

# Shape from Contour: Straight Homogeneous Generalized Cylinders and Constant Cross Section Generalized Cylinders

Fatih Ulupinar and Ramakant Nevatia, Member, IEEE

**Abstract** — We analyze the properties of Straight Homogeneous Generalized Cylinders (SHGCs) and Constant Cross Section Generalized Cylinders (CGCs), and derive the types of symmetries that the limb boundaries and cross sections of these objects produce on the image plane. The constraints on the 3D shape of the objects are formulated based on the symmetries and from the geometry of the projection models. Finally, the methods that recover the 3D shape from the image of their contours are discussed and recovered surfaces are shown for sample objects.

**Index Items** — Shape from contour, straight homogeneous generalized cones, constant cross section generalized cones, symmetry analysis, shape constraints.

## I. INTRODUCTION

**I**NFERRING 3D shape of the objects in a scene from a single image remains an important and difficult problem in computer vision. The difficulty arises, of course, from the fact that an image is a 2D projection of the scene and the process is not invertible without making some assumptions. A number of approaches for inferring 3D shape have been suggested, such as shape from shading, shape from texture, and shape from contour. We believe that of all the monocular cues, shape of the 2D contour itself is the most important one for the shape of the 3D surfaces.

Strictly speaking, not only is the inference of 3D shape from contours infinitely ambiguous, but the contours can only give shape information near the contours; shape of the surface in between can vary smoothly without producing other contours. Nonetheless, humans, when presented with contours of various, not necessarily familiar, objects, perceive complete surfaces (and even solids). Thus the challenge is in finding a set of constraints that is small, justified by arguments about the nature of the image formation, and whose results are in conformity with human judgments. (Note that, in a real sense, the results of shape from contour cannot be judged independent of human perception, as no other “ground truth” exists – even when we know that the same contours that produced one

shape could have been produced by another shape.)

In this paper we present techniques for the recovery of two classes of objects:

- 1) Straight Homogeneous Generalized Cylinders (SHGCs): These are generalized cylinders with a straight axis and cross sections of a fixed shape but of varying size. Some examples are shown in Fig. 1. Note that this class includes solids of revolutions.
- 2) Planar, Right Constant cross section Generalized Cylinders (PRCGCs): This class represents “snake” or “pipe” shaped objects. Here the cross section of the generalized cylinders is fixed in shape and size, and the axis is planar but not necessarily straight. Further, we require the cross sections to be orthogonal to the tangent of the axis (thus giving the adjective “right” in the above name). Some examples are shown in Fig. 2.

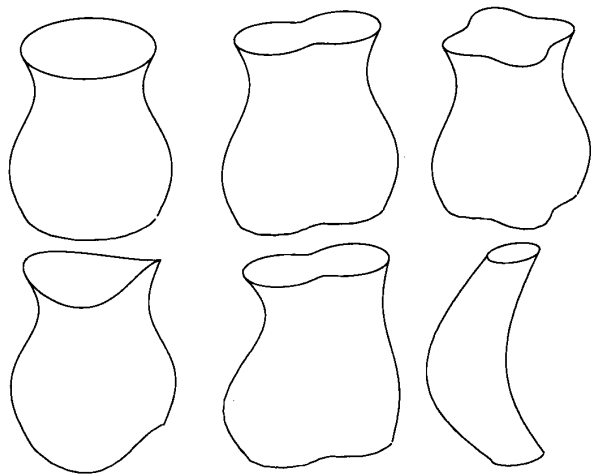


Fig. 1. Sample Straight Homogeneous Generalized Cylinders.

The above two categories follow the terminology suggested by Shafer [20] to deal with special cases of generalized cylinders first proposed by Binford [1]. Our method does not require *a priori* knowledge that the objects in the scene belong to these classes; rather, we provide a test for the presence of such objects, and if they are present we can infer their 3D shape.

Manuscript received November 7, 1991; revised May 13, 1994. Recommended for acceptance by Associate Editor Dr. T. Boult.

F. Ulupinar is with Advanced Computing Systems Company, 3000 S. Robertson Blvd., Los Angeles, CA 90034; e-mail ulupinar@acsc.com.

R. Nevatia is with Institute for Robotics and Intelligent Systems, School of Engineering, University of Southern California, Los Angeles, CA 90089-0273; e-mail nevatia@iris.usc.edu.

IEEECS Log Number P95018.

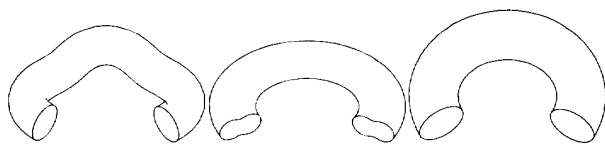


Fig. 2. Sample Planar, Right Constant cross section Generalized Cylinders (PRCGCs).

The method we describe is based on, and is a major generalization of, the technique we have described earlier for inferring shape of zero-Gaussian curvature (or ZGC) surfaces [26]. We believe that the addition of SHGCs and PRCGCs to this methodology makes it possible to infer the shape of a large fraction of the objects in our environment (particularly for indoor scenes).

Our approach is based on an analysis of the symmetries in a scene. In Section III we define the symmetries we use. Then we show how such symmetries arise naturally in images of the class of objects we study. In Section VI we summarize the constraints that derive from these symmetries and other properties of boundaries for determining the 3D shape. In Sections VII and VIII we show how these constraints and other properties of the boundary allow us to infer 3D shape of the objects in the scene and give some computational results. An earlier version of the SHGC work is published in [24] and of the CGC work is published in [25].

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 some of the previous work in our group on this topic is relevant [14]. In addition, we believe that the specific conditions needed for an object surface to be reconstructed by our method will provide further constraints for the perceptual grouping operations when surface markings and other noisy boundaries are present.

## II. PREVIOUS WORK

We provide a brief summary of previous work here; a much more detailed and critical analysis of previous work is given in our ZGC paper [26].

Early work, in the 1970s and early 1980s, on inferring 3D structure from a 2D shape was focused on analysis of line drawings of polyhedra [4], [8], [9], [11], [13]. In the 1980s, several techniques for non-polyhedral shapes were proposed (e.g., [2], [3], [22], [27], [28]), though some of these apply only to planar curves ([3], [27]). One deficiency 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.

Some of the more recent approaches are given in ([7], [15], [17]). Nalwa's method [15] is limited to analysis of solids of revolution (a subset of SHGCs); also, it does not provide a method for complete reconstruction of such objects but only gives some properties. Ponce, et al. [17], give some properties

of SHGCs which we use in this paper, but, again, the described techniques are not sufficient to recover the surface. Horaud and Brady [7] proposed a method for recovering 3D shape of the surface of revolutions from a single image of its contours. However, they fail to identify one of the free parameters of the surface (namely, the point where the axis penetrates the cross section). Therefore, their method is not complete for surface of revolutions either.

Gross and Boulton [6] provide an analysis of right SHGCs where they show that a right SHGC has two degrees of freedom if it is to be recovered from an image of its contours. They provide a method for fixing these degrees of freedom by using intensity in addition to contours. In this paper, we describe an approach to fixing these degrees of freedom by using some perceptual properties. We believe that our approach requires fewer assumptions. Use of shading information assumes uniform and known surface reflectance properties.

There is little previous work on PRCGCs. Ponce and Chelberg [16] have derived equations of the limbs of RCGC's (not necessarily planar axis) given a 3D model of the RCGC's and provide solutions to the equations for circular cross section only. Their work does not address the problem of locating such objects in images which requires study of the properties of the projection of the limbs of such objects nor do they discuss the recovery of these objects from images.

## III. DEFINITIONS AND PROPERTIES OF SURFACES

In the subsequent analysis, we will assume that the image is obtained by an orthographic projection (though some of our theorems apply to perspective projection as well) and from a general viewpoint, defined as:

**Definition 1 – General Viewpoint:** *A scene is said to be imaged from a general viewpoint, if perceptual properties of the image are preserved under slight variations of the viewing geometry.*

Specifically, the properties we are interested in are: straightness and parallelism of lines, and symmetry of curves (symmetry as defined in the following section).

In this paper we 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 3D.

### A. Symmetry Definitions

The three symmetries we define are *parallel symmetry*, *line convergent symmetry*, and *skew symmetry*. For two curves to be symmetric (parallel, line convergent or skew) certain point-wise correspondences between the two curves must exist. We call the lines joining the corresponding points on the curves the *lines of symmetry*, the locus of the mid points of

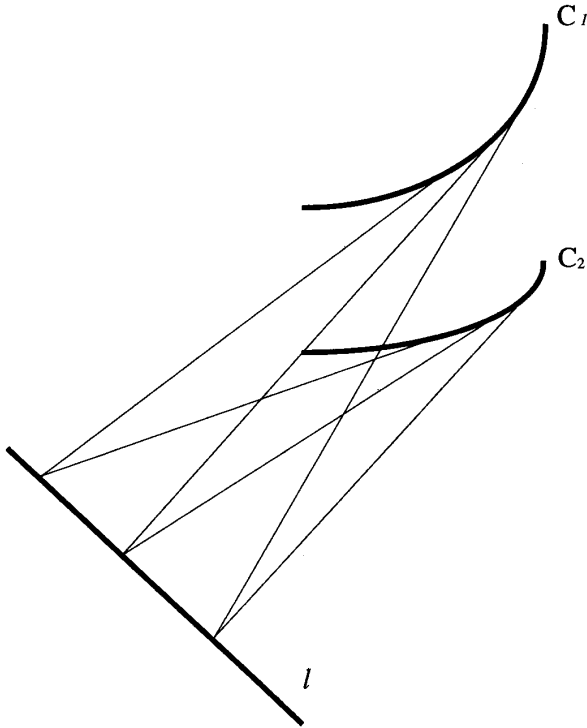


Fig. 3. Two line convergent symmetric curves.

these lines is the *axis of symmetry*, and the curves forming the symmetry are the *curves of symmetry*.

**Parallel Symmetry:** Let  $X_i(s) = (x_i(s), y_i(s), z_i(s))$ , for  $i = 1, 2$ , be two curves in 3D with a parameter of arc length  $s$ . The curves  $X_1(s)$  and  $X_2(s)$  are said to be parallel symmetric if there exists a point-wise correspondence function,  $f(s)$ , between them such that the tangent vectors to the curves, say  $T_1(s)$  and  $T_2(f(s))$ , are equal for all values of  $s$  for which  $X_1$  and  $X_2$  are defined, and  $f(s)$  is a monotonically increasing function. Note that projection of curves  $X_1$  and  $X_2$  under orthographic projection produces image curves that are parallel symmetric such that the 3D point correspondence is preserved. Computing symmetry between two curves using this definition requires estimating the function  $f(s)$  as well. Note that the correspondence function is not unique for straight lines; here, we use the unique linear correspondence function which makes the ends of the line segments correspond.

**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. An example is shown in Fig. 3. Ponce et al. [17] provide an algorithm for computing this symmetry using the Hough transform. In this manuscript we use a different algorithm based on the cross section of the SHGCs, which is computationally much less expensive.

**Skew Symmetry:** For skew 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 angle (not necessarily orthogonal) to the axis of symmetry.

We believe that the symmetries we have defined, either separately or taken together, give some qualitative as well as quantitative information about the surface shape. In [26] we have shown that if a closed contour composed of non-limb edges has a skew symmetry, the symmetry correspondences are static with respect to small changes in the viewpoint, and, if the viewpoint is general, then that contour must be planar. We also 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). In the following we show the properties that allow us to infer the presence of SHGCs and PRCGCs.

First we discuss some useful geometric properties of differentiable surfaces.

### B. Surfaces and Their Limb Edges

Let tangent line,  $L_v$ , of a surface,  $S$ , at point,  $P$ , in a given direction,  $V$ , be the line from the point  $P$  in the direction of the tangent of the curve,  $C$ , obtained by cutting the surface by a plane,  $\Pi$ , that passes through  $P$ , and contains the normal,  $N$ , of the surface at  $P$  and the direction given by the vector  $V$ . Fig. 4 shows an example.

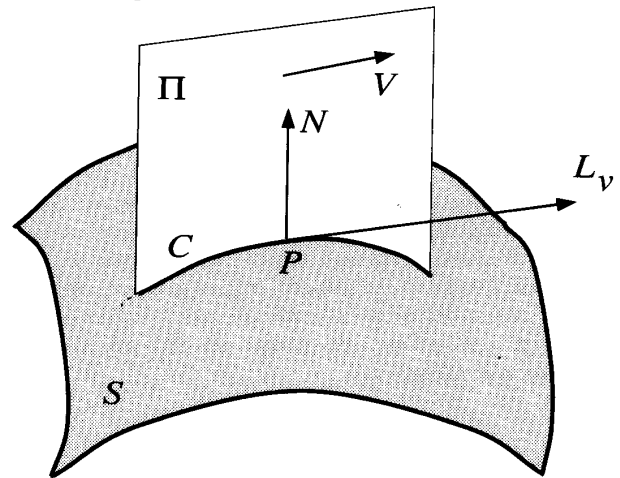


Fig. 4. Tangent line,  $L_v$ , of a surface  $S$  at point  $P$  in direction  $V$ .

It is a well known property in differential geometry [5] that the tangent lines,  $L_{v_i}$ , of a surface,  $S$ , at point,  $P$ , in all possible directions,  $V_i \in R^3$ , are on a plane,  $T_p$ , called the tangent plane of the surface at  $P$ . Moreover the plane  $T_p$  is orthogonal to the normal,  $N$ , of the surface at  $P$ . This property is shown graphically in Fig. 5.

Next, we define limb edges and their projections for smooth surfaces.

**Definition 2:** The limb edge of a surface is a viewpoint dependent curve on the surface such that at each point on the curve the surface normal is orthogonal to the viewing direction.

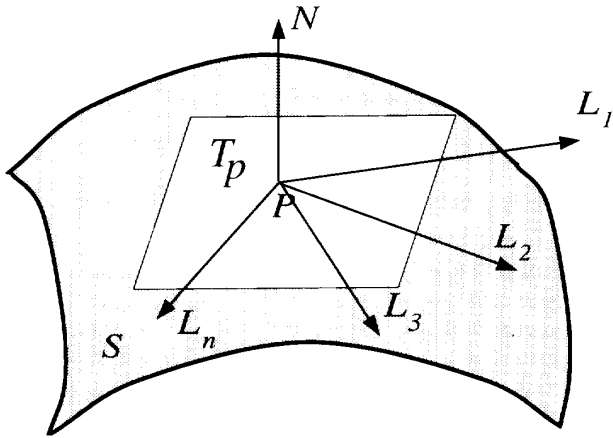


Fig. 5. Tangent plane,  $T_p$ , of a surface,  $S$ , containing all the tangent lines at point  $P$ .

The limb edges project on the image plane as the bounding curve of the surface. At these edges the surface smoothly curves around to occlude itself. Limb edges (also called “occluding contours”) can give some very important information about the 3D surface they come from; Koenderink [12] has provided a very nice analysis of limb edges. We will show how the limb edges help us recover 3D surface shape later in this paper.

**Lemma 1:** *Let  $P$  be a point on a limb edge of some  $C^1$  surface  $S$ . Then, all the tangent lines of  $S$  at point  $P$  project as the same line on the image plane under both orthographic and perspective projection.*

**Proof:** The proof involves a simple combination of the definition of limb edges and the property of tangent planes. Since the normal of the tangent plane at  $P$  (which is also the normal of the surface at  $P$ ) is orthogonal to the viewing direction, the tangent plane projects as a line on the image plane.

Therefore all the tangent lines at  $P$ , which are included in the tangent plane also project to the one line that the plane projects into.

This lemma, though simple and rather obvious, turns out to be highly useful in proving other important properties of limb boundaries.

#### IV. PROPERTIES OF SHGCs

Straight homogeneous generalized cylinders (SHGCs) are obtained by sliding a cross section, say  $C$ , along a straight axis, say  $A$ . The cross section is also scaled as it is swept along the axis by a scaling function, say  $r$ . We can parameterize the surface,  $S$ , of an SHGC (including oblique ones) as follows. Let  $C(u) = (x(u), y(u), 0)$  be the planar cross section,  $r(t)$  be the scaling function, and  $\theta$  be the tilt of the cross section with respect to the axis. Without loss of generality we can align the

axis of the SHGC along the  $z$  axis of the coordinate system and the tilt of the cross section as a rotation around the  $x$  coordinate axis. Then the equation of the surface of the SHGC is:

$$S(u, t) = (r(t)x(u), r(t)y(u)\cos(\theta), r(t)y(u)\sin(\theta) + t) \quad (1)$$

An example is shown in Fig. 6. Note that the cross section curves are generated by fixing  $t$  and varying  $u$ . We will call the curves generated by fixing  $u$  and varying  $t$  the meridians of the surface. Note that the cross sections of an SHGC are planar, by definition, and hence the meridians of the SHGC are also planar since the SHGC has no twist in its sweep. Let meridian edges of an SHGC be edges that are along the meridians of the SHGC. Images of SHGCs, with a non linear scaling function, contain meridian edges only if the cross section has a tangent discontinuity (a corner). Fig. 1 shows some sample SHGCs. The bottom left one has a meridian edge on its right side.

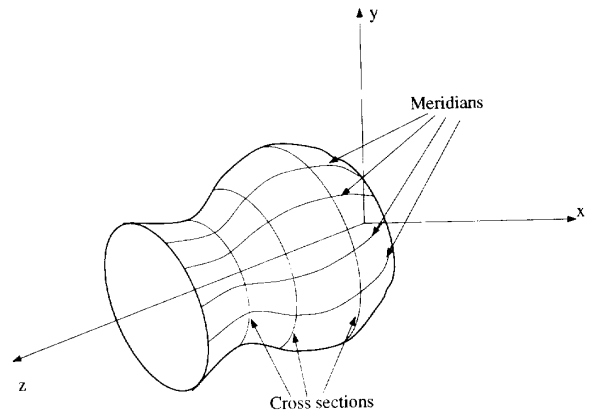


Fig. 6. An SHGC along the  $z$  coordinate axis with both meridians and cross sections marked.

**Theorem 1:** For an SHGC, the tangent lines of the surface in the direction of the axis from the points of any given cross section intersect at a common point on the axis of the SHGC.

A proof of this theorem may be found in [19]. Fig. 7(a) graphically illustrates the property.

**Corollary:** The tangents of all meridian edges at the points where they intersect a single cross section intersect the axis of the SHGC at a single point. Therefore, in the image plane, too, the tangents of the images of the meridian edges, at the point where they intersect a single cross section, intersect the image of the axis in a single point under orthographic or perspective projection.

It has been shown by Shafer [18] that the limb edges on an SHGC are not planar. Therefore the limb edges of an SHGC are necessarily not along its meridians, and the tangents of the

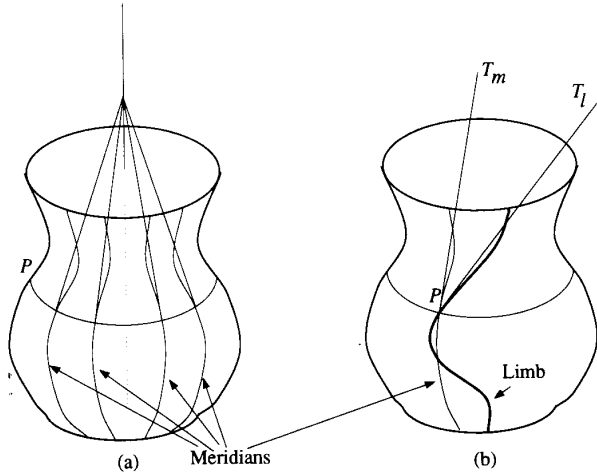


Fig. 7. (a) An SHGC, and its tangent lines, in the direction of the axis emitting from a single cross section, intersecting at a single point on the axis. (b) The tangent lines,  $T_l$ , of limb edges are not the same as the tangent lines,  $T_m$ , of the meridians in 3D.

limb boundaries at the point where they intersect the same cross section do not intersect the axis in 3D. (Fig. 7(b) shows the limb edge and its tangent for an SHGC, after rotating it, to show that in 3D the tangent of the limb edge does not intersect the axis of the SHGC.) Still, it has been shown by Ponce et al, [17] that under orthographic projection the tangents of the projections of the limb edges, at the point where they intersect the same cross section, intersect the image of the axis at a single point. Here we give a simpler proof which is independent of the projection geometry.

**Theorem 2:** *The tangents of the projections of the limb edges at the points they intersect the same cross section, when extended, intersect the image of the axis of the SHGC at the same point, i.e., the projections of the limb edges of an SHGC form a line convergent symmetry.*

**Proof:** Say the limb edge intersects a given cross section at point  $P$  (see Fig. 7). The tangent line  $T_m$  from point  $P$  in the direction of the axis of the SHGC (the tangent line of the meridian passing through the point  $P$ ) intersects the axis of the SHGC, by Theorem 1. Hence, the image of the tangent line  $T_l$  from point  $P$  in the direction of the tangent of the limb edge projects onto the same line as the tangent line  $T_m$ . Thus the image of the line  $T_l$  intersects the image of the axis at the same point where the image of the line  $T_m$  intersects.

Since Lemma 1 holds under both perspective and orthographic projection, the above theorem and the proof hold for both of the projection geometries.

In the following we show that the cross sections of an SHGC are parallel symmetric in 3D with the meridian curves joining the parallel symmetric points of the cross sections.

**Theorem 3:** *The cross sections of an SHGC are parallel symmetric in 3D with each other such that the meridian curves join the parallel symmetric points of the cross sections.*

**Proof:** We have to show that the direction of the tangent of

the cross sections is independent of the  $t$  parameter curve. Using the parameters for an SHGC given in (1), the tangent of the cross sections ( $u$  parameter curves) is given by:

$$\begin{aligned} S_u &= (r(t)x'(u), r(t)y'(u)\cos(\theta), r(t)y'(u)\sin(\theta)) \\ &= r(t)(x'(u), y'(u)\cos(\theta), y'(u)\sin(\theta)) \end{aligned} \quad (2)$$

Clearly the direction of  $S_u$  is independent of the  $t$  parameter.

**Corollary:** *The projections of the cross section curves of an SHGC are also parallel symmetric in the image plane under orthographic projection. The correspondence function is linear because cross sections are obtained by scaling a reference cross section curve without deforming it. Note that the visible parts of two cross sections may not be the same and hence the observed curves may only be partially symmetric.*

#### A. Recovering the Cross Sections

We next show how to find the projections of cross sections in the image of an SHGC, given the images of its external contours. Our method does not require complete cross sections, but only the part that lies on the visible face of the SHGC. However, we require that the SHGC be cut along its cross sections; otherwise we would not have a parallel symmetry between the image curves of the two extreme cross sections ( $C_t$  and  $C_b$  in Fig. 8). We conjecture that humans, too, do not do well if this condition is not satisfied.

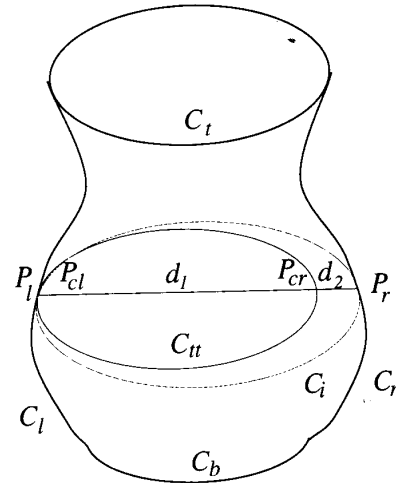


Fig. 8. Image of an SHGC cut along its cross sections. Image of the top cross section curve is  $C_t$ , the bottom one is  $C_b$ , and the limb boundaries are on the left  $C_l$ , and on the right  $C_r$ .

The following algorithm recovers the image curves  $C_i$  that correspond to the projections of the cross sections of the SHGC.

For each point  $P_i \in C_i$  do:

- 1) Find the point  $P_{cl} \in C_l$  such that  $C_l'(P_l) \equiv C_r'(P_{cl})$ .<sup>1</sup>
- 2) Translate the cross section curve  $C_l$  such that the point  $P_{cl} \in C_l$  coincides with the point  $P_r$ , obtaining the curve  $C_{tr}$ .
- 3) Scale the curve  $C_{tr}$  until it touches the right limb curve,  $C_r$ , obtaining the cross section curve  $C_i$ .<sup>2</sup>

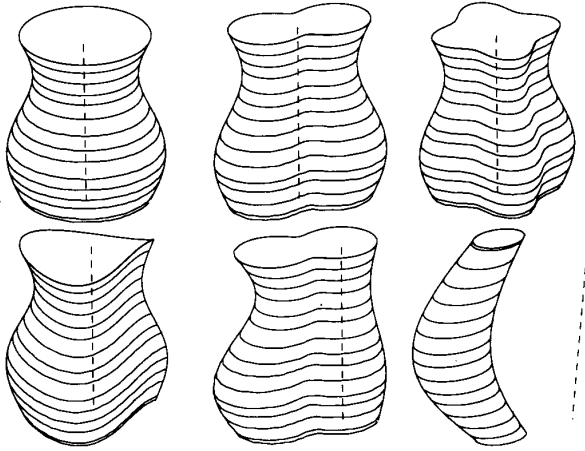


Fig. 9. Images of the cross sections and axes, the dashed lines, recovered for the SHGCs in Fig. 1.

The curve  $C_i$  obtained by this algorithm is precisely the image of the cross section curve between the points  $P_l$  and  $P_r$  of the SHGC. Once the correspondence of the points  $P_l$  and  $P_r$  between the limb edges  $C_l$  and  $C_r$  is obtained, we can directly recover the image of the axis of the SHGC by using Theorem 2. Fig. 9 shows the computed images of the cross section curves and the axes for SHGCs in Fig. 1. If the parallel, symmetric points of the cross section curves are joined, by Theorem 3 we obtain the meridian curves.

### B. Observing SHGCs

If there are two parallel symmetric curves with a linear correspondence function and bound by line convergent symmetric curves that have the correspondence computed by the above algorithm, we hypothesize that the line drawing results from an SHGC. Note that these are only necessary conditions for hypothesizing an SHGC, but not sufficient to guarantee that the viewed object must be an SHGC. However, we think it is highly unlikely that these strong properties would be satisfied by chance.

## V. PROPERTIES OF CGCS

Snakes are generalized cylinders that have a constant cross

<sup>1</sup> The  $\equiv$  operator is used for parallelity of vectors, that is, if  $V_1 \equiv V_2$  then  $V_1 = \lambda V_2$  for a scalar  $\lambda$ .  $C'$  is the tangent direction of the curve  $C$ . If more than one point with equal tangents is found, choose the one that is closest to the point chosen on the previous cross section.

<sup>2</sup> We use a simple procedure for this step: We find the point  $P_{cr} \in C_{tr}$  that minimizes the function  $f(P_{cr}) = (d_1 + d_2) / d_1$  which is the amount of scaling required to be applied on the curve  $C_{tr}$  to bring the point  $P_{cr}$  to the point  $P_r$ . The quantities  $d_1$  and  $d_2$  are the signed distances from  $P_l$  to  $P_{cr}$  and from  $P_{cr}$  to  $P_r$ . The value of the function  $f(\cdot)$  at its minimum gives the correct scaling factor for  $C_{tr}$ .

section, but the axis may be an arbitrary 3D curve. Following Shafer et al's, terminology [20], such objects may be called CGCs. We will focus on CGCs that have planar axis and that are "right", i.e., the cross sections are orthogonal to the axis; we call such objects PRCGCs. We further assume that the objects are not self-intersecting. Fig. 2 shows some examples. PRCGCs provide an important modeling tool for mostly "snake like" objects that cannot be modeled by SHGCs; however, their mathematical properties received very little attention in the past. Some researchers have studied some subclasses of PRCGCs: Saint Marc and Medioni [21] have studied a torus, which can be modeled as a PRCGC with circular cross section and circular axis. They used the property that limb edges of a torus produce parallel symmetry under orthographic projection without proving this property. They also estimated the pose of a torus from a single image of it, using the axis of the parallel symmetry of its limb edges. Rao [10] has studied PRCGCs imaged from a direction orthogonal to the plane of the axis of the PRCGC. He has shown that, from this particular viewpoint, the 3D limb edges of PRCGCs are planar. He does not provide any analysis of PRCGCs from a general viewpoint. Ponce and Chelberg [16] have worked on RCGCs (not necessarily planar axis). They have derived the equation of the limbs for such objects and provided solution of these equations for circular cross section only. They did not discuss recovery of such objects from images. In this section we provide a shape recovery from contour method for PRCGCs having arbitrary cross section and arbitrary (planar) axis.

In the following we show that limb boundaries of a PRCGC project as parallel symmetric curves under orthographic projection.

Let us choose a coordinate system such that the axis of the PRCGC lies in the  $x$ - $z$  plane and one of the cross sections, say  $C(u) = (c_x(u), c_y(u), 0)$ , is aligned with the  $x$ - $y$  plane. Let  $A(t) = (a_x(t), 0, a_z(t))$  be the axis parameterized in terms of its arc length, that is,  $|A'| = a_x'^2 + a_z'^2 = 1$  for all  $t$ . Also, let  $A(0) = (0, 0, 0)$  and since the cross section is orthogonal to the axis  $A'(0) = (0, 0, 1)$ . Then the surface of the PRCGC,  $S(u, t)$  is given by:

$$S(u, t) = R(A'(0), A'(t)) \cdot C(u) + A(t) \quad (3)$$

where  $R(V_1, V_2)$  is the rotation matrix that transforms the direction vector  $V_1$  into vector  $V_2$ . For  $A'(0) = (0, 0, 1)$  and  $A'(t) = (a_x'(t), 0, a_z'(t))$  the rotation matrix  $R$  becomes:

$$R = \begin{bmatrix} a_z'(t) & 0 & a_x'(t) \\ 0 & 1 & 0 \\ -a_x'(t) & 0 & a_z'(t) \end{bmatrix} \quad (4)$$

Note that the curves generated by fixing  $t$  and varying  $u$  are the *cross sections* of the surface  $S(u, t)$ . We will call the curves generated by fixing  $u$  and varying  $t$  the *meridians* of the surface. The meridians are also the loci of points on the cross section as the cross section is swept along the axis. Fig. 10 shows an example.

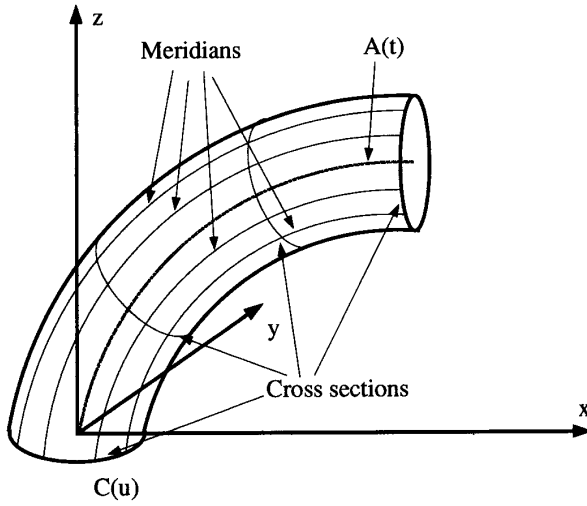


Fig. 10. A PRCGC with both meridians and cross sections marked.

**Lemma 2:** *The meridians of a PRCGC are parallel symmetric and the curves joining the parallel symmetric points of the meridians form the cross sections of the surface.*

**Proof:** We need to show that the direction of the tangents of the surface in the direction of the meridians,  $\frac{\partial S}{\partial t}$ , is independent of the parameter  $u$ .

$$\begin{aligned} \frac{\partial S(u,t)}{\partial t} &= \frac{dR}{dt} \cdot C(u) + A'(t) \\ &= \begin{bmatrix} a_x''(t) & 0 & a_z''(t) \\ 0 & 0 & 0 \\ -a_z''(t) & 0 & a_x''(t) \end{bmatrix} \begin{bmatrix} c_x(u) \\ c_y(u) \\ 0 \end{bmatrix} + \begin{bmatrix} a_x'(t) \\ 0 \\ a_z'(t) \end{bmatrix} \quad (5) \\ &= \begin{bmatrix} a_x'(t) \\ 0 \\ a_z'(t) \end{bmatrix} + c_x(u) \begin{bmatrix} a_z''(t) \\ 0 \\ -a_x''(t) \end{bmatrix} \\ &= A'(t) + c_x(u) (A''(t))^\perp \quad (6) \end{aligned}$$

where  $(A''(t))^\perp$  is a vector which is orthogonal to the vector  $A''(t)$  and is in the  $x-z$  plane. Also note that,  $A''(t) \cdot A'(t) = 0$  since

$$0 = d(1) = d(A'(t) \cdot A'(t)) = 2A'(t) \cdot A''(t) \quad (7)$$

We conclude that the vector  $(A''(t))^\perp$  is parallel to the vector  $A'(t)$ , since  $A'(t) \perp A''(t)$ ,  $A''(t) \perp (A''(t))^\perp$  and all three vectors are on a plane (the  $x-z$  plane). Then, we can rewrite  $\frac{\partial S}{\partial t}$  as:

$$\frac{\partial S(u,t)}{\partial t} = \left( 1 + c_x(u) \frac{|(A''(t))^\perp|}{|A'(t)|} \right) A'(t) \quad (8)$$

It is obvious that while the length of the vector  $\frac{\partial S}{\partial t}$  depends on the  $u$  parameter, the direction of it is independent of the  $u$  parameter.

Although the meridian curves on a PRCGC are parallel symmetric, it can be shown that the limb edges of a PRCGC are not necessarily parallel symmetric in 3D (see Fig. 11). However, the following theorem proves that the projections of the limb edges of a PRCGC are parallel symmetric under orthographic projection.

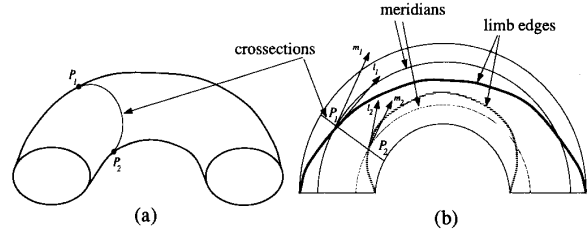


Fig. 11. A PRCGC (half a torus) (a) from a general view, and (b) semi-transparent top view with the limb edges of the previous view and the meridians passing from the points  $P_1$  and  $P_2$  marked along with their tangent lines.

**Theorem 4:** *The limb edges of a PRCGC project as parallel symmetric curves onto the image plane, and that the corresponding points are on the same cross section.*

**Proof:** Here we use the property given in Lemma 1 and in Lemma 2. Consider the points  $P_1$  and  $P_2$  in Fig. 11 such that both points are on the same cross section. As can be seen in Fig. 11(b), the tangent lines  $l_1$  and  $l_2$  from points  $P_1$  and  $P_2$  in the direction of the limb edges are not parallel symmetric in 3D. However, the tangent lines  $m_1$  and  $m_2$  from points  $P_1$  and  $P_2$  in the direction of the meridians are parallel symmetric by Lemma 2. Since the tangent line  $l_1$  projects the same line as the tangent line  $m_1$ , and tangent line  $l_2$  projects the same line as the tangent line  $m_2$ , by Lemma 1 the projection of the limb boundaries of a PRCGC are parallel symmetric. Note that in the case of straight limb edges, the PRCGC reduces to a simple cylinder. Hence, the simple unity correspondence function provides correct cross section correspondence.

#### A. Observing PRCGCs

If in the image plane there are parallel symmetric curves that are terminated by two curves (possibly closed and having skew symmetry which enhances planarity of the cross section), then we hypothesize that it is a PRCGC. The real test for the line drawing to belong to a PRCGC may be performed after the cross sections are recovered as described in Section VIII.A.

## VI. CONSTRAINTS FOR DETERMINING SURFACE SHAPE

We now give three constraints that derive from observations of the symmetries and object boundaries in the image.

### A. Curved Shared Boundary Constraint (CSBC)

This constraint relates the orientations of the two surfaces on opposite sides of an edge. It is a generalization of the constraint used in polyhedral scene analysis from the early days [13] and has been stated previously in [20], [26].

Let two surfaces,  $S_1$  and  $S_2$ , intersect along a curve,  $\Gamma$ , whose projection is the curve  $\Gamma_i(s)=(\Gamma_x(s), \Gamma_y(s))$ . Let the orientations of the surfaces  $S_1$  and  $S_2$  along the curve  $\Gamma(s)$ , in gradient space, be given by  $(p_1(s), q_1(s))$  and  $(p_2(s), q_2(s))$ . Then CSBC states that:

$$\begin{aligned} (\Gamma'_x(s), \Gamma'_y(s), z'(s)) \cdot ((p_2(s), q_2(s), 1) - (p_1(s), q_1(s), 1)) &= 0 \\ \Gamma'_x(s)(p_2(s) - p_1(s)) + \Gamma'_y(s)(q_2(s) - q_1(s)) &= 0 \end{aligned} \quad (9)$$

A stronger constraint can be obtained if we can assume that the 3D intersection curve,  $\Gamma$ , is planar. Let  $\Gamma$  lie in a plane with orientation  $(p_c, q_c)$ . With the assumption of planarity the constraint equation becomes:

$$\Gamma'_x(s)(p_c - p_i(s)) + \Gamma'_y(s)(q_c - q_i(s)) = 0, \quad i=1,2 \quad (10)$$

### B. Inner Surface Constraint (ISC)

The inner surface constraint restricts the relative orientations of the neighboring points, within a surface. Consider a regular parameterization  $C(t)=(x(t), y(t), z(t))$  of a curve on a  $C^2$  surface  $S$ . For each point  $P \in C$  associate a unit vector  $R \in T_P$ , where  $T_P$  is the tangent plane of the surface  $S$  at the point  $P$ , such that

$$C' \cdot dN_R = 0 \quad (11)$$

where  $C' = \frac{dC}{dt}$  and  $dN_R$  is the derivative of the normal  $N$  of the surface  $S$  in the direction  $R$ . Note that, except for the special case of planes, for any given surface  $S$  and the curve  $C$  at each point the associated directions  $R$  exists and is unique (up to a mirror reflection).<sup>3</sup> For a plane, since  $dN_R$  is equal to zero for any  $R$ ,  $R$  is any vector  $\in T_P$ .

**Theorem 5: Inner Surface Constraint:** *If an image curve  $C_I$  is an orthographic projection of the curve  $C$  on the surface  $S$  and  $R_I=(r_x, r_y)$  is the projection of the vector  $R$  satisfying Equation 11 and  $R$  is not parallel to the tangent of  $C$ , then the change of the orientation,  $(p, q)$ , of the surface  $S$ , along the curve  $C$ , in the  $p$ - $q$  space is restricted by the image vector  $R_I$ , as:*

$$d(p, q)_{C'} \cdot R_I = 0$$

**Proof:** Let  $X(u, v)$  be a local parameterization of the surface

<sup>3</sup> For a zero-Gaussian curvature surface, it turns out that the only solution to (11) is when  $R$  is along a ruling (which makes  $dN_R$  zero).

$S$  around the point  $P \in C(t)$  such that for  $P=X(u_0, v_0)$ , the curve  $X(u, v_0)$  is equal to the curve  $C(u)$  and  $\frac{\partial}{\partial v} X(u_0, v)$  is parallel to the direction  $R$  (i.e.,  $R = X_v(u_0, v_0)$ ). That is, at the point  $P$ ,  $u$  parameter curve is along the curve  $C$  and the tangent of the  $v$  parameter curve is in the direction  $R$ .<sup>4</sup> Here we have to show that  $\frac{d(p, q)}{du} \cdot R_I = 0$  where  $(p, q)$  is the normal of the surface in the gradient space,  $du$  is in the direction of  $C'$ ,  $R_I$  is the image,  $(x_v, y_v)$ , of the vector  $R$  under orthographic projection.

The normal,  $N$ , of this surface at any point is given by:

$$N = \frac{X_u \times X_v}{|X_u \times X_v|} \quad (13)$$

Then, the functions  $C'$  and  $dN_R$  are:

$$C' = \frac{\partial X}{\partial u} = X_u, \quad dN_R = \frac{\partial N}{\partial v} = N_v \quad (14)$$

By (11) we have  $X_u \cdot N_v = 0$ . On a surface of class  $C^k$ , where  $k \geq 2$ , we have  $X_v \cdot N_u = X_u \cdot N_v$ . Since,  $X_u$  and  $X_v$  are always orthogonal to  $N$ , by (13), we have:

$$0 = (X_v \cdot N)_u = X_{vu} \cdot N + X_v \cdot N_u \quad (15)$$

and

$$0 = (X_u \cdot N)_v = X_{uv} \cdot N + X_u \cdot N_v \quad (16)$$

Hence,  $X_v \cdot N_u = X_u \cdot N_v = 0$ , by above equations and (11) and (14).

Let the normal  $N$  of the surface around point  $P$  be represented in the  $(p, q)$  space as  $N=c(p, q, 1)$ . Where  $c$  is the scale coefficient and it is equal to  $(p^2 + q^2 + 1)^{-1/2}$ . Differentiation of  $N$  with respect to the parameter  $u$  gives:

$$N_u = c_u(p, q, 1) + c(p_u, q_u, 0) = \frac{c_u}{c} N + c(p_u, q_u, 0) \quad (17)$$

If we set  $X_v \cdot N_u = 0$  where  $X_v = (x_v, y_v, z_v)$  and  $N_u$  is given in (17), we get:

$$X_v \cdot N_u = \frac{c_u}{c} X_v \cdot N + c(x_v, y_v, z_v) \cdot (p_u, q_u, 0) = 0 \quad (18)$$

We also have  $N \cdot X_v = 0$  from (13). Therefore

$$x_v p_u + y_v q_u = \frac{d(p, q)}{du} \cdot R_I = 0 \quad (19)$$

<sup>4</sup> If  $R$  is parallel to  $C$  at  $P$  then the surface parameters cannot be this  $u - v$ ; the theorem statement excludes this case.

To apply this constraint, we need to identify a curve  $C$  in the image plane for which the orientation  $R$  can be determined. In a previous paper [26] we have shown that for zero Gaussian curvature surfaces any curve on the surface can be the curve  $C$  if the direction  $R$  is chosen to be the direction of the rulings of the surface. The following theorem shows how parallel symmetric curves can be used to identify the curve  $C$  and associated directions  $R$ . It should be noted that for each curve on a surface there is a different set of associated directions. If present, parallel symmetry provides one method of identifying them. However, it is not the only method.

**Theorem 6:** *Let a family of curves,  $\{C_i\}$ , be on a surface  $S$  such that the curves,  $C_i$ , are parallel symmetric in 3D. If the curves  $C_i$  are used as the  $C$  curves of (11), then the tangents of the curves obtained by joining the symmetric points of the curves  $C_i$  give the directions  $R$  of the ISC. Conversely, if the curves<sup>5</sup> obtained by joining the parallel symmetric points of curves  $C_i$  are used as  $C$  curves of (11) then the tangents of the curves  $C_i$  give the directions  $R$ .*

**Proof:** Consider the parametric representation  $S(u, v)$  of the surface  $S$  such that the  $u$  parameter curves are parallel symmetric to each other (the  $\{C_i\}$  family of curves) and  $v$  parameter curves join the parallel symmetric points of the  $u$  parameter curves.

For the first part of the theorem we have to show that (11) holds with the current parameters. That is,

$$S_u \cdot N_v = 0 \quad (20)$$

is true, where  $N = \frac{S_u \times S_v}{|S_u \times S_v|}$  is the unit normal of the surface.

Note that  $N \cdot S_u = N \cdot S_v = 0$  by definition. We can substitute  $-S_{uv} \cdot N$  for  $S_u \cdot N_v$  since:

$$0 = \frac{\partial(S_u \cdot N)}{\partial v} = S_{uv} \cdot N + S_u \cdot N_v \Rightarrow S_u \cdot N_v = -S_{uv} \cdot N \quad (21)$$

$S_u$  is the tangent of the  $u$  parameter curves, and, since the  $v$  parameter curves join the parallel symmetric points of  $u$  parameter curves the, direction of  $S_u(u, v)$  is independent of the  $v$  parameter. That is  $S_u(u, v) = c(v) S_u(u)$  where  $c$  is a scalar function. And:

$$S_{uv} = \frac{\partial S_u}{\partial v} = c'(v) S_u(u) \quad (22)$$

By substituting this in (21) we get

$$N_v \cdot S_u = -N \cdot S_{uv} = -c'(v)(N \cdot S_u(u)) = 0 \quad (23)$$

For the second part of the theorem we have to show that  $S_v \cdot N_u = 0$ . Using (23) we get:

<sup>5</sup> One has to make sure that these curves have regular parameters. One way to do so is by checking that the curves  $C_i$  are non intersecting with each other.

$$0 = N_v \cdot S_u = -N \cdot S_{uv} = -N \cdot S_{vu} = S_v \cdot N_u \quad (24)$$

### C. Orthogonality Constraint (OC)

The two previous constraints (CSBC and ISC) are not sufficient to determine surface orientations uniquely. To further constrain the solution, we impose an additional constraint. We require that the cross sections and the meridians of a surface (as defined in Sections IV and V) be mutually orthogonal. This constraint may be satisfied precisely for some kinds of surfaces but is not necessarily true for all surfaces; in the latter cases we maximize a measure of orthogonality (given later). This constraint is justified on perceptual observations. It may be viewed as being equivalent to slicing the surface along meridians and cross sections to obtain small skew symmetric planar patches and assuming that these patches are orthogonally symmetric in 3D, as in Kanade's analysis for polyhedra [9]. Note that when a surface containing parallel symmetric curves is cut along curves of parallel symmetry and along the curves joining the correspondence of the parallel symmetric curves, surface patches are obtained that are four sided and bounded by two parallel symmetric curves. As the patches get smaller they become more planar and the contour of the patch turns into straight lines, two of which are parallel to each other. Therefore the whole contour of the small patch becomes skew symmetric.

The orthogonality of two vectors  $A$  and  $B$ , which lie on a plane having gradient  $(p, q)$  and whose images are  $A_i = (a_x, a_y)$  and  $B_i = (b_x, b_y)$ , constrains the gradient  $(p, q)$  with the equation:

$$(a_x, a_y - (pa_x + qa_y)) \cdot (b_x, b_y - (pb_x + qb_y)) = 0 \quad (25)$$

## VII. QUANTITATIVE SHAPE RECOVERY OF SHGC SURFACES

In this section we discuss application of the constraints and orthogonality to recover the shape of the SHGCs from their contours.

### A. Application of the Constraints

To compute the shape of an SHGC along each recovered cross section curve we apply the constraints discussed in Section VI. For the following; say that there are  $m$  cross section curves and we would like to compute the orientation of the surface at  $n$  points along a cross section. Then we have  $2nm$  unknowns, initially, corresponding to the gradient  $(p, q)$  of the surface at  $nm$  points.

**CSBC.** The curved shared boundary constraint applies between the orientation,  $(p_c, q_c)$ , of the cross section curves  $C_j$  and the orientation,  $(p_i, q_i)$  of each of the points on the surface along a cross section. Note that  $(p_c, q_c)$  is the same for all cross section curves. The curved shared boundary constraint states that the line in the  $p - q$  space from the gradient  $(p_i, q_i)$  of a point  $P_i \in C_j$  to the gradient  $(p_c, q_c)$  of the cross section plane is orthogonal to the tangent,  $C_j'(P_i)$ , of the cross section  $C_j$  at

point  $P_i$ . Then the constraint equation is:

$$(p_c - p_i, q_c - q_i) \cdot C'_j(P_i) = 0 \quad \forall P_i \in C_j \quad (26)$$

This provides  $n$  constraints along each cross section curve.

**ISC.** Inner surface constraint is applied along a cross section using the tangents of the meridians at each point.<sup>6</sup> Theorem 6 indicates that ISC is applicable along the cross section curves because they are parallel symmetric by Theorem 3 with the meridian curves joining the corresponding points of the cross section curves. Computing this constraint at half points (i.e.,  $i + 1/2$ ) provides better accuracy as the difference of two consecutive points is an approximation for the derivative of the point in the middle. Inner surface constraint states that change of the orientation  $(p'_{i+1/2}, q'_{i+1/2})$  of the surface along a cross section curve  $C_j$  at point  $P_{i+1/2} \in C_j$  must be orthogonal to the tangent  $M'_{i+1/2}(P_{i+1/2})$  of the meridian that passes through the point  $P_{i+1/2}$  which is in the middle of the points  $P_i \in C_j$  and  $P_{i+1} \in C_j$ . Then the constraint equation is:

$$\begin{aligned} (p'_{i+1/2}, q'_{i+1/2}) \cdot M'_{i+1/2}(P_{i+1/2}) &= 0 \\ \forall P_i, P_{i+1/2}, P_{i+1} \in C_j \quad s.t. 1 < i < n \end{aligned} \quad (27)$$

In digital domain we approximate these half point gradient derivatives by finite differences:

$$(p'_{i+1/2}, q'_{i+1/2}) = (p_{i+1} - p_i, q_{i+1} - q_i) \quad (28)$$

The meridian half point derivative is approximated by averaging:

$$M'_{i+1/2}(P_{i+1/2}) = \frac{1}{2}(M'_i(P_i) + M'_{i+1}(P_{i+1})) \quad (29)$$

Application of ISC provides  $n - 1$  equations for each cross section curve.

There are  $2n$  unknowns for each cross section curve, and two more unknowns for the whole SHGC, the  $(p_c, q_c)$ . By combining the two constraints, we have  $2n-1$  constraints for each cross section. Then for each cross section there are three degrees of freedom as in the case of a ZGC surface discussed in [26].

**Planarity of Meridians.** The meridians of an SHGC are planar as discussed in Section IV. Then the shared boundary constraint can be applied along a meridian curve as if the curve is obtained by cutting the surface of the SHGC with a plane along the meridian. The shared boundary constraint is applied along a meridian,  $M$ , between the gradient,  $(p_m, q_m)$ , of the plane containing the meridian  $M$  and the gradient  $(p_j, q_j)$  of the points  $P_j \in M$ , using the tangent,  $M'(P_j)$  of the meridian curve at each point  $P_j \in M$ . The constraint equation is :

$$(p_m - p_j, q_m - q_j) \cdot M'(P_j) = 0 \quad \forall P_j \in M \quad (30)$$

Forcing one meridian curve to be planar forces the others to be planar, too. Therefore, the planarity is applied to only one of the meridians, giving  $m$  constraint equations at the expense of two additional unknowns. We can use any meridian curve for planarity enforcement except the ones that lie on the viewing planes, as they project into lines. For numerical stability, it is best to use the most *curved* meridian curve.

In total there are now  $2nm + 4$  unknowns,  $2nm$  for the  $(p, q)$  of  $nm$  points on the surface, two for  $(p_c, q_c)$ , two more for  $(p_m, q_m)$ , and there are  $2nm$  constraint equations,  $nm$  from the CSBC between the cross sections and the face of the surface,  $m(n - 1)$  from the ISC, and  $m$  from the CSBC of a meridian curve. That is, there are four degrees of freedom for recovering the orientation of all the points on an SHGC. Of course this degrees of freedom argument assumes that there is no interdependency between the constraint equations. In fact, for some special cases, the equations become interdependent, i.e., when the scaling function is linear (in this case it is called LSHGC, which is a ZGC surface). But in such cases the total degrees of freedom of the system also reduces. It is also possible that a particular choice of cross section curves gives an ill conditioned system of equations. We have found that sampling the limb curves sparsely avoids these problems. In an independent work, Gross and Boulton [6] have analyzed *right* SHGCs and determined that contour analysis leaves two degrees of freedom. Our analysis, valid for oblique SHGCs too, comes up with four degrees of freedom.

The four degrees of freedom can be fixed by arbitrarily choosing values for any of the four variables in the above set of equations. We have chosen to fix the four degrees of freedom by fixing the orientation,  $(p_c, q_c)$ , of the cross sections and the orientation,  $(p_m, q_m)$ , of the plane containing the chosen meridian as these parameters have some intuitive meaning. We choose these values by imposing some additional constraints described below which we believe give results that are consistent with human perception.

### B. Orthogonality

For SHGCs we use the orthogonality of the 3D tangents of the cross sections and the meridian curves, making each little patch, formed by dividing the surface along meridians and the cross sections, orthogonal. We can apply the orthogonality constraint using the equation given in (25). This constraint is not always exactly satisfied, except for surfaces of revolution. Therefore, we perform a minimization of the second orthogonality constraint, as:

$$\Xi = \sum_i \sum_j \cos(\theta_{ij}) = \frac{(C_i(P_{ij}))'_3 \cdot (M_j(P_{ij}))'_3}{\left| (C_i(P_{ij}))'_3 \right| \left| (M_j(P_{ij}))'_3 \right|} \quad (31)$$

<sup>6</sup> Note that, in reference to Theorem 5, the  $C$  curve is the cross section and the  $R$  is the tangent of the meridian. Hence, the condition of the theorem that  $R$  is not parallel  $C$  is equivalent to the condition that a meridian is not parallel to a cross section which is guaranteed by the definition of SHGCs.

where  $(C_i(P_{ij}))'_3$  and  $(M(P_{ij}))'_3$  are the 3D tangents of the cross section and meridian curves at point  $P_{ij}$ . These 3D tangents are dependent on their 2D tangents on the image and on the orientation  $(p_{ij}, q_{ij})$  of the surface at point  $P_{ij}$  as given by Equation 25. The gradients  $(p_{ij}, q_{ij})$  at each point are dependent on the four variables,  $(p_c, q_c)$  and  $(p_m, q_m)$ , discussed in the previous section. We would like to minimize the function  $\Xi$  for  $(p_c, q_c)$  and  $(p_m, q_m)$ . Unfortunately, from experimental observations, we find that the minimum is not sharply defined. However, the values obtained are always consistent with the assumption that the 3D axis of the SHGC is orthogonal to its cross section, so we enforce this is an additional explicit constraint.

The constraint that the cross sections are orthogonal to the axis of the SHGC implies that the cross section orientation  $(p_c, q_c)$  is along a line in the  $p - q$  space that passes through the origin and is in the direction of the image of the axis of the SHGC. This constraint also, in effect, forces the gradient  $(p_m, q_m)$  of the plane of the meridians to be orthogonal to the gradient  $(p_c, q_c)$  of the cross sections. That is :

$$(p_m, q_m, 1) \cdot (p_c, q_c, 1) = 0 \quad (32)$$

For simplicity, say the coordinate system is rotated such that the image of the axis of the SHGC is aligned with the  $y$  axis of the coordinate system. Then, we have;  $p_c = 0$  from the orthogonality of the axis to the cross section and  $q_m = -1/q_c$  from (32). The parameters  $p_m$  and  $q_c$  are the free variables to be fixed by minimizing the function  $\Xi$ . However, the minimum of the function  $\Xi$  does not fix the variable  $q_c$  (except for surfaces of revolution). Either the function forms a valley along  $q_c$ , making any choice as good as any other, or fixes  $q_c$  to be zero which is not a realistic solution. We use the same method for estimating  $q_c$  as we used for ZGC surfaces in [26]; it is described briefly below.

### C. Fixing $q_c$

As discussed in the previous section, application of constraints and orthogonality leaves one degree of freedom,  $q_c$ . To compute  $q_c$  we use the same algorithm we used for ZGC surfaces in [23]. Here a brief summary of this method is provided for reference.

- 1) Estimation of  $q_c$ : An ellipse is fit to the cross section contour, then the orientation of the circle  $(p_c, q_c)$ , that would project as the fitted ellipse is projected on the  $q$  axis, on the  $p-q$  plane to obtain the first approximation of  $q_c$ , call it  $q_e$ . Note that two possible orientations exist for a circle to project as the ellipse in the image plane. These two orientations of the circle can be thought of as being an instance of Necker reversal. Our Algorithm does not choose among them; instead, it provides both of the solutions. In the results section the interpretation that corresponds to a solid object, rather than a hole, is shown.

It may be necessary to segment the cross section if it is complex and repetitive. To achieve this, the concavities

of the contour are found and matched. If they match in such a way that the cross section is segmented into similar pieces, then a different ellipse is fit to each piece of the contour and the average of the ellipses is used to estimate  $q_e$ .

- 2) Updating  $q_c$ : The purpose of this updating process is to simulate the bias that humans have in orienting the cross section toward  $45^\circ$ . We update  $q_e$  to obtain the final  $q_c$  as follows (after converting  $q_e$  into degrees):

$$q_c = 45^\circ + \lambda(q_e - 45^\circ) \quad (33)$$

where  $\lambda$  is a confidence factor in the range  $[0, 1]$  and is a function of how well the ellipse approximates the cross section curve. In our implementation it is given by :

$$\lambda(\epsilon) = (1 - \epsilon^2) \quad (34)$$

where  $\epsilon$  is the ellipse fit error (in range  $[0, 1]$ ).

The algorithm derives from our observations of human perception which we have validated by an extensive comparison with human subjects.

### D. Results

To apply our method to real images we first need to find the boundaries of the objects and then the symmetries, if any, contained in them. In general we can expect object boundaries to be fragmented, and several intensity boundaries that correspond to surface markings, shadows, and noise, to be present. To separate the object boundaries from these other boundaries, and to fill in the gaps in object boundaries as appropriate, is a difficult problem in monocular image analysis and this paper is not about such analysis. The results shown here assume that labeled curves (cross section and limbs) are given to the algorithm as connected points with exact coordinates. Our experiments with integer valued positions show essentially the same results. A recent effort at USC by Zerroug and Nevatia [29] describes how the methods described in this paper can be applied to analysis of real images which contain digitization noise, surface markings, occlusion, and other effects normally found in real images.

The surface recovery algorithm is as follows:

- 1) Compute the inner cross sections and the limb boundary correspondences using the algorithm given in Section IV.A.
- 2) Given the limb boundary correspondences, compute the image of the axis using Theorem 2.
- 3) Align the axis of the SHGC with the  $y$  axis of the coordinate system (i.e.,  $p_c = 0$ ).
- 4) Compute  $q_c$  using the ellipse fit algorithm described in Section VII.C.
- 5) Compute  $p_m$  by minimizing the function  $\Xi$  (note that  $q_m = -1/q_c$ ).

- 6) Construct the local surface orientations by solving the linear set of equations generated from the constraints given in Section VII.

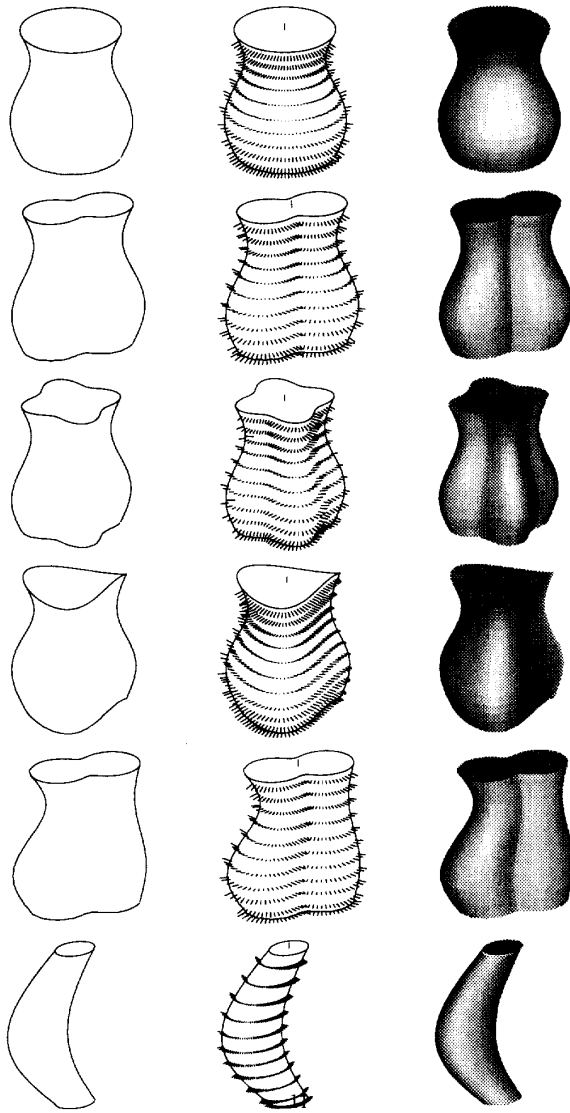


Fig. 12. The needle images and the shaded images generated with the computed gradients at each point of the SHGCs in Fig. 1.

Fig. 12 shows the result of local surface orientation reconstruction of the SHGCs in Fig. 1. Including computation of cross sections, the processing time is approximately 50 seconds for each object (40 × 40 points) on a Symbolics 3645 running lisp. We use two graphical methods to display the computed orientations. The first one shows the surface normals as oriented *needles* along the cross sections. The orientation and length of a needle is the projection of a unit surface normal at that point (for us to perceive 3D orientation from this requires solving a shape from contour problem in itself). The second method is to display the surface orientations by

constructing a synthetic shaded image from the reconstructed surface, by assuming Lambertian reflection and a point source of light (for humans to perceive this requires solving a shape from shading problem). We believe the needle diagrams to be more effective than shaded images for this purpose.

Evaluation of shape from contour results is difficult as there is no real “ground truth.” Even when contours *are* derived from a real object, or from projection of a synthetic object, the same boundaries *could* have been derived from a projection of infinitely many other real or synthetic objects. Thus, in a sense, the only good measure of the performance of our algorithms is a comparison with human performance. We present graphical results for a reader to make his/her own judgment. Unfortunately, this can only give qualitative, rather than quantitative, evaluation.

We believe that the results agree with human perception. It is also worth noting that for all the objects the computed orientation at the limb boundaries of the objects is orthogonal to the boundary, even though this is not an explicit constraint in our method.

### VIII. QUANTITATIVE SHAPE RECOVERY OF PRCGC SURFACES

Here we discuss the application of the three constraints discussed in Section VI along a cross section curve of a PRCGC, to recover the surface orientation of the PRCGC.

**CSBC.** The shared boundary constraint can be applied along the image of a cross section curve between the plane containing the cross section and the surface of the PRCGC. Let  $(p_c, q_c)$  be the gradient of the plane that contains the cross section curve,  $C(u)$ , whose image is the image curve  $C_i(u) = (c_x(u), c_y(u))$ . Let  $(p(u), q(u))$  be the orientation of the points along the cross section curve  $C(u)$ . Then the shared boundary constraint is:

$$(p_c - p(u), q_c - q(u)) \cdot (c'_x(u), c'_y(u)) = 0 \quad (35)$$

**ISC.** Theorem 6 indicates that ISC is applicable along the cross sections of a PRCGC because cross sections of a PRCGC join the parallel symmetric points of the meridian curves which are parallel symmetric as given by Lemma 2. Since the meridians of a PRCGC are parallel symmetric with cross section curves forming the correspondence, the tangent vectors of the meridians along a cross section are a constant vector which is also parallel to the axis of the PRCGC as given by (8). Let the tangent direction of the meridians along the cross section  $C(u)$  be  $A'$  and its image be  $A'_i = (a'_x, a'_y)$  (note that  $A'_i$  is independent of the  $u$  parameter). For the sake of simplicity let us assume that the coordinate system is rotated such that  $A'_i$  is along the  $y$  axis of the coordinate system, then  $a'_x = 0$ . The inner surface constraint<sup>7</sup> is :

<sup>7</sup> Note that, in reference to Theorem 5, the  $C$  curve is the cross section and the  $R$  is the tangent of the meridian. Hence, the condition of the theorem that  $R$  is not parallel  $C'$  is equivalent to the condition that a meridian is not parallel to a cross section which is guaranteed by the definition of PRCGCs.

$$\frac{d}{du}(p(u), q(u)) \cdot (a'_x, a'_y) = 0 \Rightarrow q(u)' a'_y = 0 \Rightarrow q(u) = q_0 \quad (36)$$

By combining this constraint with the CSBC given in (35) we get:

$$p(u) = \frac{c'_y(u)(q_c - q_0)}{c'_x(u)} + p_c \quad (37)$$

**Orthogonality.** The last constraint is the orthogonality of the meridians to the cross section curves. The reader can easily verify that the  $u$  and  $t$  parameter curves in (3) are orthogonal to each other for all points on the surface  $S$  of the PRCGC. Then we use the orthogonality by forcing the tangent of the meridians  $A'$ , whose image is  $A'_i = (0, a'_y)$  to be orthogonal to the tangent of the cross section curve  $C$ , whose image is  $C_i(u) = (c_x(u), c_y(u))$ , at a point on the surface whose gradient is  $(p(u), q_0)$ :

$$(0, a'_y, q_0 a'_y) \cdot (c'_x(u), c'_y(u), p(u)c'_x(u) + q_0 c'_y(u)) = 0 \quad (38)$$

By substituting  $p(u)$  given in (37) in the above equation, we get:

$$a'_y c'_y(u) (1 + q_0 q_c) + a'_y p_c c'_x(u) = 0 \quad (39)$$

Since the above equation is zero for all values of  $u$  we get both  $p_c = 0$  and

$$1 + q_0 q_c = 0 \Rightarrow q_0 = -1/q_c \quad (40)$$

Fixing  $q_c$  fixes the orientation of the surface along the cross section  $C$  together with the gradient  $(p_c, q_c)$  (which is  $(0, q_c)$  in the rotated coordinate system) of the plane containing  $C$ . However our constraint equations do not constrain  $q_c$ .

#### A. Recovering Cross Section Curves

In the previous section we discussed how to recover the surface orientation at each point on a cross section curve given the image of the cross section curve. However, it is not directly possible to replicate the images of the cross section curves of a PRCGC, such as the ones in Fig. 2, except for the ends of the PRCGC, where we assume the cross section curve is given. That is, we assume that the surface is cut along its cross sections. Here we discuss a method for recovering the cross sections when one or both ends of the PRCGC are available; the method also enables us to reconstruct the 3D PRCGC from the image of it.

At one end of the PRCGC let the image of the end cross section curve be  $C_i(u) = (c_x(u), c_y(u))$  and the image of the axis be  $A_i(t) = (a_x(t), a_y(t))$  as in the previous section. The image of the axis at the point it intersect the cross section  $C$  is  $A_i(0) = (a_x(0), a_y(0))$ . Say the coordinate system is rotated such that  $a'_x(0) = 0$ , and the orientation  $(p_c, q_c)$  of the plane containing the cross section curve  $C$  is computed using the constraints discussed in Section VIII with  $p_c = 0$ . The orientation of the

points along the cross section curve  $C$  is  $(p(u), q_0)$  where  $q_0 = -1/q_c$  and  $p(u)$  are given by Equation 37. Since the meridian curves are parallel symmetric to the axis of the PRCGC we can use the gradient  $(p(u), q_0)$  to recover the tangent of the 3D axis at  $t = 0$  as:

$$\begin{aligned} A'(0) &= (a'_x(0), a'_y(0), -(p(u)a'_x(0) + q_0 a'_y(0))) \\ &= \left(0, a'_y(0), \frac{a'_y(0)}{q_c}\right) \equiv (0, q_c, 1) \end{aligned} \quad (41)$$

That is  $A'(0)$  is parallel to normal,  $(0, q_c, 1)$ , of the plane containing the cross section  $C$ , or the plane  $\Pi_a$  containing  $A'(0)$  is orthogonal to the plane of  $C$ . Also since the axis,  $A$ , of the PRCGC is planar, the plane  $\Pi_a$  contains the whole axis curve  $A$ .

In the following we give an algorithm for recovering the 3D cross sections from the image of a PRCGC given the gradient  $(p_a, q_a)$  of the plane  $\Pi_a$  containing the axis. Then, in the next subsection, we give a method for computing  $(p_a, q_a)$  from the image.

The gradient  $(p_c, q_c)$  of the plane of the cross section  $C$  can be computed if the gradient  $(p_a, q_a)$  of  $\Pi_a$  is given. The gradient  $(p_c, q_c)$  must lie on a line that passes through the origin and in the direction of  $A_i(0)$ , in our case  $p_c = 0$ , and  $(p_c, q_c, 1)$  is orthogonal to  $(p_a, q_a, 1)$  then:

$$(0, q_c, 1) \cdot (p_a, q_a, 1) = 0 \Rightarrow q_c = -\frac{1}{q_a} \quad (42)$$

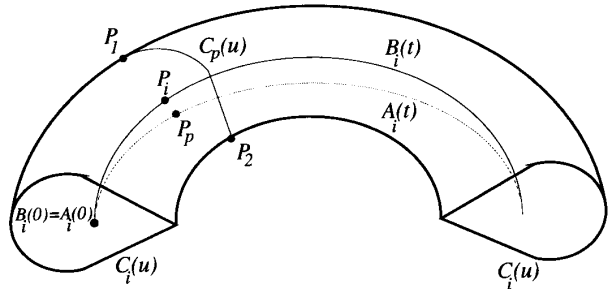


Fig. 13. A PRCGC with a non-rotationally symmetric cross section.

We can compute the 3D cross section  $C$  from the image  $C_i$  of it by backprojecting  $C_i$  to a plane having gradient  $(p_c, q_c)$ .

If the cross section is rotationally symmetric<sup>8</sup> the algorithm for recovering cross sections is much simpler. In the following we give an algorithm that applies to general, not necessarily rotationally symmetric, cases.

It can be shown that the image of the axis of the PRCGC,  $A_i(t)$ , is not always the same as the axis,  $B_i(t)$ , of the parallel symmetry of the image of the limb edges, where the axis of the PRCGC is the trace of a single point on the cross section as the cross section is swept. This is shown in Fig. 13. However, the

<sup>8</sup> A planar cross section is rotationally symmetric iff the lines passing through the center of the cross section intersect both sides of the cross section at equal distances. }

image curves  $A_i(t)$  and  $B_i(t)$  are always parallel symmetric to each other such that the corresponding points are on the same cross section. By using Lemma 2 and Theorem 4, we conclude that the images of the limb edges are parallel symmetric to each other (and, of course, to its axis) as well as to the images of the meridians of the surface, and, as the meridians of the surface are parallel symmetric to the axis of the PRCGC by (8), so are their images. Therefore the axis of the image of the limb edges, the  $B_i(t)$  curve, is parallel symmetric to the image of the axis of the PRCGC, the  $A_i(t)$  curve.

The above does not define the position of the axis  $A$  uniquely. Without loss of generality, we require the axis  $A$  to pass through the backprojection of  $B_i(0)$  onto the cross section plane  $C$ , i.e.,  $A_i(0) = B_i(0)$ . By Theorem 4, the points  $P_1$  and  $P_2$ , the intersection of the cross section  $C_p(u)$  with the limb edges, are given by the corresponding points of the parallel symmetry of the limb edges.

Now, we can recover the 3D cross section,  $C_p(u)$  corresponding to a given point  $P_i$ , on the image axis  $B_i$  by the following procedure.

- 1) Backproject image axis tangents  $B'_i(0)$  and  $B'_i(P_i)$  onto the axis plane  $\Pi_a$ . Let the resulting vectors be called  $B'(0)$  and  $B'(P)$ , respectively.
- 2) Rotate the cross section  $C$  and the point  $A(0)$  by the matrix  $R(B'(0), B'(P))$  to give us a new curve, say,  $\hat{C}_p(u)$  and a new point, say,  $\hat{P}_p$ .  $\hat{C}_p(u)$  is parallel to  $C_p(u)$ .
- 3) Search for two points, say  $Q_1$  and  $Q_2$ , on  $\hat{C}_p(u)$  whose tangents to  $\hat{C}_p(u)$  project to vectors parallel to the tangents to the limb boundaries at  $P_1$  and  $P_2$  (as the cross section and the limbs must be tangential to each other).
- 4) Translate  $\hat{C}_p(u)$  such that the projections of points  $Q_1$  and  $Q_2$  coincide with points  $P_1$  and  $P_2$ , and the point  $\hat{P}_p$  lies on the axis plane  $\Pi_a$ . The resulting curve is the desired  $C_p(u)$ .

### B. Computing $(p_a, q_a)$

The gradient  $(p_a, q_a)$  of the plane  $\Pi_a$  containing the axis is computed by performing a search in the gradient plane. The objective of the search is to compute  $(p_a, q_a)$  that gives a valid reconstruction. A valid construction is one that makes the projection of the cross section points  $\sqrt{\quad}$  and  $C_p(P_2)$  exactly the same as the points  $P_1$  and  $P_2$  on the image plane (see Fig. 13). We form an objective function which is the average distance, on the image plane, of the reconstructed and projected point  $C_p(P_2)$  to the point  $P_2$  when  $C_p(P_1)$  and  $P_1$  are aligned exactly. Then this objective function is minimized for  $(p_a, q_a)$ .

The search is facilitated by finding a good initial point for  $(p_a, q_a)$  using the shapes of the end cross sections. The analysis in Section VIII shows that the gradient  $(p_c, q_c)$  of the cross section at one end is constrained to be on a line in the gradient space. A particular value on that line may be chosen by using

the ellipse fit discussed in Section VII.C. Similar analysis applies to the other end of the PRCGC (if available). Say the orientation of the plane containing the other end cross section  $C_n$  is  $(p_n, q_n)$ . Then the plane of  $C_n$  is orthogonal to the plane  $\Pi_a$ . If  $(p_n, q_n)$  is not equal to  $(0, q_c)$  we can compute an initial normal  $N_a = (p_a, q_a, 1)$  of  $\Pi_a$  as  $N_a \equiv (p_n, q_n, 1) \times (0, q_c, 1)$ . If the other end cross section  $C_n$  is not available then the gradient  $(p_a, q_a)$  is constrained to be on a line by its orthogonality to  $(0, q_c)$ . The equation of the line containing  $(p_a, q_a)$  is  $(0, q_c, 1) \cdot (p_a, q_a, 1) = 0$ . Any particular value of  $(p_a, q_a)$  may be chosen on this line as the initial  $(p_a, q_a)$ . Fig. 14 shows that perception is more definite when both ends are available, which confirms the above observation that two ends are more informative than only one.

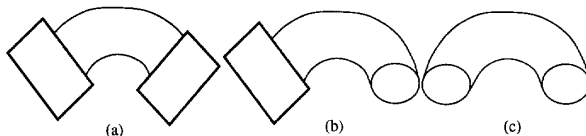


Fig. 14. A schematic drawing of a PRCGC with (a) none, (b) one, and (c) both end cross sections available.

### C. Results

We have implemented the cross section recovery method described in Section VIII.A on synthetic contour data. As in the case of SHGCs, the input to the algorithm is a segmented set of curves representing the contours of the PRCGCs. Each curve is given by a list of points with exact coordinates. In the implementation, first the orientations  $(p_c, q_c)$  and  $(p_n, q_n)$  of the end cross sections are computed using the ellipse fit method described in Section VII.C. Then the normal  $N_a$  of  $(p_c, q_c, 1) \times (p_n, q_n, 1)$  that gives a valid reconstruction as described in Section VIII.B. The 3D position of each cross section is then found by translating the end cross section, rotating and aligning it with the limb boundaries and the plane of the axis  $\Pi_a$  as described in Section VIII.A.

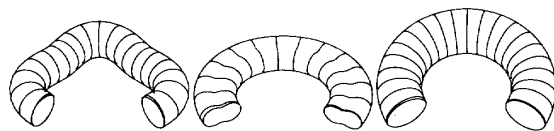


Fig. 15. The recovered cross sections for the PRCGCs in Fig. 2.

Fig. 15 shows the recovered cross sections and Fig. 16 shows the recovered orientations by both needle and shaded images for the PRCGCs given in Fig. 2. The computation time is approximately one minute per object running in LISP on a Symbolics 3645. As mentioned in Section VII.D there is no real ground truth to compare to the computed surface normals. Since many objects could produce the shown contours and the specific model used to generate the image contours has no particular importance, the real comparison is with human perception. In our opinion the results are in agreement with human perception. We provide graphical display for readers to judge the results on their own.

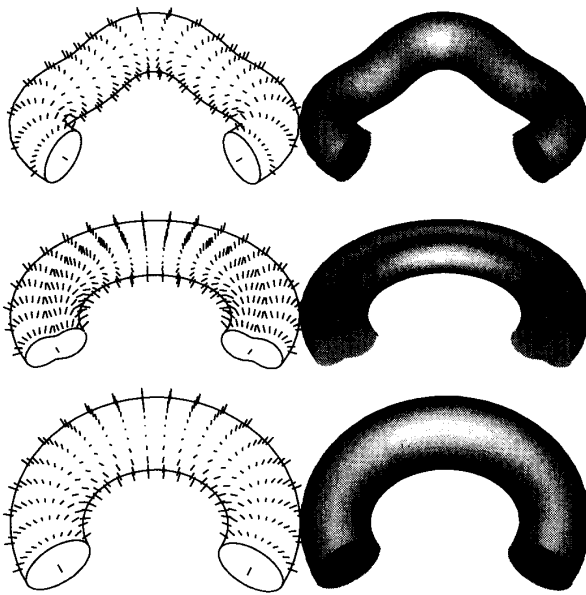


Fig. 16. The recovered orientations shown by both needle image and by shading the objects for the PRCGCs in Fig. 2.

### IX. CONCLUSIONS

We have presented a methodology for inferring 3D shape from contour for two broad classes of curved objects, namely SHGCs and PRCGCs, provided they exhibit certain symmetry properties. Together with our earlier work on zero-Gaussian curvature surfaces, we believe that this methodology covers a large fraction of the surfaces found in man-made environments.

Our theory does make certain assumptions, as must all such methods, to extract shape from contour. We feel that our assumptions are minimal and justified, both by the geometrical analysis and by the agreement with human perception. We have argued that, in a certain sense, the only evaluation can be with human perception, although such comparison is difficult to quantify.

We have presented results on several synthetic examples. To apply our method to complex real images where surface markings, shadows, and highlights may be present, will require further processing to identify the boundaries that contribute to the object shape and, perhaps, to fill in some missing contours. We believe that our methodology will be helpful here too, as it provides specific conditions that a process may search for. We intend to pursue this in our future research.

### ACKNOWLEDGMENTS

We would like to thank Dr. Terry Boulton, the associate editor of *Trans. on PAMI*, for his careful and critical reading of this paper and many suggestions for improving our presentation. In particular, his comments have helped us make the statements and proofs of the mathematical properties in the paper more rigorous.

This research was supported by the U.S. Advanced Research Projects Agency under contract number F49620-90-C-0078, monitored by the U.S. Air Force Office of Scientific Research. The United States Government is authorized to reproduce and distribute reprints for governmental purposes, notwithstanding any copyright notation hereon.

### REFERENCES

- [1] T.O. Binford, "Visual perception by computer," *IEEE Conf. on Systems and Controls*, Miami, Fla., Dec. 1971.
- [2] H.G. Barrow and J.M. Tenenbaum, "Interpreting line drawings as three dimensional surfaces," *Artificial Intelligence*, vol. 17, pp. 75-116, 1981.
- [3] M. Brady and A. Yuille, "An extremum principle for shape from contour," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 6, pp. 288-301, 1984.
- [4] M.B. Clowes, "On seeing things," *Artificial Intelligence*, vol. 2, no. 1, pp. 79-116, 1971.
- [5] M.P. Do Carmo, *Differential Geometry of Curves and Surfaces*, Prentice Hall, 1976.
- [6] A. Gross and ? Boulton "Straight homogeneous generalized cylinders: constraints from contour," *Proc. of the Image Understanding Workshop*, pp. 573-582, Pennsylvania, 1990.
- [7] R. Horaud and M. Brady, "On the geometric interpretation of image contours," *Artificial Intelligence*, vol. 37, pp. 333-353, 1988.
- [8] D.A. Huffman, "Impossible objects as nonsense sentences," *Machine Intelligence*, vol. 6, pp. 295-323, 1971.
- [9] T. Kanade, "Recovery of the three-dimensional shape of an object from a single view," *Artificial Intelligence*, vol. 17, pp. 409-460, 1981.
- [10] R. Kashipati, *Shape Description from Sparse and Imperfect Data*, Ph.D. thesis, Univ. of Southern Calif., 1988.
- [11] T. Kanade and J.R. Kender, "Mapping image properties into shape constraints: Skew symmetry, affine transformable patterns, and the shape from texture paradigm," J. Beck, B. Hope, and A. Rosenfeld, ed., *Human and Machine Vision*, pp. 237-257, Academic Press, 1983.
- [12] J.J. Koenderink, "What does the occluding contour tell us about solid shape," *Perception*, vol. 13, pp. 321-330, 1984.
- [13] A.K. Mackworth, "Interpreting pictures of polyhedral scenes," *Artificial Intelligence*, vol. 4, pp. 121-137, 1973.
- [14] R. Mohan and R. Nevatia, "Segmentation and description based on perceptual organization," *Proc. of Computer Vision and Pattern Recognition Conf.*, pp. 333-341, San Diego, Calif., 1989.
- [15] V. Nalwa, "Line-drawing interpretation: bilateral symmetry," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 11, pp. 1117-1120, 1989.
- [16] J. Ponce and D. Chelberg, "Finding the limbs and cusps of generalized cylinders," *Int'l J. of Computer Vision*, vol. 1, pp. 195-210, 1987.
- [17] J. Ponce, D. Chelberg, and W.B. Mann, "Invariant properties of straight homogeneous generalized cylinders and their contours," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 11, no. 9, pp. 951-966, 1989.
- [18] S.A. Shafer, "Shadow geometry and occluding contours of generalized cylinders," Tech. Report CS-83-131, Carnegie Mellon Univ., 1983.
- [19] S.A. Shafer and T. Kanade, "The theory of straight homogeneous generalized cylinders", Tech. Report CS-083-105, Carnegie Mellon Univ., 1983.
- [20] S.A. Shafer, T. Kanade and J.R. Kender, "Gradient space under orthography and perspective," *Computer Vision, Graphics and Image Processing*, vol. 24, pp. 182-199, 1983.
- [21] P. Saint-Marc and G. Medioni, "B-spline contour representation and symmetry detection," *First European Conf. on Computer Vision*, pp. 64-606, Antibes, France, April 1990.
- [22] K.A. Stevens, "The visual interpretations of surface contours," *Artificial Intelligence*, vol. 17, pp. 47-73, 1981.
- [23] F. Ulupinar and R. Nevatia, "Inferring shape from contour for curved surfaces," *Proc. of the 10th Int'l. Conf. on Pattern Recognition*, pp. 147-154, Atlantic City, NJ, 1990.

- [24] F. Ulupinar and R. Nevatia, "Shape from contour: SHGCs," *Proc. of the Third ICCV*, Japan, 1990.
- [25] F. Ulupinar and R. Nevatia, "Recovering shape from contour for constant cross section generalized cylinders," *Proc. of Computer Vision and Pattern Recognition Conf.*, Maui, Hawaii, 1991.
- [26] F. Ulupinar and R. Nevatia, "Perception of 3D surfaces from 2D contours," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 15, no. 1, pp. 3-18, 1993.
- [27] I. Weiss, "3D shape representation by contours," *Computer Vision, Graphics and Image Processing*, vol. 41, pp. 80-100, 1988.
- [28] G. Xu and S. Tsuji, "Inferring surfaces from boundaries," *Proc. of the First ICCV*, pp. 716-720, London, 1987.
- [29] M. Zerroug and R. Nevatia, "Scene segmentation and volumetric descriptions of SHGCs from a single intensity image," *Proc. of the Image Understanding Workshop*, pp. 905-916, Washington, D.C., 1993.



**Fatih Ulupinar** received the B.Sc. degree in computer engineering in 1985 from the Middle East Technical University, Ankara, Turkey, and M.Sc. and Ph.D. degrees in computer engineering from the University of Southern California, Los Angeles, Calif., in December, 1986, and August, 1991, respectively.

Dr. Ulupinar conducted research on various areas of computer vision at the Institute of Robotics and Intelligent Systems (IRIS), University of Southern California, from 1986-1991. He authored various research papers and book chapters during his stay at

IRIS.

Dr. Ulupinar worked as an assistant professor in the Computer and Information Science Department of the Bilkent University, Ankara, Turkey, from 1992 to 1993. He is currently a member of the technical staff at Advanced Computing Systems Company, Los Angeles, Calif. His research interests include computer vision, multimedia, and computer networks.



**Ramakant Nevatia** received his B.S. degree from the University of Bombay, India, and the M.S. and Ph.D. from Stanford University, Stanford, Calif., all in electrical engineering.

Dr. Nevatia has been with the University of Southern California, Los Angeles, Calif., since 1975. He is currently a professor of computer science and electrical engineering and director of the Institute for Robotics and Intelligent Systems. He spent the academic year 1981-1982 at Stanford University as a visiting faculty member.

Dr. Nevatia is the author of two books, *Machine Perception and Computer Analysis of 3D Curved Objects*, and has contributed chapters to several other books. His research interests include computer vision, artificial intelligence, and robotics.

Dr. Nevatia is a fellow of the American Association for Artificial Intelligence and a member of the Association for Computing Machinery. He is an associate editor of *Pattern Recognition and Computer Vision, Graphics and Image Processing* and just recently completed his term as an associate editor of *IEEE Transactions on Pattern Analysis and Machine Intelligence*. He has also served as a technical editor in the areas of robot vision and inspection systems for the *IEEE Journal of Robotics and Automation*.