On the Geometric Interpretation of Image Contours

Radu Horaud
LIFIA, 46, avenue F. Viallet, 38031 Grenoble, France

Michael Brady
University of Oxford, Department of Engineering Science, Parks Road, Oxford, OX1 3PJ, United Kingdom

ABSTRACT

In this paper we suggest a computational model for the 3D interpretation of a 2D view based on contour classification and contour interpretation. We concentrate on those contours arising from discontinuities in surface orientation. We combine a generic surface description well suited for visual tasks with a model of the image formation process in order to derive image contour configurations that are likely to be interpreted in terms of surface contours. Next we describe a computer algorithm which attempts to interpret image contours on the following grounds. First, an image analysis process produces a description in terms of contours and relationships between them. Second, among these contours, we select those which form a desired configuration. Third, the selected contours are combined with constraints available with the image formation process in order to be interpreted in terms of discontinuities in surface orientation. As a consequence, there is a dramatic reduction in the number of possible orientations of the associated scene surfaces.

1. Introduction

An image is a two-dimensional (2D) projection of a three-dimensional (3D) scene. Many scene characteristics such as discontinuity in surface orientation, changes in surface reflectance, shadows, and textures give rise to image contours. An image contour simply characterizes the variation of image brightness. The analysis of this variation cannot, by itself, provide a correct interpretation in terms of the scene attributes mentioned above.

In spite of this apparent ambiguity, human beings have no difficulty in properly interpreting images. We can easily recognize complex objects, find their spatial orientation, and even more, learn a great deal about objects never seen before. In this paper we suggest a computational model for the 3D interpretation of a 2D view based on contour classification and contour interpretation. We concentrate on those contours arising from discontinuities in

Artificial Intelligence **37** (1988) 333–353
0004-3702/88/$3.50

surface orientation. We combine a generic surface description well suited for visual tasks with a model of the image formation process in order to derive image contour configurations that are likely to be interpreted in terms of surface contours.

Next we describe a computer algorithm which attempts to interpret image contours on the following grounds. First, an image analysis process produces a description in terms of contours and relationships between them. Second, among these contours, we select those which form a desired configuration. Third, the selected contours are combined with constraints available from the image formation process in order to be interpreted in terms of discontinuities in surface orientation. As a consequence, there is a dramatic reduction in the number of possible orientations of the associated scene surfaces.

1.1. Background and approach

Among many surface description alternatives, generalized cylinders and generalized cones have been introduced by Binford [4] as a convenient object representation within the context of visual recognition. Furthermore, Marr [11] suggested an argument in support of this representation. He argued that under a set of assumptions, there is a one-to-one map between an *occluding contour* and a generalized cylinder. The recognition strategy suggested by Marr is based on the observation that the axis of symmetry of the occluding contour is the image of the generalized cylinder's axis. This result is true only under parallel projection and, more restrictive, when the viewing direction is perpendicular to the cylinder's axis. For a foreshortened view, one of Marr's assumptions is not valid anymore, namely that the *contour generator* of the occluding contour is planar.

This restriction can be partially overcome if an occluding contour is further decomposed into *extremal* and *discontinuity* contours. This distinction has been introduced by Barrow and Tenenbaum [3] and is made clear on Fig. 1. An occluding boundary separates an object from the background. A discontinuity boundary occurs whenever a smooth surface terminates or intersects with another surface. An extremal boundary occurs where a curved surface turns smoothly away from the viewer. Hence, 3D shape recovery from image contours is twofold: interpretation of extremal *and* discontinuity contours. On this line of investigation a certain number of geometric and topological properties have been reported which relate extremal contours to surfaces. See for example Koenderink [9] and Brady et al. [6]. Discontinuity contours are harder to interpret mainly because they correspond to surface intersections whose orientations in space are not constrained by the viewing direction.

A number of researchers have concentrated on the 3D interpretation of an isolated image contour. If this contour is thought of as the projection of a space (scene) contour, the recovery of the space contour shape is not trivial. This is

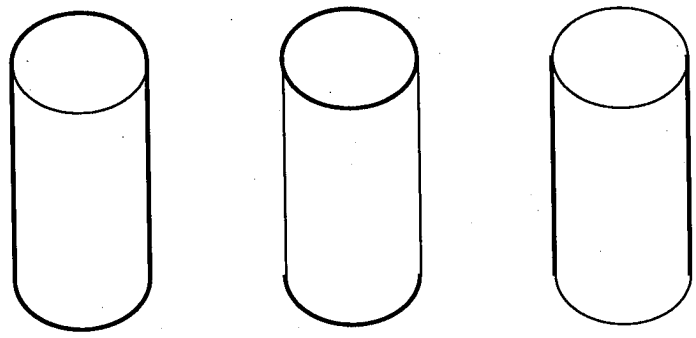

Fig. 1. The distinction between *occluding*, *discontinuity*, and *extremal* boundaries.

referred to as "shape from contour". Existing methods generally operate on a closed contour and, with a few exceptions they assume planar space curves. The problem is reduced now to estimating the orientation of this plane with respect to the viewer. Since the problem is underconstrained, additional assumptions about the shape of the space contour are required. Standard approaches make use of the following assumptions: uniform distribution of tangent directions (Witkin [13]), uniformity of curvature (Barrow and Tenenbaum [3]), minimal entropy shape (Barnard [1]), most compact figure assumption (Brady and Yuille [7]), and symmetry (Kanade [8]). In the case of nonplanar interpretations Witkin [13] and Barnard and Pentland [2] offer some suggestions.

Shape from contour methods provide a unique solution, namely the orientation of the plane in which the space curve lies. This solution corresponds to the minimization or maximization of the criterion being used. This standard approach needs to be adapted to the case where the contour in question lies on an object, at the intersection of two surfaces. Rather than producing just one solution, an alternative approach may attempt to represent explicitly a range of possible solutions and to select the one which best matches constraints derived from various sources. Hence, the key idea of our approach is first to interpret separately extremal and discontinuity contours and second to combine their contribution. Instead of producing unique solutions for each contour, we derive constraints in the form of one- or two-parameter loci. These constraints can be combined on the basis of a postulated geometric relationship between the contour interpretations which may lead to a dramatic reduction in the number of solutions, i.e., a small finite set instead of a continuous locus. Moreover, if it turns out that none of these solutions is correct, a different set of solutions can rapidly be hypothesized by applying a different geometric relationship.

2. Relating Image Contours to Surfaces

We analyze the origins of the image intensity changes referred to in this paper as *image contours*. From the physical processes underlying their formation, image contours fall into four classes (see [12]):

(1) discontinuities in distance from the viewer;
(2) discontinuities in surface orientation;
(3) changes in surface reflectance;
(4) illumination effects.

Interestingly, all these classes contribute to the recovery of 3D surfaces. However, we are interested here in those contours that are intrinsically related to the geometry of surfaces being viewed, that is class (2). As has been already mentioned, these contours can be further classified into extremal and discontinuity contours. The former occur wherever a surface turns smoothly away from the viewer while the latter occur where smooth surfaces terminate or intersect. Moreover, reconstruction techniques such as "shape from stereo" and "shape from motion" have difficulties with this class of contours.

2.1. Definition of generalized cylinders

It has been suggested that generalized cylinders are well suited for surface representation. Let's analyze in detail how class (2) image contours may be related to the geometry of a generalized cylinder. A generalized cylinder is a special case of a more general 3D shape, a generalized cone. A generalized cone is defined by:

- a planar cross-section curve $x = f(s)$ and $y = g(s)$, where s denotes the curvilinear abscissa;
- an eccentricity angle ψ;
- a spine function;
- an expansion function h.

A generalized cylinder is a generalized cone with a straight spine (straight axis) and zero eccentricity (the cross-section is perpendicular to the axis). See Fig. 2. Hence, the cross-section and expansion function are sufficient to define a generalized cylinder which can be written as:

$$\boldsymbol{r}(s, z) = h(z)f(s)\boldsymbol{i} + h(z)g(s)\boldsymbol{j} + z\boldsymbol{k}\,, \tag{1}$$

Where $\boldsymbol{i}$, $\boldsymbol{j}$ and $\boldsymbol{k}$ are the unit vectors associated with a cylinder centered reference frame: The cross-section lies in the $\boldsymbol{i}$-$\boldsymbol{j}$ plane (x-y plane) and $\boldsymbol{k}$ (the z-axis) is parallel to the cylinder's straight axis. It is important to notice that this is simply a generative definition which does not guarantee real surfaces. Also, h, f and g must be analytical.

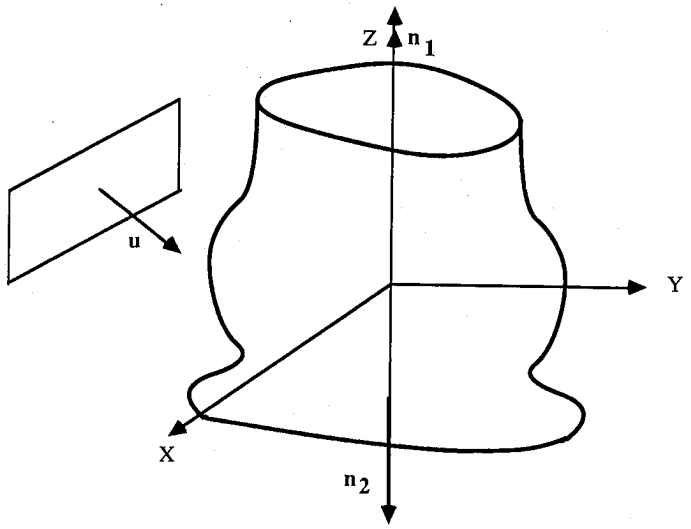

Fig. 2. A generalized cylinder intersected by two planes with normal unit vectors n_1 and n_2 parallel to the z-axis

2.2. Extremal contours

Let n be the unit vector normal to the cylinder's surface at (s, z); this vector is given by [6]:

$$n(s, z) = h(z)\frac{dg}{ds}\,i - h(z)\frac{df}{ds}\,j + h(z)\frac{dh}{dz}\left(g(s)\frac{df}{ds} - f(s)\frac{dg}{ds}\right)k\,. \quad (2)$$

Consider now a unit vector u parallel to the viewing direction and assume orthographic projection. The *contour generator* associated with the viewing direction u is the surface contour defined such as the viewing vector is tangent to the surface:

$$n(s, z)\cdot u = 0\,. \quad (3)$$

The projection of the contour generator onto the image plane defines the extremal contour associated with the generalized cylinder and a viewing direction. We now analyze the extent to which an extremal contour constrains the orientation in space of the generalized cylinder.

We restrict the generalized cylinder to be a surface of revolution. Many complex real objects are not *globally* symmetric but nevertheless are composed

of parts which have an axis of symmetry. Moreover, flexible objects, such as a hand or a snake preserve this *local* symmetry. This surface of revolution has the same expansion function as the initial cylinder and a circular cross-section; its axis is parallel to the cylinder's axis:

$$\boldsymbol{r}_u(\theta, z) = h(z)\cos\theta\,\boldsymbol{i} + h(z)\sin\theta\,\boldsymbol{j} + z\boldsymbol{k}\,. \tag{4}$$

It is impossible to determine the actual shape of the cross-section by observing an extremal contour. Let's call this surface the "perceived surface of revolution" or PSR. Since we deal with a surface of revolution, its cross-section axes are defined up to a rotation about the z-axis. There is no loss of generality in choosing the y-axis such that the viewing vector lies in the y-z plane, e.g., Fig. 3. The view vector can be written in the PSR's frame as:

$$\boldsymbol{u} = \cos\sigma\,\boldsymbol{j} + \sin\sigma\,\boldsymbol{k}\,, \tag{5}$$

where σ is the usual *slant* angle (the angle between $\boldsymbol{u}$ and $\boldsymbol{k}$). The vector normal to the PSR can be easily derived from equations (2) and (4):

$$\boldsymbol{n}_u = \cos\theta\,\boldsymbol{i} + \sin\theta\,\boldsymbol{j} - \frac{\mathrm{d}h}{\mathrm{d}z}\,\boldsymbol{k}\,. \tag{6}$$

A surface point is constrained to belong to the contour generator if it satisfies equation (3) which can be written as:

$$\cos\sigma\sin\theta = \sin\sigma\,\frac{\mathrm{d}h}{\mathrm{d}z}\,, \tag{7}$$

which gives rise to the following analysis:

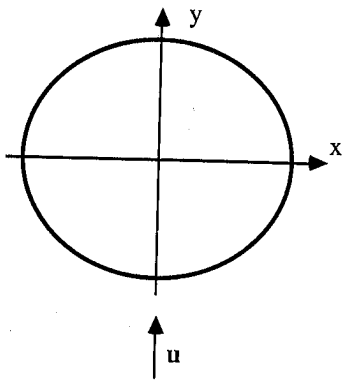

Fig. 3. The x- and y-axes of the perceived surface of revolution are chosen such that the viewing direction lies in the y-z plane.

(1) $\sigma = 0$. This case corresponds to a *side view* and θ may take the values 0 or π. The equation of the contour generator is obtained by replacing θ with either of these two values in the expression of $\boldsymbol{r}_u$ given by equation (4):

$$\boldsymbol{c}_u(z) = \pm h(z)\boldsymbol{i} + z\boldsymbol{k} . \tag{8}$$

(2) $\sigma = \frac{1}{2}\pi$. This case corresponds to a *top view* in which case equation (7) is reduced to $\mathrm{d}h/\mathrm{d}z = 0$. The contour generator is the *skeleton* of the PSR.

(3) In the general case, a solution for θ exists only if $|\tan\sigma\, \mathrm{d}h/\mathrm{d}z| \leq 1$ in which case there are two solutions for θ:

$$\theta_1 = \arcsin\left(\tan\sigma \frac{\mathrm{d}h}{\mathrm{d}z}\right) \quad \text{and} \quad \theta_2 = \pi - \theta_1 .$$

The contour generator can be written as:

$$\boldsymbol{c}_u(z) = \pm h(z)\cos\theta_1 \boldsymbol{i} + h(z)\sin\theta_1 \boldsymbol{j} + z\boldsymbol{k} . \tag{9}$$

The following conclusions are interesting within the context of image contour interpretation:

(i) A contour generator (and hence an extremal contour) cannot be defined for an arbitrary generalized cylinder and for an arbitrary viewing direction; hence, if an image contour is interpreted as an extremal contour, it constrains both the shape of the viewed surface and its spatial orientation relatively to the viewer.

(ii) For a side view, the contour generator is a planar curve lying in a plane parallel to the image plane and is identical to the expansion function; this situation has been studied by Marr [11].

(iii) In the more general case the contour generator is not planar; however for surfaces whose expansion functions are linear (which are of great practical importance), $\mathrm{d}h/\mathrm{d}z$ vanishes and hence the contour generator is planar. This is the case for simple man-made object primitives such as cones, cylinders and wedges.

(iv) The contour generator is symmetric about the y-z plane, that is, the plane defined by the viewing direction and the axis of the PSR; it follows that its image projection, i.e., the extremal boundary, is symmetric about the projection of the y-z plane.

(v) The axis of symmetry of the extremal contour is equal to the image of the PSR's axis. Since the PSR is defined such that its axis is parallel to the generalized cylinder's axis, it follows that the axis of symmetry of the extremal contour is parallel to the image of the generalized cylinder's axis.

The last conclusion has practical importance. It means that whenever an

image contour has a straight axis of symmetry it can be interpreted as an extremal contour and moreover, the axis of the cylinder which gave rise to this contour is constrained to lie in a plane defined by the contour's axis of symmetry and the viewing direction. This set of conclusions could probably be derived from the more general surface representation envisaged by Koenderink [9]. The reason for which we devised a theory restricted to generalized cylinders is to emphasize the relationship between symmetry detection in an image and a corresponding 3D interpretation. Indeed, it has been recently shown that axes of symmetry are a well-suited two-dimensional shape representation [5]. Relating 2D symmetry to surface representation is therefore an important component of a computational model for image understanding. Under certain assumptions local symmetry is invariant under projection and hence it is an important visual cue.

2.3. Discontinuity contours

We turn now to a more realistic situation: a finite object. Such an object is bounded by piecewise-smoothed surfaces. If the expansion function of a generalized cylinder is defined over a finite domain, the shape vanishes naturally. This is the case with such objects as eggs or fruits. In the more general case the domain of the expansion function is infinite. We bound the object by two planes with normal unit vectors $\boldsymbol{n}_1$ and $\boldsymbol{n}_2$. In general these planes may or may not be parallel to the x-y plane. *We restrict our analysis to the case where the two bounding planes are parallel to the x-y plane*, i.e., Fig. 2.

The intersection of a plane with a surface is a planar curve. The projection of this *space contour* onto the image gives rise to a discontinuity contour. Unlike extremal contours, there is no constraint which relates the discontinuity contour to the spatial orientation of an object. It is therefore necessary to provide an additional piece of knowledge about the shape of the space contour.

One way to express knowledge about the global shape of a closed planar curve is to compute the ratio of the enclosed area to the square of the perimeter:

$$M = \frac{\text{Area}}{(\text{Perimeter})^2} .$$

This is a scale-invariant number characterizing the curve. M is maximized by a circle (the most compact shape). This gives the measure an upper bound of $1/4\pi$. Its lower bound is zero and it is achieved by a straight line. The quantity M is commonly used in pattern recognition and industrial vision systems as an attribute that measures the compactness of an object.

Brady and Yuille [7] have used this measure within the shape from contour paradigm. They have developed an extremum principle for determining the orientation of the most symmetric and most compact space contour interpreta-

tion. This is achieved for the maximum value of M. For example an ellipse is interpreted as a tilted circle and a parallelogram as a tilted and rotated square. However, apart from psychophysical arguments, there is no reason a priori for selecting this unique interpretation. Instead one may compute M explicitly at each possible orientation and express M as a two-dimensional function with azimuth and elevation as parameters. If a specific shape is expected to occur, the value of the compactness measure determines a one-parameter locus of points on the surface M (see next paragraph for an example).

Let's denote by C the space contour lying in a plane with unit normal vector $\boldsymbol{n}$ and let C_u be its image projection. Let $\boldsymbol{u}$ be the unit normal vector associated with the z-axis of the camera which is also the viewing direction. We want to express M for C as a function of $\boldsymbol{n}$ and C_u. The contour C is described by a set of successive points $V_0, V_1, \ldots, V_m$ where $V_0 = V_m$. Let O' be the intersection of the camera's z-axis with the space plane containing C and let $\boldsymbol{v}_i$ be a vector from O' to V_i. Refer to Fig. 4. The area enclosed by C is:

$$\text{Area} = \left| \sum_{i=1}^{m-1} A_i \right| , \tag{10}$$

where the area element is given by

$$A_i = \tfrac{1}{2} \boldsymbol{n} \cdot (\boldsymbol{v}_i \times \boldsymbol{v}_{i+1}) . \tag{11}$$

Similarly, the perimeter is given by:

$$\text{Perimeter} = \sum_{i=1}^{m-1} \| \boldsymbol{v}_i - \boldsymbol{v}_{i+1} \| . \tag{12}$$

A point V on C projects in the image at P on C_u. Let's compute the vector $\boldsymbol{v}$ from O' to V as a function of its image projection, that is the vector $\boldsymbol{p}$ from O to P. We shall first use perspective projection, establish the formula for $\boldsymbol{v}$ and show that M is not defined for any contour and for any spatial orientation. Next, we shall consider orthographic projection as for extremal contours.

Let F be the focal point, f be the focal length (the distance from F to the image plane), d be the distance from O to O', $\boldsymbol{u}$ be the viewing direction and $\boldsymbol{u}_p$ be the unit vector associated with the direction from F to P. See Fig. 4. $\boldsymbol{v}$ is given by the following formula (see Appendix A for details concerning its derivation):

$$\boldsymbol{v} = \frac{d+f}{f} \frac{(\boldsymbol{n} \cdot \boldsymbol{u}_p)\boldsymbol{p} - (\boldsymbol{n} \cdot \boldsymbol{p})\boldsymbol{u}_p}{\boldsymbol{n} \cdot \boldsymbol{u}_p} . \tag{13}$$

Notice that $\boldsymbol{v}$ is not defined whenever $\boldsymbol{n} \cdot \boldsymbol{u}_p = 0$ which means that for every

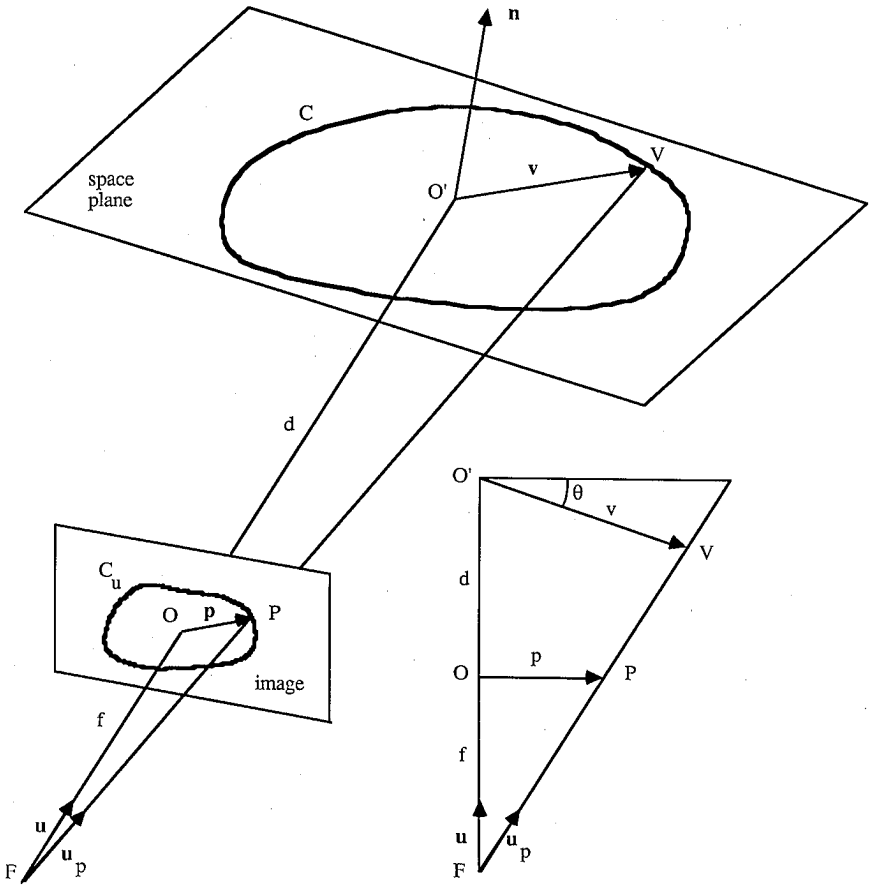

Fig. 4. The geometry of projecting a space contour C onto the image plane. C lies in a plane with normal unit vector $\boldsymbol{n}$. Equation (13) expresses $\boldsymbol{v}$ $(O'V)$ as a function of d, f, $\boldsymbol{n}$, $\boldsymbol{p}$ (OP) and $\boldsymbol{u}_p$.

image curve point P_i there exists a locus of orientations for which the space contour is infinitely elongated. Moreover, if $\boldsymbol{n}$ is perpendicular to the plane defined by $\boldsymbol{p}$ and $\boldsymbol{u}_p$ we have: $\boldsymbol{n} \cdot \boldsymbol{u}_p = \boldsymbol{n} \cdot \boldsymbol{p} = 0$; for these orientations the space contour *crosses the image*! Consequently, M has a lot of singularities depending on the shape of the image contour. Although it is worth investigating the behaviour of M within the perspective camera model context, let's consider parallel projection.

In the case of parallel projection f tends to infinity and hence $(d+f)/f$ tends to 1 and $\boldsymbol{u}_p$ tends to $\boldsymbol{u}$. Therefore equation (13) becomes:

$$\boldsymbol{v} = \frac{(\boldsymbol{n} \cdot \boldsymbol{u})\boldsymbol{p} - (\boldsymbol{n} \cdot \boldsymbol{p})\boldsymbol{u}}{\boldsymbol{n} \cdot \boldsymbol{u}}, \tag{14}$$

which is the formula used in [7]. Notice that $\boldsymbol{v}$ is not defined where $\boldsymbol{n} \cdot \boldsymbol{u} = 0$. Since $\boldsymbol{u}$ is the viewing vector, the locus of spatial orientations for which the space contour is infinitely elongated is a great circle bounding the visible half of the Gaussian sphere. For these orientations M is null. Therefore, the parallel camera model is to be preferred to the perspective model since it involves less singularities.

By combining equations (10), (12) and (14) one can express M as a function of $\boldsymbol{n}$, $\boldsymbol{u}$ *and the shape of* C_u described by the image points $P_0, P_1, \ldots, P_m$. Notice however that M *does not depend on* d, the distance from the viewer to the space plane.

To summarize, for a fixed camera position, and for a given image closed curve, M can be written as a two-parameter function $M(\alpha, \beta)$ where the two parameters are any suitable Gaussian sphere coordinates of $\boldsymbol{n}$. The domain of M is the set of surface orientations visible from the camera viewpoint. Assuming opaque objects, this domain is the hemisphere oriented toward the viewer.

2.4. Examples

Let's consider two examples, one of interpreting a pair of symmetric extremal contours and one of interpreting a closed (discontinuity) contour. Figure 5 shows two lines that may be interpreted as extremal contours arising from the projection of a cone. According to the last conclusion of Section 2.2, the axis

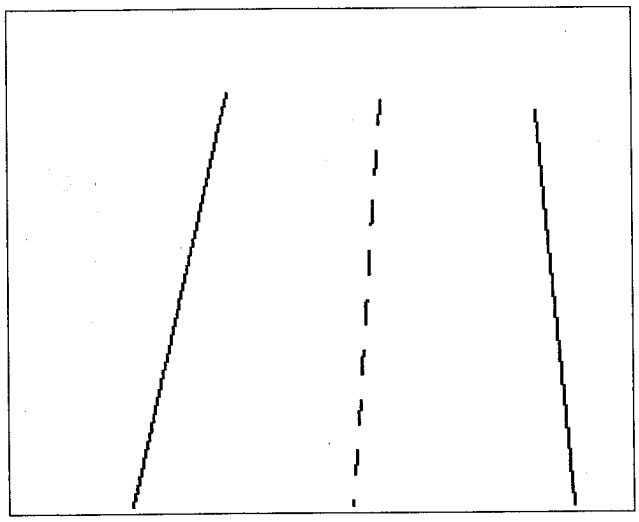

Fig. 5. Two image lines and their axis of symmetry.

of this cone should lie in the plane defined by the viewing direction and the axis of symmetry of the extremal contours. Let $\boldsymbol{u}$ be the viewing direction, $\boldsymbol{s}$ be the unit vector associated with the axis of symmetry and $\boldsymbol{k}$ be the unit vector associated with the cone's axis. The orientation of $\boldsymbol{k}$ is given by the following equation:

$$\boldsymbol{k} \cdot (\boldsymbol{u} \times \boldsymbol{s}) = 0 . \tag{15}$$

That is, the cone's axis lies in the plane defined by $\boldsymbol{u}$ and $\boldsymbol{s}$. The tip of $\boldsymbol{k}$ lies on a circle given by (15). Figure 6 shows this circle projected on a planar grid as a function of α (azimuth) and β (elevation). α varies from $\frac{1}{2}\pi$ to $\frac{3}{2}\pi$ and β varies from $-\frac{1}{2}\pi$ to $\frac{1}{2}\pi$.

Consider now a closed contour such as that shown in Fig. 7. For this contour M has been computed at every possible orientation. Figure 8 shows a perspective plot of M as a function of azimuth and elevation. This function has two maxima corresponding to two possible orientations of the image contour being interpreted as the most compact shape. Figure 9 shows loci of possible orientations for space contours of a predetermined compactness. Locus 1 corresponds to $M = 0.04$ (an elongated shape), locus 2 corresponds to $M = 0.05$ while locus 3 corresponds to $M = 0.076$. The two maxima correspond in this case to $M = 0.077$. Notice that for a circle $M = 0.079$.

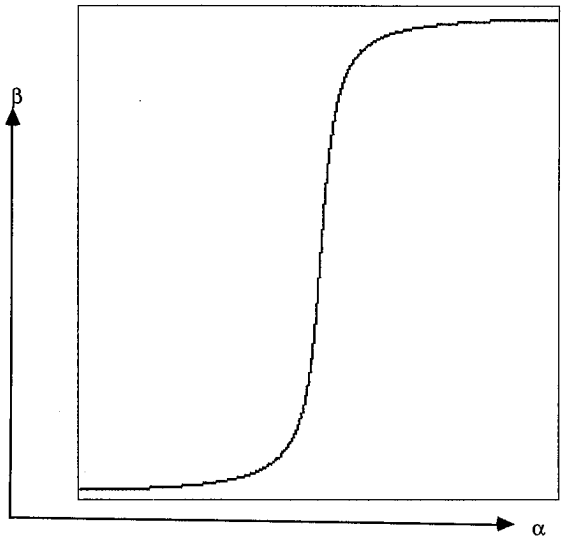

Fig. 6. The locus of possible orientations of the cone's axis lying in the plane defined by the viewing vector and the dotted line of the previous figure.

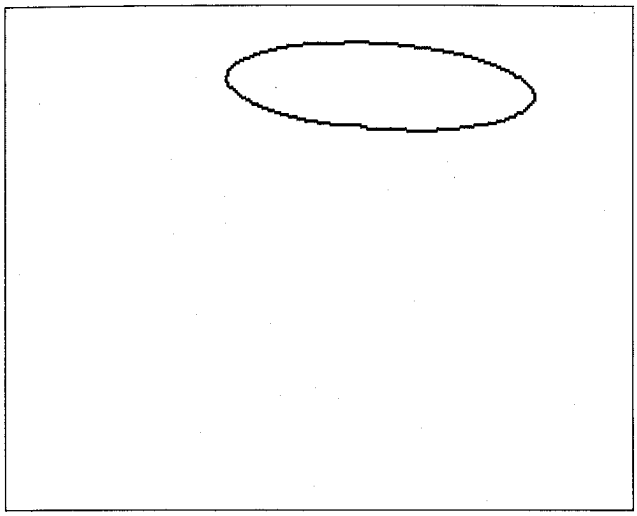

Fig. 7. An image contour. This contour may be interpreted as the projection onto the image of a space contour C lying in a space plane whose associated normal unit vector is $\boldsymbol{n}$.

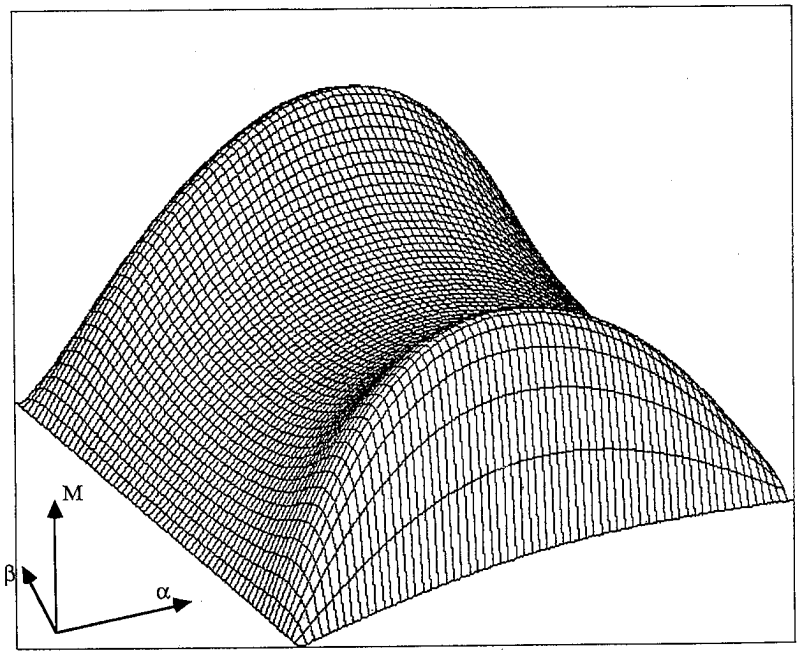

Fig. 8. M (area over the square of the perimeter) for the contour shown in Fig. 7 as a function of the deprojection of the contour. The locations of the maxima of this surface correspond to the orientation of the most compact interpretations of the image contour.

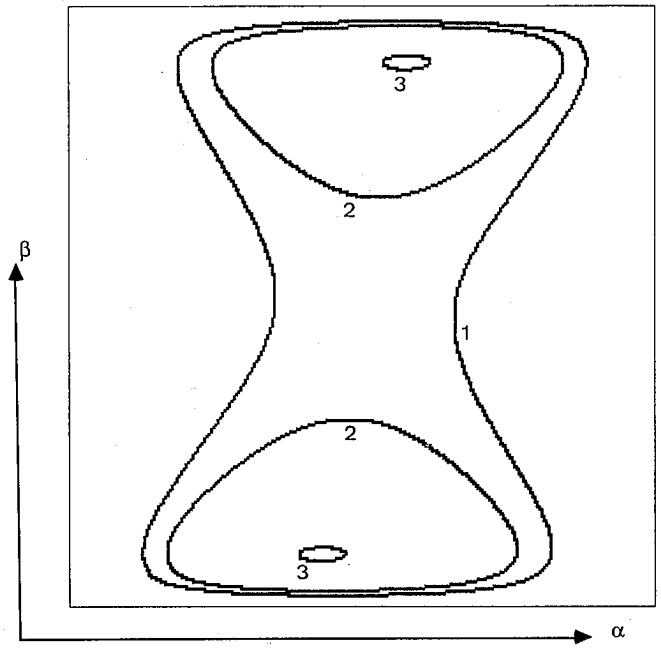

Fig. 9. Loci of possible orientations of the space contours which have the same compactness. 1: $M = 0.04$, 2: $M = 0.05$, 3: $M = 0.076$. Notice that in the case of a circle $M = 0.079$.

3. Interpreting Image Contours

We summarize the main results of the previous section. We have developed a method allowing certain image contours to be interpreted in terms of contours arising from a generalized cylinder. A pair of extremal contours having a straight axis of symmetry constrain both the expansion function of the corresponding generalized cylinder and the spatial orientation of its axis.

A closed discontinuity contour combined with a compactness measure may be interpreted as a closed planar 3D contour. The latter is in fact the cross-section of a generalized cylinder.

Let's consider now an isolated image contour. This contour cannot provide by itself enough information such that it may be classified into extremal, discontinuity, or something else. This section is partially devoted to this problem. First we recall a contour labeling scheme recently suggested by Malik [10]. Next we describe a contour classification algorithm based on this scheme. Finally we suggest a way of combining extremal and discontinuity contours in order to further constrain the shape and orientation of the perceived surface.

3.1. Contour labeling

This section summarizes the labeling scheme developed by Malik [10]. Assume that the boundary of an object is a C^3 piecewise-smooth surface. A generalized cylinder belongs to this class if its cross-section and expansion functions are C^3. Moreover, any object that is a union of such primitives belongs to the class as well. If two C^3 surfaces intersect at a point where the surfaces have distinct tangent planes, then it can be shown that the intersection of the two surfaces is a C^3 arc. Such an arc is called an edge. A point of intersection of three or more edges is called a *vertex*. The projection of edges and vertices onto the image gives rise to *junctions*. The possible geometric configurations of these junctions are given by the following analysis.

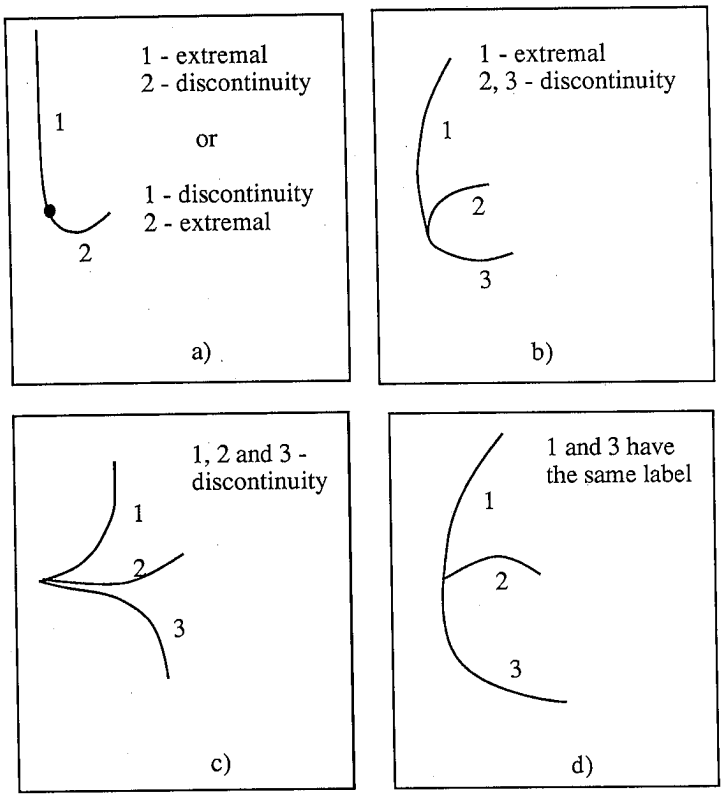

Fig. 10. A catalogue of image junctions. (a) *Curvature-L* may be labeled: 1: extremal and 2: discontinuity, or 1: discontinuity and 2: extremal. (b) *Three-tangent* is labeled: 1: extremal, 2: discontinuity and 3: discontinuity. (c) Projection of a vertex neighborhood: 1, 2 and 3 are all discontinuity boundaries. (d) For a *T-junction* 1 and 3 must have the same label (extremal or discontinuity).

(1) *Projection of an edge neighborhood.* Let S_1, S_2 be the two surfaces meeting at the edge and let P be a point on the edge. Assuming general viewpoint, the viewing direction $\boldsymbol{u}$ defines a contour generator C_u on the first surface. If the contour generator passes through P, the following image junctions may arise:

(a) *Curvature-L.* This is a curvature discontinuity point along a curve. The tangent is continuous because the projection of the intersection curve is tangent to the projection of the contour generator. Figure 10(a) shows the possible labels of such a junction.

(b) *Three-tangent.* Three curves with a common tangent. Two have the same curvature. This junction has a unique labeling as shown on Fig. 10(b).

(2) *Projection of a vertex neighborhood.* The projection of a vertex locally looks like the projection of an equivalent polyhedral vertex formed by replacing each of the surface elements by their tangent planes. If there is no contour generator through the vertex (which would violate the general viewpoint assumption) all the image segments meeting at such a junction are discontinuity contours. An example of such a junction is shown on Fig. 10(c).

To this analysis one may add *T-junctions* where two of the three arcs have the same tangent and curvature and hence they have the same label. See Fig. 10(d). Notice that this junction indicates occlusion.

3.2. A contour classification algorithm

The contour classification algorithm begins with a classical contour detection process: edge detection and edge linking. Next, contours are approximated with straight lines and arcs (segments). This analysis is quickly expanded to include the connecting segments. Each segment has two sets of connecting segments associated with each one of its ends. A junction is defined as a set of at least two segments passing through a common point. Within a junction, colinear lines, arcs with a common tangent, a line tangent to an arc and arcs with the same curvature are detected carefully.

This network of segments and junctions is then mapped into a graph representation. A junction is mapped into a node and a segment is mapped into an arc. Both nodes and arcs are labeled as follows. A node may have a label out of Malik's catalogue: curvature-L, three-tangent, vertex or T-junction. An arc may be labeled discontinuity, extremal, unknown or ambiguous.

Notice that this graph is planar. The planarity is a direct consequence of the fact that two contours never cross each other (a junction is detected whenever contours intersect) and hence, two arcs in the graph never cross. The planarity of the graph implies that the complexity of graph exploration is highly reduced. Contour classification consists of exploring this graph in order to propagate the constraints available with the junction type. The result is a reduction in the number of possible labelings of each arc.

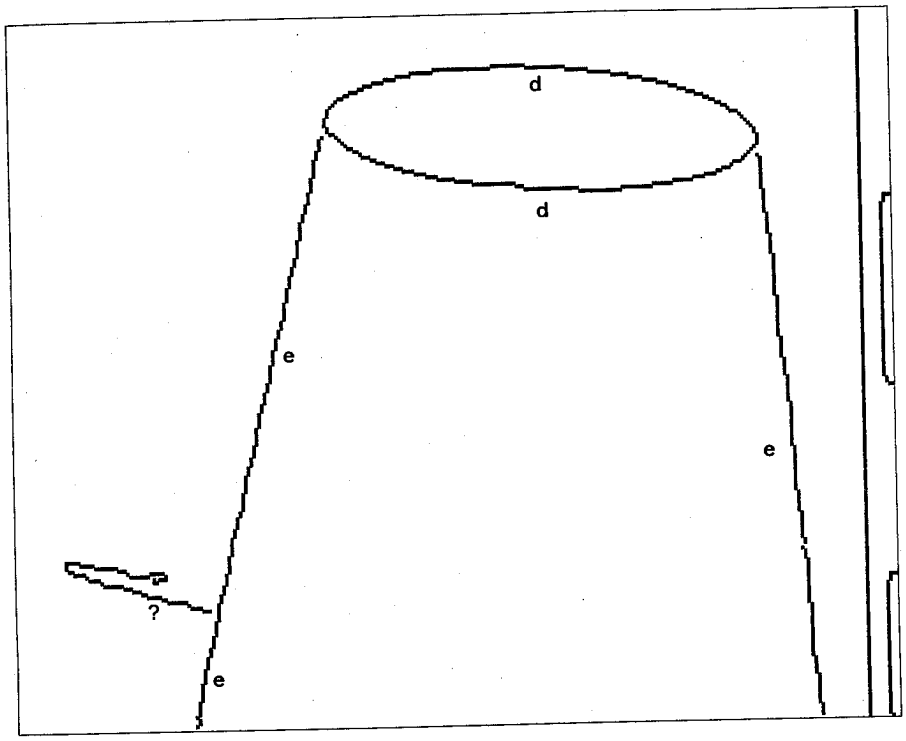

Fig. 11. A line drawing to be interpreted with two three-tangent junctions and one T-junction. The labeling shown on the figure is unique: *d*, discontinuity, *e* extremal.

Consider for example the drawing of Fig. 11. The algorithm described above detected two three-tangent junctions and one T-junction. The final contour labeling is shown on Fig. 11.

3.3. Combining contour interpretations

Let's try now to interpret the set of discontinuity and extremal contours of Fig. 11 in terms of a 3D surface, namely a generalized cylinder. Figure 12 shows the constraint (the *S-curve*) derived from the axis of symmetry of the extremal contours *overlapped* onto the constraints derived from the discontinuity contour. The intersections of these constraints provide solutions for the space orientation of the generalized cylinder. If one eliminates exaggeratedly slanted interpretations, it follows that there is no solution for a cross-section with $M = 0.04$, two solutions for a cross-section with $M = 0.05$, four solutions for $M = 0.076$ and two solutions for a circular cross-section.

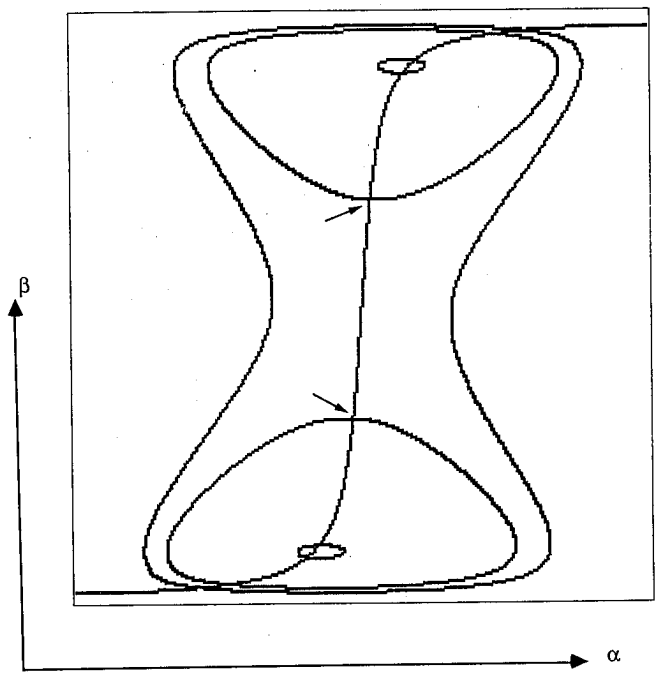

Fig. 12. Combining orientation constraints derived from extremal and discontinuity boundaries. The arrows indicate two distinct solutions corresponding to a generalized cylinder whose cross-section has a compactness of $M = 0.05$.

3.4. Discussion

The Gaussian sphere (the α-β space) provides a common representation for interpreting both extremal and discontinuity contours. This enables to *intersect* constraints derived from various image contours. This common representation provides as well a framework for discussing the relative importance of extremal and discontinuity contours.

Independently of any particular interpretation, a pair of extremal contours provides a one-parameter locus of orientations for the axis of a generalized cylinder. Any selected solution *must* lie along or in the vicinity of this locus. A discontinuity contour provides a two-parameter locus of orientations for the plane containing the deprojected contour. If the compactness of this space contour is known, this locus is reduced to a curve. Moreover, there are only two solutions for the most compact interpretation of a discontinuity contour. These two solutions lie or do not lie along the S-curve. In the first case, the obvious interpretation is an object with a cross-section perpendicular to the axis of the cylinder. The second case may suggest that the cross-section makes an eccentricity angle ψ with the cylinder's axis.

Finally, our approach requires for the discontinuity contours to be closed. In conclusion, extremal contours provide stronger constraints than discontinuity contours.

4. Conclusions

In this paper we have analyzed the extent with which image contours constrain both the shape and orientation of a scene surface. We have focused our attention on those contours arising from a particular surface type, i.e., generalized cylinders. Pairs of extremal contours constrain the axis of the generalized cylinder to one degree of freedom. The shape from contour paradigm may be applied to closed discontinuity contours.

Next we have shown how image contours can be mapped into a graph representation and labeled in terms of surface contours. The labeled contours are backprojected onto the Gaussian sphere where their corresponding constraints are combined.

In the future we plan to extend this approach such that we can deal with generalized cones and with even more complex objects such as a concatenation of generalized cylinders and cones. We also plan to investigate more thoroughly the relationship between two-dimensional symmetry as defined by Brady and Asada [5] and the projection of three-dimensional surfaces. We would also like to devise a method for interpreting nonclosed discontinuity contours and to extend the shape from contour paradigm to nonplanar space curves.

Surface contour interpretation is an important component of computer vision systems since alternative techniques such as stereo and motion have difficulties with interpreting this class of contours.

Appendix A. Derivation of equation (13)

Referring to Fig. 4 we have:

$$\boldsymbol{v} = \|\boldsymbol{v}\| \cdot \boldsymbol{v}'/\|\boldsymbol{v}'\| .$$

We also have:

$$\boldsymbol{v}' = \boldsymbol{n} \times (\boldsymbol{p} \times \boldsymbol{u}_p) = (\boldsymbol{n} \cdot \boldsymbol{u}_p)\boldsymbol{p} - (\boldsymbol{n} \cdot \boldsymbol{p})\boldsymbol{u}_p$$

and

$$\|\boldsymbol{v}\| = \frac{(d+f)\|\boldsymbol{e}\|}{(d+f)\cos\theta + \|\boldsymbol{e}\|\sin\theta} .$$

The length of $\boldsymbol{e}$ is $\|\boldsymbol{e}\| = \|\boldsymbol{p}\|(d+f)/f$. We also have:

$$\cos\theta = \frac{\boldsymbol{v}' \cdot \boldsymbol{p}}{\|\boldsymbol{v}'\| \, \|\boldsymbol{p}\|} \quad \text{and} \quad \sin\theta = \frac{\|\boldsymbol{v}' \times \boldsymbol{p}\|}{\|\boldsymbol{v}'\| \, \|\boldsymbol{p}\|} .$$

We obtain:

$$\boldsymbol{v} = \frac{(d+f)\|\boldsymbol{p}\|^2}{f(\boldsymbol{v}' \cdot \boldsymbol{p}) + \|\boldsymbol{p}\| \, \|\boldsymbol{v}' \times \boldsymbol{p}\|} \boldsymbol{v}' .$$

From the expression of $\boldsymbol{v}'$ we have:

$$\boldsymbol{v}' \cdot \boldsymbol{p} = (\boldsymbol{n} \cdot \boldsymbol{u}_p)\|\boldsymbol{p}\|^2 - (\boldsymbol{n} \cdot \boldsymbol{p})(\boldsymbol{p} \cdot \boldsymbol{u}_p) ,$$
$$\|\boldsymbol{v}' \times \boldsymbol{p}\| = (\boldsymbol{n} \cdot \boldsymbol{p})\|\boldsymbol{p} \times \boldsymbol{u}_p\| .$$

The expression of $\boldsymbol{v}$ becomes:

$$\boldsymbol{v} = \frac{(d+f)\|\boldsymbol{p}\|^2}{f(\boldsymbol{n} \cdot \boldsymbol{u}_p)\|\boldsymbol{p}\|^2 - f(\boldsymbol{n} \cdot \boldsymbol{p})(\boldsymbol{p} \cdot \boldsymbol{u}_p) + (\boldsymbol{n} \cdot \boldsymbol{p})\|\boldsymbol{p}\| \, \|\boldsymbol{p} \times \boldsymbol{u}_p\|} \boldsymbol{v}' .$$

Let the coordinates of the image point P be x, y and f. Hence, the coordinates of $\boldsymbol{p}$ are x, y and 0 and the coordinates of $\boldsymbol{u}_p$ are x/s, y/s and f/s with $s = (x^2 + y^2 + f^2)^{1/2}$. We easily obtain:

$$\boldsymbol{p} \cdot \boldsymbol{u}_p = \|\boldsymbol{p}\|^2/s \quad \text{and} \quad \|\boldsymbol{p} \times \boldsymbol{u}_p\| = f\|\boldsymbol{p}\|/s .$$

The last two terms of the denominator of $\boldsymbol{v}$ vanish. The expression of $\boldsymbol{v}$ becomes:

$$\boldsymbol{v} = \frac{d+f}{f} \frac{\boldsymbol{v}'}{\boldsymbol{n} \cdot \boldsymbol{u}_p} .$$

By replacing $\boldsymbol{v}'$ by its expression, we finally obtain equation (13).

REFERENCES

1. Barnard, S.T., Interpreting perspective images, *Artificial Intelligence* **21** (1983) 435–462.
2. Barnard, S.T. and Pentland, A., Three-dimensional shape from line drawings, in: *Proceedings Image Understanding Workshop*, Arlington, VA (1983) 282–284.
3. Barrow, H.G. and Tenenbaum, J.M., Interpreting line drawings as three-dimensional surfaces, *Artificial Intelligence* **17** (1981) 75–116.
4. Binford, T.O., Visual perception by computer, in: *Proceedings IEEE Conference on Systems and Control*, Miami, FL (1971).
5. Brady, M. and Asada, H., Smoothed local symmetries and their implementation, *Int. J. Rob. Res.* **3** (3) (1984) 36–61.

6. Brady, M., Ponce, J., Yuille, A. and Asada, H., Describing surfaces, *Comput. Vision Graph. Image Process.* **32** (1985) 1–28.
7. Brady, M. and Yuille, A., An extremum principle for shape from contour, *IEEE Trans. Pattern Anal. Mach. Intell.* **6** (3) (1984) 288–301.
8. Kanade, T., Recovery of the three-dimensional shape of an object from a single view, *Artificial Intelligence* **17** (1981) 409–460.
9. Koenderink, J., The internal representation of solid shape based on the topological properties of the apparent contour, in: W. Richards and S. Ullmann (Eds.), *Image Understanding 1986* (Ablex, Norwood, NJ, 1986).
10. Malik, J., Labeling line drawings of curved objects, in: *Proceedings Image Understanding Workshop*, Miami Beach, FL (1985) 209–218.
11. Marr, D., Analysis of occluding contour, *Proc. Roy. Soc. London B* **197** (1977) 441–475.
12. Marr, D., *Vision* (Freeman, San Francisco, CA, 1982).
13. Witkin, A.P., Recovering surface shape and orientation from texture, *Artificial Intelligence* **17** (1981) 17–45.

Received May 1987; revised version received January 1988