



ELSEVIER

Artificial Intelligence 67 (1994) 1-28

---

---

Artificial  
Intelligence

---

---

## Recovery of 3-D objects with multiple curved surfaces from 2-D contours\*

Fatih Ulupinar\*\*

*Department of Computer and Information Sciences, Faculty of Engineering, Bilkent University, Ankara, Turkey*

Ramakant Nevatia

*Institute for Robotics and Intelligent Systems, School of Engineering, University of Southern California, Los Angeles, CA 90089-0273, USA*

Received November 1991; revised December 1992

---

### Abstract

Inference of 3-D shape from 2-D contours in a single image is an important problem in machine vision. Often, techniques to solve this problem examine each surface in the scene separately whereas our perception of their shapes clearly depends on the interplay between them as well. In this paper, we describe a technique that attempts to recover the shapes of all the surfaces of an object simultaneously, though it is limited to objects made of zero-Gaussian curvature surfaces. Our technique is based on an analysis of three kinds of symmetries defined in the paper and the constraints that derive from them, and from other boundaries. This technique uses some of the constraints developed in an earlier paper that was limited to examining a zero-Gaussian curvature surface cut by parallel planes. This restriction has been removed here and the constraints have been reformulated to allow integration of constraints from all the neighboring surfaces. Results on some complex examples are shown.

---

### 1. Introduction

One of the basic goals of mid-level vision is to recover the local orientations of the surfaces of the objects in a scene. Of the many cues available to aid in this

---

\* This research was supported by the Advanced Research Projects Agency of the Department of Defense and was monitored by the Air Force Office of Scientific Research under Contract No. F49620-90-C-0078. The United States Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright notation hereon.

\*\* Corresponding author. Present address: F. Ulupinar, ACSC, 3000 S. Robertson Bl., Suite 400, Los Angeles, CA 90034, USA. E-mail: ulupinar@iris.usc.edu.

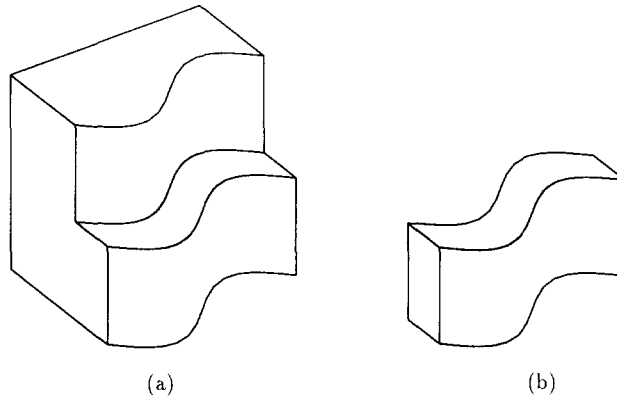


Fig. 1. (a) An object consisting of multiple planar and curved surfaces, and (b) the front part of the object in isolation.

process, we believe that shape of the 2-D contour itself is the most important and robust one. Of course, inferring shape from contour is highly ambiguous and cannot be done without making some assumptions. The goal in shape from contour methods is to minimize the number of needed assumptions and to achieve results consistent with human perception. For shape from contour analysis, the only *ground truth* is really in human perception, for even if the given contour was obtained by a real object, it *could* have been also obtained by any number of other objects as well.

Early work on inferring 3-D structure from a 2-D shape was focused on analysis of line drawings of polyhedra [3,6–9]. In the 80s, several techniques for non-polyhedral shapes were proposed (e.g. [1,2,5,10,11,13,18,19]). One characteristic of most of these methods is that they examine only a single surface in the scene at a time whereas our perception of a surface can be strongly influenced by our perception of the entire object.

In earlier papers, we have examined the recovery of 3-D surface shape of a variety of curved surfaces cut by planes [14,15,17]). Our techniques rely on observed symmetries in the image and the analysis depends on the interplay between constraints imposed by the curved surface and the planes cutting it. We showed successful shape recovery for the following kinds of surfaces: zero-Gaussian curvature surfaces [14], surfaces of straight homogeneous generalized cylinders [15] and surfaces of planar, right, constant cross-section generalized cylinders [17].

Complex objects, however, are composed of a number of curved surfaces and planar patches. Our perception of each of these surfaces is affected by the presence of the others. For example consider the object in Fig. 1(a) which appears to be a composite of two objects, one in front of the other. If the front object is viewed in isolation as in Fig. 1(b), the interpretation for its top surface is ambiguous (it may be planar or not). However, in the context of the whole object (a), this ambiguity disappears (the said surface must be planar). This is an example of how a remote surface changes the perception of some

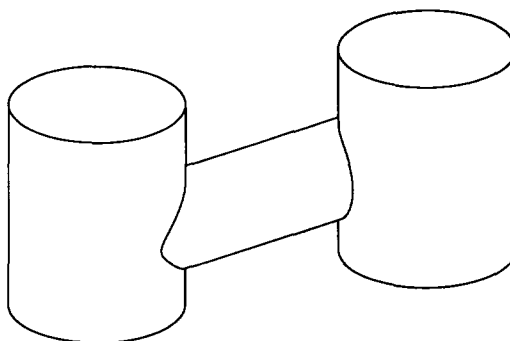


Fig. 2. Three cylinders intersected such that the middle cylinder has no planar cross-section. The perception of the middle cylinder, as being a circular cylinder, is much weaker compared to the side cylinders.

other surface drastically. In general, even if the perception of a surface may not be affected by the neighboring surfaces this drastically, its perception would still be affected by small amounts to make the whole object more consistent (surfaces obeying inter-surface constraints).

This paper explores 3-D surface inference by including interplay between many surfaces that may comprise a complex object. Our technique is limited to a combination of zero-Gaussian curvature (ZGC) surfaces and planar surfaces, such that the intersection curves are planar. We can actually test the planarity of an intersection curve by using symmetries, since the symmetries presented in this paper only exist when the intersection curves are planar, except for very specific intersections violating the generality assumptions. We also conjecture that the shape information from contours degrades if the intersection of the surface produces non-planar curves. Fig. 2 shows an example.

Our method uses constraints similar to those used in our earlier work on shape from contour for ZGCs [14]. The previous work, however, was limited to analysis of a ZGC surface cut by two parallel planes. In this paper, we first develop techniques that allow analysis of a ZGC surface cut by non-parallel planes and then show how multiple ZGC surface shapes can be recovered simultaneously while influencing each other. To accomplish this, we have found it more convenient to change some of the representations used in our earlier work as well as to devise an additional form of symmetry.

In Section 2 we define three kinds of symmetries and discuss the occurrence of these symmetries in the context of planar and ZGC surfaces. In Section 3 we discuss the constraint equations that are used in the shape recovery. In Section 4 the representation of the surfaces and single-surface recovery is discussed. In Section 5 combined shape recovery of multiple surfaces is discussed in detail. In Section 6 we discuss an implementation of the shape recovery algorithms and present some results.

Our method assumes that clean, closed boundaries are given (or can be extracted from the real image). We do not address the issue of separating object

boundaries from surface markings, or other *perceptual grouping* operations here, though we believe that constraints required for shape inference by our methods will aid in the perceptual organization process itself. Such research is being currently pursued in our laboratory separately.

We assume orthographic projection throughout the paper unless specifically mentioned otherwise. (In a separate paper, we have shown how many of the constraints for orthographic projection can be transformed to the case of perspective projection [16].)

In this paper we will use *gradient space* to represent the orientation of surfaces (given by their normals). To review, the normal,  $N$ , of a plane  $ax + by + cz + d = 0$  is given by the vector  $N = (a, b, c)$ . This can be rewritten as  $(p, q, 1)$ , where  $p = a/c$  and  $q = b/c$  (Note that this excludes cases where  $c = 0$ , however, such planes are parallel to the line of sight and are not imaged as planes under orthographic projection anyway.)  $(p, q)$  can be thought of as defining a two-dimensional space, called the gradient space, such that every point in this space corresponds to the normal of a plane in 3-D.

## 2. Surfaces and symmetries

In this paper we concentrate on shape from contour for objects composed of planar and zero-Gaussian curvature surfaces. A zero-Gaussian curvature (ZGC) surface is one where the Gaussian curvature (the product of the maximum and minimum principal curvatures) of the surface is zero everywhere. Cylinders and cones are examples of a ZGC surface. These surfaces are also called developable surfaces since they can be generated from a piece of paper by rolling and/or bending without cutting. We feel that ZGC surfaces comprise a large and useful class and that they represent a natural step up in complexity from the study of planar surfaces that have dominated previous work in the field. Lines of minimum curvature for a ZGC surface, also called rulings, are straight, i.e., it is possible to embed straight lines on a ZGC surface along these rulings.

### 2.1. Symmetries

We define three types of symmetries that we call *parallel* symmetry, *line-convergent* symmetry and *skew* symmetry. We show when such symmetries can be expected to occur and how they can be used to infer qualitative shape properties.

For curves to be symmetric, certain point-wise correspondences between two curves must exist. We will call the lines joining the corresponding points on the curves the *lines of symmetry*, the locus of the mid-points of these lines the *axis of symmetry*, and the curves forming the symmetry the *curves of symmetry*.

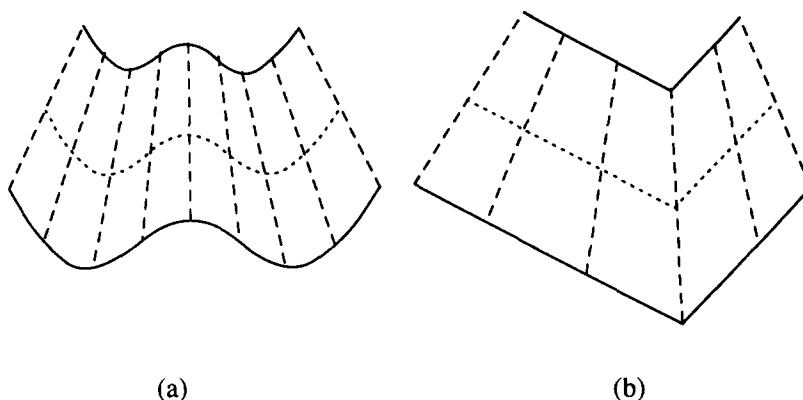


Fig. 3. Examples of (a) parallel symmetry with curved contours, and (b) parallel symmetry with straight contours. The dotted curves are axes of symmetry and the dashed lines are lines of symmetry.

### 2.1.1. Parallel symmetry

Consider two curves  $X_i(s) = (x_i(s), y_i(s))$ , for  $i = 1, 2$ , parameterized by arc length  $s$ . Let  $T_i(s) = (x'_i(s), y'_i(s))$  be the unit tangent of the curves. Then,  $X_1(s)$  and  $X_2(s)$  are said to be parallel symmetric if there exists a correspondence function  $f(s)$  between them such that  $T_1(s) = T_2(f(s))$  for all values of  $s$  for which  $X_1$  and  $X_2$  are defined and  $f(s)$  is a continuous monotonic function. A useful special case is when  $f(s)$  is restricted to be a linear function. In that case, the symmetry condition becomes:  $T_1(s) = T_2(as + b)$ , where  $a$  and  $b$  are constant ( $a$  may be thought of as a scale parameter). Parallel symmetry with linear correspondence is found in cylindrical and conic surfaces (subsets of ZGC surfaces). In the context of ZGC surfaces parallel symmetry is obtained when ZGC surfaces are intersected by parallel planes [14]. Some examples are given in Fig. 3.

### 2.1.2. Line-convergent symmetry

Two image curves  $C_1$  and  $C_2$  are line-convergent symmetric if the tangents of  $C_1$  and  $C_2$ , at the corresponding points, intersect along a line, say  $l$ , on the image plane. This is shown in Fig. 4. Parallel symmetry may be thought of as a limiting case of line-convergent symmetry where the line of intersection is at infinity. We show later, in Section 2.2, that this symmetry is found in curves obtained by cutting a ZGC surface with two non-parallel planes. A parallel symmetry also turns into line-convergent symmetry under perspective projection [16]. It is also present for limbs of straight homogeneous generalized cylinders [15].

### 2.1.3. Skew symmetry

In this symmetry, the point-wise correspondence should be such that the axis of the symmetry is straight, and the lines of symmetry are at a constant

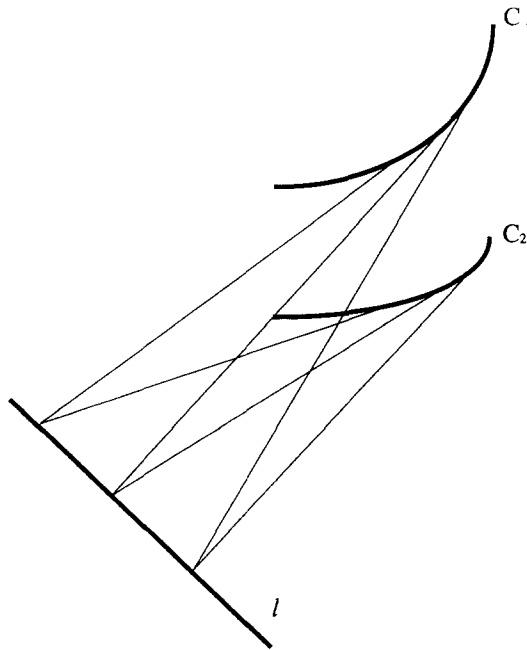


Fig. 4. Two line-convergent symmetric curves.

angle (not necessarily orthogonal) to the axis of symmetry. Skew symmetry was first proposed by Kanade [7] and used in the analysis of scenes of polyhedral objects. Examples are given in Fig. 5. In [14] we prove that, if a 3-D contour, formed by non-limb edges, produces a skew symmetric line drawing in the image plane such that the 3-D correspondence is invariant under small perturbations of the viewpoint, then the 3-D contour must be planar (under the assumption of general viewpoint).

## 2.2. Symmetries in surfaces

The symmetries discussed in the previous sections are present in ZGC and planar surfaces. Symmetries also provide strong information about the type of the surface. In [14] we showed that, if a closed contour composed of non-limb edges has a skew symmetry, then the contour has to be planar under the assumption of general viewpoint and, if the correspondence is static, with respect to changing viewpoint. A ZGC surface cut by parallel planes produces parallel symmetry. Moreover, we showed that a figure bounded by one parallel symmetry and one skew symmetry with straight lines of symmetry must be a ZGC surface (assuming general viewpoint in both cases). Line-convergent symmetry is produced when a ZGC surface is intersected by non-parallel planes.

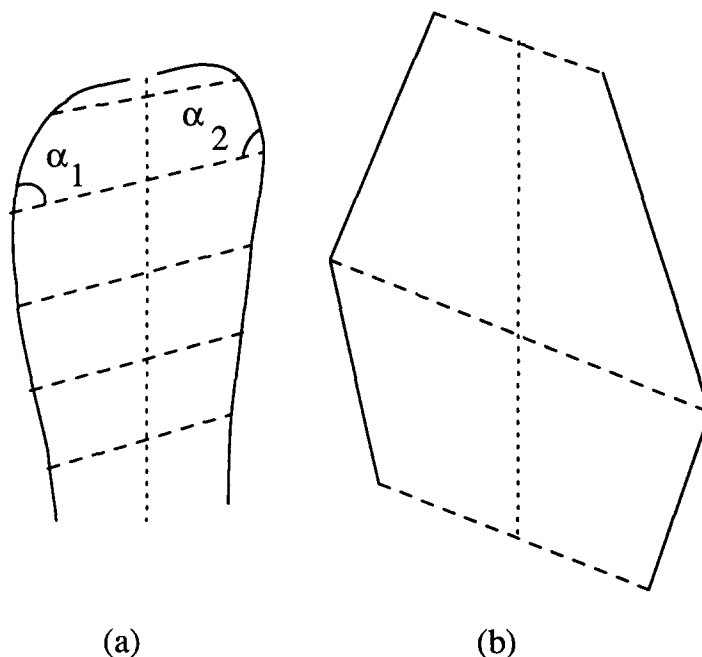


Fig. 5. Examples of (a) skew symmetry with curved contours, (b) and skew symmetry with straight contours. The dotted curves are axis of symmetry and the dashed lines are lines of symmetry.

**Theorem 2.1.** *Curves  $C_1$  and  $C_2$  obtained by cutting a ZGC surface  $S$  by two non-parallel planes  $\Pi_1$  and  $\Pi_2$  project as line-convergent symmetric curves such that the lines joining the corresponding points of the image curves are the projections of the rulings of  $S$  and the line  $l$  formed on the image plane by joining the intersection points of the tangent lines of the line-convergent symmetric curves is the projection of the 3-D intersection line of the planes  $\Pi_1$  and  $\Pi_2$ .*

**Proof.** The above theorem is visualized in Fig. 6. The key to the proof of this theorem is that the tangent plane, plane  $T$ , in Fig. 6 of the ZGC surface  $S$  is the same along the rulings of  $S$ . Therefore, both the tangent lines,  $t_1$  and  $t_2$ , of the curves  $C_1$  and  $C_2$  from points  $P_1$  and  $P_2$  are on plane  $T$ . Also the tangent line  $t_1$  is on plane  $\Pi_1$  and  $t_2$  is on  $\Pi_2$ . Therefore intersection of  $t_1$  and  $t_2$  is necessarily at the intersection point of the three planes  $\Pi_1$ ,  $\Pi_2$  and  $T$ . For other rulings the same things repeat for a different  $T$  plane, and all the tangent line intersections take place along the line  $l$ , the intersection line for planes  $\Pi_1$  and  $\Pi_2$ . Hence, on the image plane too the intersection of the tangents takes place on the projection of the line  $l$ .  $\square$

Note that the reverse of this theorem, that line-convergent symmetry curves *must* come from non-parallel planar cuts of ZGC surfaces, is not valid. However, we believe that it is reasonable to infer that line-convergent symmetry

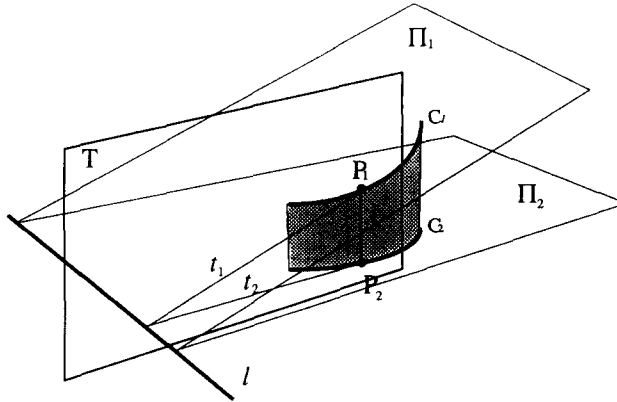


Fig. 6. Formation of the line-convergent symmetry with a ZGC surface and two non-parallel planes.

curves are planar cross-sections of ZGC surfaces, if they are terminated by line segments (corresponding to the rulings). For a ZGC surface to produce line-convergent symmetry with non-planar cuts, the cuts must be in a very specific direction, considerably decreasing the probability of a non-planar cut.

### 3. Constraints on surface shape

We will solve the recovery of shape from contour problem as a constraint minimization problem. The constraints discussed here are the building blocks of constraints and error terms of the minimization. The constraints are originally stated in gradient space. However, the gradient space is not uniform, i.e., a constant shift at the center of the gradient space corresponds to a larger vector difference in 3-D than the same shift somewhere further away from the center. For the constraints discussed, the uniformity of the constraint function implies that the error returned by the constraint function, when not satisfied exactly, depends on the 3-D vector differences rather than the differences in gradient space. When minimization on the gradient space points is employed, the minimization terms should actually be on 3-D angle differences not on Euclidean distances on the gradient space. The purpose of the normalization is to eliminate the nonuniform contributions by the gradient space points. A gradient  $(p, q)$  corresponds to a 3-D vector of  $v = (p, q, 1)$ . The projection of  $v$  on the unit sphere is given by

$$v_s = \frac{1}{\sqrt{p^2 + q^2 + 1}}(p, q, 1). \quad (1)$$

The vector  $v_s$  is the normalized (i.e.,  $|v_s| = 1$ ) form of the vector  $v$ , and  $v_s$  is only dependent on the orientation of the vector  $v$ , it has no length information. Eq. (1) shows that a constant shift in parameters  $p$  and  $q$  of  $v_s$  has less significance, i.e., affects the components of  $v_s$  less, as  $p$  and  $q$  get



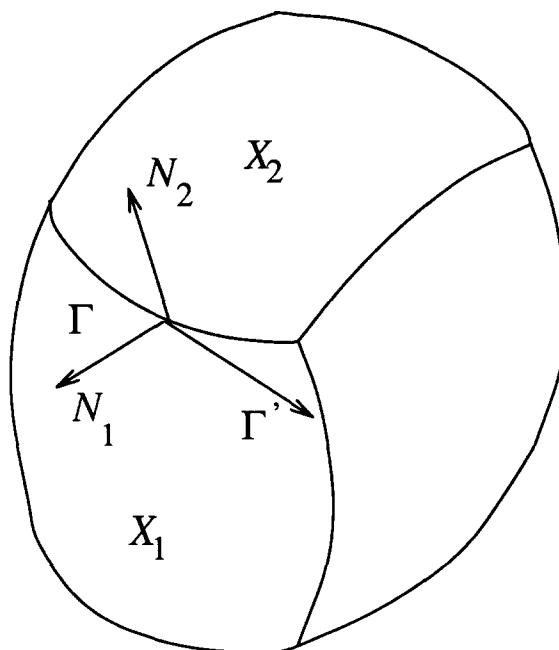


Fig. 7. Two curved surfaces meeting along a curve  $\Gamma$ .

larger. The normalized constraint error functions are formulated to compensate for this behavior of gradient space. The drawback of this normalization is that linear constraint functions are no longer linear.

There are three basic constraint equations that are used in the shape recovery process. We call them the shared boundary constraint, the orthogonality constraint, and the equality constraint. Shared boundary and equality constraints are derived from geometry, the orthogonality constraint is derived from perceptual observations. We first state the constraint and give the equation of the constraint, then we give a reformulation of the constraint such that the error given by the constraint is uniform and normalized, i.e., in the range  $[0, 1.0]$ .

### 3.1. Shared boundary constraint (SBC)

This constraint relates the orientations of the two surfaces on opposite sides of an edge. The planar version of this constraint has been used since early days in polyhedral scene analysis [9]. Shafer et al. [12] first extended it to the case of intersection of curved surfaces.

Consider two surfaces  $X_1(u, v)$  and  $X_2(u, v)$  meeting along a curve  $\Gamma(s) = (x(s), y(s), z(s))$  as in Fig. 7. Let  $N_1(u, v)$  and  $N_2(u, v)$  be the normals of  $X_1$  and  $X_2$  respectively. Along the curve  $\Gamma(s)$  we can represent the normals  $N_1$  and  $N_2$  as  $N_i(s) = N_i(u_i(s), v_i(s))$ . Let the normals  $N_i(s)$  be represented in  $p$ - $q$  space as  $N_i(s) = (p_i(s), q_i(s), 1)$ . In [14] we prove the following constraint:

$$\begin{aligned} (x'(s), y'(s), z'(s)) \cdot ((p_2(s), q_2(s), 1) - (p_1(s), q_1(s), 1)) &= 0, \\ x'(s)(p_2(s) - p_1(s)) + y'(s)(q_2(s) - q_1(s)) &= 0. \end{aligned} \quad (2)$$

We call this the shared boundary constraint (SBC) which states that along the curve  $\Gamma(s)$  the orientation of the surfaces  $X_1$  and  $X_2$  are constrained by the tangent,  $(x'(s), y'(s))$  of the image of the curve  $\Gamma(s)$  under orthographic projection. Specifically, the line joining  $N_1(s)$  and  $N_2(s)$  must be orthogonal to the tangent of the image of the curve  $\Gamma(s)$ .

A stronger constraint can be obtained if we can assume that the intersection curve,  $\Gamma$ , is planar. Say,  $\Gamma$  lies in a plane with orientation  $(p_c, q_c)$ . With the assumption of planarity the constraint equation becomes

$$x'(s)(p_c - p(s)) + y'(s)(q_c - q(s)) = 0. \quad (3)$$

For ZGC surfaces, we will assume that the parallel or line-convergent symmetric curves are planar.

The normalized form of this constraint is a nonlinear equation and it is applied between two gradients  $(p_1, q_1)$  and  $(p_2, q_2)$ , and a 2-D vector  $(x, y)$ . The constraint is

$$SBC(p_1, q_1, p_2, q_2, x, y) = \frac{((p_2 - p_1)x + (q_2 - q_1)y)^2}{((p_2 - p_1)^2 + (q_2 - q_1)^2 + 1)(x^2 + y^2)}. \quad (4)$$

### 3.2. Orthogonality constraint

The orthogonality assumption has been used in various forms by many authors, based on the belief that the human visual system prefers certain orthogonal interpretations. It was first studied for skew symmetric planar contours by Kanade [7]. We will assume orthogonality between the axis of parallel symmetry and the lines of parallel symmetry. For a ZGC surface, this is equivalent to slicing the surface along rulings to obtain thin skew symmetric planar strips and assuming that these strips are orthogonally symmetric in 3-D. We justify the orthogonality by observation of human perception. A circular cylinder cut by a non-orthogonal plane is, for example, perceived as an orthogonal cylinder [14]. In [14] a detailed discussion of the orthogonality assumption is provided which is omitted here.

Consider Fig. 8, on the surface analyzed, say the image of the tangent vector  $A$  makes an angle  $\alpha$  with the horizontal and the image of the tangent vector  $B$  makes an angle  $\beta$  at some point on the surface, such that  $A$  and  $B$  are hypothesized to be orthogonal in 3-D. Let the normal of the surface be  $N = (p, q, 1)$  at that point. Since the 3-D tangent vectors  $A$  and  $B$  lie on the tangent plane of the surface they can be represented as

$$\begin{aligned} A &= (\cos(\alpha), \sin(\alpha), p \cos(\alpha) + q \sin(\alpha)), \\ B &= (\cos(\beta), \sin(\beta), p \cos(\beta) + q \sin(\beta)). \end{aligned} \quad (5)$$

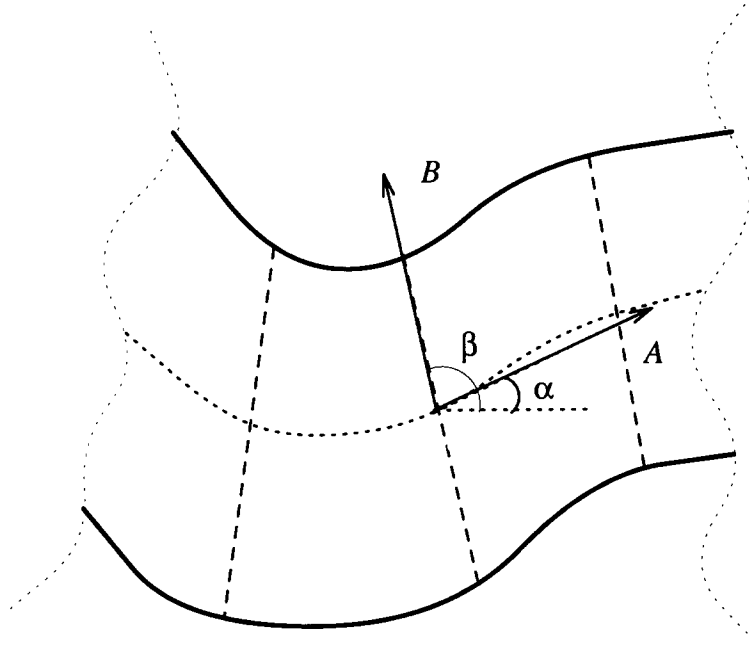


Fig. 8. Orthogonality constraint.

From the orthogonality of the 3-D vectors  $A$  and  $B$  we get  $A \cdot B = 0$ ; this is the equation of a hyperbola in the  $p$ - $q$  plane since

$$\cos(\alpha - \beta) + (p \cos \alpha + q \sin \alpha)(p \cos \beta + q \sin \beta) = 0. \quad (6)$$

As in the case of the shared boundary constraint, this constraint is not normalized either. The normalized constraint is a function of two image directions  $(x_1, y_1) = (\cos(\alpha), \sin(\alpha))$ ,  $(x_2, y_2) = (\cos(\beta), \sin(\beta))$  and the surface gradient  $(p, q)$  given by

$$O(p, q, x_1, y_1, x_2, y_2) = \frac{((x_1, y_1, px_1 + qy_1) \cdot (x_2, y_2, px_2 + qy_2))^2}{|(p_1, q_1, 1)|^2 |(p_2, q_2, 1)|^2}. \quad (7)$$

### 3.3. Equality constraint

This constraint is applied when two gradients are hypothesized to be equal. For two gradients  $(p_1, q_1)$  and  $(p_2, q_2)$  the unnormalized constraint is

$$|(p_1, q_1) - (p_2, q_2)| = 0. \quad (8)$$

Since the gradient space is not uniform, using Euclidean distance between vectors  $(p_1, q_1)$  and  $(p_2, q_2)$  is not a normalized and uniform error measure. Therefore, the square of the sin of the 3-D unit vectors corresponding to the gradients  $(p_1, q_1)$  and  $(p_2, q_2)$  is used:

$$Eq(p_1, q_1, p_2, q_2) = 1 - \frac{((p_1, q_1, 1) \cdot (p_2, q_2, 1))^2}{|(p_1, q_1, 1)|^2 |(p_2, q_2, 1)|^2}. \quad (9)$$

#### 4. Analysis of ZGCs cut by non-parallel planes

In order to be able to include the contributions of the constraints from each surface and the inter-surface constraints into a pool of constraints, an appropriate parameter representation for each surface is necessary. The parameterization we use is described below.

##### 4.1. Parameterization of surfaces

For planar surfaces the gradient space representation  $(p, q)$  of the surface normal of the plane is used. This is the natural and most versatile (for our purposes) representation for planar surfaces.

In the following we discuss the parameterization and computation of local surface normals, for the case of ZGC surfaces cut by non-parallel planes. As discussed in Section 2.2 a ZGC surface cut by parallel planes produces parallel symmetric curves and a ZGC surface cut by non-parallel planes produces line-convergent symmetric curves. The non-parallel cut case is the more general case.

In order to compute line-convergent symmetry between two curves on a general ZGC surface we must try all possible monotonic point correspondences between the curves. This is a very costly search. However, for the case of cylindrical and conic surfaces, the computation of line-convergent symmetry is much simpler. Most of the ZGC surfaces that we encounter in our environment are in fact cylindrical or conic surfaces. It is known [4] that a ZGC surface consists of cylindrical, conic and tangent surfaces. Tangent surfaces are quite uncommon in the real world, therefore we concentrate on cylindrical and conic surfaces. To ease the job of symmetry finding, a general ZGC surface can be segmented into cylindrical and conic surfaces. However, the constraints and solutions are modeled for a general ZGC surface. Also, there is no problem with over-segmentation as long as the constraints are properly identified between surfaces. The suggested segmentation is along the rulings that passes through the inflexion points of the cross-section curves. The main reason is that, a ruling which passes through the inflexion point of a cross-section curve has the property that the intersection point of this ruling with any curve on the surface is an inflexion point of the curve. This property indicates that inflexion points of two cross-section curves correspond (in the sense of symmetry), providing a corresponding point pair without computing symmetry. Moreover, empirically, inflexion points of the cross-section curves are good candidates for the change of surface type.

A rigorous segmentation can be obtained if a robust linear correspondence line-convergent symmetry finder can be developed. Such a symmetry finder can

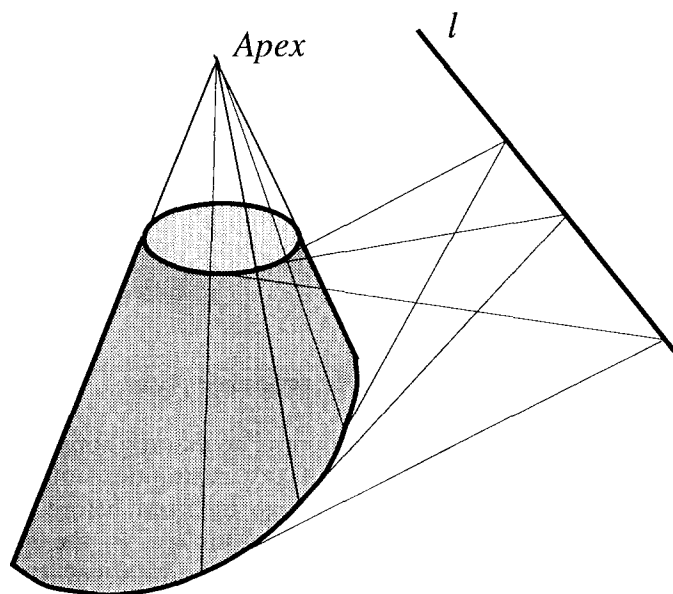


Fig. 9. A conic surface with line-convergent symmetry.

decide on a segmentation point when the symmetry correspondence becomes nonlinear.

For conic surfaces correspondence of line-convergent symmetry is restricted to be along the lines that pass through a common point, the apex of the cone. In Fig. 9 line-convergent symmetry is shown for a cone. The computation of line-convergent symmetry for conic surfaces is, therefore, restricted to checking the correspondences between the curves such that the rulings, when extended, intersect at a single point on the image plane. The process is further simplified when the end points of the curves are available. In that case the apex point can easily be computed on the image plane and it is only needed to check the correspondence for that apex point. For cylindrical surfaces the apex point is at infinity, therefore, the direction of the apex is used rather than the location of it.

To recover the local surface normals for ZGCs cut by non-parallel planes, we need to decide which cross-section curve is to be made orthogonal to the rulings. Fig. 10 shows three possibilities. In (a) the general preference is to make the top cross-section curve orthogonal, in (b) the bottom one is preferred and in (c) we prefer the middle curve, which is the axis of line-convergent symmetry. In our implementation the curve that “looks” orthogonal to the image axis, which is the line joining the mid-points of the lines joining the end points of the line-convergent symmetry curves, is chosen as the curve that will be made orthogonal. How orthogonal a curve looks is determined by the angle between the image axis and the line joining two ends of the symmetry curve.

In [14] we show that, a ZGC surface cut by parallel planes, (i.e., it has par-

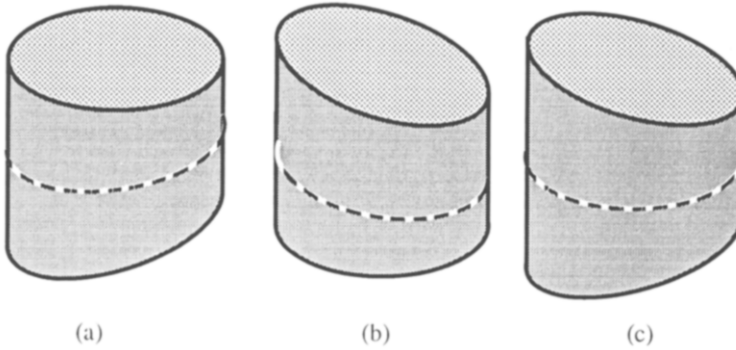


Fig. 10. Three ZGCs cut by non-parallel planes.

allel symmetry), has three degrees of freedom if no assumptions are used. That is, after the constraints are applied without any assumptions of orthogonality or regularity there remain three parameters to be fixed before local surface normals can be computed at all points on the surface. In the more general case of ZGC surfaces cut by non-parallel planes two additional parameters are involved: the gradient of the second cross-section plane. In total a ZGC surface cut by non-parallel planes has five degrees of freedom. Of the five degrees of freedom, four parameters are the gradients  $(p_t, q_t)$  and  $(p_b, q_b)$  of the top and the bottom planes cutting the ZGC surface. The fifth parameter is discussed below. Here we use a slightly different parameterization and approach for computing surface normals than the one we proposed in [14]. This formulation permits the computation of local surface normals of ZGC surfaces at random locations without computing the surface normal at all points on the surface. This property is needed in the combined shape recovery algorithm.

We can model a conic surface by using any 3-D axis that goes through the apex of the cone. We use the 3-D axis that projects as the 2-D axis of the straight edges of the cone. If the image direction of the axis is  $(a_x, a_y)$ , then the 3-D direction of the axis in gradient space representation is  $(ua_x, ua_y)$ , where  $u$  is a free variable; this is the fifth parameter of the ZGC representation. Given  $(p_t, q_t)$ ,  $(p_b, q_b)$ , and  $u$ , the surface gradient can be computed at any point on the surface. In the next subsection we discuss the computation of the local surface normals of a ZGC surface for a particular instance of its parameters.

#### 4.2. Recovering a ZGC surface from its parameters

Consider Fig. 11, let  $(p_t, q_t)$  be the gradient of the cross-section plane that is chosen to be made orthogonal to the surface. The gradient  $(p, q)$  of the surface along the ruling  $r$  is given by the combination of the two linear constraints given below. The first one is the shared boundary constraint, given in Section 3.1, between gradients  $(p, q)$  and  $(p_t, q_t)$  using the tangent  $(x', y')$  of the intersection curve at the point the curve touches the ruling  $r$ . The

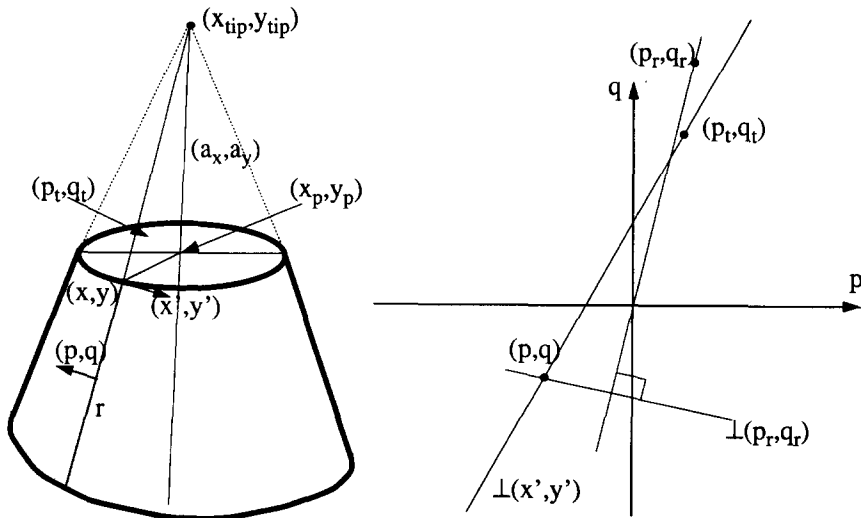


Fig. 11. The parameters of a ZGC surface and the constraints in the gradient  $(p, q)$  of the surface along the ruling  $r$ .

equation of the constraint is

$$(p - p_t, q - q_t) \cdot (x', y') = 0. \tag{10}$$

This constraint is shown in the gradient space by the line labelled  $\perp(x', y')$  in Fig. 11.

The second constraint is that the 3-D gradient  $(p_r, q_r, 1)$  of the ruling  $r$  must be orthogonal to the 3-D gradient,  $(p, q, 1)$ , of the surface along ruling  $r$ , that is,

$$(p, q, 1) \cdot (p_r, q_r, 1) = 0. \tag{11}$$

In Fig. 11 this constraint is shown by the line labelled  $\perp(p_r, q_r)$ , which is the orthogonal line of the gradient  $(p_r, q_r)$ , i.e., the gradient of the set of the directions that are orthogonal to  $(p_r, q_r)$  in 3-D. Note that this line is also orthogonal to the 2-D direction of the image of the ruling.

From (10) and (11) the gradient  $(p, q)$  of the surface along the ruling  $r$  is given by

$$\begin{aligned} p &= \frac{x'p_tq_r + y'q_tq_r + y'}{q_r x' - p_r y'}, \\ q &= \frac{1 + p_r p}{q_r}. \end{aligned} \tag{12}$$

The gradient  $(p_r, q_r)$  of the ruling  $r$  is obtained by reconstructing the axis line  $(a_x, a_y)$ , and the line between points  $(x, y)$  and  $(x_p, y_p)$  in 3-D (i.e., computing the  $z$ -coordinates of these image points). Since the gradient of the axis line is  $(ua_x, ua_y)$ , fixing  $z_p = 0$ , the  $z$ -coordinate of the point  $(x_{tip}, y_{tip})$ ,  $z_{tip}$ , is given by

$$z_{\text{tip}} = \frac{x_{\text{tip}} - x_p}{ua_x}. \quad (13)$$

The  $z$ -coordinate of the point  $(x, y)$  is computed, using the gradient  $(p_t, q_t)$ , as

$$z = p_t(x - x_p) + q_t(y - y_p). \quad (14)$$

Then the gradient  $(p_r, q_r)$  of the ruling is given by

$$(p_r, q_r) = \frac{1}{z_{\text{tip}} - z} (x_{\text{tip}} - x, y_{\text{tip}} - y). \quad (15)$$

These formulas are exactly the same for ZGCs cut by parallel planes. In that case the result is independent of the parallel symmetry curve used.

Compared to the case of a ZGC surface cut by parallel planes, for a ZGC surface cut by non parallel planes there are two additional unknowns, which are the gradient parameters of the second cutting plane. Also, there are two additional constraints on the orientation of the planes cutting the ZGC surface. These constraints are not needed to compute the local surface normals of a ZGC surface. In fact the gradient of the second plane (i.e., the plane that is not chosen to be made orthogonal to the surface) is not needed at all for that purpose. However, since for the multiple-surface recovery algorithm the gradient of the second plane is needed so are these constraints. Here we state the constraints, they will be used in Section 5.1.

The first one is a shared boundary constraint; for a ZGC surface  $S$ , with line-convergent symmetry, let  $(p_t, q_t)$  be the gradient of the top plane, let  $(p_b, q_b)$  be the gradient of the bottom plane, and let the intersection line have direction  $(l_x, l_y)$  on the image plane. Since the top and the bottom planes actually intersect each other along the line  $l$  in 3-D we have the shared boundary constraint:

$$SBC(p_t, q_t, p_b, q_b, l_x, l_y) = 0. \quad (16)$$

Consider Fig. 12; let  $(p_r, q_r)$  be the local surface gradient of the surface  $S$  along ruling  $r$ . This constraint enforces that the 3-D lines  $t_1$  and  $t_2$  be on the same tangent plane having gradient  $(p_r, q_r)$ . The constraint is:

$$t_1 \times r \equiv t_2 \times r; \quad (17)$$

or in long form:

$$\begin{aligned} & ((x_1 - x, y_1 - y, -(p_b(x_1 - x) + q_b(y_1 - y))) \\ & \times (x_2 - x_1, y_2 - y_1, -(p_r(x_2 - x_1) + q_r(y_2 - y_1))) \\ & \equiv ((x_2 - x, y_2 - y, -(p_t(x_2 - x) + q_t(y_2 - y))) \\ & \times (x_2 - x_1, y_2 - y_1, -(p_r(x_2 - x_1) + q_r(y_2 - y_1))), \end{aligned} \quad (18)$$

where  $\equiv$  denotes the parallelity of 3-D curves.



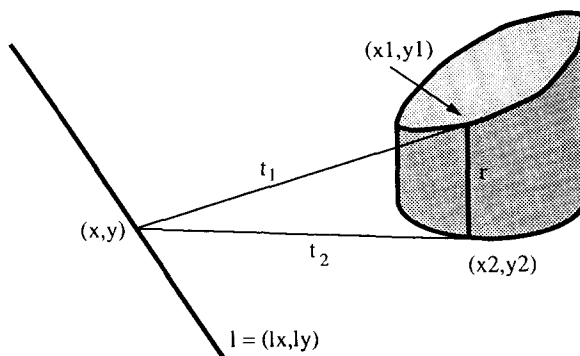


Fig. 12. Constraints on the orientation of the cutting planes of a ZGC surface.

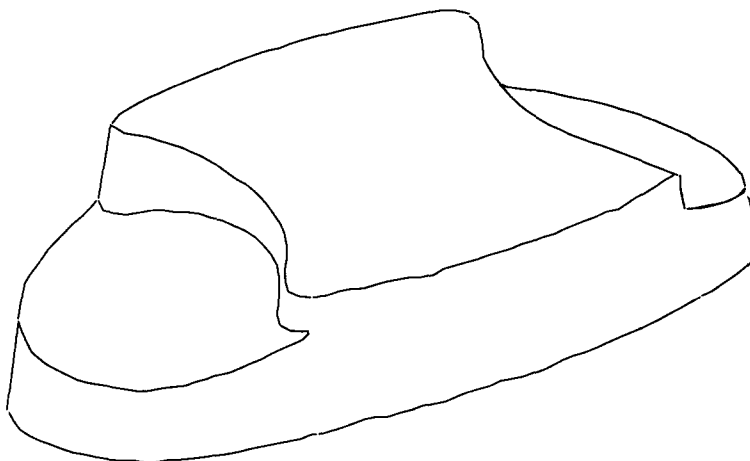


Fig. 13. The telephone example.

## 5. Combined shape recovery

Many objects of interest consist of several curved surfaces. Here the recovered 3-D individual surfaces must be in agreement with the neighboring surfaces, i.e., surfaces sharing a common boundary. We describe a technique for such an integrated multiple surface recovery for objects consisting of planar and ZGC surfaces. Figs. 1 and 13 show some sample objects. In the previous section we discussed parameterization of the surfaces and computation of the local surface normals from the parameters of the surface. In this section we discuss a method that sets the parameters of every surface that would be consistent with the neighboring surfaces. The problem of finding a consistent shape for all the surfaces is formulated as a constraint optimization problem, where the shared boundary constraints between surfaces are satisfied exactly and minimization is performed on assumption-driven constraints of orthogonality of surfaces.

The shape of all the surfaces is recovered simultaneously by finding appropriate values for the parameters of each surface. The values of the surface

parameters are computed by solving the following constraint minimization problem:

$$\text{minimize } E_i \quad \text{subject to } E_x = 0, \quad (19)$$

where  $E_i$  stands for error terms resulting from internal constraints of each surface and the  $E_x$  are the external terms, that is, the constraints obtained by intersection of surfaces.

### 5.1. Internal constraints

The internal constraints are the constraints obtained from the regularity assumptions of each surface. In general they have the following form:

$$E_i = \sum w_p E_p + \sum w_o E_o + \sum w_c E_c, \quad (20)$$

where each  $w$  is a weight and  $E_p$  is the error term for the orthogonality constraint of the planes,  $E_o$  is the error term for the orthogonality constraint of the ZGC surfaces and  $E_c$  is the error term for the implicit constraint of the parameters of ZGC surfaces. These error terms are described in more detail in the following.

$E_p$  is the error term for the orthogonality constraint of planes. If a planar surface has a skew symmetry, then this is the orthogonality function of the lines and axis of skew symmetry as given in (7), where  $(x_1, y_1)$  and  $(x_2, y_2)$  are the image directions of the lines of symmetry and the axis of symmetry and  $(p, q)$  is the gradient of the plane.  $w_p$  is the weight of  $E_p$  and is proportional to the total length of the contour enclosing the surface. The formula used for  $w_p$  is  $w_p = \sqrt{l_c}$  where  $l_c$  is the total length of the curve enclosing the surface. If the surface does not have skew symmetry  $E_p$  is zero.

$E_o$  is the error term for the orthogonality of ZGC surfaces. As in our previous work on ZGC surfaces [14], we choose to make the directions of the rulings orthogonal to the tangents of the parallel or chosen line-convergent symmetry curve in 3-D:

$$E_o = \frac{1}{N} \sum_i O(p_i, q_i, x'_i, y'_i, x_{\text{tip}} - x_i, y_{\text{tip}} - y_i). \quad (21)$$

Here  $O(\cdot)$  is the orthogonality constraint given in (7) for  $i \in [0, N-1]$ , at the  $i$ th location;  $(p_i, q_i)$  is the local surface normal represented in gradient space,  $(x'_i, y'_i)$  is the tangent of the line-convergent symmetry curve that is chosen to be orthogonal to rulings,  $(x_i, y_i)$  is the location of the point where the ruling meets with the line-convergent symmetry curve, and  $(x_{\text{tip}}, y_{\text{tip}})$  is the location of the apex of the cone. These are denoted as  $(p, q)$ ,  $(x', y')$ ,  $(x, y)$ ,  $(x_{\text{tip}}, y_{\text{tip}})$  respectively in Fig. 11. However computing  $E_o$  requires the computation of the local normal at each point on the surface at each iteration of the minimization, which is very time-consuming. Also, since it is a huge nonlinear expression, it creates many local minima around the true minima, where the minimization

routine gets stuck. Therefore from our experiments we use an approximation of  $E_o$ ,  $\hat{E}_o$ , such that  $\hat{E}_o$  is quadratic and the error given by  $\hat{E}_o$  is very similar to the one given by  $E_o$ .  $\hat{E}_o$  has the following form:

$$\hat{E}_o = E_{ax} + E_t. \quad (22)$$

Here  $E_{ax} = \cos^2(\alpha)$ , where  $\alpha$  is the angle between the gradient  $(p_t, q_t)$  of the plane containing the parallel (or line-convergent) symmetry curve that is decided to be made orthogonal and the direction of the image axis  $(a_x, a_y)$  of Fig. 11. Motivation for  $E_{ax}$  is based on the observation that the minimum of  $E_o$  in general occurs when  $(p_t, q_t)$  of Fig. 11 is along a line in  $p$ - $q$  space that passes through the origin and is parallel to the image axis.  $E_t = (u_{int} - u)^2$ , where  $u$  is the  $u$ -parameter of the ZGC surface and  $u_{int}$  is set at the initialization by minimizing the orthogonality error  $E_o$  given in (21). In our implementation we tried both  $E_o$  and  $\hat{E}_o$ . It turns out that  $\hat{E}_o$  performs better (in the sense of stability) because it is a simpler error function and creates fewer local minima. One can use  $\hat{E}_o$  as an initializer for  $E_o$ , but in our experiments another run of minimization with  $E_o$  over  $\hat{E}_o$  was not necessary.  $w_o$  is the weight of the orthogonality term and is proportional to the total length of the perimeter of the surface,  $w_o = \sqrt{l_c}$ , where  $l_c$  is the total length of the contour enclosing the surface.

$E_c$  is the error term for implicit constraints of the parameters of ZGC surfaces. Let  $(p_t, q_t)$  and  $(p_b, q_b)$  be the gradients of the planes containing the two parallel (or line-convergent) symmetry curves of the ZGC surface. If the ZGC surface has a parallel symmetry, then  $(p_t, q_t)$  should be equal to  $(p_b, q_b)$ , therefore,  $E_c = Eq(p_t, q_t, p_b, q_b)$ , where  $Eq()$  is given in (9). If the ZGC surface has an line-convergent symmetry, then  $E_c$  is the addition of the constraints given in (16) and (18).  $w_c$  is the weight and is inversely proportional to the eccentricity of the parallel (or line-convergent) symmetry curves. If the parallel (or line-convergent) symmetry curves are highly eccentric, i.e., they are almost straight, then the weight of this constraint is low. The formula for  $w_c = 1/ecc$ , where  $ecc$  is the eccentricity of the total cross-section curve. (The eccentricity of a curve is given by  $\sqrt{e_1/e_2}$  where  $e_1$  and  $e_2$  are the first and second eigenvalues of the covariance matrix of the curve given in (29).)

## 5.2. External constraints

External constraints are the inter-surface restrictions imposed by each surface on neighboring surfaces. Extremal constraints have the following form:

$$E_x = \sum w_{pp} E_{pp} + \sum w_{pz} E_{pz} + \sum w_{zz} E_{zz}, \quad (23)$$

where  $w_{ij}$  is the weight of each constraint and is equal to  $\sqrt{l_c}$  where  $l_c$  is the length of the curve produced by intersection of the surfaces.  $E_{pp}$ ,  $E_{pz}$  and  $E_{zz}$  are the error terms for shared boundary constraints between planes,

between planes and ZGC surfaces, and between ZGC surfaces respectively. In the following, the individual error terms are given in detail.

$E_{pp}$  is the error of the shared boundary constraint between the gradients of the two intersecting planes as given in (4).

$E_{pz}$  is the error term for the shared boundary constraint between a plane and a ZGC surface. There are two possibilities: the intersection is along a ruling of the ZGC surface or the intersection is along a parallel (or line-convergent) symmetry of the ZGC surface. If the intersection is along the ruling of the ZGC surface, then  $E_{pz}$  is the shared boundary constraint as given in (4) between the gradient of the plane and the local surface normal of the ZGC surface at the ruling of intersection. If the intersection is along one of the parallel (or line-convergent) symmetry curves, then

$$E_{pz} = Eq(p, q, p_t, q_t) \quad (24)$$

where  $(p_t, q_t)$  is the parameter of the ZGC surface, that is, the gradient of the plane containing the intersection curve, and  $(p, q)$  is the gradient of the planar surface.

$E_{zz}$  is the error term for the shared boundary constraint between two ZGC surfaces. There are various ways two ZGC surfaces may intersect each other. Here we only handle the intersections that produce a planar intersection curve. There are two types of such intersections: along the rulings of the ZGC surfaces or along the parallel (or line-convergent) symmetry of the ZGC surfaces. If the intersection is along the rulings of the ZGC surfaces, then the shared boundary constraint given in (4) is applied between the local surface normals of the ZGC surfaces at the ruling of the intersection. If the intersection is along the parallel (or line-convergent) symmetry curves, then let  $(p_1, q_1)$  and  $(p_2, q_2)$  be the gradients of the planes containing the intersection curve in the representations of the first and the second intersecting ZGC surfaces. The error term is:

$$E_{zz} = Eq(p_1, q_1, p_2, q_2). \quad (25)$$

When two ZGC surfaces intersect each other along their parallel symmetry (or line-convergent symmetry) curves, how orthogonal both surfaces can be made depends on how parallel their image axes are. Therefore we form a new orthogonality error term  $E_{on}$  for the intersecting ZGC surfaces to replace their original orthogonality error terms (the  $E_o$ ). Let  $\alpha$  be the angle between the image axes of these surfaces, let  $E_{o1}$  and  $E_{o2}$  be the error terms for the orthogonality of the intersecting ZGC surfaces. Then the new combined orthogonality error term is

$$E_{on} = \cos^2(\alpha)(E_{o1} + E_{o2}) + \sin^2(\alpha)(E_{o1}E_{o2}). \quad (26)$$

$E_{on}$  emphasizes the orthogonality of both of the ZGC surfaces when the image axes are almost parallel to each other, and it emphasizes the orthogonality of either of the ZGC surfaces when the image axes are almost orthogonal to each other.

### 5.3. Solving constraint equations

The total error function  $E$  is solved using a constraint minimization technique, where  $E_x$  consists of "must-satisfy" external constraints and  $E_i$  consists of assumption-driven error terms as defined earlier. To solve this constraint minimization (where the constraints are nonlinear), the problem is converted into a minimization form as follows:

$$\lim_{\lambda \rightarrow \text{inf}} \min E = \lim_{\lambda \rightarrow \text{inf}} \min (E_i + \lambda E_x), \quad (27)$$

that is,  $E$  is minimized for successively larger values of  $\lambda$ , thus, emphasizing  $E_x$  more at each minimization cycle. At the end,  $E_x$  constraints are satisfied almost exactly and  $E_i$  constraints are minimized to the extent possible. In our implementation we increased  $\lambda$  from 1 to 100 in exponential steps of 3.5 (that is  $\lambda = 1, 3.5, 12.25, \text{etc.}$ ). For the minimization, a gradient descent algorithm is used. The set of parameters of the surfaces ( $(p, q)$ 's and  $u$ 's) minimizing  $E$  is taken as the solution set and used to reconstruct the local surface gradients.

Initial values of the parameters of  $E$ , i.e., the parameters of all the surfaces involved in  $E$ , are computed by an initializer. The initializer starts with an arbitrary ZGC surface, and sets its parameters as if it is an isolated surface. Then the initializer sets the parameters of the neighboring surfaces by keeping them consistent with the first surface, and the neighbors of these surfaces are processed progressively until the parameters of all the surfaces are initialized.

## 6. Implementation and results

For the results shown in this section the following implementation is used. The input to the program are segmented curves—represented as a list of points—that define the contour of each object. However we do not assume that the input curves are noise-free. These segmented curves are grouped into closed regions using continuity. Each closed region is taken to correspond to an object surface. Next, we find symmetries among segments of a surface. Every segment bounding a surface is checked for symmetry (parallel, line-convergent or skew) against every other segment in the surface. Two segments are considered to be symmetric, if they return a *low* symmetry error defined for each of these symmetries. For parallel symmetry the symmetry error is given by

$$\frac{1}{l_2} \int_0^{l_2} \sqrt{(x_2'(s) - x_1'(\rho s))^2 + (y_2'(s) - y_1'(\rho s))^2} ds, \quad (28)$$

where the segments  $C_1(s) = (x_1(s), y_1(s))$  and  $C_2(s) = (x_2(s), y_2(s))$  are parameterized in terms of their arc length  $s$ ,  $\rho = l_1/l_2$  is a scaling parameter where  $l_1$  and  $l_2$  are the lengths of the segments  $C_1$  and  $C_2$ .

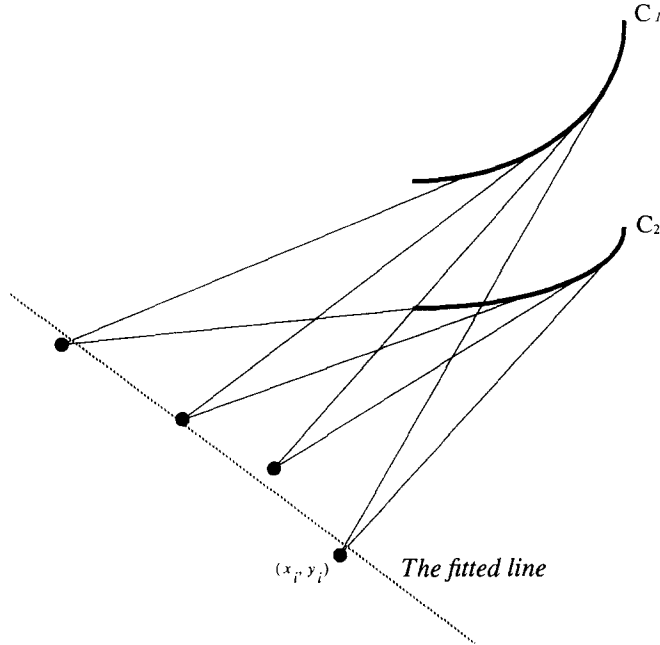


Fig. 14. Computation of line-convergent symmetry.

Line-convergent symmetry is computed in a similar way. First the apex point of the conic surface is computed using the end points of the two curves that are checked for line-convergent symmetry. Then the correspondences of the points on the curves are given by extending the lines from the apex that intersect both of the curves. Let  $\{(x_i, y_i)\}$  be the set of the intersections of the tangents of the corresponding points on the two curves  $C_1$  and  $C_2$ , as shown in Fig. 14. The goodness of line-convergent symmetry is given by how well a line fits to the points  $\{(x_i, y_i)\}$ . The goodness of line fit is computed by computing the scattering matrix of the points as

$$S = \begin{bmatrix} \frac{1}{n} \sum (x_i - \bar{x})^2 & \frac{1}{n} \sum (x_i - \bar{x})(y_i - \bar{y}) \\ \frac{1}{n} \sum (x_i - \bar{x})(y_i - \bar{y}) & \frac{1}{n} \sum (y_i - \bar{y})^2 \end{bmatrix}, \quad (29)$$

where  $(\bar{x}, \bar{y})$  is the mean of the points  $\{(x_i, y_i)\}$ . The scattering of these contour points is given by the eigenvalues  $e_1$  and  $e_2$  of the covariance matrix  $S$  ( $e_1$  being the larger one), and the goodness of fit is given by  $\sqrt{e_1/e_2}$ . Ponce [11] provides a Hough transform-based approach for computing this symmetry, which is much more expensive in terms of complexity.

The above symmetry computations are effective only if the entire lengths of two segments are symmetric to each other. Also, this measure is limited to parallel or line-convergent symmetry found in cylindrical and conic surfaces only.

Segments are also checked for having the same curvature sign at the corresponding points. This measure is especially useful when the segments are almost straight, in which case the error measure given in the above equations may be low even if the segments are not parallel or line-convergent symmetric.

We compute skew symmetry by performing a one-dimensional search in the direction of the lines of symmetry. For two curves  $C_1(s)$  and  $C_2(s)$  the algorithm is as follows.

*Step 1.* Use the direction of the line joining  $C_1(0)$  and  $C_2(0)$  as the initial direction  $r$ .

*Step 2.* Compute the correspondences of curves  $C_1$  and  $C_2$  such that the lines of symmetry  $\{l_i\}$  are parallel to direction  $r$ .

*Step 3.* Compute the mid-points,  $\{(x_i, y_i)\}$ , of the lines  $\{l_i\}$ .

*Step 4.* Fit a line to the points  $\{(x_i, y_i)\}$  by using the method described in the computation of the line-convergent symmetry.

*Step 5.* Stop if the error of the line fit is at a minimum with respect to direction  $r$ . Otherwise move direction  $r$  in the minimum error direction and go to Step 2.

If the final line fit error is low, then curves  $C_1$  and  $C_2$  are accepted as skew symmetric with  $r$  being the direction of the lines of symmetry and the line fitted in Step 4 as the axis of symmetry.

The surfaces containing a parallel or line-convergent symmetric segment pair are treated as curved and others are treated as planar. For curved surfaces the curves joining parallel symmetric curves are checked if they are straight to confirm that the surface is a ZGC.

Some surfaces are a combination of various curved surfaces and there is no distinctive boundary between them. This is the case for the curved surfaces of the object in Fig. 13. Such surfaces contain more than one parallel (or line-convergent) symmetry and they are segmented into smaller surfaces containing only one parallel (or line-convergent) symmetry. Fig. 15 shows the segmented surfaces and the symmetries (skew, parallel or line-convergent) for each surface.

The constraints for each surface and the inter-surface constraints including the ones for the newly formed intersections are extracted forming the error function  $E$ . Then  $E$  is minimized by the constraint minimization technique discussed in Section 5.3. Figs. 16 and 17 show the output of the steps of the minimization process for increasing  $\lambda$  for the objects in Figs. 1(a) and 13 respectively. For both figures,  $\lambda$ -values are 0, 1, 3.5, 12, 42 from left to right, top to bottom.  $\lambda = 0$  corresponds to the initial state. The computed surface normals are shown by needle diagrams, as needles sticking to the surface in the direction of the local surface normals. For planar surfaces, a small coordinate frame, with a triangle at the base, is used to better show the computed surface normal. Note that, for the object in Fig. 17 the result of the very first step of the minimization ( $\lambda = 1$ ) is very similar to the final result. Whereas for the object in Fig. 16 most of the parameter determination happens at later stages of the minimization (i.e., for larger  $\lambda$ -values). This is because, for the object

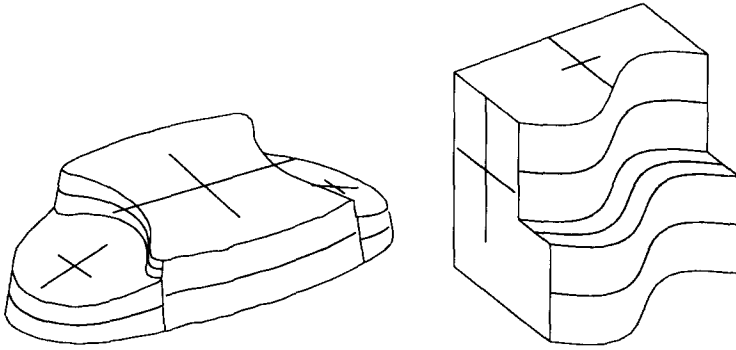


Fig. 15. The segmented surfaces, and the symmetries computed for each surface. The skew symmetry of planar surfaces is shown by crosses, the long line is the axis of symmetry and the short one is the direction of the lines of symmetry. Parallel and line-convergent symmetries are shown by their curved axes only.

in Fig. 16, the middle surface is initially classified as a curved surface due to the parallel symmetry it has. However, it's perceived as a planar surface affected by the planarity of the top surface as discussed in the Introduction. Therefore, strong inter-surface constraints are needed to force the planarity

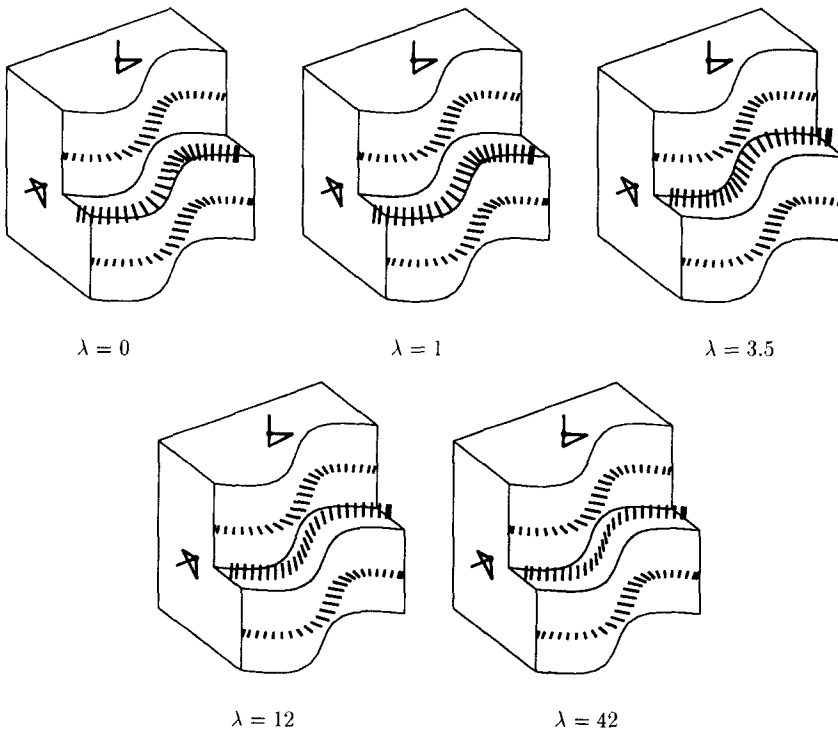


Fig. 16. Surface normals at the end of each step of the minimization, as  $\lambda$  increases, for the object in Fig. 1(a).



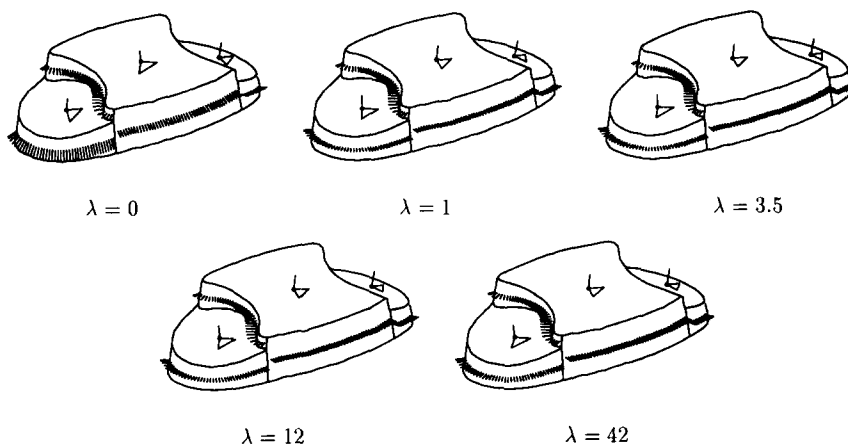


Fig. 17. Surface normals at the end of each step of the minimization, as  $\lambda$  increases, for the object in Fig. 13.

of the middle surface, and increasing  $\lambda$  emphasizes inter-surface constraints. The final result of the minimization, in fact, shows the middle surface as planar within error bounds of the minimization. For the object in Fig. 17 no surface imposes a drastic shape change, from the initial classification, on other surfaces. Therefore, for this figure, even weaker inter-surface constraints provide a reasonably consistent shape.

Figs. 18 and 19 show the final results for the objects in Figs. 1(a) and 13 respectively. This process takes approximately 2 minutes for an object on a

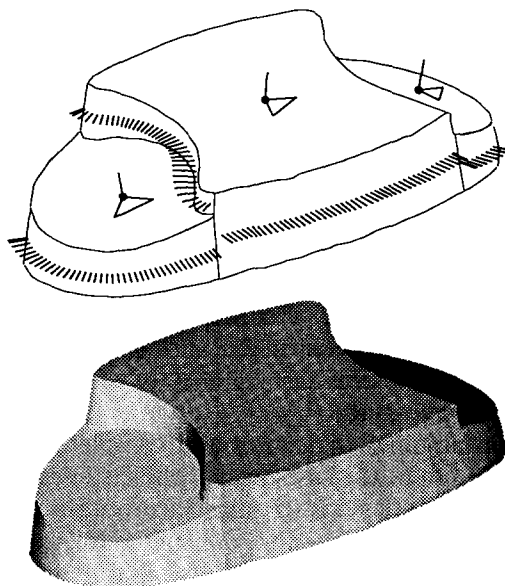


Fig. 18. The needle and the shaded images obtained from the computed surface normals for the object in Fig. 13.

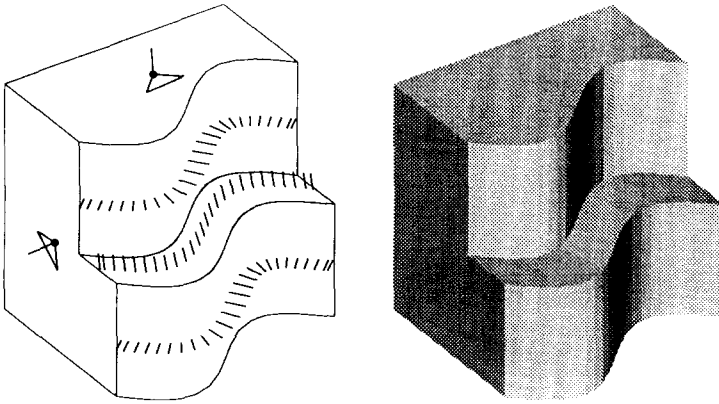


Fig. 19. The needle and the shaded images obtained from the computed surface normals for the object in Fig. 1(a).

Symbolics 3645 computer running in LISP. Besides the needle diagrams, we also provide the shaded images of the objects computed by using the surface normals, a Lambertian reflectance model and a point light source. We believe that the computational results are in agreement with human perception, though we have not attempted a quantitative comparison.

## 7. Conclusion

We have described a technique for recovering 3-D shape of objects consisting of zero-Gaussian curvature (and planar) surfaces. Our method incorporates the constraints imposed by all the surfaces simultaneously. As shown by an example, this can result in fixing the shapes of some surfaces which are ambiguous otherwise.

We have attempted to show the accuracy of our results by comparison with human perception. For shape from contour analysis, the only *ground truth* is really in human perception, for even if the given contour was obtained by a real object, it *could* have been also obtained by any number of other objects as well. Our system has a certain notion of preference that is based partly on geometrical analysis and partly on perceptual observations (i.e., assumptions). The agreement of the final reconstructed surfaces in the examples with the usual human perception confirms the viability of the perceptual observations, like orthogonality, we have made.

The current system assumes that clean and complete boundaries are given. This cannot be expected to be the case for real objects in complex environments. To cope with these difficulties will require incorporation of some perceptual organization techniques. We believe that our methodology will help in this step too, as we are able to provide strong constraints that hypotheses of a perceptual organization system can be tested against. This is the topic of separate, current research in our laboratory.

A possible research direction is incorporating non-ZGC surfaces. We have worked on recovery of other types of surfaces such as straight homogeneous generalized cylinders [15] and constant cross-section generalized cylinders [17]. The main issues are to identify the possible ways such surfaces may come together, and to form the necessary constraints between them.

Our current method assumes orthographic projection. Perspective projection introduces various additional problems. The biggest difficulty is that additional ambiguities are introduced. Parallel symmetry, under perspective, turns into a line-convergent symmetry. Line-convergent symmetry remains line-convergent. Hence, we can no longer distinguish among the two cases without higher level reasoning. The constraint equations themselves can be generalized for perspective projection as shown in [16]. In fact once the symmetries are identified, the perspective analysis provides tighter constraints, thus allowing us to relax other assumptions.

## References

- [1] H. Barrow and J. Tenenbaum, Interpreting line drawings as three dimensional surfaces, *Artif. Intell.* **17** (1981) 75–116.
- [2] M. Brady and A. Yuille, An extremum principle for shape from contour, *IEEE Trans. Pattern Anal. Mach. Intell.* **6** (1984) 288–301.
- [3] M.B. Clowes, On seeing things, *Artif. Intell.* **2** (1) (1971) 79–116.
- [4] M.P. Do Carmo, *Differential Geometry of Curves and Surfaces* (Prentice Hall, Englewood Cliffs, NJ, 1976).
- [5] R. Horaud and M. Brady, On the geometric interpretation of image contours, *Artif. Intell.* **37** (1988) 333–353.
- [6] D.A. Huffman, Impossible objects as nonsense sentences, in: B. Meltzer and D. Michie, eds., *Machine Intelligence 6* (Edinburgh University Press, Edinburgh, Scotland, 1971) 295–323.
- [7] T. Kanade, Recovery of the three-dimensional shape of an object from a single view, *Artif. Intell.* **17** (1981) 409–460.
- [8] T. Kanade and J.R. Kender, Mapping image properties into shape constraints: skew symmetry, affine transformable patterns, and the shape from texture paradigm, in: J. Beck, B. Hope and A. Rosenfeld, eds., *Human and Machine Vision 237–257* (Academic Press, NY, 1983).
- [9] A.K. Mackworth, Interpreting pictures of polyhedral scenes, *Artif. Intell.* **4** (1973) 121–137.
- [10] V. Nalwa, Line drawing interpretation: bilateral symmetry, in: *Proceedings Image Understanding Workshop*, Los Angeles, CA (1987) 956–967.
- [11] J. Ponce, D. Chelberg and W.B. Mann, Invariant properties of straight homogeneous generalized cylinders and their contours, *IEEE Trans. Pattern Anal. Mach. Intell.* **11** (9) (1989) 951–966.
- [12] S.A. Shafer, T. Kanade and J. Kender, Gradient space under orthography and perspective, *Comput. Vision, Graph. Image Process.* **24** (1983) 182–199.
- [13] K.A. Stevens, The visual interpretations of surface contours, *Artif. Intell.* **17** (1981) 47–73.
- [14] F. Ulupinar and R. Nevatia, Inferring shape from contour for curved surfaces, in: *Proceedings 10th International Conference on Pattern Recognition*, Atlantic City, NJ (1990) 147–154.
- [15] F. Ulupinar and R. Nevatia, Shape from contour: SHGCs, in: *Proceedings Third ICCV*, Japan (1990).
- [16] F. Ulupinar and R. Nevatia, Constraint for interpretation of line drawings under perspective projection, *Comput. Vision, Graph. Image Process.* **52** (1991) 674–676.

- [17] F. Ulupinar and R. Nevatia, Recovering shape from contour for constant cross section generalized cylinders, in: *Proceedings Computer Vision and Pattern Recognition Conference, Maui, HI* (1991).
- [18] I. Weiss, 3-D shape representation by contours, *Comput. Vision, Graph. Image Process.* **41** (1988) 80–100.
- [19] G. Xu and S. Tsuji, Inferring surfaces from boundaries, in: *Proceedings First ICCV, London* (1987) 716–720.