sight at the separated convex segments as $P_1(X_1,Z_1)$ and $P_2(X_2,Z_2)$, we have

$$(X_1, Z_1)\bullet(\cos\theta, \sin\theta) = \omega(\theta) \qquad (5.1)$$

$$(X_2, Z_2)\bullet(\cos\theta, \sin\theta) = \omega(\theta) \quad (5.2).$$

Taking the left derivative about θ at point P_1 and the right derivative about θ at point P_2, we obtain

$$L'_-(\theta): \; -X_1\sin\theta + Z_1\cos\theta = \frac{\partial\omega}{\partial\theta_-} \quad (6.1)$$

$$L'_+(\theta): \; -X_2\sin\theta + Z_2\cos\theta = \frac{\partial\omega}{\partial\theta_+} \quad (6.2).$$

Solving for X_1, X_2, Z_1, and Z_2 by Eqs. (5-6), we obtain $P_1(X_1,Z_1)$ and $P_2(X_2,Z_2)$ as

$$\begin{bmatrix} X_1 \\ Z_1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} \omega \\ \partial\omega/\partial\theta_- \end{bmatrix} \quad (7)$$

$$\begin{bmatrix} X_2 \\ Z_2 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} \omega \\ \partial\omega/\partial\theta_+ \end{bmatrix} \quad (8)$$

$$\begin{bmatrix} X_2 - X_1 \\ Z_2 - Z_1 \end{bmatrix} = \left(\frac{\partial\omega}{\partial\theta_+} - \frac{\partial\omega}{\partial\theta_-} \right) \begin{bmatrix} -\sin\theta \\ \cos\theta \end{bmatrix} \quad (9)$$

The vector jumping from $P_1(X_1, Z_1)$ to $P_2(X_2, Z_2)$ thus has the length $\omega'+ - \omega'-$ (Eq. (9)), and we can compute its two margin points that indicate the unexposed segment. For a smooth trace where $\omega'+ = \omega'- = \omega'$, the jump is zero and the formulas for computing the points on the planar curve are the same as Eqs. (4.1) and (4.2). If we find discontinuity in the first derivative $\omega'(\theta)$ and, correspondingly, a linear segment connecting two tangent points, we can assert that there is a concave, linear, or occluded convex planar curve between these points. It is important to notice from Eq. (9) that the degree of nonsmoothness due to a concave valley is equal to the real span of the bridge over that valley.

2.3 Qualitative Detection and Quantitative Extraction of Silhouette Unexposed Regions

Suppose that a resolution Δ required for a 3D model is expressed as the size of the smallest patch on the model. This resolution's component on the rotational plane is denoted as the length δ. For a given resolution component δ, we only need to find, at the curve on each rotational plane, the concave valleys that are wider than δ.

According to the given δ, we identify areas unexposed as contours by locating the corner points of $\omega(\theta)$. Because the span over an unexposed segment is equal to the difference between the left and right first derivatives of $\omega(\theta)$, we have the criterion

$$\text{Span} = \frac{\partial\omega}{\partial\theta_+} - \frac{\partial\omega}{\partial\theta_-} > \delta \quad (10)$$

As θ increases, $\omega(\theta)$ is obtained continuously by tracking a contour point. We feed the sequence $\omega(\theta)$ to a connected pair of differential operators, which has the same effect as a second-order differential filter (denoted F in Fig. 6). The first convolution with the operator

yields a step-type signal at a corner point and then the second convolution yields a peak at the corner point in the output.

It is interesting to note that corner points on the trace of a right-side silhouette are always leftward, and vice versa. We can therefore find nonsmooth points of $\omega(\theta)$ from the right-side silhouette by extracting the local maxima in the filtered sequence $F(\omega(\theta))$. To ignore the concave or linear segments that have spans shorter than the required length δ, we set the threshold of $F(\omega(\theta))$ to δ before finding the local maxima.

As the contour is being tracked, we set a window moving over $\omega(\theta)$ to compute precise first-derivative values $\omega'(\theta)$ at smooth places for using Eq. (4). If a silhouette unexposed region is detected, the window size, as shown in Fig. 7, is adapted near the extracted corner point so as to exclude corner points from the estimation of $\omega'_-(\theta)$ and $\omega'_+(\theta)$ in using Eqs. (7-9). We simply use the least squares approximation to fit a line to the data inside the window, so that we can obtain its derivative as the tangent.

This processing for the shape extraction and identification of an unknown area can be carried out almost instantaneously as images come in continuously, with the only delay being the size of the filter $F(\omega(\theta))$.

3. General Approach of Using Contours

3.1 Relations between Contours and 3D Points

Research concerning shape from contour thus far has only studied the following. A surface can be a parameterization of silhouettes according to the differential geometry, or intuitively, the surface has continuous derivatives up to the second order as well as be convex. There exists a one-to-one correspondence between a silhouette in an image and a surface point. However, it is not enough to construct a complete model from images. We also have to deal with those parts that do not satisfy the conditions of parameterization; although their shapes can not be directly computed from silhouettes, they are still constrained by silhouettes to a certain degree.

There are several kinds of relationships between contours in an image sequence and 3D points on an observed object (Fig. 8). We can categorize 3D shapes by using two attributes: the 2D shape of silhouettes and the viewing condition of the silhouette sequence. These two factors are described in terms of the derivatives of a silhouette point about the y coordinate and time (or rotation angle).

Let us first look at the case where a contour in the image is not parallel to the rotational plane. In this case, the derivative of the silhouette about the y coordinate is not infinite. When the viewing ray shifts on a smooth convex shape as a tangent line on the rotational plane, there is a one-to-one correspondence between a contour point and a surface point. Differentiating Eq. (4), we obtain the distance between the connected 3D points as

$$\begin{bmatrix}\Delta X\\ \Delta Z\end{bmatrix} = \left(\omega + \frac{\partial^2\omega}{\partial\theta^2}\right)\begin{bmatrix}-\sin\theta\\ \cos\theta\end{bmatrix}\Delta\theta \qquad (11),$$

if an interval $\Delta\theta$ in the angle of observation is given. The density of the estimated 3D points on the planar curve is

$$\text{density} = \frac{1}{\sqrt{\Delta X^2 + \Delta Z^2}} = \frac{1}{(\omega + \partial^2\omega/\partial\theta^2)\Delta\theta} = \frac{c}{\Delta\theta} \qquad (12)$$

According to Giblin's [1] results, c is the curvature of the planar curve; therefore, the density is proportional to the curvature of the curve. This is desired for data reduction when shape represented by 3D patches in graphics, which assigns more points to places with high curvatures.

In the extreme case of a shape with infinite curvature [11,15,20], e.g. a convex corner point on a planar curve, more than one line of sight goes through the point and the recovered 3D positions coincide. The viewed corner point does not move on the curve as its projected silhouette point moves in continuous images. The derivative of the 3D point with time is thus zero. This forms a many-to-one correspondence between contours and the 3D point. This kind of correspondence may allow us to refine the estimation of the 3D point through redundant data fusion.

Moreover, in the case of an unexposed planar curve, there is a one-to-many correspondence between a contour point, through which a line of sight touches two convex curves, and points that are occluded. The shape, invisible as contours, should be investigated by using other visual cues.

If the contour is, on the other hand, parallel to the rotational plane, a line of sight through the contour may touch many 3D points; if that is the case, a 3D point will be projected to other contours in more than one image. This can be observed at the top plane of the box. This produces a many-to-many correspondence between silhouettes and 3D points. A plane coincident with the rotational plane is one such case. This kind of shape can be detected in each image at the discontinuities of contour $\omega(y)$ with respect to the y coordinate.

The distribution of the estimated 3D points in the space, as a result, is related to the surface curvature. Figure 9 summarizes possible shapes whose projection in the image is not parallel to the rotational plane.

3.2 Data Representation and Implementation

We choose a situation where a fixed camera takes images of rotational objects to implement our method. This geometry needs little space, is easy to set up, and allows the use of a static background to reduce the cost in image processing. Moreover, almost no calibration is needed to get the rotational axis. The camera has a long focal length to approximate orthogonal projection. It can also produce the effect of blurring background (when the subject is sharply

focused) to prevent small background noise influencing the figure-ground separation.

Our algorithm starts tracking contours from image to image and generates a matrix $W(\theta,y)$ that describes the distribution of $\omega(\theta,y)$. Figure 10(a) shows the contour distribution of the objects given in Fig. 4: the period here is 2π and the value of ω is displayed in the gray level. The greater $\omega(\theta,y)$ is, the brighter the pixel at (θ,y) looks. Representing a contour sequence in $W(\theta,y)$ has the following advantages. First, it allows us to access the sequence freely according to (θ,y). Secondly, a matrix can be intuitively displayed as an image and handled by image processing modules. Further, it can easily be saved as a file in a database and transmitted in the communication network.

If there is a hole visible inside a silhouette closed region over a certain viewing period, an extra matrix must be added for the contour distribution of the hole. However, we neglect this in the following, because it is identical in principle to generating a shape from an outline contour.

As a result, the trajectory of a contour point in Fig. 5 is represented as a horizontal data line in the $W(\theta,y)$ image, with the distance from the rotation axis as the gray level. A corner point in the wave of the right trace in Fig. 5 becomes a local minimum in the horizontal line of $W(\theta,y)$. We can see some vertical valleys which are connected corner points at different heights. To find the precise positions of these vertical valleys, the dynamically yielded $W(\theta,y)$ is filtered horizontally by the operator $F(\omega(\theta))$ given in Fig. 6. After thresholding the output with the value $k\delta$, where k is a weighted constant, the local maxima are detected in the filtered wave and their positions are marked as corner points.

At the same time, we locate horizontal distributed edge points in $W(\theta,y)$, where discontinuities in the vertical direction appear in the continuous contour, by convoluting with a normal vertical differential operator. The resulting peak positions are marked for exclusion from shape estimation by Eqs. (4) and (7-9). For the distribution $W(\theta,y)$ in Fig. 10(a), the computed vertically distributed corner points (horizontally nonsmooth, tangent discontinuous) and horizontal edges are shown in Fig. 10(b). These singular points segment the whole silhouette distribution into many sub-distributions which correspond to surfaces of the sphere, cylinder, and four rigid edges of the box. Each of them can yield a shape from the silhouette values.

Surface points are estimated as follows: A small adaptive window shifts over $W(\theta,y)$ at places without singular points. The derivative value is approximated inside the window by the linear least squares method. The 3D positions are computed either by Eq. (4) or by Eqs. (7-9), with each unexposed region detected between two margin points. The obtained 3D points are connected with their neighbors by triangular patches, i.e. the simplest patches for graphics display. Figure 11(a) shows two views of the estimated shapes of the objects in Fig. 4, one in patches and the other in shaded form. In Fig. 11(b), the unexposed areas are shown as gray

patches.

The occluded regions may contain some concave segments or even convex segments extending from both sides. Due to the difference in surface orientation between the convex segment and a segment behind it, the convex segment can still be extracted as an edge point (see Fig. 3); it can be calculated as a self-occluding edge by Eq. (4). If the concave part is shallow, however, it is difficult to locate self-occluding contours in a long period of rotation. We need some verification of the results. Fixed points due to the discontinuity in surface albedo or surface orientation can help to determine the shape inside a silhouette unexposed region [11,15,20].

4. Generalization of the Method to Perspective Projection

4.1 Shape from Silhouettes with no Epipolar Plane Constraint

In this section, we will consider the shape from silhouette under perspective projection. We first deal with surfaces (or curves) which can be a parameterization of continuous silhouettes. Then, a whole silhouette distribution is segmented so that each part of it corresponds to a surface exposed as silhouettes.

What we have now is a silhouette distribution located in a spatio-temporal volume (Fig. 12). It is represented as a matrix $W(\theta,y)$ whose element $\omega(\theta,y)$ is the distance of the contour point from the rotation axis at height y in the θ-th image. The trace of a silhouette point on a horizontal X-Θ slice in the volume is referred to as the θ curve in this paper, which corresponds to one horizontal line in matrix $W(\theta,y)$. A contour in an image (X-Y slice) is a closed curve named the s curve here. We can find the two series of curves located in the spatio-temporal volume in Fig. 12. We will estimate a 3D shape from a series of θ curves and s curves by using their positions as well as their derivatives.

In order to track a contour point along a certain line in the images, an epipolar plane is used in shape estimation. If the employed camera has a linear motion, a 3D point is constrained in the epipolar plane which is determined only from the camera translation and its initial image position. For one spatial point, the epipolar planes yielded from any two camera positions are all coincident for linear motion. The move of this point hence becomes predictable in the image frames. This makes tracking of the point easier on the epipolar line over the image sequence [2,11,15,20].

If the camera motion, however, contains rotation (for example, the camera moves along a circle), the epipolar planes through the spatial point and every two consecutive camera positions are no longer coincident with each other (Fig. 13(a)). An epipolar plane must be relocated again if we want to maintain a point on the corresponding epipolar line in the consecutive frames (see Fig. 13(b)).

The constraint is, hence, feasible and efficient only for the case where only a few images

rather than a sequence are employed. In fact, if images are taken at sufficiently near positions, we have a Ruled Surface determined by rulings (the lines of sight) through the camera trace and the surface points (Fig. 13(c)). We call this surface an Epipolar Surface because any two successive rulings determine an epipolar plane. The connected epipolar planes become the tangent planes of the surface at each ruling. The rulings are tangent to a 3D curve P on the object surface; therefore, the epipolar surface is also a tangent surface of the curve P. The surface can not be predicted from only the camera motion; nevertheless, it is also related to spatial points or the shape P itself where the camera looks at.

The intrinsic viewing scheme of shape from contour is the relative rotation between the camera and objects, since we need the produced silhouettes to cover as large an area as possible for the objects. A surface point viewed as a contour point shifts on the surface. There is no particular necessity to kept it in an epipolar plane during the relative motion. Instead of estimating epipolar lines, we simply use the θ curve at each image height to recover 3D points whose projections are on the curve. This makes the tracking of silhouette points easier since we only need to search points horizontally in the images.

The recovered 3D points compose a 3D curve of the objects, named the Θ curve, or possibly several broken Θ curves when some concave region exists. We call a 3D curve recovered from a silhouette (s curve in an image) an S curve. If the surface is not smooth, it may appear as several segmented ones (discussed in the following sections). Generally, an S curve and a Θ curve will not intersect orthogonally, nor will an S curve and a Θ curve be assured of existing in one plane completely, although both an s curve and a θ curve are planar curves in the spatio-temporal volume.

4.2 Surface Parameterized by Contour

In the following, we derive equations to estimate 3D points from local sets of silhouette data, assuming the surface is smooth at each point and is continuously viewed as a contour in the images. The surface normal and curvature are determinable. Let O-XYZ be an object-centered coordinates system located at the rotation axis. Without losing generality, the camera has a relative rotation on the plane $y=0$ starting from $C_0=(0,0,D)$, where $-D>0$ is the diameter of the camera trace. As shown in Fig. 14, a surface point P(**X**) on a rim is viewed from an angle θ, and its projection on the image frame is $p(\omega,y)$. The plane of sight through the camera focus and the tangent of the contour touches the surface at P. The normal of the plane of sight is denoted as **N**; for the smooth surface here, it becomes the surface normal.

For the camera centered coordinates system C-XYZ, the tangent direction of the contour at p (in the image) and the viewing ray through p to P are denoted as $\mathbf{t}_0$ and $\mathbf{v}_0$, respectively. Assuming that the contour at p is not horizontal in the image frame, i.e. it has a unique horizontal coordinate ω, $\mathbf{t}_0$ and $\mathbf{v}_0$ can be expressed as:

$$\mathbf{t}_0 = \begin{bmatrix} \frac{\partial \omega}{\partial y} & 1 & 0 \end{bmatrix}^T \qquad \mathbf{v}_0 = [\omega \quad y \quad \lambda]^T \tag{13}$$

where λ is a constant denoting the ratio of camera focal length to pixel size, which is obtained through calibration. If we denote the rotation matrix as R, the location of the camera center **C** and the rotated vectors **t**, **v** of $\mathbf{t}_0$, $\mathbf{v}_0$ in the coordinates system O-XYZ become:

$$\mathbf{C} = R\,\mathbf{C}_0 \qquad \mathbf{t} = R\,\mathbf{t}_0 \qquad \mathbf{v} = R\,\mathbf{v}_0 \tag{14}$$

The normal of the plane of sight at P can be determined from vectors **t** and **v** by:

$$\mathbf{N} = \mathbf{t} \times \mathbf{v} = R\,\mathbf{t}_0 \times R\,\mathbf{v}_0 = R\,\mathbf{N}_0 \tag{15}$$

$$\mathbf{N}_0 = \mathbf{t}_0 \times \mathbf{v}_0 = \begin{bmatrix} \lambda & -\lambda \frac{\partial \omega}{\partial y} & y\frac{\partial \omega}{\partial y} - \omega \end{bmatrix}^T \tag{16}$$

where $\mathbf{N}_0$ is the normal N represented in the coordinates system C-XYZ. Because the viewing ray is in the plane of sight through P(**X**), it is perpendicular to the normal **N**. Hence, the next equation holds.

$$\mathbf{N} \cdot (\mathbf{X} - \mathbf{C}) = 0 \tag{17}$$

We will finally derive the result of **X**, that is, the 3D position of P in the O-XYZ coordinates system. Because we have assumed the surface at P has good properties in differentiation (differentiable up to the second order) and is visible as a contour, partial derivatives exist for both surface position **X** and normal **N**. Computing the partial derivatives of Eq. (17) about θ and y, we obtain,

$$\frac{\partial \mathbf{N}}{\partial \theta} \cdot (\mathbf{X} - \mathbf{C}) + \mathbf{N} \cdot (\frac{\partial \mathbf{X}}{\partial \theta} - \frac{\partial \mathbf{C}}{\partial \theta}) = 0 \tag{18.1}$$

$$\frac{\partial \mathbf{N}}{\partial y} \cdot (\mathbf{X} - \mathbf{C}) + \mathbf{N} \cdot (\frac{\partial \mathbf{X}}{\partial y} - \frac{\partial \mathbf{C}}{\partial y}) = 0 \tag{18.2}$$

Because normal **N** is perpendicular to any vector in the plane of sight (tangent plane for smooth surface here), and the relative rotation is on the plane y=0 changing the angle θ, we have the following relations.

$$\mathbf{N} \cdot \frac{\partial \mathbf{X}}{\partial y} = 0 \qquad \mathbf{N} \cdot \frac{\partial \mathbf{X}}{\partial \theta} = 0 \tag{19.1}$$

$$\frac{\partial R}{\partial y} = 0 \qquad \frac{\partial \mathbf{C}}{\partial y} = 0 \tag{19.2}$$

Then, Eqs. (18.1, 18.2) can be written as:

$$\frac{\partial \mathbf{N}}{\partial \theta} \cdot (\mathbf{X} - \mathbf{C}) - \mathbf{N} \cdot \frac{\partial \mathbf{C}}{\partial \theta} = 0 \tag{20.1}$$

$$\frac{\partial \mathbf{N}}{\partial y} \cdot (\mathbf{X} - \mathbf{C}) = 0 \tag{20.2}$$

On the other hand, because we track a contour point on horizontal image lines and

parameterize the surface by the s and θ curves (rather than a sequence of epipolar lines which are not always horizontal in successive images from rotation), we have

$$\frac{\partial y}{\partial \theta} = 0 \tag{21}$$

for the tracked points. Therefore, the partial derivatives of **N** can be computed from Eqs. (15, 16) and they become

$$\frac{\partial \mathbf{N}}{\partial \theta} = \frac{\partial \mathrm{R}}{\partial \theta}\begin{bmatrix} \lambda & -\lambda\frac{\partial \omega}{\partial y} & y\frac{\partial \omega}{\partial y} - \omega \end{bmatrix}^{\mathrm{T}} + \mathrm{R}\begin{bmatrix} 0 & -\lambda\frac{\partial^2 \omega}{\partial \theta \partial y} & y\frac{\partial^2 \omega}{\partial \theta \partial y} - \frac{\partial \omega}{\partial \theta} \end{bmatrix}^{\mathrm{T}} \tag{22.1}$$

$$\frac{\partial \mathbf{N}}{\partial y} = \mathrm{R}\begin{bmatrix} 0 & -\lambda\frac{\partial^2 \omega}{\partial y^2} & y\frac{\partial^2 \omega}{\partial y^2} \end{bmatrix}^{\mathrm{T}} \tag{22.2}$$

which are the normal curvatures for the surface along the **Θ** curve and **S** curve [21]. Putting these results into Eq. (20) and putting Eq. (15) into Eq. (17), we obtain the next important equations.

$$\mathrm{R}\begin{bmatrix} \lambda & -\lambda\frac{\partial \omega}{\partial y} & y\frac{\partial \omega}{\partial y} - \omega \end{bmatrix}^{\mathrm{T}} \cdot (\mathbf{X} - \mathbf{C}) = 0$$

$$\mathrm{R}\begin{bmatrix} 0 & -\lambda\frac{\partial^2 \omega}{\partial y^2} & y\frac{\partial^2 \omega}{\partial y^2} \end{bmatrix}^{\mathrm{T}} \cdot (\mathbf{X} - \mathbf{C}) = 0$$

$$\left(\left(\frac{\partial \mathrm{R}}{\partial \theta}\right)\begin{bmatrix} \lambda & -\lambda\frac{\partial \omega}{\partial y} & y\frac{\partial \omega}{\partial y} - \omega \end{bmatrix}^{\mathrm{T}} + \mathrm{R}\begin{bmatrix} 0 & -\lambda\frac{\partial^2 \omega}{\partial \theta \partial y} & y\frac{\partial^2 \omega}{\partial \theta \partial y} - \frac{\partial \omega}{\partial \theta} \end{bmatrix}^{\mathrm{T}}\right) \cdot (\mathbf{X} - \mathbf{C})$$

$$= \mathrm{R}\begin{bmatrix} \lambda & -\lambda\frac{\partial \omega}{\partial y} & y\frac{\partial \omega}{\partial y} - \omega \end{bmatrix}^{\mathrm{T}} \cdot \frac{\partial \mathrm{R}}{\partial \theta}\mathbf{C}_0 \tag{23}$$

Now, we derive this in detail by substituting a value for the rotation transform matrix. The standard rotation matrix R in the object centered coordinates system O-XYZ can be written as:

$$\mathrm{R} = \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix} \tag{24}$$

where θ is the angle of C from $\mathbf{C}_0$. From this matrix, we further deduce that

$$\frac{\partial \mathrm{R}}{\partial \theta} = \begin{bmatrix} -\sin\theta & 0 & -\cos\theta \\ 0 & 0 & 0 \\ \cos\theta & 0 & -\sin\theta \end{bmatrix} \tag{25}$$

Using these results, we reorganize Eq. (23) as

$$\begin{bmatrix} \lambda\cos\theta - \sin\theta\left(y\frac{\partial \omega}{\partial y} - \omega\right) & -\lambda\frac{\partial \omega}{\partial y} & \lambda\sin\theta + \cos\theta\left(y\frac{\partial \omega}{\partial y} - \omega\right) \\ -\sin\theta y\frac{\partial^2 \omega}{\partial y^2} & -\lambda\frac{\partial^2 \omega}{\partial y^2} & \cos\theta y\frac{\partial^2 \omega}{\partial y^2} \\ \mathrm{A} & -\lambda\frac{\partial^2 \omega}{\partial \theta \partial y} & \mathrm{B} \end{bmatrix} (\mathbf{X} - \mathbf{C}) = \begin{bmatrix} 0 \\ 0 \\ \mathrm{K} \end{bmatrix}$$

(26.1)

where terms A, B, K are written as

$$A = -\lambda \sin\theta - \cos\theta\left(y\frac{\partial\omega}{\partial y} - \omega\right) - \sin\theta\left(y\frac{\partial^2\omega}{\partial\theta\partial y} - \frac{\partial\omega}{\partial\theta}\right)$$

$$B = \lambda\cos\theta - \sin\theta\left(y\frac{\partial\omega}{\partial y} - \omega\right) + \cos\theta\left(y\frac{\partial^2\omega}{\partial\theta\partial y} - \frac{\partial\omega}{\partial\theta}\right)$$

$$K = \left[\lambda\cos\theta - \sin\theta\left(y\frac{\partial\omega}{\partial y} - \omega\right) \quad -\lambda\frac{\partial\omega}{\partial y} \quad \lambda\sin\theta + \cos\theta\left(y\frac{\partial\omega}{\partial y} - \omega\right)\right]\begin{bmatrix}-\cos\theta\\0\\-\sin\theta\end{bmatrix}D = -\lambda D$$

(26.2)

The |D| here is the distance between the camera and the rotation axis; it has been defined at the beginning of this section. We can now directly compute the value of **X−C** from the determinants of Eq. (26), which are

$$\Delta = -\lambda\frac{\partial^2\omega}{\partial y^2}\left(\lambda^2 - \omega y\frac{\partial\omega}{\partial y} + \omega^2 - \lambda\frac{\partial\omega}{\partial\theta}\right) \tag{27.1}$$

$$\Delta x = -\lambda^2 D\frac{\partial^2\omega}{\partial y^2}(\lambda\sin\theta - \omega\cos\theta) \tag{27.2}$$

$$\Delta y = \lambda^2 Dy\frac{\partial^2\omega}{\partial y^2} \tag{27.3}$$

$$\Delta z = \lambda^2 D\frac{\partial^2\omega}{\partial y^2}(\lambda\cos\theta + \omega\sin\theta) \tag{27.4}$$

The line of sight reaching the surface point P from the camera focus C is

$$\mathbf{X} - \mathbf{C} = \begin{bmatrix}\Delta x/\Delta\\ \Delta y/\Delta\\ \Delta z/\Delta\end{bmatrix} = \frac{-\lambda D}{\left(\lambda^2 + \omega^2 - \omega y\frac{\partial\omega}{\partial y} - \lambda\frac{\partial\omega}{\partial\theta}\right)}\begin{bmatrix}-\lambda\sin\theta + \omega\cos\theta\\ y\\ \lambda\cos\theta + \omega\sin\theta\end{bmatrix} \tag{28.1}$$

And the 3D surface point in the coordinates system O-XYZ is then at

$$\mathbf{X} = \frac{-\lambda D}{\left(\lambda^2 + \omega^2 - \omega y\frac{\partial\omega}{\partial y} - \lambda\frac{\partial\omega}{\partial\theta}\right)}\begin{bmatrix}-\lambda\sin\theta + \omega\cos\theta\\ y\\ \lambda\cos\theta + \omega\sin\theta\end{bmatrix} + \begin{bmatrix}-\sin\theta\\0\\\cos\theta\end{bmatrix}D$$

$$= \frac{D}{\left(\omega y\frac{\partial\omega}{\partial y} + \lambda\frac{\partial\omega}{\partial\theta} - \lambda^2 - \omega^2\right)}\begin{bmatrix}\lambda\omega\cos\theta - \sin\theta\left(\omega y\frac{\partial\omega}{\partial y} + \lambda\frac{\partial\omega}{\partial\theta} - \omega^2\right)\\ \lambda y\\ \lambda\omega\sin\theta + \cos\theta\left(\omega y\frac{\partial\omega}{\partial y} + \lambda\frac{\partial\omega}{\partial\theta} - \omega^2\right)\end{bmatrix} \tag{28.2}$$

In this equation, parameters D and λ are measured through calibration and rotation angle θ

is determined from the control of the system. The ω is extracted from images at the height of y. The partial derivative of ω about the height y in an image can be estimated from contour points near p. The partial derivative about the rotation angle θ can be estimated from the tracked ω sequence in a short period containing p (which is a tangent edge in the x-θ slice of the spatio-temporal volume). This result is similar to that obtained by Eq. (4) in the orthogonal projection; scaled by some factor and taking into account the contour inclined to the y axis in the image. If we let $D\to\infty$, $\lambda\to\infty$, $D/\lambda\to 1$, and the derivative of ω about the y coordinate be zero in Eq. (28.2), we can obtain the exact same equation for the orthogonal projection as Eq.(4).

The first- and second-order differential properties are not difficult to obtain after we completely know the surface. In fact, the surface normal is obtained before the 3D position of a point is computed; it is given by Eqs. (15,16) and Eq. (24). It becomes

$$\mathbf{N}=\left[\lambda\cos\theta-\sin\theta\left(y\frac{\partial\omega}{\partial y}-\omega\right)\quad -\lambda\frac{\partial\omega}{\partial y}\quad \lambda\sin\theta+\cos\theta\left(y\frac{\partial\omega}{\partial y}-\omega\right)\right]^{T} \tag{29}$$

in the coordinates system O-XYZ. As to the Gaussian Curvature and principle curvature, they can be considered in the following way. We obtained two normal curvatures from the Θ and S curves (see Eq. (22)). The Θ curve and S curve do not intersect orthogonally. The angle between them (angle between the tangent directions of the curves) can also be given from vectors $\partial\mathbf{X}/\partial\theta$ and $\partial\mathbf{X}/\partial y$ by differentiating Eq. (28.2), which contains second-order derivatives of ω(θ,y).

Besides the two normal curvatures from the Θ and S curves, we also need normal curvatures along two other directions for complete determination of the maximum and minimum curvatures using Euler's formula. Arbitrarily selecting two curves in W(θ,y) that go through point p(θ,y) neither horizontally nor vertically, say two lines $y_1=\theta_1-\theta+y$, and $y_2=-\theta_2+\theta+y$, we obtain two surface curves $\mathbf{X}_1(\theta_1)$ and $\mathbf{X}_2(\theta_2)$ that go through P(**X**) by replacing θ and y in Eq. (28.2) with θ_i and y_i. Differentiating Eq. (28.2) with respect to θ_i will yield tangent directions and normal curvatures. In a real situation, however, the second-order derivatives from digitized noisy image data are extremely inaccurate. We therefore do not expect the result of the curvature from the image data.

4.3 Categorizing Shapes According to Silhouette Attributes

In subsection 4.2, we assumed that a silhouette is smooth and is viewed as a smoothly changed silhouette. In modeling an object, however, these conditions might not be satisfied at some place. We might have to ask "how many kinds of shapes can appear when we look at silhouettes successively during one move?", or "how are these shapes combined into a whole object?" We therefore categorize shapes according to their behaviors in observed silhouette sequences.

A number of works have computed the aspect view of an object [22,23]. The criterion is to look at the qualitative changes in the line drawings [19], given a completely known model. In our categorization, we stress the detectability of visual events in images. The smoothness of three attributes ---- surface, 2D silhouette and its image motion, are used for this categorization; they can be extracted in our defined rotation. Figure 15 illustrates categorized shapes. The horizontal lines in each image at corners of the cube are pixel lines that are searched when silhouettes are being extracted.

If the s curve and θ curve are both differentiable, we have graphs (011) and (111) in Fig. 15; here, the normal of plane of sight is differentiable about y and θ. The surface normal, however, may be either continuous or discontinuous. The continuous case (111) was analyzed in the previous section. The discontinuous case (011), however, can be distinguished by examining curvatures [11,15] of the S and Θ curves. Because changing either y or θ might cause the focused point **X** to shift on a 3D edge, the S and Θ curves are coincident at the edge. The edge is then parameterized by y or θ. We can use this constraint to verify the normal discontinuity.

In the case of nonsmooth silhouette motion (θ curve is not differentiable), some region unexposed as a silhouette is detected. Two kinds of shapes, a concave shape and a planar shape, can be considered. If a line of sight touches two surface points as example (110) in Fig. 15 shows, which is the first case, the unknown region should contain at least one concave point, or disconnected shapes. If the line of sight touches an infinitive number of surface points as (110) shows, which is the second case, these points are on a plane or a parabolic surface (zero curvature along the line of sight). The proper ways of filling the unknown region include the shape from motion applied to surface marks, the shape from occluding contours applied to self-occluding contours, and the shape from shading applied to an area with uniform albedo in it.

A discontinuity on s curves also includes two situations. First, the silhouette is not smooth about its y coordinates such as (001). Second, the silhouette itself is discontinuous along the y coordinate (101), where a segment parallel to the rotation plane appears on the silhouette.

Thus far, we have analyzed some basic types of different shapes determined by silhouettes. By merging different shapes, other combinations such as (000) and (100) may be possible.

4.4 Segmentation of Silhouette Distribution and Detection of Unknown Regions

In order to estimate a shape from a silhouette distribution using the equations given in subsection 4.2, we need to first separate a whole distribution into sub-distributions satisfying the required conditions. The approach is basically the same as that described for the orthogonal projection.

1. In the distribution $W(\theta,y)$ obtained from contour tracking, a differential operator is

convoluted vertically to find discontinuities. The value to threshold for edge points is determined from a required model resolution. The filtered image of W(θ,y) is convoluted with the differential operator again for detecting nonsmooth points in the s-curve. During the second convolution, the edge points already obtained in W(θ,y) are masked because the derivative about y does not exist there.

2. The distribution W(θ,y) is filtered by using another horizontal differential operator two times for detecting nonsmooth points on θ-curves at different heights. The extracted points are also masked.

3. The results from steps 1 and 2 are merged so as to exclude those singular edges. The computation of the partial derivatives of ω about θ and y is the same approach as that for the orthogonal projection. Linear approximation is used within an adaptive window. The window squeezes automatically near the masked edges in order to exclude the singular edges in the estimation.

4. The obtained derivative values are put into Eq. (28) for locating 3D points. A silhouette unexposed area exist in between two points on Θ-curves whose projections are masked nonsmooth points on the corresponding θ-curves. The obtained 3D points P(θ,y) are connected to their neighbors P(θ-1,y-1), P(θ-1,y), P(θ,y-1) to produce two triangular patches [P(θ,y), P(θ-1,y-1), P(θ-1,y)] and [P(θ,y), P(θ-1,y-1), P(θ,y-1)]. Those points on different sides of the masked singular points but immediate to them are also connected at their 3D positions. The obtained patches are masked in a different color in order to indicate unknown properties of the regions. The generated patches can be directly displayed on a graphics screen.

It is necessary to confirm quantitatively the filtering applied in step 2 above. Under the perspective projection, a vector connecting two margin points $P_1(X_1)$, $P_2(X_2)$ of an unexposed region can be written as

$$\Delta X = -\lambda D\left(\frac{1}{\lambda^2+\omega^2-\omega y\frac{\partial\omega}{\partial y_2}-\lambda\frac{\partial\omega}{\partial\theta_+}}-\frac{1}{\lambda^2+\omega^2-\omega y\frac{\partial\omega}{\partial y_1}-\lambda\frac{\partial\omega}{\partial\theta_-}}\right)\begin{bmatrix}-\lambda\sin\theta+\omega\cos\theta\\ y\\ \lambda\cos\theta+\omega\sin\theta\end{bmatrix} \tag{30}$$

Substituting the coefficients with their Taylor expansions, we have

$$\Delta X = \frac{-\lambda D}{\lambda^2+\omega^2}\left(\left(1-\frac{1}{\lambda^2+\omega^2}\left(\omega y\frac{\partial\omega}{\partial y_2}+\lambda\frac{\partial\omega}{\partial\theta_+}\right)....\right)-\left(1-\frac{1}{\lambda^2+\omega^2}\left(\omega y\frac{\partial\omega}{\partial y_1}+\lambda\frac{\partial\omega}{\partial\theta_-}\right)....\right)\right)\times$$

$$\begin{bmatrix}-\lambda\sin\theta+\omega\cos\theta\\ y\\ \lambda\cos\theta+\omega\sin\theta\end{bmatrix}$$

$$= \frac{-\lambda D}{\lambda^2+\omega^2}\left(\frac{1}{\lambda^2+\omega^2}\left(\omega y\frac{\partial\omega}{\partial y_1} - \omega y\frac{\partial\omega}{\partial y_2} + \lambda\frac{\partial\omega}{\partial\theta_-} - \lambda\frac{\partial\omega}{\partial\theta_+}\right)....\right)\begin{bmatrix}-\lambda\sin\theta+\omega\cos\theta\\ y\\ \lambda\cos\theta+\omega\sin\theta\end{bmatrix} \tag{31}$$

The condition needed for this expansion is

$$\frac{1}{\lambda^2+\omega^2}\left(\omega y\frac{\partial\omega}{\partial y}+\lambda\frac{\partial\omega}{\partial\theta}\right)\approx 0 \tag{32}$$

which is not difficult to be verified since

$$\frac{1}{\lambda^2+\omega^2}\left(\omega y\frac{\partial\omega}{\partial y}+\lambda\frac{\partial\omega}{\partial\theta}\right) = \sin\xi\sin\psi\frac{\partial\omega}{\partial y}+\frac{\cos\xi}{\sqrt{\lambda^2+\omega^2}}\frac{\partial\omega}{\partial\theta} \tag{33}$$

where ξ is the angle between the plane x=0 in the camera coordinates system C-XYZ and the projection of the line of sight on the plane y=0, and ψ is the angle of the line of sight with respect to the plane y=0 in C-XYZ. For a camera, these angles are usually very small; $\partial\omega/\partial y$ has a limited bond (otherwise, it should already be detected in step 1 above), and $\sqrt{\lambda^2+\omega^2}$ is on the order of several hundred. Therefore, condition (32) approximately holds. The length of the vector becomes

$$\|\Delta X\| \approx \frac{\lambda|D|\sqrt{\lambda^2+\omega^2+y^2}}{\left(\lambda^2+\omega^2\right)^2}\left|\omega y\left(\frac{\partial\omega}{\partial y_2}-\frac{\partial\omega}{\partial y_1}\right)+\lambda\left(\frac{\partial\omega}{\partial\theta_+}-\frac{\partial\omega}{\partial\theta_-}\right)\right| \tag{34}$$

which is approximately related to the differences of contour derivatives. We are able to determine the threshold for extracting nonsmooth points according to the smallest span, i.e. the resolution of the model.

5. Experiments

Experiments were carried out on plaster models and real people under both orthogonal projection and perspective projection. For the orthogonal projection, the camera was fixed a far distance away from the subjects (7-10 m) and a lens with a long focal length was used. A model is put on a turntable controlled step-by-step by a computer. The background was set with a piece of color paper. There was no special illumination in the sensing.

Silhouette points were tracked along the horizontal direction in the images. The search, from background toward figure, was done at a position near the obtained silhouette in the previous image. The results generated from a total of 360 images are shown in Figs. 16-19.

The process is very simple compared with many other shape recovery methods. The structure of the algorithm is clear and promising for real-time applications. It provides accurate results for sensing surfaces without many image features on them.

For the perspective projection, on the other hand, we calibrate the camera at the beginning. Figure 20 shows an example of acquiring the shape of a model car. The distance between the camera and rotation axis is 450 mm and the sampling at the rotation angle is 1 degree. The

image center is at (124, 132) and λ is 524 pixels.

In a more difficult experiment, people sat in a swivel chair and image sequences (6-7 seconds, about 200 images of size 256×256 pixels) were taken by a video camera as the chair is rotated. Some system topics will be introduced briefly next.

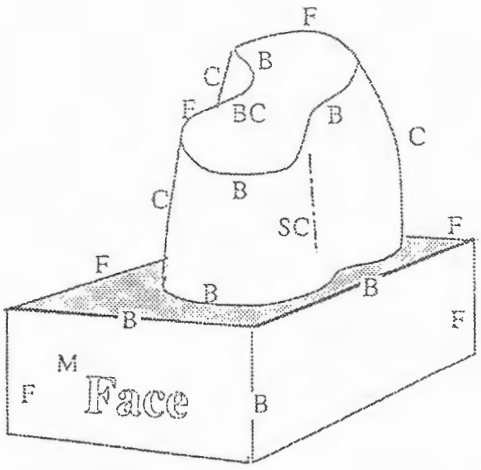
Rotation: We do not have control of the chair so the (changeable) rotation speed and unknown axis position have to be obtained from an image sequence. We track two marks attached to the subjects' shoulders and succeed in estimating rotational information from the trajectories of the marks in the images [14].

Texture: The texture of a face is collected in a Dynamic Projection Image [6,7,18]. It is then mapped onto the established model for display. When image t is taken, the center $c_y(t)$ of both the left and right silhouette points at a given height y is located. As the subject rotates (t increases), the texture is sampled successively from the face at the vertical pixel lines through moving center $c_y(t)$. These pixel lines are arranged along the horizontal axis in the order of time so that a 2D image containing texture around the whole face is obtained. This image corresponds to a vertical cylinder surface in the spatio-temporal volume determined by curve $c_y(t)$. The texture data are mapped onto the established 3D model again at the positions they are sampled, by making a correspondence between each memorized $c_y(t)$ and the 3D model point at the height y given previously [14].

A real system has been built and we have modeled many people. Figure 21 shows one example where the system is making the face model of an author; 256×256 size images are processed on a SUN4. The recovered model, both in triangular patches and mapped with texture, are shown in Fig. 22. It is constructed from 60×70 θ curves and s curves and yields about 4000 3D points. It is displayed on a Silicon Graphics Machine using about 8000 patches.

6. Conclusion

We have established 3D models from silhouette sequences (from rotational objects and human faces). A whole model has been obtained from silhouettes for the first time, instead of only local shapes with some good attributes in smoothness and visibility. We analyzed the behavior of contours so that regions unexposed as any contour can be qualitatively detected and quantitatively located. Identification of areas that an algorithm is unable to model will allow other vision modules to further plan active investigations. We think this ability is very important for an algorithm in visual world understanding. We have presented a general approach in order to use contours in shape recovery. The entire process is efficient and can be applied to a real system, and the model construction is promising for real-time application. The established facial model will be displayed in a graphics-generated virtual environment, where it can be moved and deformed to show head motions and facial expressions for the purpose of visual communications.



		Edges Extractable in Image		
		Depth discontinuous		Depth continuous
		Internal contour	Silhouette	
Rims in Space	Normal discontinuous	BC	F	B
	Normal continuous	SC	C	M surface mark

Fig. 1. Categorization of observable edges on objects.

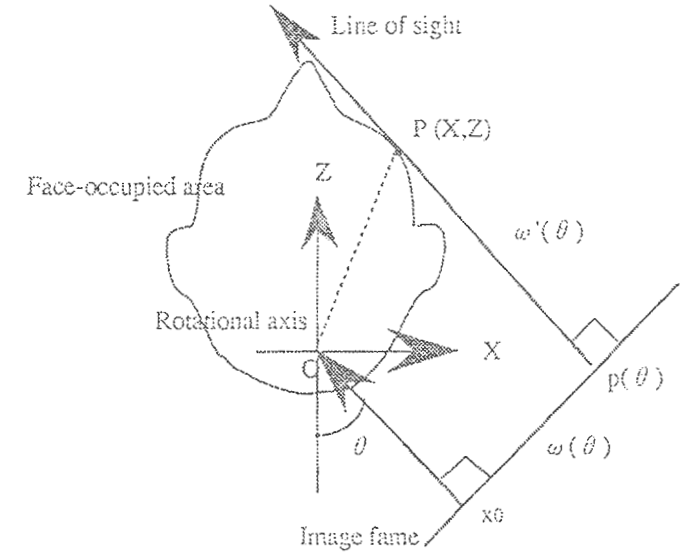


Fig. 2. Recovering shape of planar curve from occluding contours.

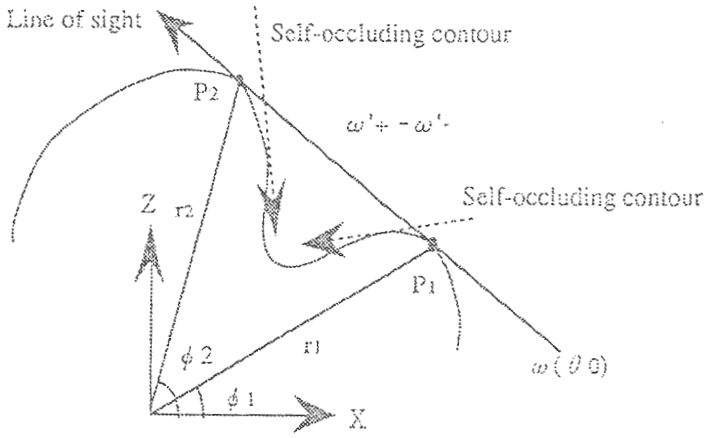


Fig. 3. Detecting a planar curve unexposed to contours.

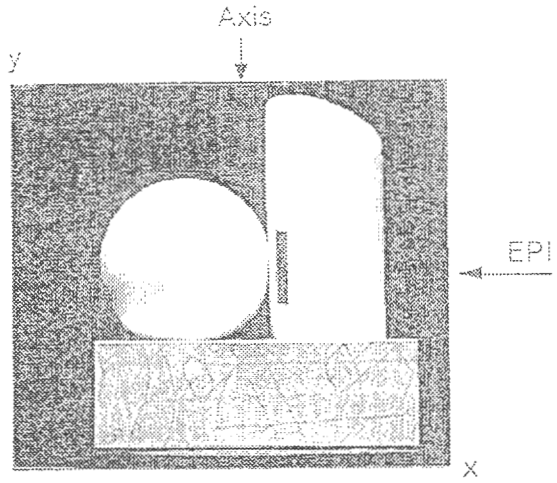


Fig. 4. Objects on a turntable.

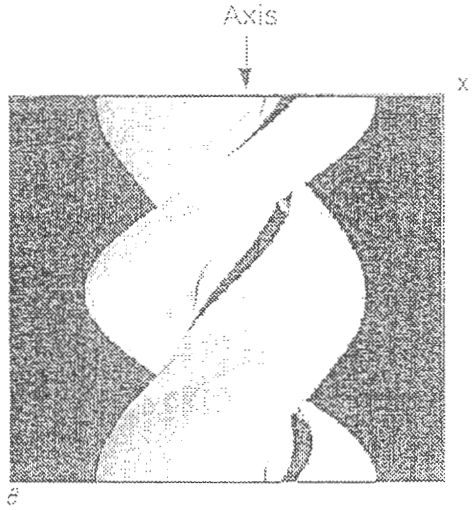


Fig. 5. An x-θ plane image in the spatio-temporal volume at the height of the sphere shown in Fig. 4.

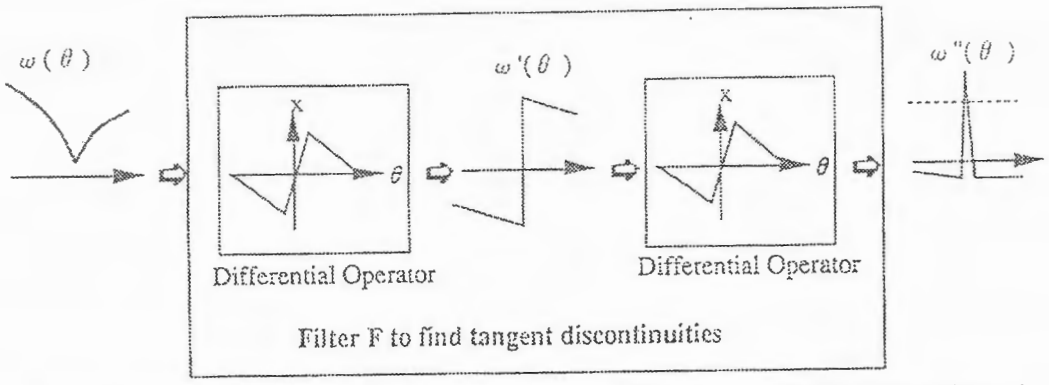


Fig. 6. Detecting corner points in a contour distribution. The contour distribution is filtered and the result is thresholded with a value related to model resolution.

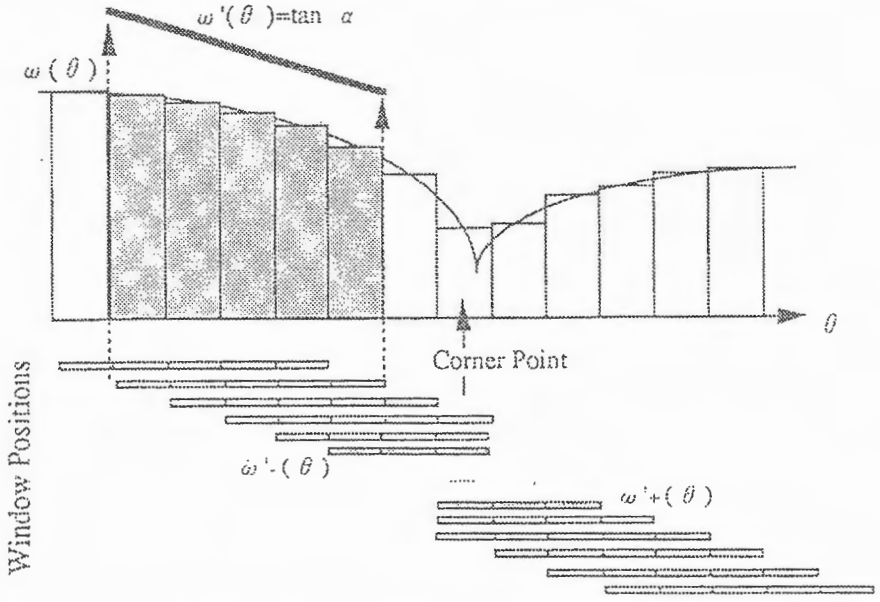
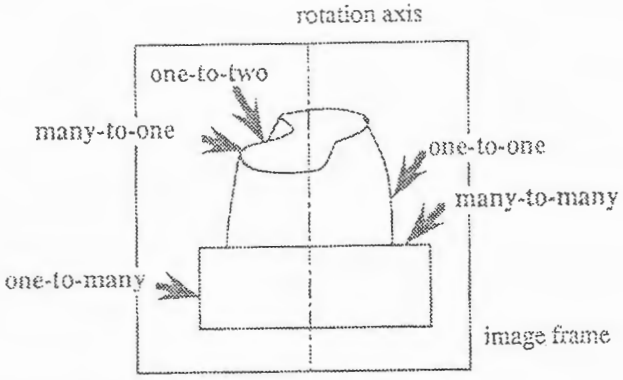


Fig. 7. A window shifts across contour distribution ω for estimating the first-order derivative values. A line is fitted data inside the window using the least square approximation. The window is adaptive near the extracted corner points.

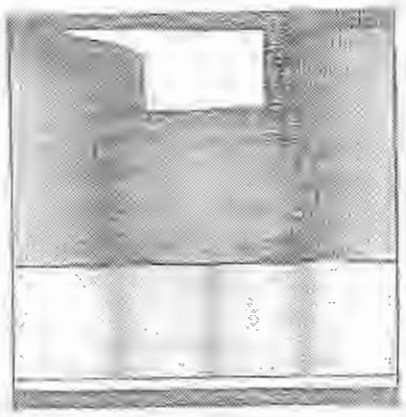


	$\frac{\partial \omega}{\partial \theta}$ exists		$\frac{\partial \omega}{\partial \theta}$ does not exist	
	corner point	convex	plane	concave
$\frac{\partial \omega}{\partial y}$	$\frac{\partial X}{\partial \theta} = 0$	$\frac{\partial X}{\partial \theta}$ exist	ΔX undeterminable $\frac{\partial X}{\partial \theta}$ does not exist	ΔX determinable $\frac{\partial X}{\partial \theta} = \infty$
	many-to-one	one-to-one	one-to-many	one-to-two
$\frac{\partial \omega}{\partial y} = \infty$	ΔX undeterminable		$\frac{\partial X}{\partial \theta}$ does not exist	many-to-many

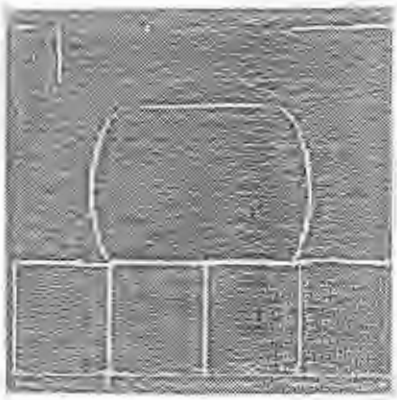
Fig. 8 Relation between silhouette points and surface points.

Curvature	$+\infty$	> 0	0	< 0	$-\infty$
Shape	corner	convex	plane	concave	inverse cusp point
Image behavior	*sin* trace	smooth contour trace	corner	invisible	*sin* trace
Move on surface	fixed point, stationary	shift	scan across	jump	fixed point, invisible as contour

Fig. 9. Planar curves of different shapes and their image properties.

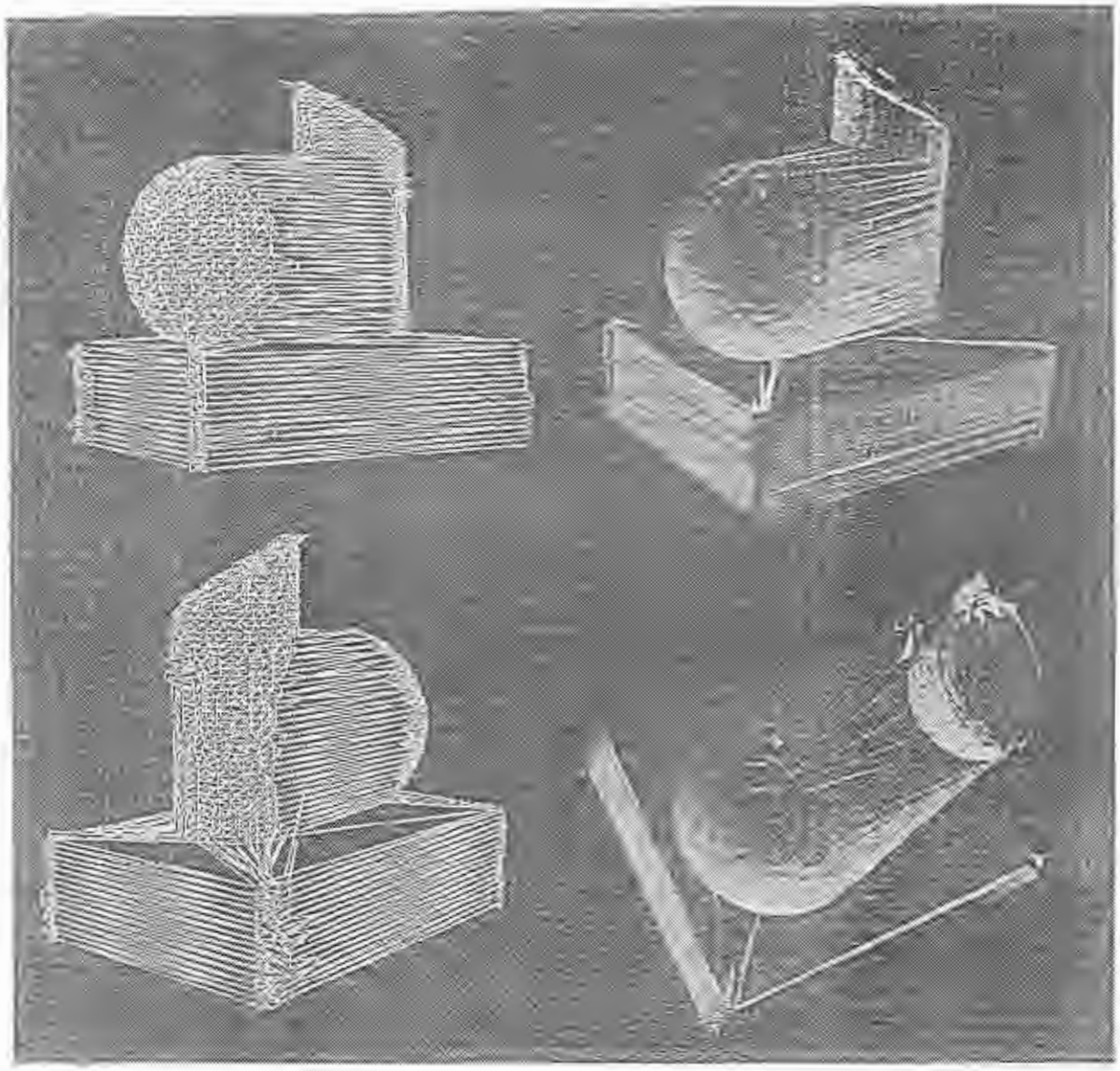


10(a). Right silhouette distribution W(θ,y).

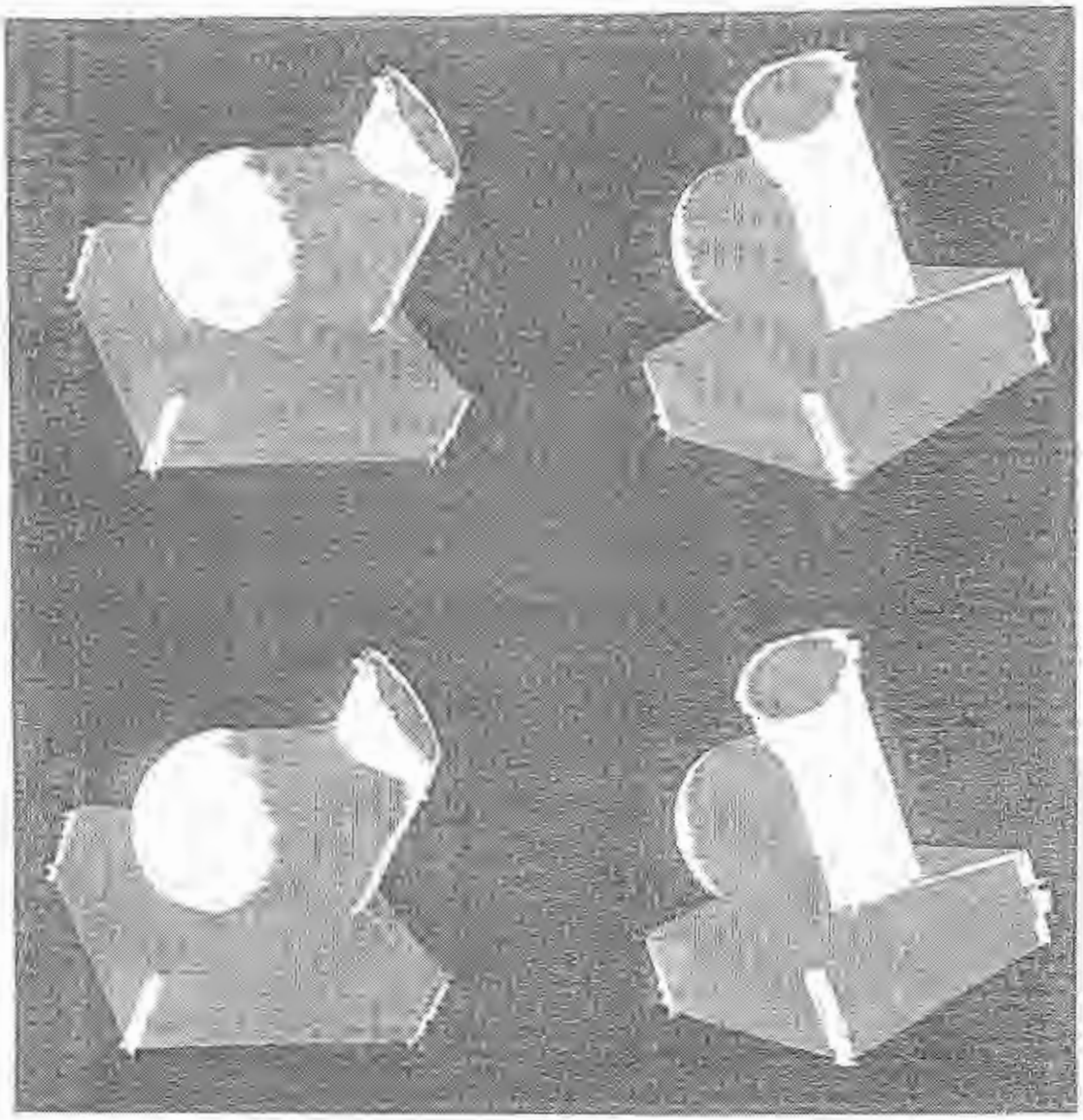


10(b). Corner points & discontinuous points segment the contour distribution into many sub-distributions corresponding to different surfaces or edges on objects.

Fig. 10. Analysis of contour distribution of objects shown in Fig. 3.



11(a). Recovered 3D shape .



11(b). Areas unexposed to contours are assigned with a different color (A patch has its color as the average of the connected points).

Fig. 11. 3D shape recovered from contours in Fig. 10 and occluded areas identified.

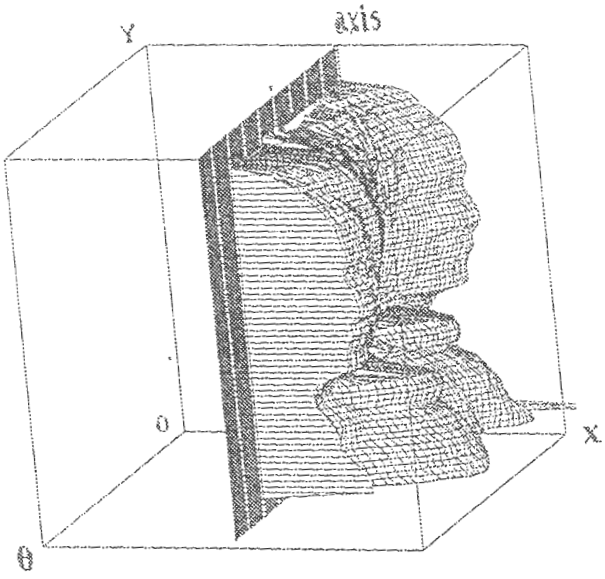
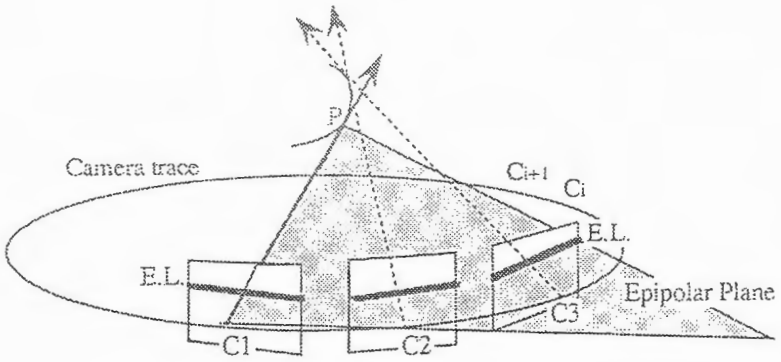
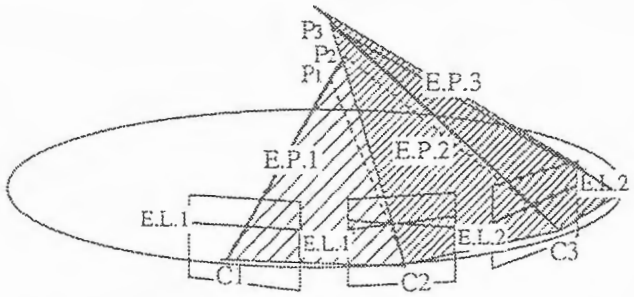


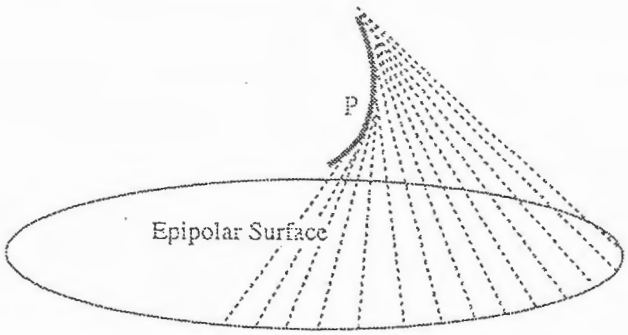
Fig. 12. Silhouette distribution W(θ, y) obtained continuously when images are taken. The s curve and θ curve are used for shape recovery.



a. One epipolar plane (determined by two view points C1, C2 and a spatial point P) is considered as the constraint for rotational camera motion. The corresponding epipolar lines (E.L.) will not be kept in all image frames; as a result, it is not feasible for tracking points in a whole image sequence.



b. Epipolar planes (E.P.) are updated at each successive camera position according to tracked silhouette points (P1, P2, ...). For example, EP1(C1C2P1) is for finding P2, then it is updated to EP2(C2C3P2) for finding P3..... This updating requires a large computation cost and is inaccurate when the image sequence is dense.



c. If an image sequence is sufficiently dense, we can consider the extreme situation that (b) shows. The updated epipolar planes yield a ruled surface (its rulings as the lines of sight tangent to a curve P (P1,P2,P3...)). However, the problem is that this ruled surface is indeterminable from camera motion only. It is also related to the unknown curve P. As a result, the surface is not predictable.

Fig. 13. Difficulty of using epipolar plane constraint in a sequence of contours.

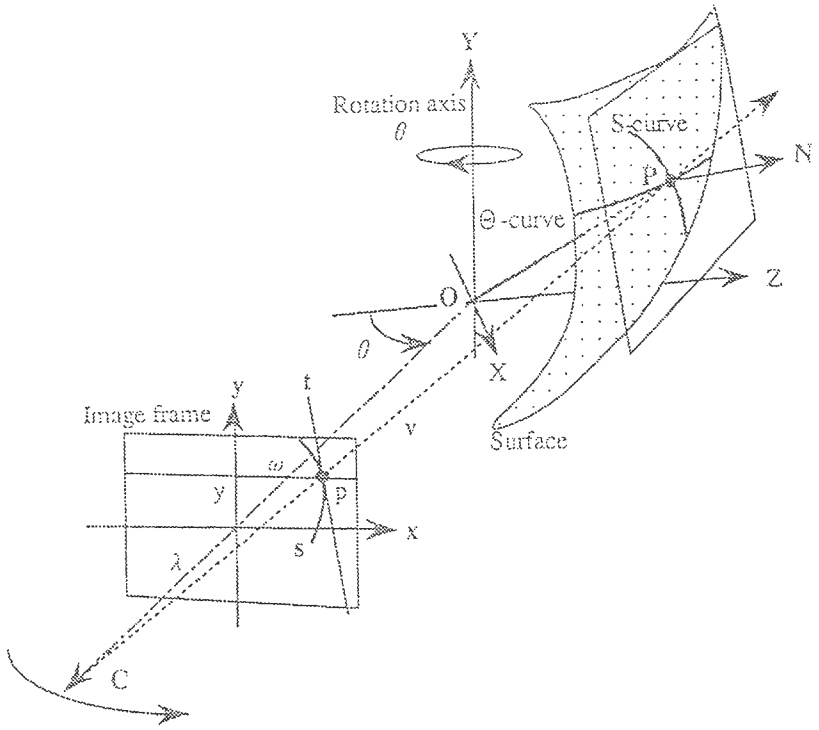


Fig. 14. Perspective projection for silhouettes.

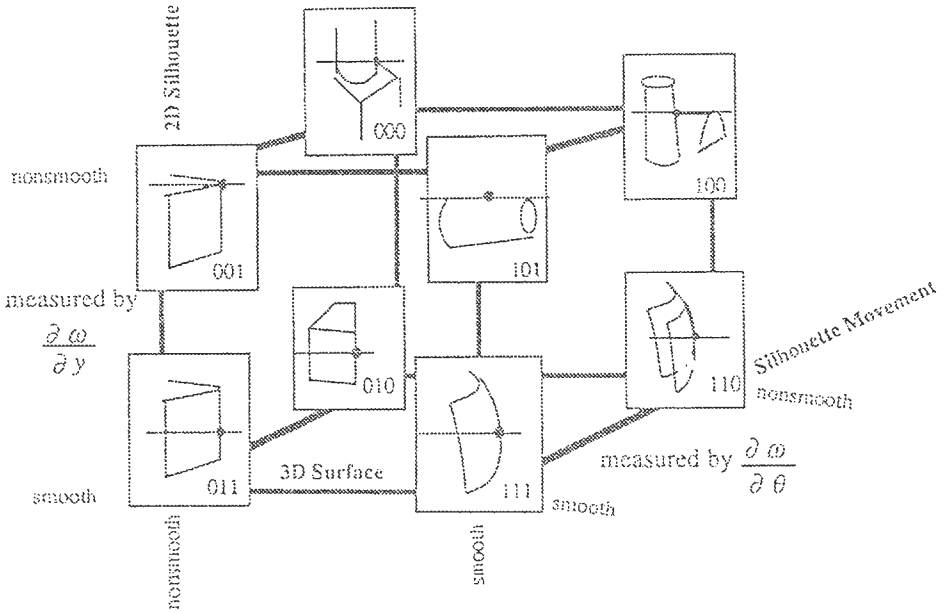
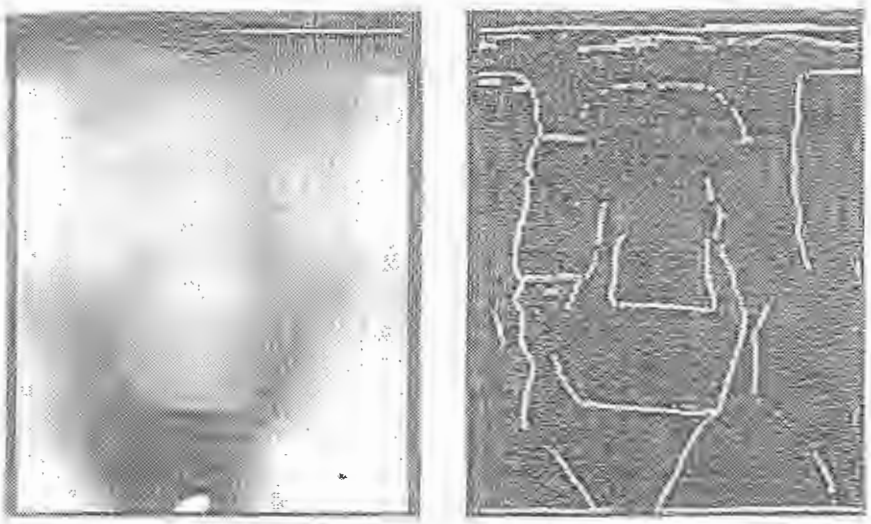


Fig. 15. Different shapes viewed as silhouettes and their relations according to 3D and 2D properties. Dots indicate silhouette points on a given horizontal line.

Fig. 16. A plaster figure for modeling.



a b

17(a). Contour distribution W(θ,y) of the figure in Fig. 16. The image size is 360*512.

17(b). Vertically distributed horizontal corner points and horizontally distributed vertical discontinuous points extracted from distribution W(θ,y).

Fig. 17. Contour distribution and its singular points.

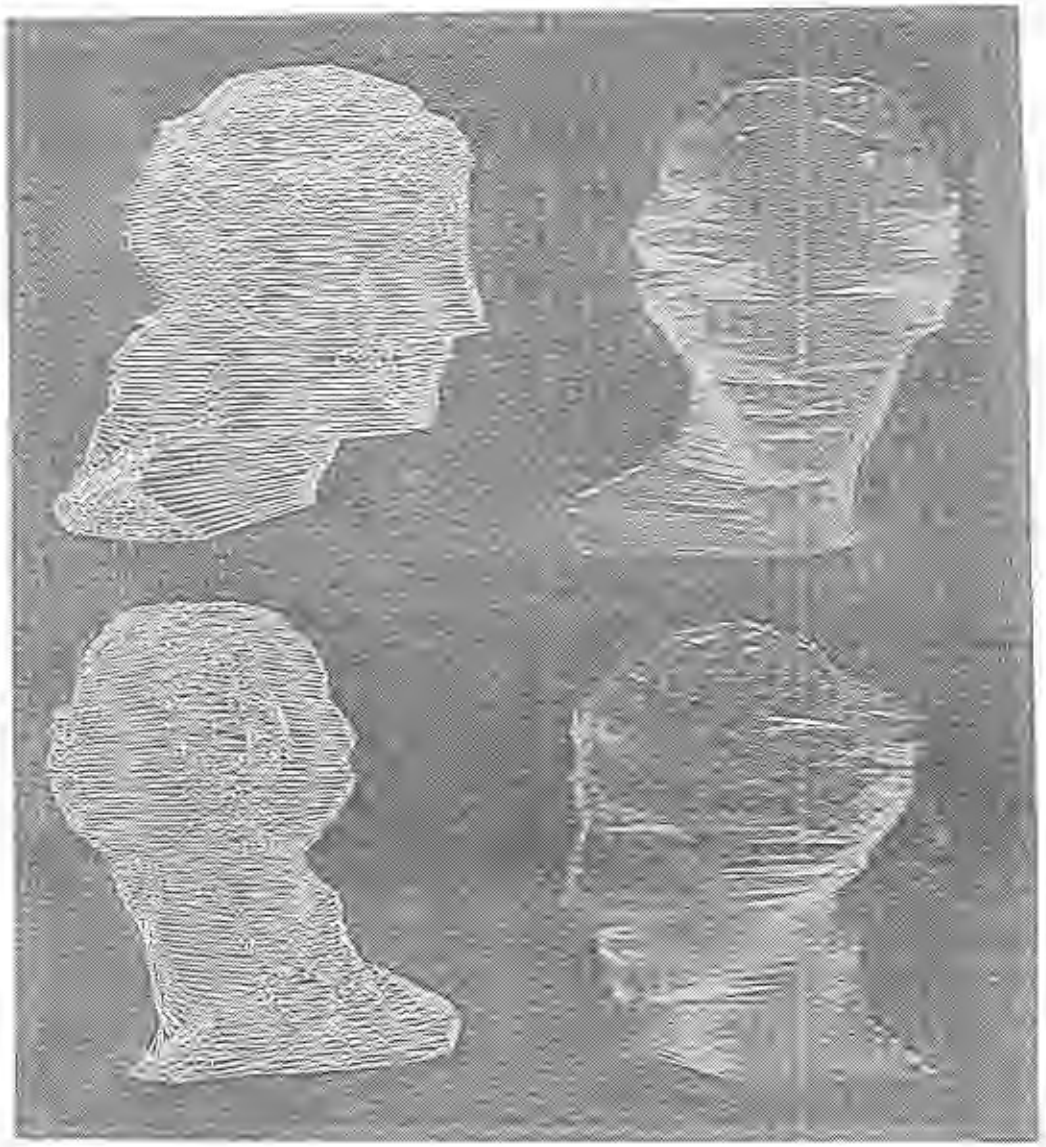


Fig. 18. Wire frame model and shaded model of a recovered figure.

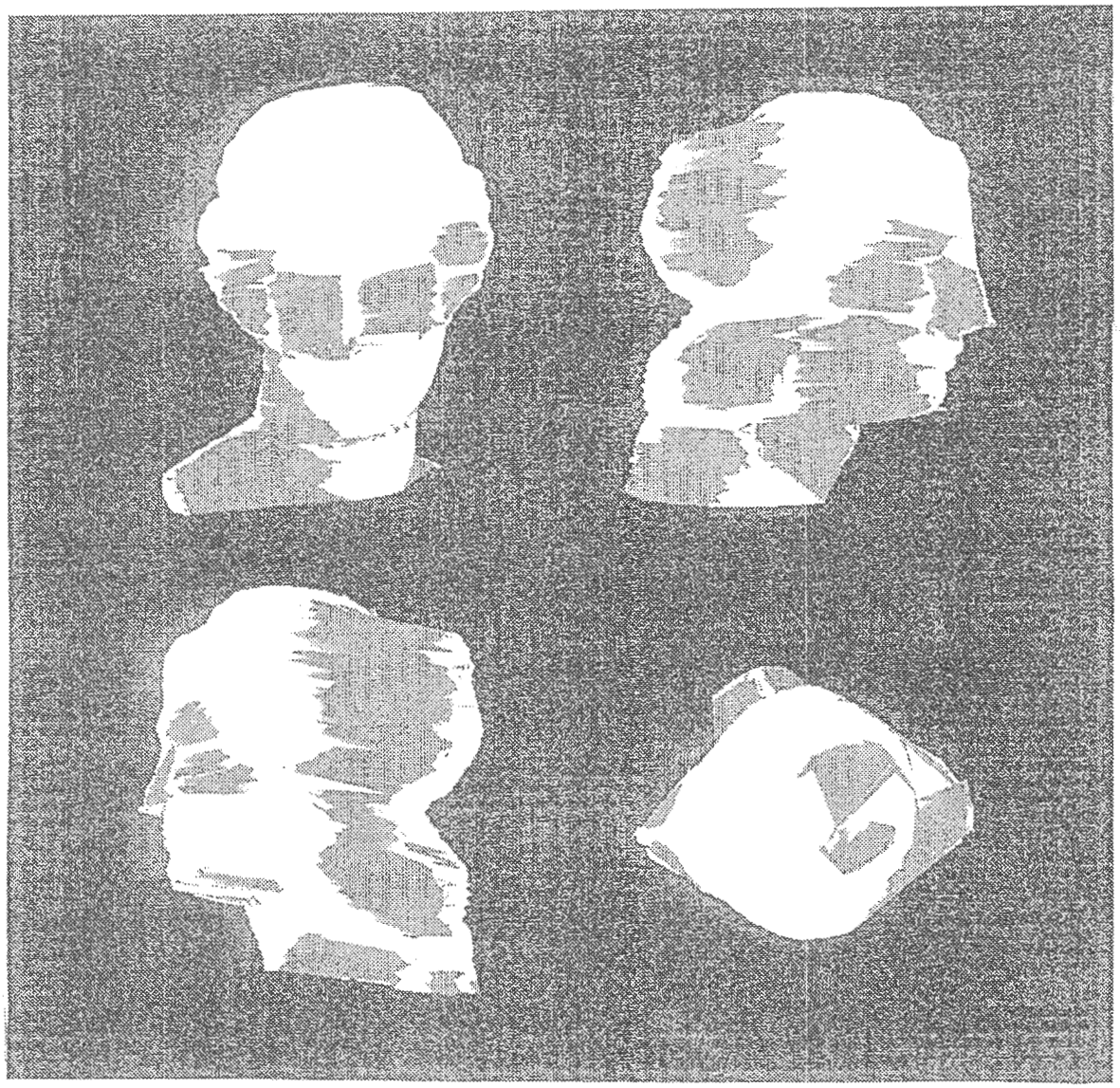


Fig. 19. Area unexposed to any contour on the plaster figure.

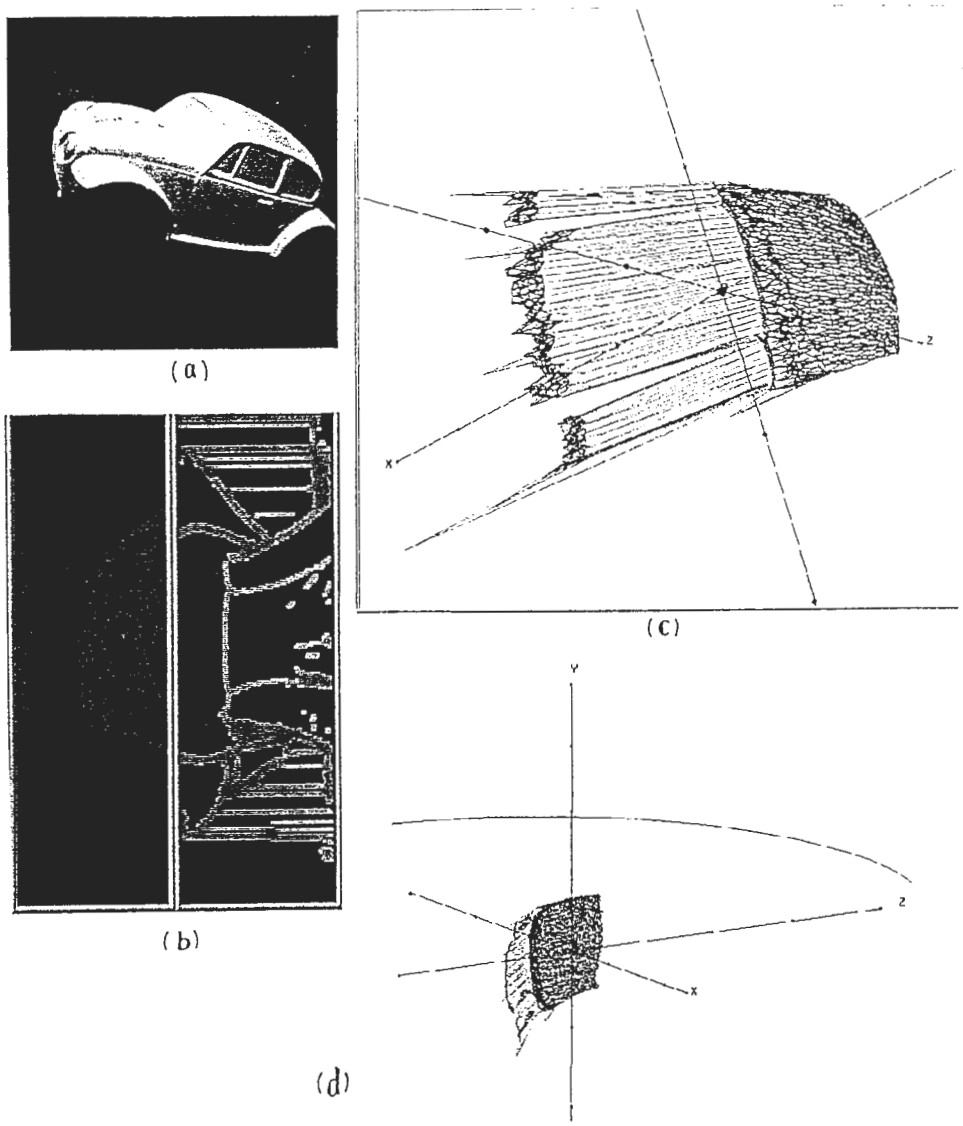


Fig. 20. Shape of a model car obtained from 160 images each with a size of 256*256 pixels under perspective projection. The camera rotates degree by degree in a plane about which the car is symmetrical. (a) A car model. (b) Silhouette distribution and extracted singular points under perspective projection. The horizontal axes represent image number. (c) The Θ and S curves are plotted on the object. The gray part represents a unknown region between the top and the front of the car. (d) The circular line represents the relative move of camera with respect to the car.

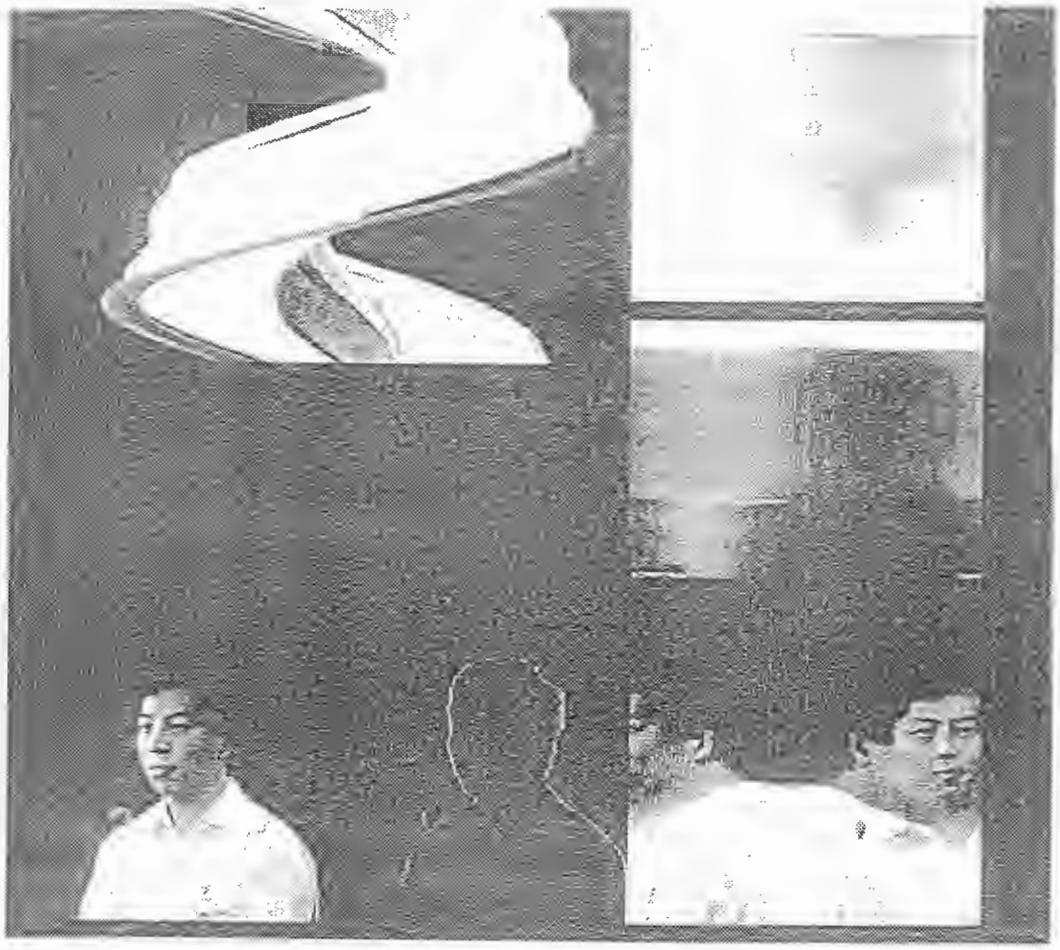
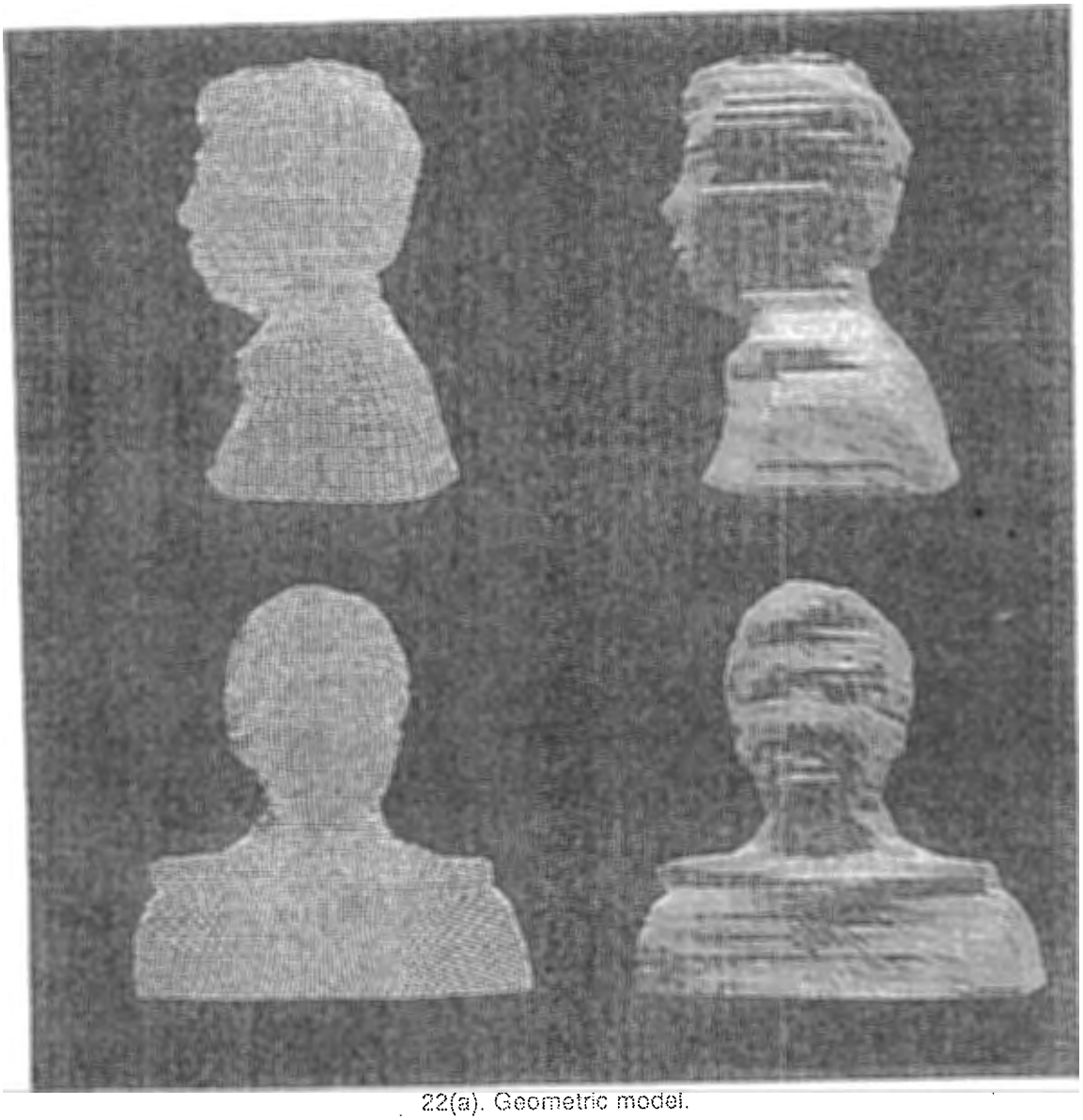
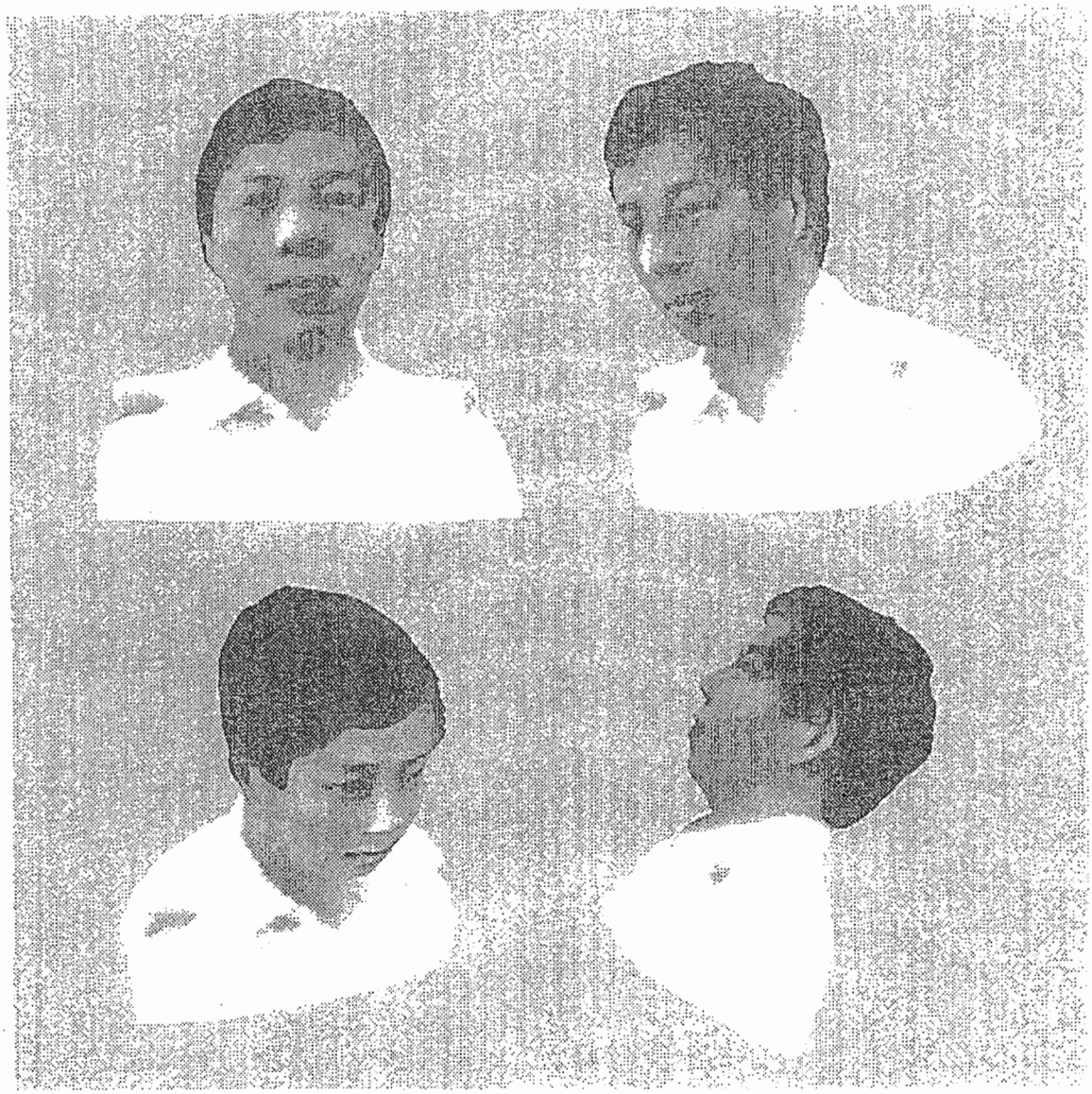


Fig. 21. A process extracting information for building a face model. Lower-left: a window showing input image. Lower-middle: tracked silhouette from the input image. Upper-right: right silhouette distribution where the x coordinate of the silhouette is displayed in the gray level (the horizontal axis is the frame number and the vertical axis is the y axis in the image). Middle-right: left silhouette distribution. Lower-right: a dynamic projection image containing texture data of the face (the horizontal axis is the time axis). Upper-left: an x-t slice at the height of shoulders, where two marks are attached for estimating the rotation angle of the subject when each image is taken.



22(a). Geometric model.



22(b). Texture mapped head.

Fig. 22. A 3D face model obtained from a real person with only a swivel chair and a homogeneous background behind the subject. Several views of the recovered face model are shown. It contains about 60*70 3D points.