

# Constraints for Recognizing and Locating Curved 3D Objects from Monocular Image Features \*

David J. Kriegman,<sup>1</sup> B. Vijayakumar,<sup>1</sup> Jean Ponce<sup>2</sup>

<sup>1</sup> Center for Systems Science, Dept. of Electrical Engineering, Yale University, New Haven, CT 06520-1968, USA

<sup>2</sup> Beckman Institute, Dept. of Computer Science, University of Illinois, Urbana, IL 61801, USA

**Abstract.** This paper presents viewpoint-dependent constraints that relate image features such as t-junctions and inflections to the pose of curved 3D objects. These constraints can be used to recognize and locate object instances in the imperfect line-drawing obtained by edge detection from a single image. For objects modelled by implicit algebraic equations, the constraints equations are polynomial, and methods for solving these systems of constraints are briefly discussed. An example of pose recovery is presented.

## 1 Introduction

While in the “classical approach” to object recognition from images, an intermediate  $2\frac{1}{2}$ D or 3D representation is constructed and matched to object models, the approach presented in this paper bypasses this intermediate representation and instead directly matches point image features to three dimensional vertices, edges or surfaces. Similar approaches to recognition and positioning of polyhedra from monocular images have been demonstrated by several implemented algorithms [3, 4, 8] and are based on the use of the so-called “rigidity constraints” [1, 2] or “viewpoint consistency constraints” [8]. This feature-matching approach is only possible, however, because most observable image features are the projections of object features (edges and vertices). In contrast, most visible features in the image of a curved object depend on viewpoint and cannot be traced back to particular object features. More specifically, the image contours of a smooth object are the projections of limb points (occluding contours, silhouette) which are regular surface points where the viewing direction is tangent to the surface; they join at t-junctions and may also terminate at cusp points which have the additional property that the viewing direction is an asymptotic direction of the surface.

In this paper, we show how matching a small number of point image features to a model leads to a system of polynomial equations which can be solved to determine an object’s pose. Hypothesized matches between image features and modelled surfaces, edges and vertices can be organized into an interpretation tree [2], and the mutual existence of these features can be verified from a previously computed aspect graph [5, 11]. The image features emphasized in this paper and shown in figure 1.a are generic viewpoint dependent point features and include vertices, t-junctions, cusps, three-tangent junctions, curvature-L junctions, limb inflections, and edge inflections [9]. This is an exhaustive list of the possible contour singularities and inflections which are stable with respect to viewpoint; for almost any viewpoint, perturbing the camera position in a small ball around the original viewpoint will neither create nor destroy these features. More details of the presented approach can be found in [7].

---

\* This work was supported by the National Science Foundation under Grant IRI-9015749.

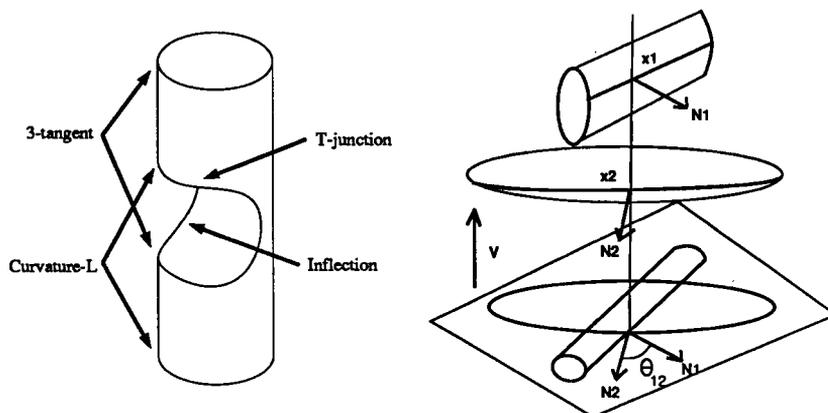


Fig. 1. a. Some viewpoint dependent image features for piecewise smooth objects. b. A t-junction and the associated geometry.

## 2 Object Representation and Image Formation

In this paper, objects are modelled by algebraic surfaces and their intersection curves. We consider implicit surfaces given by the zero set of a polynomial

$$f(\mathbf{x}) = f(x, y, z) = 0. \quad (1)$$

The surface will be considered nonsingular, so the unnormalized surface normal is given by  $\mathbf{n}(\mathbf{x}) = \nabla f(\mathbf{x})$ . Note that rational parametric surface representations, such as Bezier patches, non-uniform rational B-splines, and some generalized cylinders, can be represented implicitly by applying elimination theory, and so the presented constraints readily extend to these representations [6, 12]. The intersection curve between two surfaces  $f$  and  $g$  is simply given by the common zeros of the two defining equations:

$$\begin{cases} f(\mathbf{x}) = 0 \\ g(\mathbf{x}) = 0. \end{cases} \quad (2)$$

In this paper, we assume scaled orthographic projection though the approach can be extended to perspective; the projection of a point  $\mathbf{x} = [x, y, z]^t$  onto the image plane  $\tilde{\mathbf{x}} = [\tilde{x}, \tilde{y}]^t$  can be written as:

$$\tilde{\mathbf{x}} = \mathbf{x}_0 + \mu[\mathbf{w} \mid \mathbf{u}]^t \mathbf{x} \quad (3)$$

$\mathbf{w}, \mathbf{u}$  form an orthonormal basis for the image plane, and  $\mathbf{v} = \mathbf{w} \times \mathbf{u}$  is the viewing direction;  $\mathbf{x}_0 = [x_0, y_0]^t$  and  $\mu$  respectively parameterize image translation and scaling.

## 3 Viewpoint-Dependent Feature Constraints

We now consider the constraints that relate a pair of points on an object model to measured image features in terms of a system of  $n$  equations in  $n$  unknowns where the unknowns are the coordinates of the model points. While these constraints hold in general, they can be manipulated into systems of polynomial equations for algebraic surfaces. To solve these systems, we have used the global method of homotopy continuation to find all roots [10] as well as a combination of table lookup and Newton's method to only find the real roots. For each pair of points, the parameters of the viewing transformation can be easily calculated. Below, constraints are presented for all of the features found in Malik's junction catalogue [9]. Additionally, constraints are presented for inflections of image contours which are easily detected in images. Pose estimation from three vertices (viewpoint independent features) has been discussed elsewhere [4]. Note that other pairings of the same features are possible and lead to similar constraints.

### 3.1 T-junctions

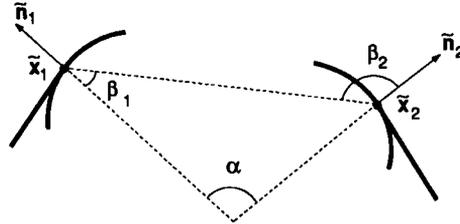
First, consider the hypothesis that an observed t-junction is the projection of two limb points  $\mathbf{x}_1, \mathbf{x}_2$  as shown in figure 1.b which provides the following geometric constraints:

$$\begin{cases} f_i(\mathbf{x}_i) = 0 \\ (\mathbf{x}_1 - \mathbf{x}_2) \cdot \mathbf{N}_i = 0 \\ \mathbf{N}_1 \cdot \mathbf{N}_2 = \cos \theta_{12}, \end{cases} \quad (4)$$

where  $i = 1, 2$ ,  $\mathbf{N}_i$  denotes the unit surface normals, and  $\cos \theta_{12}$  is the observed angle between the image normals. In other words, we have five equations, one observable  $\cos \theta_{12}$ , and six unknowns ( $\mathbf{x}_1, \mathbf{x}_2$ ). In addition, the viewing direction is given by  $\mathbf{v} = \mathbf{x}_1 - \mathbf{x}_2$ . When another t-junction is found, we obtain another set of five equations in six unknowns  $\mathbf{x}_3, \mathbf{x}_4$ , plus an additional vector equation:  $(\mathbf{x}_1 - \mathbf{x}_2) \times (\mathbf{x}_3 - \mathbf{x}_4) = \mathbf{0}$  where only two of the scalar equations are independent. This simply expresses the fact that the viewing direction should be the same for both t-junctions. Two observed t-junctions and the corresponding hypotheses (i.e., "t-junction one corresponds to patch one and patch two", and "t-junction two corresponds to patch three and patch four") provide us with 12 equations in 12 unknowns. Such a system admits a finite number of solutions in general. For each solution, the viewing direction can be computed, and the other parameters of the viewing transformation are easily found by applying eq. (3). Similar constraints are obtained for t-junctions that arise from the projection of edge points by noting that the 3D curve tangent, given by  $\mathbf{t} = \nabla f \times \nabla g$ , projects to the tangent of the image contour.

### 3.2 Curvature-L and Three-tangent Junctions

For a piecewise smooth object, curvature-L or three-tangent junctions are observed when a limb terminates at an edge, and they meet with a common tangent; observe the top and bottom of a coffee cup, or consider figure 1. Both feature types have the same local geometry, however one of the edge branches is occluded at a curvature-L junction.



**Fig. 2.** The image plane geometry for pose estimation from three-tangent and curvature-L junctions: The curved branch represents the edge while the straight branch represents a limb.

Consider the two edge points  $\mathbf{x}_i, i = 1, 2$  formed by the surfaces  $f_i, g_i$  that project to these junctions.  $\mathbf{x}_i$  is also an occluding contour point for one of the surfaces, say  $f_i$ . Note the image measurements (angles  $\alpha, \beta_1$  and  $\beta_2$ ) shown in figure 2. Since  $\mathbf{x}_i$  is a limb point of  $f_i$ , the surface normal is aligned with the measured image normal  $\tilde{\mathbf{n}}_i$ . Thus, the angle  $\alpha$  between  $\tilde{\mathbf{n}}_1$  and  $\tilde{\mathbf{n}}_2$  equals the angle between  $\mathbf{n}_1$  and  $\mathbf{n}_2$ , or  $\cos \alpha = \mathbf{n}_1 \cdot \mathbf{n}_2 / |\mathbf{n}_1| |\mathbf{n}_2|$ . Now, define the two vectors  $\Delta = \mathbf{x}_1 - \mathbf{x}_2$  and  $\tilde{\Delta} = \tilde{\mathbf{x}}_1 - \tilde{\mathbf{x}}_2$ . Clearly the angle between  $\tilde{\mathbf{n}}_i$  and  $\tilde{\Delta}$  must equal the angle between  $\mathbf{n}_i$  and the projection of  $\Delta$  onto the image plane  $\tilde{\Delta}$  which is given by  $\tilde{\Delta} = \Delta - (\Delta \cdot \hat{\mathbf{v}}) \hat{\mathbf{v}}$  where  $\hat{\mathbf{v}} = \mathbf{n}_1 \times \mathbf{n}_2 / |\mathbf{n}_1 \times \mathbf{n}_2|$  is the normalized viewing direction. Noting that  $\mathbf{n}_i \cdot \hat{\mathbf{v}} = 0$ , we have  $|\mathbf{n}_i| |\tilde{\Delta}| \cos \beta_i = \mathbf{n}_i \cdot \Delta$ . However,  $\tilde{\Delta}$  is of relatively high degree and a lower degree equation is obtained by taking the ratio of  $\cos \beta_i$  and using the equation for  $\cos \alpha$ . After squaring and rearrangement, these equations

$$\begin{cases} (\mathbf{n}_1 \cdot \mathbf{n}_1)(\mathbf{n}_2 \cdot \mathbf{n}_2) \cos \alpha - (\mathbf{n}_1 \cdot \mathbf{n}_2)^2 = 0, \\ \frac{\cos \beta_1}{\cos \beta_2} (\mathbf{n}_2 \cdot \Delta)(\mathbf{n}_1 \cdot \mathbf{n}_2) - \cos \alpha (\mathbf{n}_2 \cdot \mathbf{n}_2)(\mathbf{n}_1 \cdot \Delta) = 0. \end{cases} \quad (5)$$

along with the edge equations (2) form a system of six polynomial equations in six unknowns whose roots can be found; the pose is then determined from (3).

### 3.3 Inflections

Inflections (zeros of curvature) of an image contour can arise from either limbs or edges. In both cases, observing two such points is sufficient for determining object pose.

As Koenderink has shown, a limb inflection is the projection of a point on a parabolic line (zero Gaussian curvature) [5], and for a surface defined implicitly, this is

$$f_x^2(f_{yy}f_{zz} - f_{yz}^2) + f_y^2(f_{xx}f_{zz} - f_{xz}^2) + f_z^2(f_{xx}f_{yy} - f_{xy}^2) + 2f_xf_y(f_{xz}f_{yz} - f_{zz}f_{xy}) \\ + 2f_yf_z(f_{xy}f_{xz} - f_{xx}f_{yz}) + 2f_xf_z(f_{xy}f_{yz} - f_{yy}f_{xz}) = 0, \quad (6)$$

where the subscripts indicate partial derivatives. Since both points  $\mathbf{x}_1, \mathbf{x}_2$  are limbs, equation (6) and the surface equation for each point can be added to (5) for measured values of  $\alpha, \beta_1$  and  $\beta_2$  as depicted in figure 2. This system of six equations in  $\mathbf{x}_1, \mathbf{x}_2$  can be solved to yield a set of points, and consequently the viewing parameters.

In the case of edges, an image contour inflection corresponds to the projection of an inflection of the space curve itself or a point where the viewing direction is orthogonal to the binormal. Space curve inflections typically occur when the curve is actually planar, and can be treated like viewpoint independent features (vertices). When inflections arise from the binormal  $\mathbf{b}_i$  being orthogonal to the viewing direction, as in figure 1, two measured inflections are sufficient for determining pose. It can be shown that the projection of  $\mathbf{b}_i$  is the image contour normal, and for surfaces defined implicitly, the binormal is given by  $\mathbf{b} = [t^t H(g)t] \nabla f - [t^t H(f)t] \nabla g$  where  $H(f)$  is the Hessian of  $f$ . By including the curve equations (2) with (5) after replacing  $\mathbf{n}_i$  by  $\mathbf{b}_i$ , a system of six equations in  $\mathbf{x}_1, \mathbf{x}_2$  is obtained. After solving this system, the pose can be readily determined.

### 3.4 Cusps

Like the other features, observing two cusps in an image is sufficient for determining object pose. It is well known that cusps occur when the viewing direction is an asymptotic direction at a limb point which can be expressed as  $\mathbf{v}^t H(\mathbf{x}_i) \mathbf{v} = 0$  where the viewing direction is  $\mathbf{v} = \nabla f_1(\mathbf{x}_1) \times \nabla f_2(\mathbf{x}_2)$ . While the image contour tangent is not strictly defined at a cusp (which is after all a singular point), the left and right limits of the tangent as the cusp is approached will be in opposite directions and are orthogonal to the surface normal. Thus, the cusp and surface equations can be added to the system (5) which is readily solved for  $\mathbf{x}_1$  and  $\mathbf{x}_2$  followed by pose calculation.

## 4 Implementation and Results

Fig. 3.a shows an image of a cylinder with a cylindrical notch and two inflection points found by applying the Canny edge detector and fitting cubic splines. The edge constraints of section 3.3 lead to a system of six polynomial equations with 1920 roots. However, only two roots are unique, and figs. 3.b and 3.c show the corresponding poses. Clearly the pose in fig. 3.c could be easily discounted with additional image information. As in [6], elimination theory can be used to construct an implicit equation of the image contours of the intersection curve parameterized by the pose. By fitting this equation to all detected edgels on the intersection curve using the previously estimated pose as initial conditions for nonlinear minimization, the pose is further refined as shown in fig. 3.d. Using continuation to solve the system of equations required nearly 20 hours on a SPARC Station 1, though a recently developed parallel implementation running on network of SPARC stations or transputers should be significantly faster. However, since there are only a few

real roots, another effective method is to construct a table offline of  $\alpha, \beta_i$  as a function of the two edge points. Using table entries as initial conditions to Newton's method, the same poses are found in only two minutes. Additional examples are presented in [7].

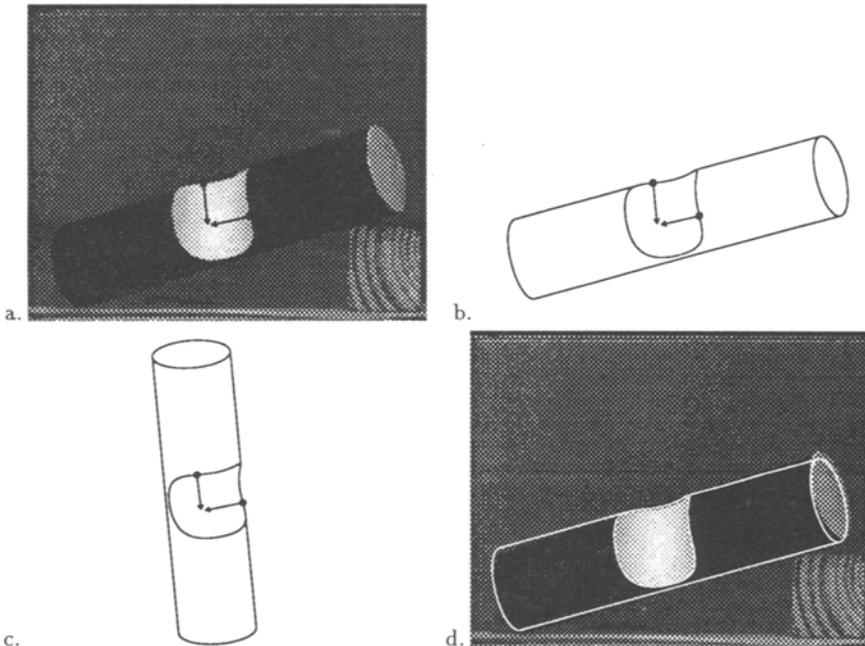


Fig. 3. Pose estimation from two inflection points. Note the scale difference in c.

#### Acknowledgments:

Many thanks to Darrell Stam for his distributed implementation of continuation.

#### References

1. O. Faugeras and M. Hebert. The representation, recognition, and locating of 3-D objects. *Int. J. Robot. Res.*, 5(3):27–52, Fall 1986.
2. W. E. L. Grimson. *Object Recognition by Computer: The Role of Geometric Constraints*. MIT Press, 1990.
3. R. Horaud. New methods for matching 3-D objects with single perspective views. *IEEE Trans. Pattern Anal. Mach. Intelligence*, 9(3):401–412, 1987.
4. D. Huttenlocher and S. Ullman. Object recognition using alignment. In *International Conference on Computer Vision*, pages 102–111, London, U.K., June 1987.
5. J. Koenderink. *Solid Shape*. MIT Press, Cambridge, MA, 1990.
6. D. Kriegman and J. Ponce. On recognizing and positioning curved 3D objects from image contours. *IEEE Trans. Pattern Anal. Mach. Intelligence*, 12(12):1127–1137, 1990.
7. D. Kriegman, B. Vijayakumar, and J. Ponce. Strategies and constraints for recognizing and locating curved 3D objects from monocular image features. Technical Report 9201, Yale Center for Systems Science, 1992.
8. D. G. Lowe. The viewpoint consistency constraint. *Int. J. Computer Vision*, 1(1), 1987.
9. J. Malik. Interpreting line drawings of curved objects. *Int. J. Computer Vision*, 1(1), 1987.
10. A. Morgan. *Solving Polynomial Systems using Continuation for Engineering and Scientific Problems*. Prentice Hall, Englewood Cliffs, 1987.
11. J. Ponce, S. Petitjean, and D. Kriegman. Computing exact aspect graphs of curved objects: Algebraic surfaces. In *European Conference on Computer Vision*, 1991.
12. T. W. Sederberg, D. Anderson, and R. N. Goldman. Implicit representation of parametric curves and surfaces. *Comp. Vision, Graphics, and Image Proces.*, 28:72–84, 1984.